2020

API Guide : Helical Insight

API Guide of Helical Insight BI Tool

Version: 3.1

API Guide is an in-depth document informing about all the APIs used in detail along with examples.

Table of Contents

Prerequisites

Product Information

This API document is applicable for Helical Insight version: **Helical Insight 3.1**

Server side requirement

Java	Java version 1.8
Operating System	Windows 7/8/10, Linux, Mac OS
Database	MySQL / Postgresql / Derby
application Server	Any servlet container (eg: Apache Tomcat version 6 or higher) or full blown Java EE application server (eg: Jboss) or jetty 9.2

Minimum Hardware Requirements :

RAM	4 GB
Processor	2.5GHz + multi-core Pentium for Windows, Mac, and Linux
Disk Space	10 GB

Browser Settings

- **Browser:** Mozilla Firefox , Google Chrome,Safari
- Javascript should be enabled
- Cookies have to be enabled

Preinstallation Configuration

When the application is installed with the configured database then two default users (**super user** and **super admin**) are created in the database. Following are the list of tables where user details are stored :

- h_users
- organization
- role
- user_role
- profile

Two different roles are configured in setting.xml file which is located at ${SOLUTION DIRECTORY}/ System / Admin directory. The default configured roles are **ROLE_ADMIN** and **ROLE_USER**. If the respective roles are changed in the setting.xml then WEB-INF/spring-security.xml file also need to be updated before installing the application.

Snippet of setting.xml

```
<defaultRoleNames>
<roleUser email="user@helicaltech.com" name="hiuser">ROLE_USER</roleUser>
<roleAdmin email="admin@helicaltech.com" name="hiadmin">ROLE_ADMIN</roleAdmin>
</defaultRoleNames>
```

Here, username / email / default roles can be configured as required.

Super admin: This user has no organization and its default role is ROLE_ADMIN. This user has authority to modify organization / user details such as add / remove and so on.

Super user: This user does not belong to any organization and its default role is ROLE_USER. The superuser has the authority to share reports / files with all the users registered and can access reports / files which are being shared with superuser.

Organization Admin: This user has an organization (created by superadmin) and its default role is ROLE_ADMIN of that respective organization.This user has authority to modify user details of that organization.

Organization User: This user (created by organization admin) has an organization and has default role ROLE_USER of that respective organization. This user has authority to view reports/files that has been shared with the user as well as can share reports/ files within the organization.

After logging into the application, the server sets JSESSIONID attribute in the client cookie. The server time, session expiry details, current time is also set in the cookie. Thereon the browser uses these cookies with every request.

Following are the cookie details:

- **currentTime 1449500912771**
- **serverTime 1449490981479**
- **sessionExpiry 1449490981479**
-

Note: The above cookies are set in the path of the context.

The server also stores additional information in the session. The spring security mechanism stores the user information such as user role, user profile, email and other user details.

Pre-requisite to Test API service:

>We are using POSTMAN tool to test API service.
>Select **'x-www-form-urlencoded'** unde**r Body** while sending the parameters using **POST** method

>While passing formData: value make sure that the value should be passed in single line.

For Ex. ***formData:{"name":"ROLE_developer","organisation":"118"}***

>Before running API service, Authenticated user should Login. **[Refer login module]** , If the user is not logged in then you will get login page.

Definations:

HTTP Request MethodS :

GET : HTTP Request to get data from server using URL.

POST : HTTP Request to send data to server using URL.

HTTP Request Key : HTTP Request Key is the json object key.

HTTP Request Value: HTTP Request Key is the json object value for particular HTTP Request Value.

SERVICE STATUS:

200	Success OK
302	Redirection error
304	Not modified
404	Not Found
401	Access Denied / unauthorized
500	Internal Server

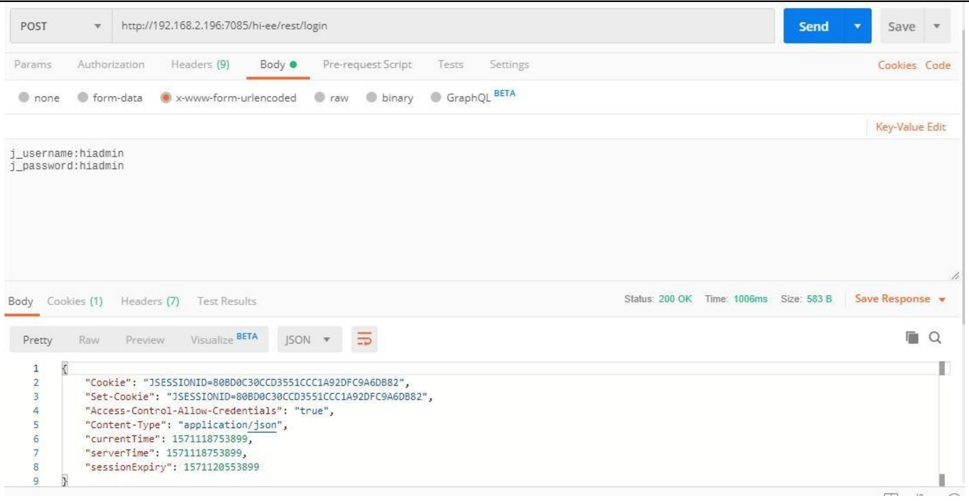
1. Login Module

The Login Module is a portal module that allows users to type a user name and password and organization name(optional) to log in to the **"Helical Insight application".** The application is no longer available to users after they have logged in.

1.1 REST Login

URL	rest/login	
Description	With REST login we can login to helical insight application. The REST login stores the JSESSIONID , serverTime,sessionExpiry information in cookies.	
Pre-requisite	The Helical Insight Application should be up.	
Accessible for	Every user.	
HTTP Request Method	**POST**	
Example	**Access through Brower :** http://192.168.2.196:7085/hi-ee/rest/login **Access through Curl command :** curl -d "j_username=hiadmin&j_password=hiadmin" http://192.168.2.196:7085/hi-ee/rest/login -v	
HTTP Request Key	**HTTP Request Value**	**Description**
j_organization:	(optional)	Leave the organization name as blank.
j_username:	hiadmin	Enter user name
j_password:	hiadmin	Enter the password
Response Output	*{ "Cookie": "JSESSIONID=80BD0C30CCD3551CCC1A92DFC9A6DB82", "Set-Cookie": "JSESSIONID=80BD0C30CCD3551CCC1A92DFC9A6DB82", "Access-Control-Allow-Credentials": "true", "Content-Type": "application/json", "currentTime": 1571118753899, "serverTime": 1571118753899, "sessionExpiry": 1571120553899 }*	

Description of Response Output:	REST login returns the stored cookie information as response
Service Status	200 OK

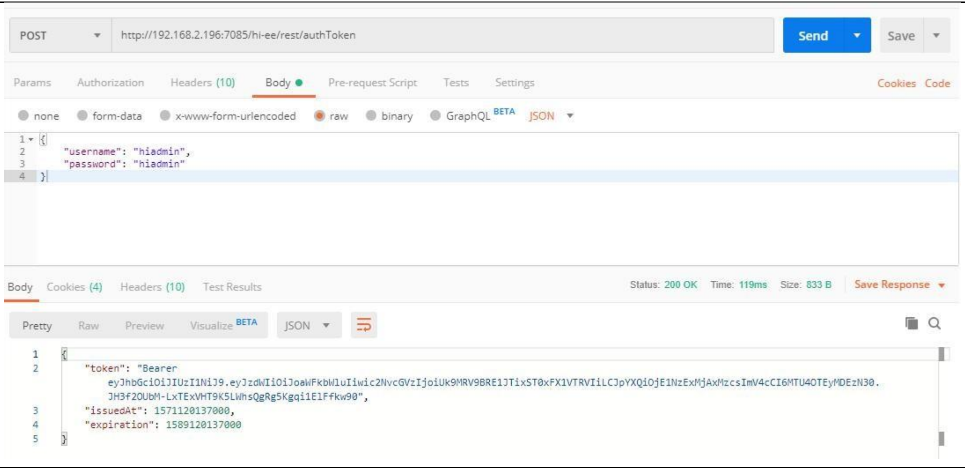
Screenshot	 POST http://192.168.2.196:7085/hi-ee/rest/login
	x-www-form-urlencoded
	`j_username:hiadmin`
	`j_password:hiadmin`
	Status: 200 OK Time: 1006ms Size: 583 B
	`{`
	`"Cookie": "JSESSIONID=80BD0C30CCD3551CCC1A92DFC9A6DB82",`
	`"Set-Cookie": "JSESSIONID=80BD0C30CCD3551CCC1A92DFC9A6DB82",`
	`"Access-Control-Allow-Credentials": "true",`
	`"Content-Type": "application/json",`
	`"currentTime": 1571118753899,`
	`"serverTime": 1571118753899,`
	`"sessionExpiry": 1571120553899`
	`}`

Post-action	After REST login you can access the Helical Insight application using cookie information

1.1.1 REST Login using JWT (JSON Web Token) Authentication

URL	rest/authToken	
Description	With REST login using JWT authentication we can login to helical insight application using token value. The REST login JWT authentication returns the token ,issued time and expiry time.	
Pre-requisite	The Helical Insight Application should be up.	
Accessible for	Every user.	
HTTP Request Method	**POST**	
Example	**Access through Brower :** http://192.168.2.156:8085/hi-ee/rest/authToken	
HTTP Request Key	**HTTP Request Value**	**Description**
username:organization(optional)	hiadmin	Enter user name.In case of organization user need to provide it with username.For Ex. User_name:Organization_name
password:	hiadmin	Enter the password
Response	{	

Output	*"token": "Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoaWFkbWluIiwic2NvcGVzIjoiUk9MRV9BRE1JTixST0xFX1VTRVIiLCJpYXQiOjE1NzExMjAxMzcsImV4cCI6MTU4OTEyMDEzN30.JH3f2OUbM-LxTExVHT9K5LWhsQgRg5Kgqi1ElFfkw90",* *"issuedAt": 1571120137000,* *"expiration": 1589120137000* *}*
Description of Response Output:	REST login JWT token authentication returns the bearer type token along with token issued and token expiry time-stamp.
Service Status	200 OK

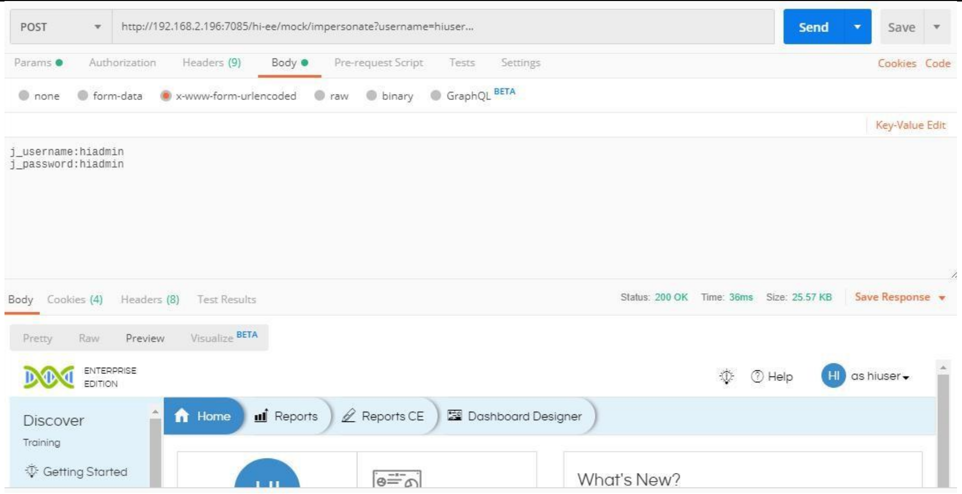
Screenshot	
	POST http://192.168.2.196:7085/hi-ee/rest/authToken
	Body: raw JSON
	`{`
	`"username": "hiadmin",`
	`"password": "hiadmin"`
	`}`
	Status: 200 OK Time: 119ms Size: 833 B
	`{`
	`"token": "Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJoaWFkbWluIiwic2NvcGVzIjoiUk9MRV9BRE1JTixST0xFX1VTRVIiLCJpYXQiOjE1NzExMjAxMzcsImV4cCI6MTU4OTEyMDEzN30.JH3f2OUbM-LxTExVHT9K5LWhsQgRg5Kgqi1ElFfkw90",`
	`"issuedAt": 1571120137000,`
	`"expiration": 1589120137000`
	`}`
	 Send
	Save
	Params
	Authorization
	Headers (10)
	Body
	Pre-request Script
	Tests
	Settings
	Cookies
	Code
	none
	form-data
	x-www-form-urlencoded
	raw
	binary
	GraphQL BETA
	Cookies (4)
	Headers (10)
	Test Results
	Save Response
	Pretty
	Raw
	Preview
	Visualize BETA

Post-action	After REST login JWT authentication you can access the Helical Insight application using bearer type token

1.2 Impersonate/MIMIC Login

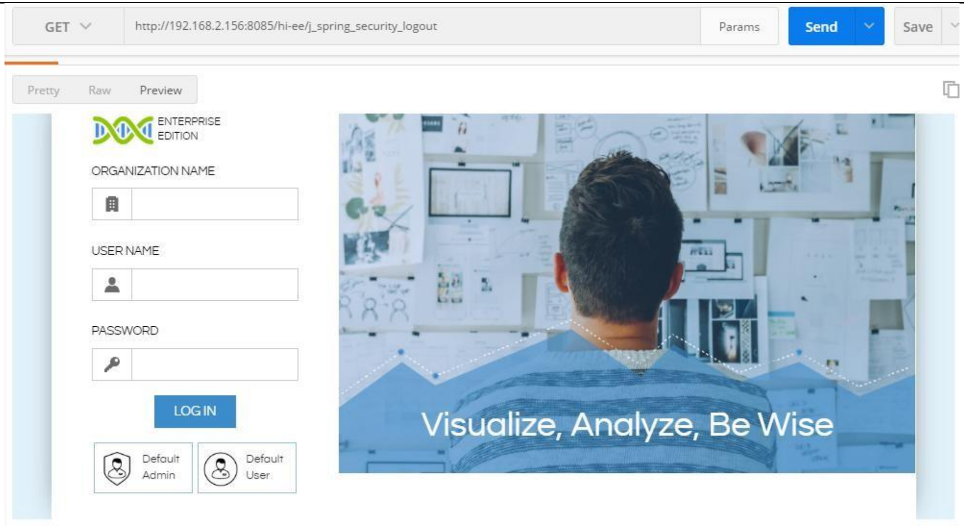
URL	mock/impersonate?username=user_name:organization_name
Description	User can impersonate/MIMIC any user through Superadmin user/ROLE_Admin.
Pre-requisite	The Helical Insight Application should be up.
Accessible for	Every user except ROLE_ADMIN user/Superadmin
HTTP Request Method	**POST**
Example	**Access through Brower :** http://192.168.2.196:7085/hi-ee/mock/impersonate?username=hiuser

HTTP Request Key	**HTTP Request Value**	**Description**
username:organization(optional)	hiadmin	Enter user name.In case of organization user need to provide it with username.For Ex. User_name:Organization_name
Response Output	Response of service API is nothing but the HTML contents of welcome.html for impersonate user.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.196:7085/hi-ee/mock/impersonate?username=hiuser...
	 Body: x-www-form-urlencoded
	 `j_username:hiadmin`
	 `j_password:hiadmin`
	 Status: 200 OK Time: 36ms Size: 25.57 KB Send
	Save
	Params
	Authorization
	Headers (9)
	Body
	Pre-request Script
	Tests
	Settings
	Cookies Code
	none
	form-data
	raw
	binary
	GraphQL BETA
	Key-Value Edit
	Body
	Cookies (4)
	Headers (8)
	Test Results
	Save Response
	Pretty
	Raw
	Preview
	Visualize BETA
	ENTERPRISE EDITION
	Help
	HI as hiuser
	Discover
	Training
	Getting Started
	Home
	Reports
	Reports CE
	Dashboard Designer
	What's New?	

1.3 Logout

URL	j_spring_security_logout
Description	We can logout from helical insight application if its already logged in.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	Every user.
HTTP Request Method	**GET,POST**
Example	**Access through Brower :** http://192.168.2.156:8085/hi-ee/j_spring_security_logout **Access through Curl command :** curl http://192.168.2.156:8085/hi-ee/j_spring_security_logout -v

Response Output	*> GET /hi-ee/j_spring_security_logout HTTP/1.1* *> Host: 192.168.2.156:8085* *> User-Agent: curl/7.52.1* *> Accept: */** *>* *< HTTP/1.1 302 Found* *< Set-Cookie: JSESSIONID=;Version=1;Path=/hi-ee;Expires=Thu, 01-Jan-1970 00:00:00 GMT;Max-Age=0* *< Expires: Thu, 01 Jan 1970 00:00:00 GMT* *< Location: http://192.168.2.156:8085/hi-ee/login.html* *< Content-Length: 0* *< Server: Jetty(9.2.z-SNAPSHOT)*
Description of Response Output:	j_spring_security_logout returns the cookie information as response with HTTP 302 Found .
Service Status	302
Screenshot	 GET http://192.168.2.156:8085/hi-ee/j_spring_security_logout Params Send Save Pretty Raw Preview ENTERPRISE EDITION ORGANIZATION NAME USER NAME PASSWORD LOG IN Default Admin Default User Visualize, Analyze, Be Wise

2. Admin Module

Admin module offers users the possibility to define different levels of access to information in the application.It allows to Create and manages the Organizations, Users, Roles and profiles. Allows to manage Scheduled jobs. Deals with Application configurations such as set Logger settings, Reloading Configurations. Refresh and Delete 'Temp Directory', 'Resources in memory', 'Cached reports', 'cached datasources' .

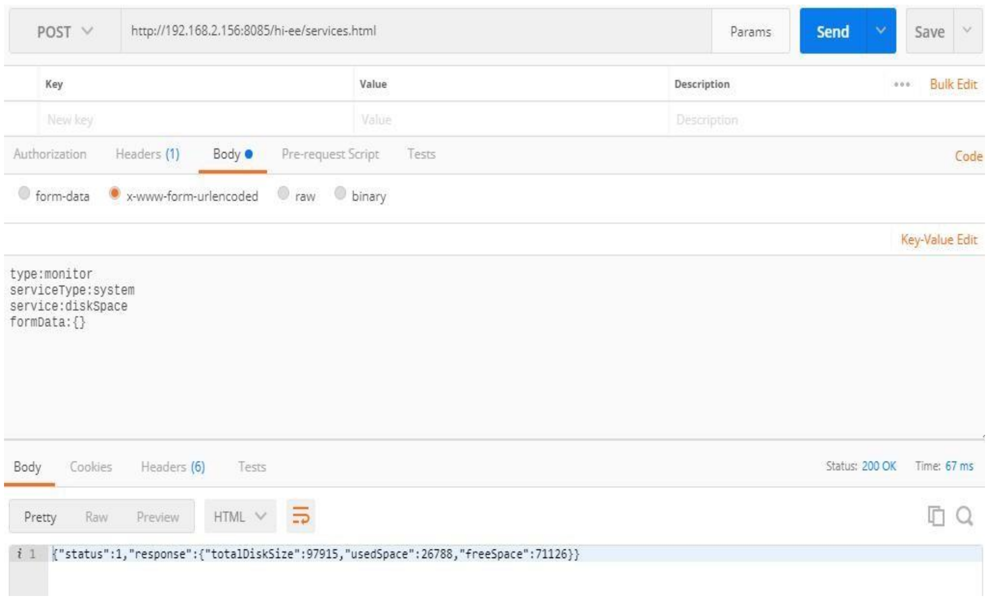
2.1 Overview

2.1.1 Disk Space Refresh

URL	services.html	
Description	It allows super admin to check the disk space status of the system. Expected output should be the used disk space , free disk space and the total disk space.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through Brower :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=diskSpace&formData:={}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	diskSpace	Service name as diskSpace
formData:	{}	JSON object containing disk space information

Response Output (JSON format)	{
	"status":1,
	"response":
	{"totalDiskSize":97215,"usedSpace":26788,"freeSpace":71126}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns the some parameter values as response like **totalDiskSize :** Total disk size of system. **usedSpace:** Used disk space of system. **freeSpace:** Free/available space on the system.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/services.html

```
type:monitor
serviceType:system
service:diskSpace
formData:{}
```

Status: 200 OK Time: 67 ms

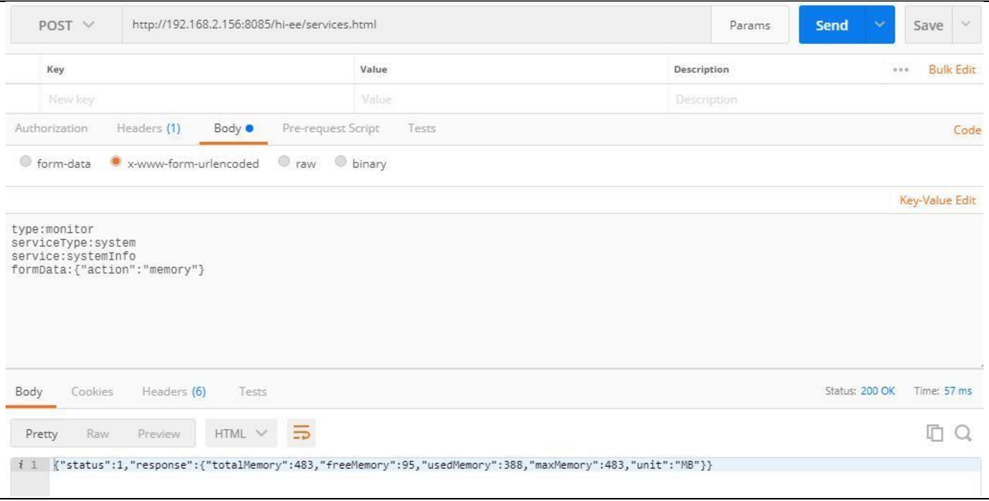
```
{"status":1,"response":{"totalDiskSize":97915,"usedSpace":26788,"freeSpace":71126}}
```

2.1.2 JVM Memory Refresh

URL	services.html	
Description	It allows super admin to check the JVM memory status of the system. It gives you the information like total JVM memory , used memory , free memory and the unit of memory etc.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=systemInfo&formData:={'action':'memory'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	systemInfo	Service name as systemInfo
formData:	{"action":"memory"}	JSON object containing action as memory.

Response Output (JSON format)	{
	"status":1,"
	response":{"totalMemory":483,"freeMemory":95,"usedMemory":388,"maxMemory":483,"unit":"MB"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns some parameter values as response like **totalMemory**:Total memory allocated to java virtual machine **usedMemory:** The memory consumed by java runtime environment. **freeMemory :** Available memory in java virtual machine. **maxMemory:** Maximum memory in java virtual machine. **unit**:Unit of the memory.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:system
service:systemInfo
formData:{"action":"memory"}
```

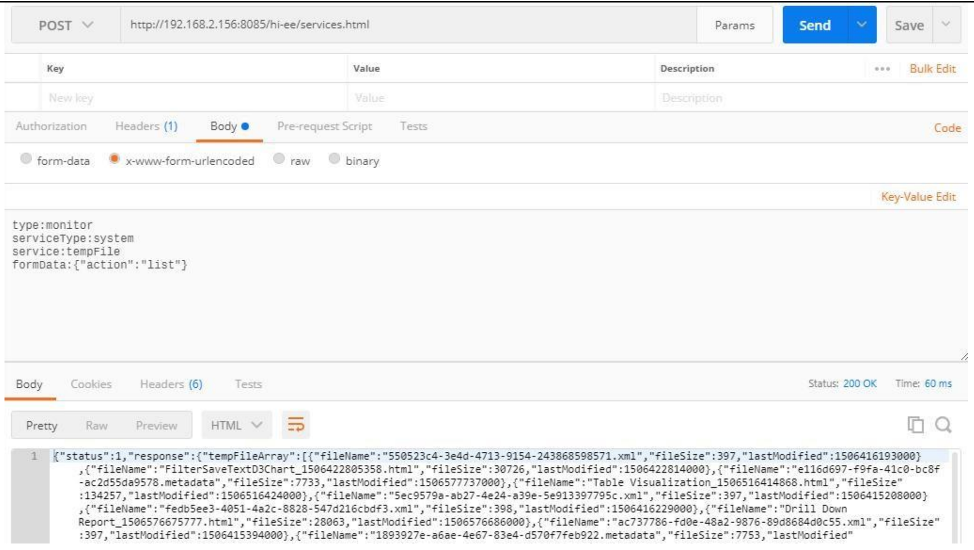
Status: 200 OK Time: 57 ms

```
{"status":1,"response":{"totalMemory":483,"freeMemory":95,"usedMemory":388,"maxMemory":483,"unit":"MB"}}
```

2.1.3 Temp Directory

2.1.3.1 Temp Directory Refresh

URL	services.html	
Description	It allows super admin to refresh the the recent files or the application downloaded file present in System/Temp directory. For Example : If the user exports any report, it will get stored in temp directory.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=tempList&formData:={'action':list}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	tempFile	Service name as tempFile
formData:	{"action":"list"}	JSON object containing action list of temp directory.
Response Output (JSON format)	{"status":1,"response":{"tempFileArray":[{"fileName":"1505914529229.csv","fileSize":197,"lastModified":1505914529241},{"fileName":"1505914550680.xls","fileSize":4608,"lastModified":1505914551824},{"fileName":"1505914607504.xls","fileSize":4608,"lastModified":1505914607536},{"fileName":"1505914938565.xls","fileSize":4608,"lastModified":1505914938593},{"fileName":"5e1972af-2d10-4f9c-9cbf-4c4a92d2e45a.metadata","fileSize":7753,"lastModified":1505901994993},{"fileName":"hiadmin_1505900889255.crt","fileSize":510,"lastModified":1505900889323},{"fileName":"hiadmin_1505901027634.crt","fileSize":510,"lastModified":1505901027653},,{"fileName":"Sample EFW Report_1505905860632.jpg","fileSize":54420,"lastModified":1505905886154},{"fileName":"Simple Bubble Chart_1505914491387.xls","fileSize":5207,"lastModified":1505914501098}}]	
Description of response output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns temporary files array with files details like:	

	filename : Name of the file **filesize** : Size of the file in KB. **lastModified** : Last modified date time of file.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:system
service:tempFile
formData:{"action":"list"}
```

Status: 200 OK Time: 60 ms

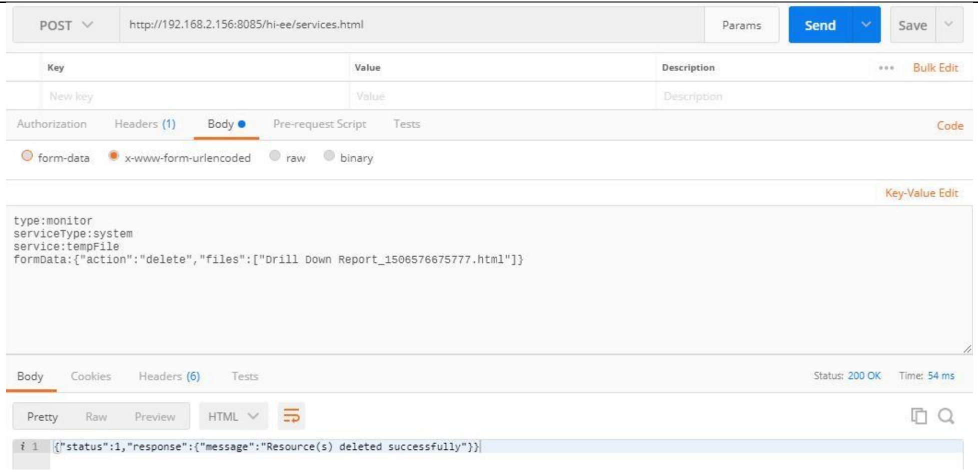
```
{"status":1,"response":{"tempFileArray":[{"fileName":"550523c4-3e4d-4713-9154-243868598571.xml","fileSize":397,"lastModified":1506416193000}
,{"fileName":"FilterSaveTextD3Chart_1506422805358.html","fileSize":30726,"lastModified":1506422814000},{"fileName":"e116d697-f9fa-41c0-bc8f
-ac2d55da9578.metadata","fileSize":7733,"lastModified":1506577737000},{"fileName":"Table Visualization_1506516414868.html","fileSize"
:134257,"lastModified":1506516424000},{"fileName":"5ec9579a-ab27-4e24-a39e-5e913397795c.xml","fileSize":397,"lastModified":1506415208000}
,{"fileName":"fedb5ee3-4051-4a2c-8828-547d216cbdf3.xml","fileSize":398,"lastModified":1506416229000},{"fileName":"Drill Down
Report_1506576675777.html","fileSize":28063,"lastModified":1506576686000},{"fileName":"ac737786-fd0e-48a2-9876-89d8684d0c55.xml","fileSize"
:397,"lastModified":1506415394000},{"fileName":"1893927e-a6ae-4e67-83e4-d570f7feb922.metadata","fileSize":7753,"lastModified"
```

2.1.3.2 Temp Directory Delete

URL	services.html	
Description	It allows super admin to delete the selected/particular file from the temp directory of the system. It deletes selected file from temporary files which are present in Temp directory.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=systemInfo&formData:={'action':'delete','files':['Drill Down Report_1506576675777.html']}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	tempFile	Service name as tempFile
formData:	{"action":"delete","files":["Drill Down Report_1506576675777.html"]}	Action: selected file to be delete from list of temp directory.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Resource(s) deleted successfully"}
	}	

Description of response output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as message as resources deleted successfully.	
Service Status	200 OK	

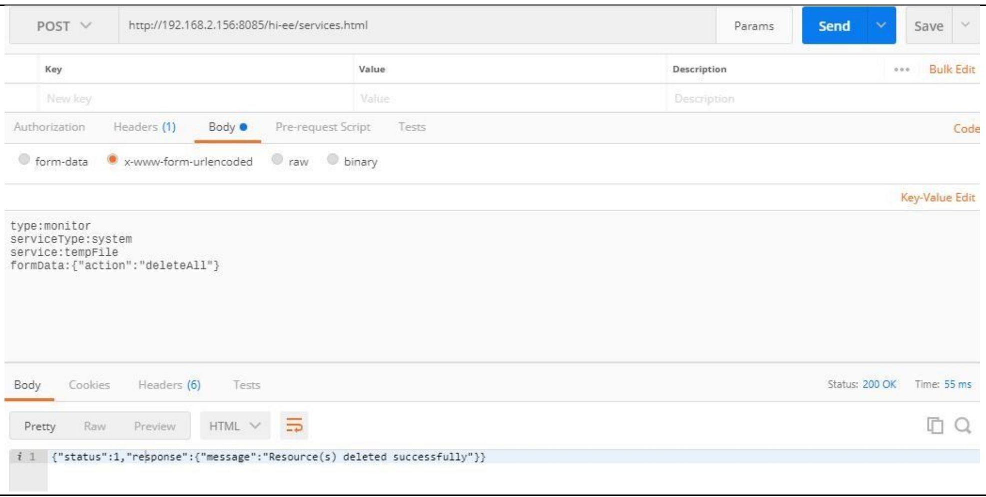
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	x-www-form-urlencoded
	`type:monitor`
	`serviceType:system`
	`service:tempFile`
	`formData:{"action":"delete","files":["Drill Down Report_1506576675777.html"]}`
	Status: 200 OK Time: 54 ms
	`{"status":1,"response":{"message":"Resource(s) deleted successfully"}}`

2.1.3.3 Temp Directory DeleteAll

URL	services.html	
Description	It allows super admin to delete all the files from the temp directory of the system. It deletes all the file from temporary files which are present in Temp directory.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=tempFile&formData:={'action':'deleteAll'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	tempFile	Service name as tempFile
formData:	{"action":"deleteAll"}	Action : to deleteAll files from list of temp directory.

Response output (JSON format)	{
	"status":1,
	"response":{"message":"Resource(s) deleted successfully"}
	}	

Description of response output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as message as resources deleted successfully.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	 Body: x-www-form-urlencoded
	 `type:monitor`
	 `serviceType:system`
	 `service:tempFile`
	 `formData:{"action":"deleteAll"}`
	 Status: 200 OK Time: 55 ms
	 `{"status":1,"response":{"message":"Resource(s) deleted successfully"}}`

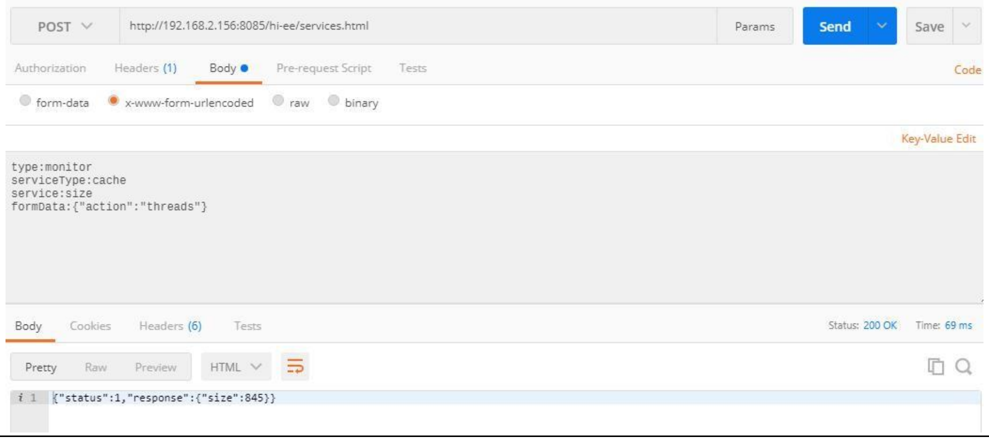
2.1.4 In Memory Cache

2.1.4.1 In Memory Cache Refresh

URL	services.html	
Description	It allows super admin to refresh the in memory cache data and calculates the size of cache.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=size&formData:={'action':'threads'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTTP Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	size	Service name as size
formData:	{"action":"threads"}	Action to refresh the threads to refresh the in memory cache.

Response Output(JSON format)	{
	"status":1,
	"response":{"size":845}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as size of cache memory. **size**: size of cache memory after refresh.	
Service Status	200 OK	

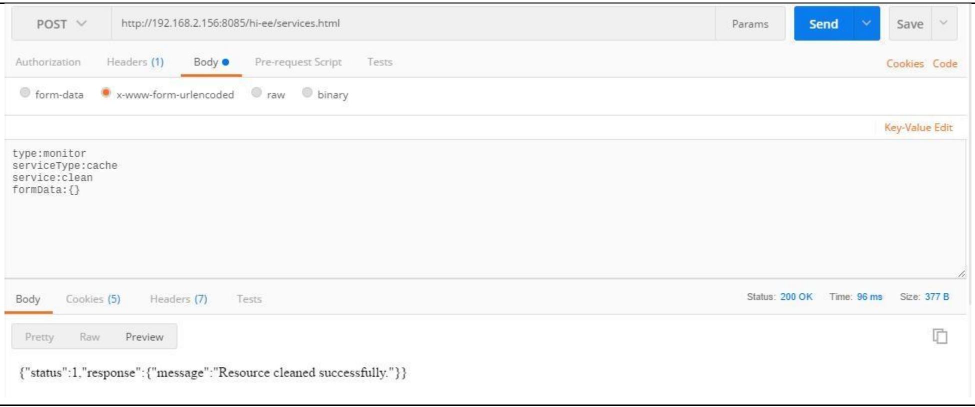
ScreenShot	 POST http://192.168.2.156:8085/hi-ee/services.html
	x-www-form-urlencoded
	`type:monitor`
	`serviceType:cache`
	`service:size`
	`formData:{"action":"threads"}`
	Status: 200 OK Time: 69 ms
	`{"status":1,"response":{"size":845}}`

2.1.4.2 In Memory Cache DeleteAll

URL	services.html	
Description	It allows super admin to delete all the in memory cache data.The system all in memory cache get deleted/clean. Note : After the in memory cache deletion automatically in memory cache get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=clean&formData:={}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	clean	Service name as clean
formData:	{}	Action to clean all the threads from the in memory cache.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Resource cleaned successfully."}
	}	

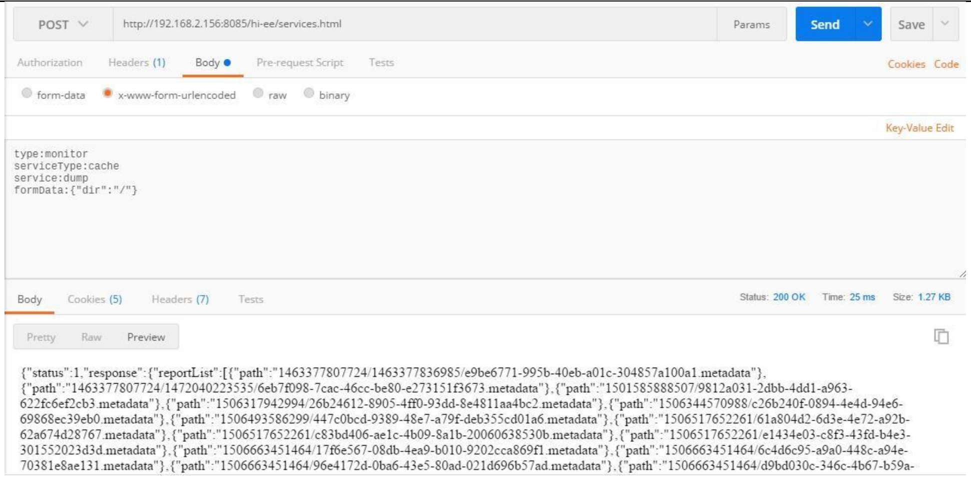
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:monitor`
	`serviceType:cache`
	`service:clean`
	`formData:{}`
	Status: 200 OK Time: 96 ms Size: 377 B
	`{"status":1,"response":{"message":"Resource cleaned successfully."}}`

2.1.5 Cache Reports

2.1.5.1 Cache Reports Refresh

URL	services.html	
Description	It allow super admin to view the latest cached reports size from of the system. For Ex. In case if you opened one report in another tab which will get added in cache reports , so after cache refresh you can see the updated list of cached reports.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=dump&formData:={'dir':'/'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	dump	Service name as dump
formData:	{"dir":"/"}	Action to refresh the cached reports.
Response Output : (JSON format)	{"status":1,"response":{"reportList":[{"path":"1463377807724/1463377836985\\e9be6771-995b-40eb-a01c-304857a100a1.metadata"},{"path":"1463377807724/1463377978248/Sample EFW Report\\sample_report.efw"},{"path":"1463377807724\\1463377836985\\e9be6771-995b-40eb-a01c-304857a100a1.metadata"},{"path":"1463377807724\\1463377978248\\Sample EFW Dashboard\\sample_dashboard.efw"}]}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as reportList array mentioned below : **path** : path of the file(report,metadata etc)	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:cache
service:dump
formData:{"dir":"/"}
```

Status: 200 OK Time: 25 ms Size: 1.27 KB

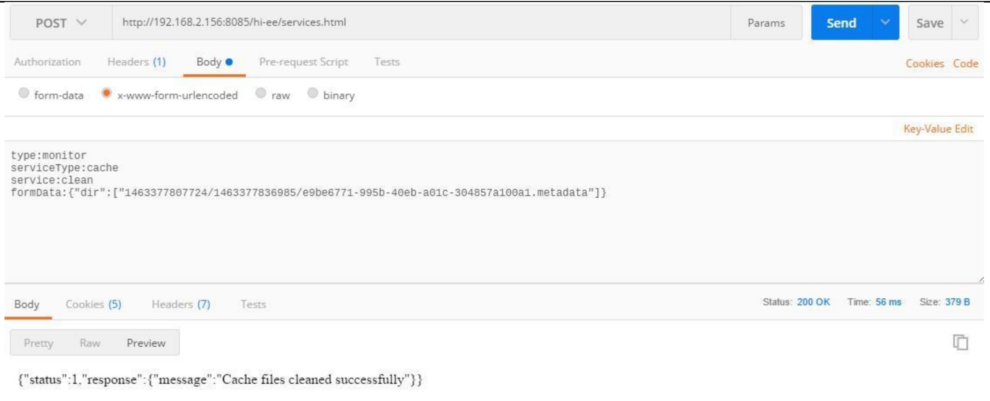
```
{"status":1,"response":{"reportList":[{"path":"1463377807724/1463377836985/e9be6771-995b-40eb-a01c-304857a100a1.metadata"},
{"path":"1463377807724/1472040223535/6eb7f098-7cac-46cc-be80-e273151f3673.metadata"},{"path":"1501585888507/9812a031-2dbb-4dd1-a963-
622fc6ef2cb3.metadata"},{"path":"1506317942994/26b24612-8905-4ff0-93dd-8e4811aa4bc2.metadata"},{"path":"1506344570988/c26b240f-0894-4e4d-94e6-
69868ec39eb0.metadata"},{"path":"1506493586299/447c0bcd-9389-48e7-a79f-deb355cd01a6.metadata"},{"path":"1506517652261/61a804d2-6d3e-4e72-a92b-
62a674d28767.metadata"},{"path":"1506517652261/c83bd406-ae1c-4b09-8a1b-20060638530b.metadata"},{"path":"1506517652261/e1434e03-c8f3-43fd-b4e3-
301552023d3d.metadata"},{"path":"1506663451464/17f6e567-08db-4ea9-b010-9202cca869f1.metadata"},{"path":"1506663451464/6c4d6c95-a9a0-448c-a94e-
70381e8ae131.metadata"},{"path":"1506663451464/96e4172d-0ba6-43e5-80ad-021d696b57ad.metadata"},{"path":"1506663451464/d9bd030c-346c-4b67-b59a-
```

2.1.5.2 Cache Reports Delete

URL	services.html	
Description	It allows super admin to delete the selected/particular reports from cached reports. You can select the report which you want to remove from cached reports.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=clean&formData:={'dir':['1463377807724/1463377836985/e9be6771-995b-40eb-a01c-304857a100a1.metadata']}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	clean	Service name as clean
formData:	{"dir":["1463377807724/1463377836985/e9be6771-995b-40eb-a01c-304857a100a1.metadata"]}	Action to delete selected cached reports from the directory

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Cache files cleaned successfully"}
	}	

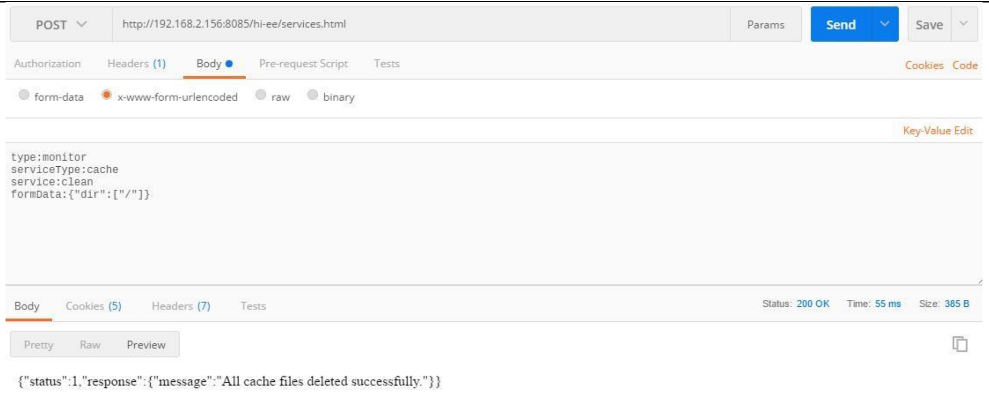
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message for cache file.	
Service Status	200 OK	

Screenshot	

```
POST  http://192.168.2.156:8085/hi-ee/services.html
Body: x-www-form-urlencoded

type:monitor
serviceType:cache
service:clean
formData:{"dir":["1463377807724/1463377836985/e9be6771-995b-40eb-a01c-304857a100a1.metadata"]}

Status: 200 OK   Time: 56 ms   Size: 379 B

{"status":1,"response":{"message":"Cache files cleaned successfully"}}
```

2.1.5.3 Cache Reports DeleteAll

URL	services.html	
Description	It allows super admin to delete all the cached reports from of the system. All Cached reports get deleted from the system. Note : After all cached reports deletion automatically cached report get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=clean&formData:={'dir':['/']}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	clean	Service name as clean
formData:	{"dir":["/"]}	Action to delete all cached reports from the directory

Response Output (JSON format)	{ "status":1, "response":{"message":"All cache files deleted successfully."} }
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message for all deleted cache file.
Service Status	200 OK
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html Body: x-www-form-urlencoded `type:monitor` `serviceType:cache` `service:clean` `formData:{"dir":["/"]}` Status: 200 OK Time: 55 ms Size: 385 B `{"status":1,"response":{"message":"All cache files deleted successfully."}}`

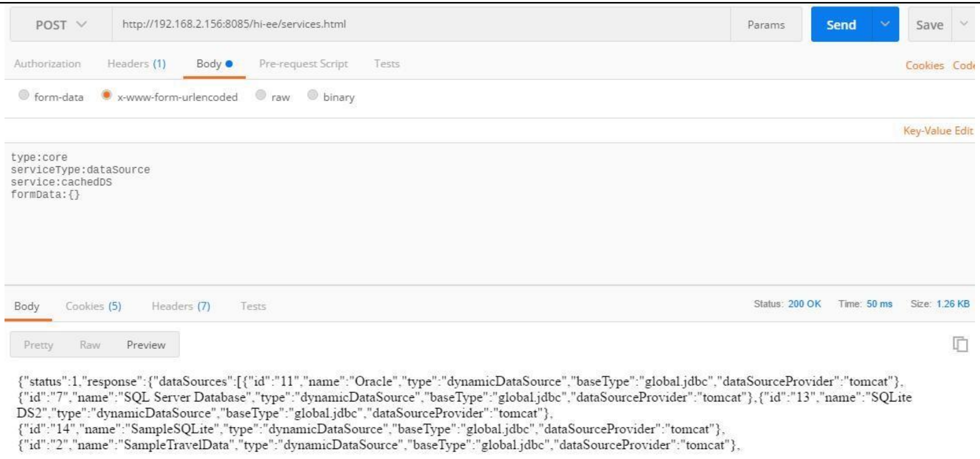
2.1.6 Cache Datasource

2.1.6.1 Cache Datasource Refresh

URL	services.html	
Description	It allows super admin to view the size of cached datasources from of the system. For Ex. In case if new managed datasource is added in another tab which will get added in cache Datasource, so after Datasources cache refresh you can see the updated list of cached Datasources.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=cachedDS&formData:={}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	Type of the Operation
serviceType:	datasource	serviceType as datasource
service:	cachedDS	Service name as cachedDS
formData:	{}	Action to refresh the cached datasources.

Response Output (JSON format)	{
	"status":1,"response":{"dataSources":[{"id":"11","name":"Oracle","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},{"id":"7","name":"SQL Server Database","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},{"id":"13","name":"SQLite DS2","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as datasource list after datasources are cached shown below : **id**: id of the datasource	

	name: Name of datasource **type**: Type of datasource **baseType** : base type of datasource **dataSourceProvider**: provider of datasource For Ex.tomcat
Service Status	200 OK
Screenshot	

```
type:core
serviceType:dataSource
service:cachedDS
formData:{}
```

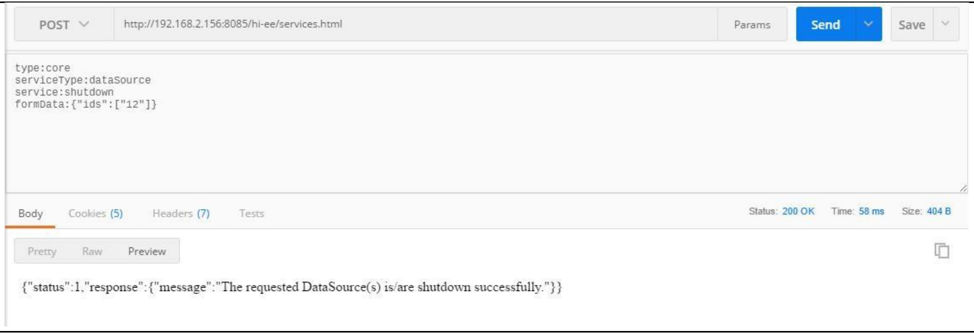
```
{"status":1,"response":{"dataSources":[{"id":"11","name":"Oracle","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},
{"id":"7","name":"SQL Server Database","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},{"id":"13","name":"SQLite
DS2","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},
{"id":"14","name":"SampleSQLite","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},
{"id":"2","name":"SampleTravelData","type":"dynamicDataSource","baseType":"global.jdbc","dataSourceProvider":"tomcat"},
```

2.1.6.2 Cache Datasource Delete

URL	services.html	
Description	It allows super admin to delete the selected datasources from cached datasources . Selected Cached datasources get deleted from the system. Note : After the selected cached datasources deletion, automatically cached report get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=shutdown&formData:={'ids':['12']}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	Type of the Operation
serviceType:	datasource	serviceType as datasource
service:	shutdown	Service name as shutdown
formData:	{"ids":["12"]}	Action to delete selected cached datasource from the directory using its id.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"The requested DataSource(s) is/are shutdown successfully."}
	}	

Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message after deletion of requested cached datasource.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html Params Send Save
	`type:core`
	`serviceType:dataSource`
	`service:shutdown`
	`formData:{"ids":["12"]}`
	Body Cookies (5) Headers (7) Tests Status: 200 OK Time: 58 ms Size: 404 B
	Pretty Raw Preview
	`{"status":1,"response":{"message":"The requested DataSource(s) is/are shutdown successfully."}}`

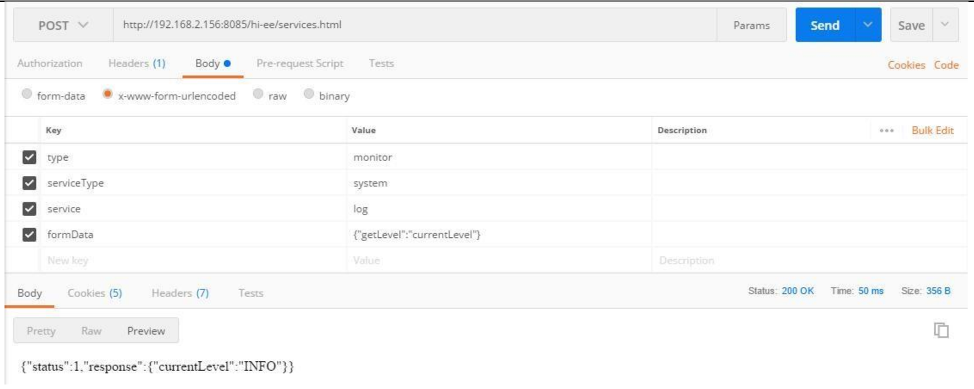
2.1.7 Logger Settings

2.1.7.1 Logger Settings Refresh

URL	services.html	
Description	It allows super admin to refresh current level of the logger settings. Current level of the logger settings get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=log&formData:={'getLevel':'currentLevel'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	log	Service name as log
formData:	{"getLevel":"currentLevel"}	Action to refresh current level of the logger settings.

Response Output: (JSON format)	{
	"status":1,
	"response":{"currentLevel":"INFO"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as currentLevel as INFO as default logger settings.	
Service Status:	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

Key	Value	Description
✓ type	monitor	
✓ serviceType	system	
✓ service	log	
✓ formData	{"getLevel":"currentLevel"}	

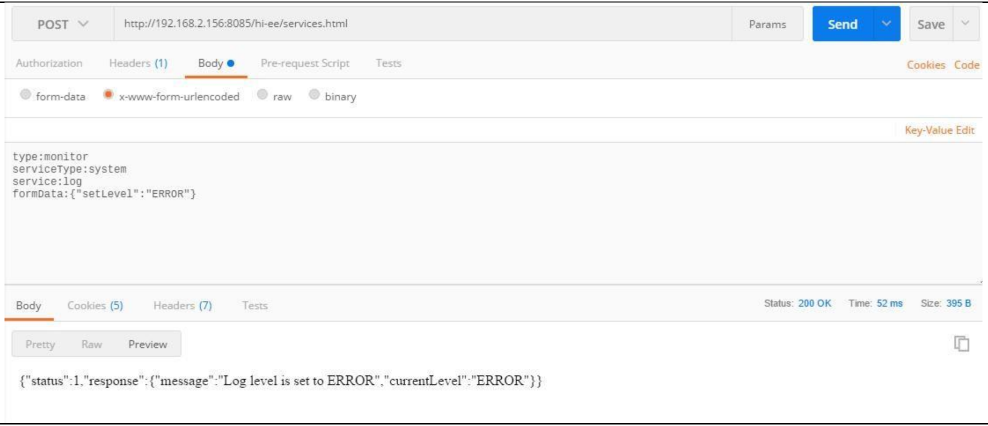
Status: 200 OK Time: 50 ms Size: 356 B

```
{"status":1,"response":{"currentLevel":"INFO"}}
```

2.1.7.2 Logger Settings :: Change Log level

URL	services.html
Description	It allows super admin to change the current log level to ERROR. Note : Default logger level is INFO , if you want change logger level , this API is used.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=log&formData:={'setLevel':'ERROR'}" http://192.168.2.156:8085/hi-ee/services.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	log	Service name as log
formData:	{"setLevel":"ERROR"}	Action to set the logger settings current log level to ERROR.

Response Output (JSON format)	{ "status":1, "response":{"message":"Log level is set to ERROR","currentLevel":"ERROR"} }
Description of Response output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with the current level of log as “ERROR”. Current level of the logger settings get changed to ERROR level. **currentLevel** : Selected log level.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:system
service:log
formData:{"setLevel":"ERROR"}
```

Status: 200 OK Time: 52 ms Size: 395 B

```
{"status":1,"response":{"message":"Log level is set to ERROR","currentLevel":"ERROR"}}
```

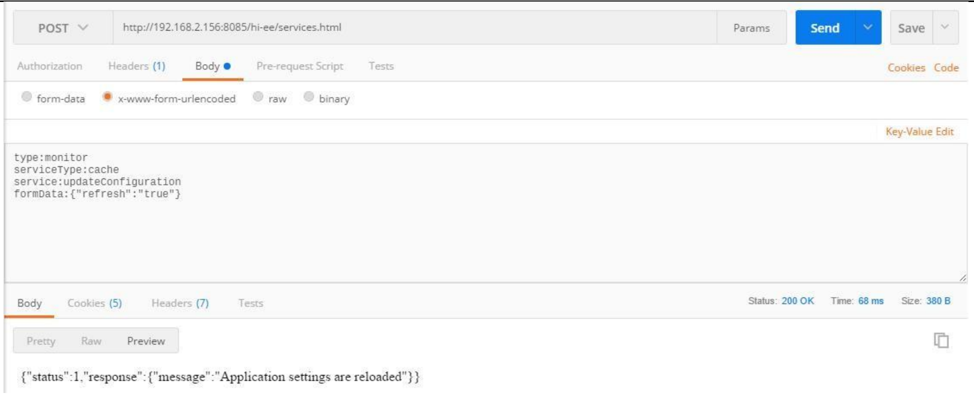
2.1.8 Reload Configurations

2.1.8.1 Application Configuration

URL	services.html	
Description	It allows super admin to reload the setting.xml changes(application settings) If you want to get updated changes in setting.xml related to application , you can just reload the using this API.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=updateConfiguration&formData:={'refresh':'true'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	updateConfiguration	Service name as updateConfiguration
formData:	{"refresh":true}	Action to set refresh as true to reload the setting.xml changes.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Application settings are reloaded"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message after application reload. Changes in setting.xml get reloaded.	
Service Status	200 OK	

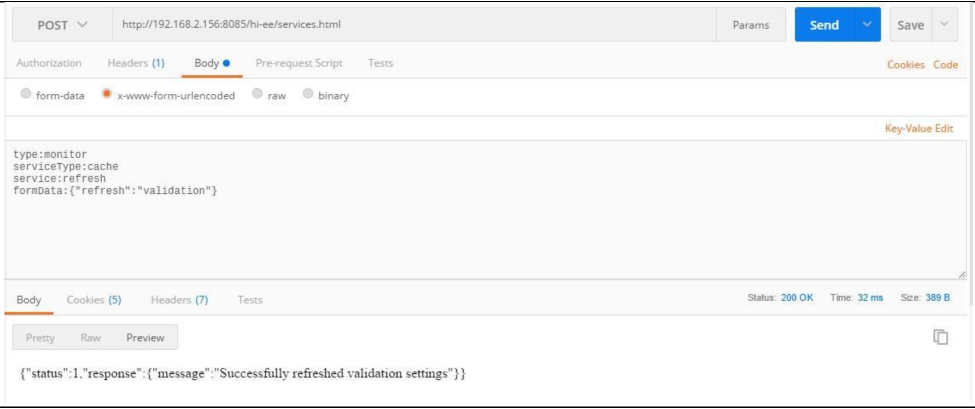
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html x-www-form-urlencoded
	`type:monitor`
	`serviceType:cache`
	`service:updateConfiguration`
	`formData:{"refresh":"true"}`
	Status: 200 OK Time: 68 ms Size: 380 B
	{"status":1,"response":{"message":"Application settings are reloaded"}}

2.1.8.2 Validation Configuration

URL	services.html	
Description	It allows super admin to reload the validation.xml changes(validation settings) If you want to get updated changes in validation.xml related to application , you can just reload the using this API.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=refresh&formData:={'refresh':'validation'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	refresh	Service name as refresh
formData:	{"refresh":"validation"}	Action to set refresh as validation to reload the validation.xml changes.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Successfully refreshed validation settings"}
	}

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message after validation reload. Changes in validation.xml get reloaded.
Service Status	200 OK

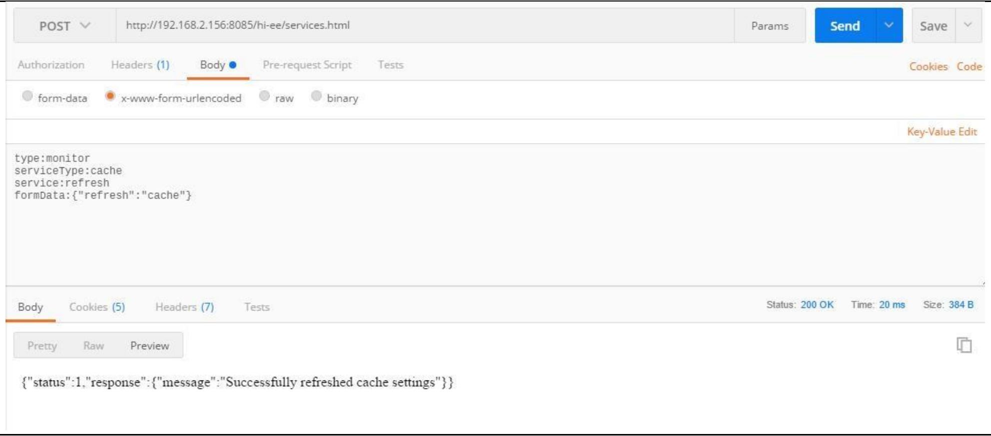
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:monitor`
	`serviceType:cache`
	`service:refresh`
	`formData:{"refresh":"validation"}`
	Status: 200 OK Time: 32 ms Size: 389 B
	{"status":1,"response":{"message":"Successfully refreshed validation settings"}}

2.1.8.3 Cache Configuration

URL	services.html	
Description	It allows super admin to reload the cache.xml changes(cache settings) If you want to get updated changes in cache.xml related to cache configuration, you can just reload the using this API.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=cache&service=refresh&formData:={'refresh':'cache'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	cache	serviceType as cache
service:	refresh	Service name as refresh
formData:	{"refresh":"cache"}	Action to set refresh as cache to reload the cache.xml changes.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Successfully refreshed cache settings"}
	}	

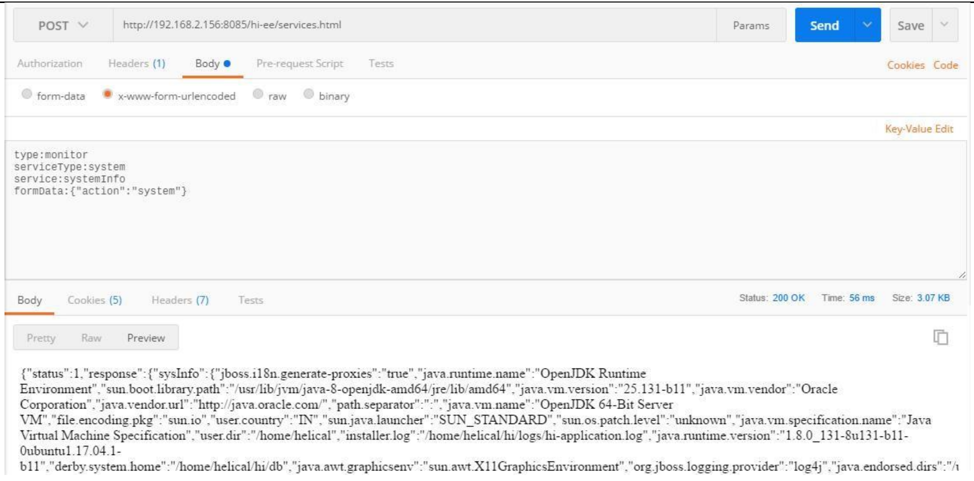
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message after cache reload. Changes in cache.xml get reloaded.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:monitor`
	`serviceType:cache`
	`service:refresh`
	`formData:{"refresh":"cache"}`
	Status: 200 OK Time: 20 ms Size: 384 B
	{"status":1,"response":{"message":"Successfully refreshed cache settings"}}

2.2 System

2.2.1 Refresh OS Details

URL	services.html	
Description	It allows super admin to refresh the OS details/informaion of the system. All OS details all the application get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=systemInfo&formData:={'action':'system'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	systemInfo	Service name as systemInfo
formData:	{"action":"system"}	Action to refresh the system / OS details.
Response Output (JSON format)	{"status":1,"response":{"sysInfo":{"jboss.i18n.generate-proxies":"true","java.runtime.name":"Java(TM) SE Runtime Environment","sun.boot.library.path":"C:\\Program Files\\Java\\jre1.8.0_92\\bin","java.vm.version":"25.92-b14","java.vm.vendor":"Oracle Corporation","java.vendor.url":"http://java.oracle.com/","path.separator":";","java.vm.name":"Java HotSpot(TM) 64-Bit Server VM","file.encoding.pkg":"sun.io","user.country":"US","user.script":"","sun.java.launcher":"SUN_STANDARD"}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. Response you get as the sysInfo array where all os details are returned after refresh like : **Java.runtime.name :** Name of the java runtime **sun.boot.library.path:** java library path **java.vm.vendor**: Java vender name. etc	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/services.html

```
type:monitor
serviceType:system
service:systemInfo
formData:{"action":"system"}
```

Status: 200 OK Time: 56 ms Size: 3.07 KB

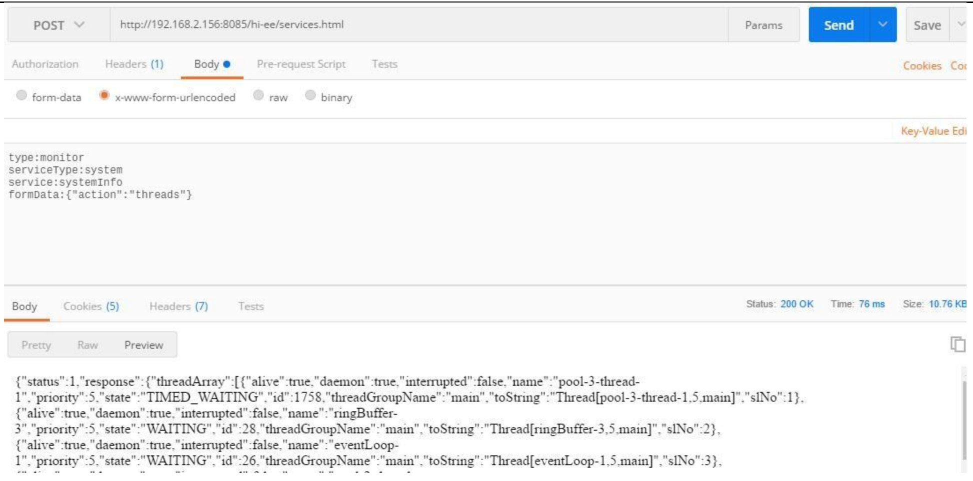
```
{"status":1,"response":{"sysInfo":{"jboss.i18n.generate-proxies":"true","java.runtime.name":"OpenJDK Runtime
Environment","sun.boot.library.path":"/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/amd64","java.vm.version":"25.131-b11","java.vm.vendor":"Oracle
Corporation","java.vendor.url":"http://java.oracle.com/","path.separator":":","java.vm.name":"OpenJDK 64-Bit Server
VM","file.encoding.pkg":"sun.io","user.country":"IN","sun.java.launcher":"SUN_STANDARD","sun.os.patch.level":"unknown","java.vm.specification.name":"Java
Virtual Machine Specification","user.dir":"/home/helical","installer.log":"/home/helical/hi/logs/hi-application.log","java.runtime.version":"1.8.0_131-8u131-b11-
0ubuntu1.17.04.1-
b11","derby.system.home":"/home/helical/hi/db","java.awt.graphicsenv":"sun.awt.X11GraphicsEnvironment","org.jboss.logging.provider":"log4j","java.endorsed.dirs":"/
```

2.2.2 Refresh JVM thread details

URL	services.html	
Description	It allows super admin to refresh the JVM thread details. We can refresh the JVM threads to check the thread details like its priority,state etc.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=systemInfo&formData:={'action':'threads'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	systemInfo	Service name as systemInfo
formData:	{"action":"threads"}	Action set as threads to refresh the JVM thread details.

Response Output (JSON format)	{
	"status":1,"response":{"threadArray":[{"alive":true,"daemon":true,"interrupted":false,"name":"pool-3-thread-1","priority":5,"state":"TIMED_WAITING","id":1758,"threadGroupName":"main","toString":"Thread[pool-3-thread-1,5,main]","slNo":1},{"alive":true,"daemon":true,"interrupted":false,"name":"ringBuffer-3","priority":5,"state":"WAITING","id":28,"threadGroupName":"main","toString":"Thread[ringBuffer-3,5,main]","slNo":2}]}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. Response you get as the threadArray where all jvm thread related details(alive status ,dameon is there are not,thread name,priority,state etc.) are returned after refresh. JVM thread details of the system get refreshed. **alive**: alive status of thread **daemon:** daemon status	

	interrupted : Interrupted status of thread **name** : Name of thread **priority** :Thread priority **state:** current state of thread etc.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:system
service:systemInfo
formData:{"action":"threads"}
```

Status: 200 OK Time: 76 ms Size: 10.76 KB

```
{"status":1,"response":{"threadArray":[{"alive":true,"daemon":true,"interrupted":false,"name":"pool-3-thread-
1","priority":5,"state":"TIMED_WAITING","id":1758,"threadGroupName":"main","toString":"Thread[pool-3-thread-1,5,main]","slNo":1},
{"alive":true,"daemon":true,"interrupted":false,"name":"ringBuffer-
3","priority":5,"state":"WAITING","id":28,"threadGroupName":"main","toString":"Thread[ringBuffer-3,5,main]","slNo":2},
{"alive":true,"daemon":true,"interrupted":false,"name":"eventLoop-
1","priority":5,"state":"WAITING","id":26,"threadGroupName":"main","toString":"Thread[eventLoop-1,5,main]","slNo":3},
```

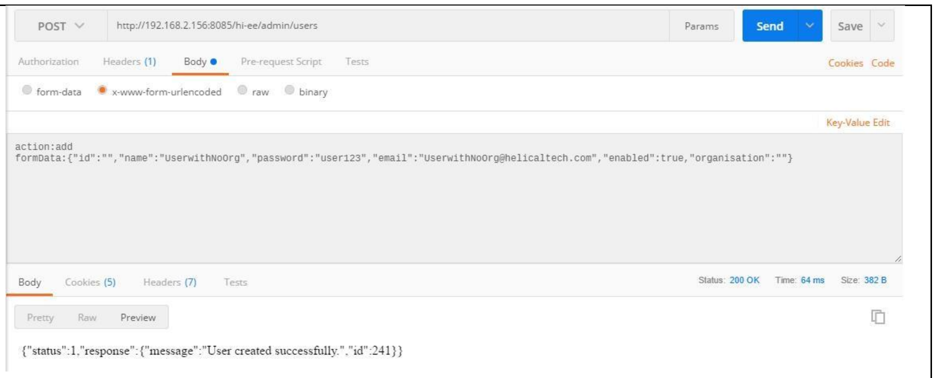
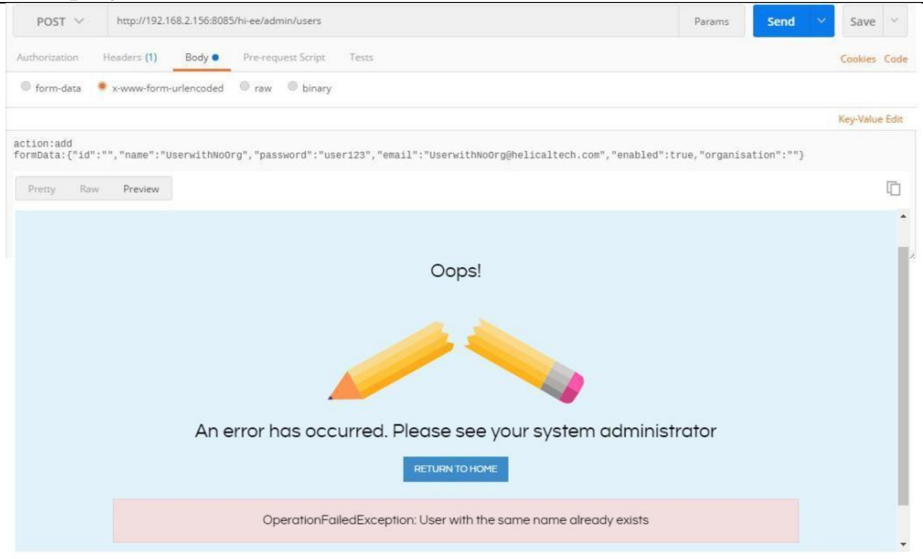
2.3 User Management

2.3.1 Add user under Super organization

URL	admin/users	
Description	It allows super admin to add user with Super organization. To add any user without assigning the oraganization to that user we can create user.Just keep organisation key as blank value	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'id':'','name':'UserwithNoOrg','password':'user123','email':'UserwithNoOrg@helicaltech.com','enabled':true,'organisation':''}" http://192.168.2.156:8085/hi-ee/admin/users -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	action as add
formData:	{"id":"","name":"UserwithNoOrg","password":"user123","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":""}	formData: contains the all details required to add user with Super organization.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"User created successfully.","id":9}
	}	

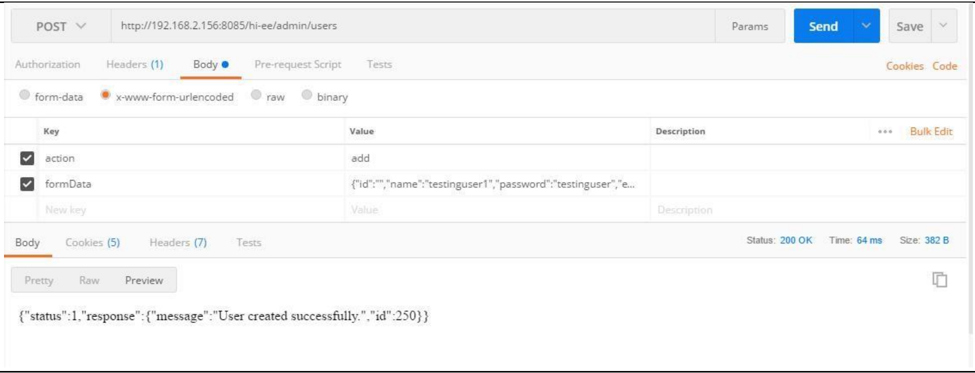
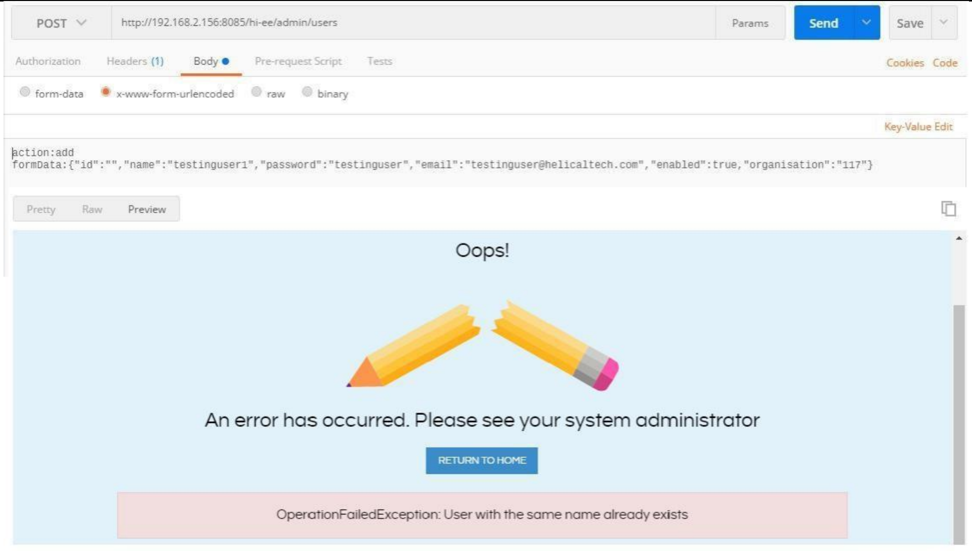
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id for the user. User get added under Super organization.	
Service Status	200 OK	

Screenshot(Success)	 POST `http://192.168.2.156:8085/hi-ee/admin/users` Body: x-www-form-urlencoded `action:add` `formData:{"id":"","name":"UserwithNoOrg","password":"user123","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":""}` Status: 200 OK Time: 64 ms Size: 382 B `{"status":1,"response":{"message":"User created successfully.","id":241}}`
Possible Error	*Note : If the user with same name already exist then an error "OperationFailedException: User with the same name already exists" is displayed*
Screenshot(Error)	 POST `http://192.168.2.156:8085/hi-ee/admin/users` Body: x-www-form-urlencoded `action:add` `formData:{"id":"","name":"UserwithNoOrg","password":"user123","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":""}` Oops! An error has occurred. Please see your system administrator RETURN TO HOME OperationFailedException: User with the same name already exists

2.3.2 Add user under organization

URL	admin/users	
Description	It allows super admin to create a new user for the respective organization. Moreover, • Superadmin can create user for any organization available in the list. • Organization admin can create user for its organization. • While passing organization name for particular user we need to pass the *Note : organization id which we get after creation of organization using POST Method. Refer Get OrganisationList*	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'id':'','name':'testinguser1','password':'testimguser','email':'testinguser@helicaltech.com','enabled':true,'organisation':'117'}" http://192.168.2.156:8085/hi-ee/admin/users -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add new user
formData:	{"id":"" "name":"testinguser1", "password":"testinguser", email":"testinguser@helicaltech.com" "enabled":true, "organisation":"117"}	JSON object containing user information like id which is auto generated , name of user,password ,email,enabled value as true and the organisation id which you want assign to user.

Response Output: (JSON format)	{
	  "status": 1,
	  "response": {
	    "message": "User created successfully.",
	    "id": 250
	  }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id for the user. The user is created successfully
Service Status	200 OK
Screenshot(success)	 POST http://192.168.2.156:8085/hi-ee/admin/users · Body: x-www-form-urlencoded · Key: action, Value: add · Key: formData, Value: `{"id":"","name":"testinguser1","password":"testinguser","e...` · Status: 200 OK Time: 64 ms Size: 382 B · `{"status":1,"response":{"message":"User created successfully.","id":250}}`
Possible Error	*Note: If the user name already exist then an error "OperationFailedException: User with the same name already exists" is displayed*
Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/users · Body: x-www-form-urlencoded · `action:add` `formData:{"id":"","name":"testinguser1","password":"testinguser","email":"testinguser@helicaltech.com","enabled":true,"organisation":"117"}` · Oops! An error has occurred. Please see your system administrator · RETURN TO HOME · OperationFailedException: User with the same name already exists

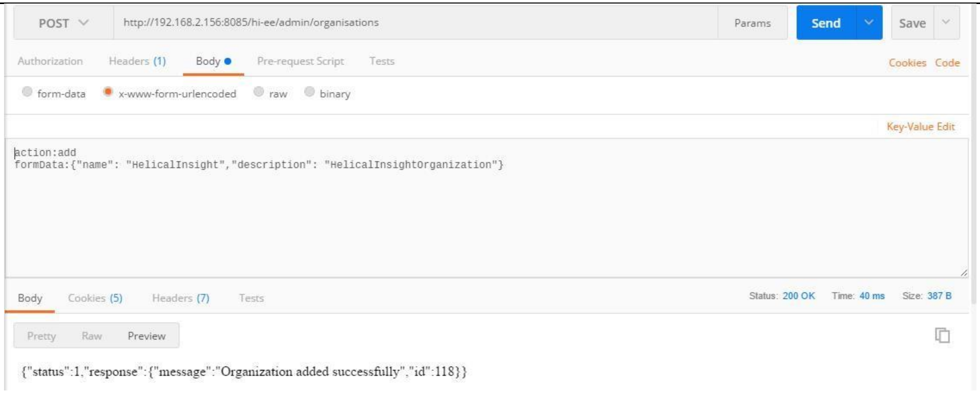
2.3.3 Add organization

URL	admin/organisations
Description	It allows super admin to add new organization in the existing list. While adding new organization we need to set the organization name and the description for organization.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.

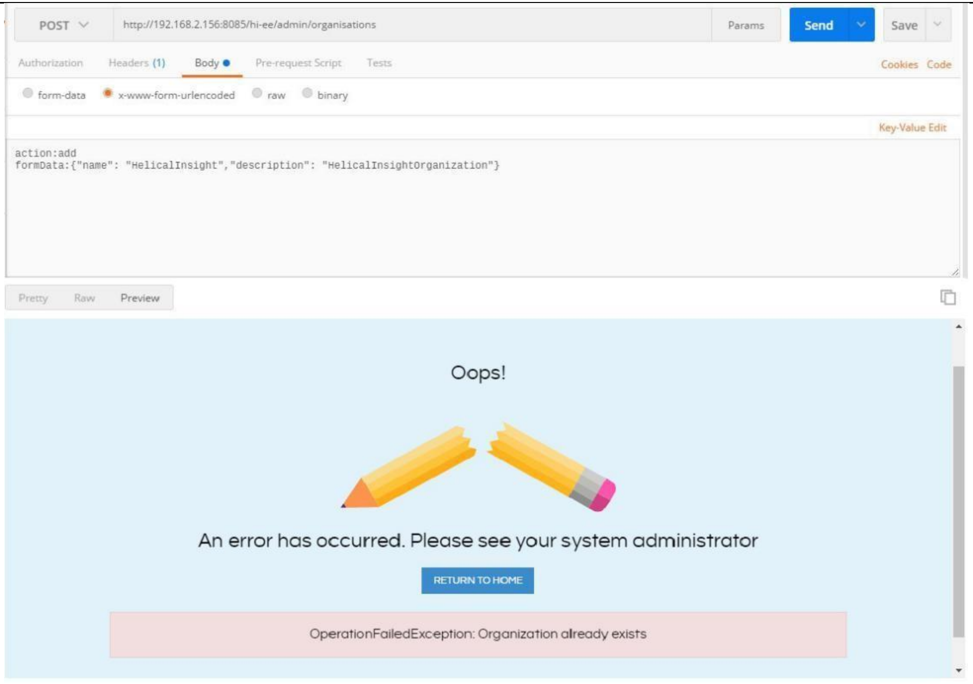
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'name':'HelicalInsight','description':'HelicalInsightOrganization'}" http://192.168.2.156:8085/hi-ee/admin/organisations -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add organization
formData:	{"name": "HelicalInsight", "description": "HelicalInsightOrganization"}	JSON object containing organization information

Response Output (JSON format)	{
	    "status": 1,
	    "Response": {
	        "message": "Organization added successfully",
	        "id": 118
	    }
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id to organization. . Organization is created and stored in database.By default ROLE_ADMIN and ROLE_USER are created under this organization automatically.	
Service Status	200 OK	

Screenshot(success)	 POST http://192.168.2.156:8085/hi-ee/admin/organisations
	`action:add`
	`formData:{"name": "HelicalInsight","description": "HelicalInsightOrganization"}`
	Status: 200 OK Time: 40 ms Size: 387 B
	{"status":1,"response":{"message":"Organization added successfully","id":118}}

Possible Error	*Note: If the organization name already exist then an error "OperationFailedException: Organization already exists" is displayed*

Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/organisations
	`action:add`
	`formData:{"name": "HelicalInsight","description": "HelicalInsightOrganization"}`
	Oops!
	An error has occurred. Please see your system administrator
	RETURN TO HOME
	OperationFailedException: Organization already exists

Post-action	Organization details modification / deletion and assign created organization to particular user.

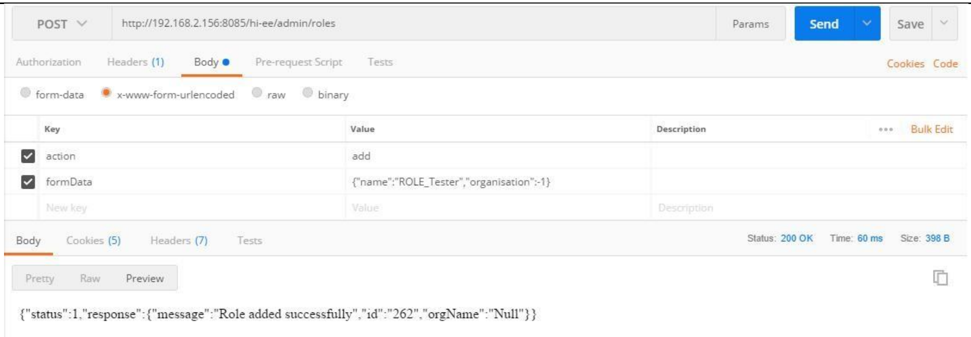
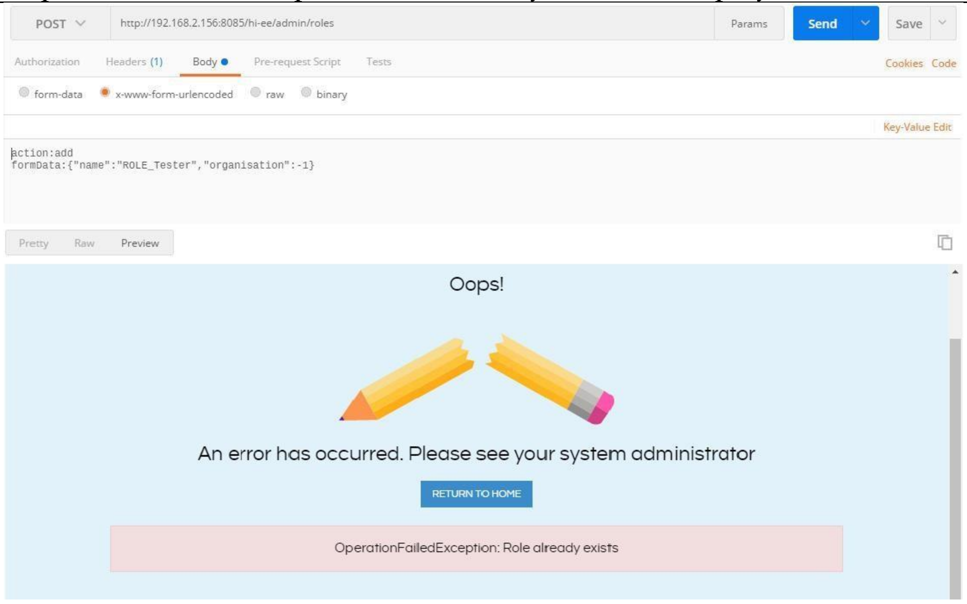
2.3.4 Add role under Super organization

URL	admin/roles
Description	It allows super admin to add role to an User. Moreover, • Superadmin/organization admin has the authority to add role for any user. • At user level, user admin has the authority to add roles for that respective user. *Note: At a time single role can be added and to add role under Super organization "-1" value is used as organisation value.*

Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/roles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'name':'ROLE_Tester','organisation':'-1'}" http://192.168.2.156:8085/hi-ee/admin/roles -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add roll
formData:	{"name":"ROLE_Tester","organisation":-1}	JSON object containing role information which get added in Super organization. *Note: At a time single role can be added and to add role under Super organization "-1" value is used as organisation value.*

Response Output (JSON format)	{
	"status":1,
	"response":
	{"message":"Role added successfully",
	"id":"11",
	"orgName":"Null"
	}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned **id** for created role with **Super organization**. New role is added to the Super organization and is saved in database.	
Service Status	200 OK	

Screenshot(success)	 POST http://192.168.2.156:8085/hi-ee/admin/roles x-www-form-urlencoded Key: action, Value: add Key: formData, Value: {"name":"ROLE_Tester","organisation":-1} Status: 200 OK Time: 60 ms Size: 398 B {"status":1,"response":{"message":"Role added successfully","id":"262","orgName":"Null"}}
Possible Error	*Note: If the role name already exist then an error "OperationFailedException: Role already exists" is displayed*
Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/roles x-www-form-urlencoded action:add formData:{"name":"ROLE_Tester","organisation":-1} Oops! An error has occurred. Please see your system administrator RETURN TO HOME OperationFailedException: Role already exists
Post-action	Can modify/delete role details and assign created role to particular user.

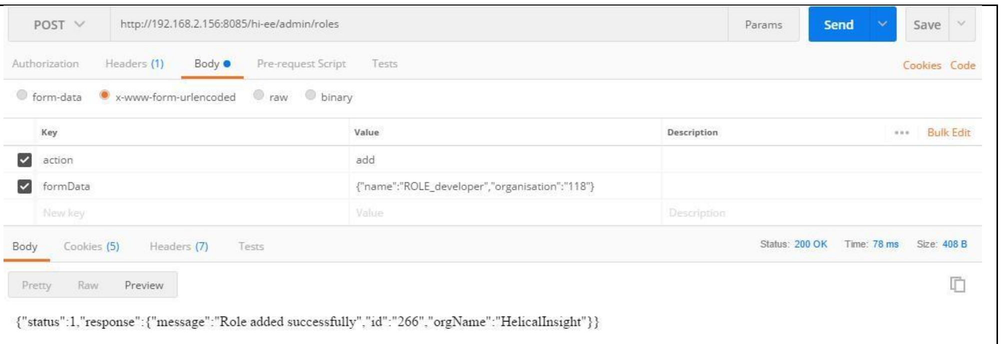
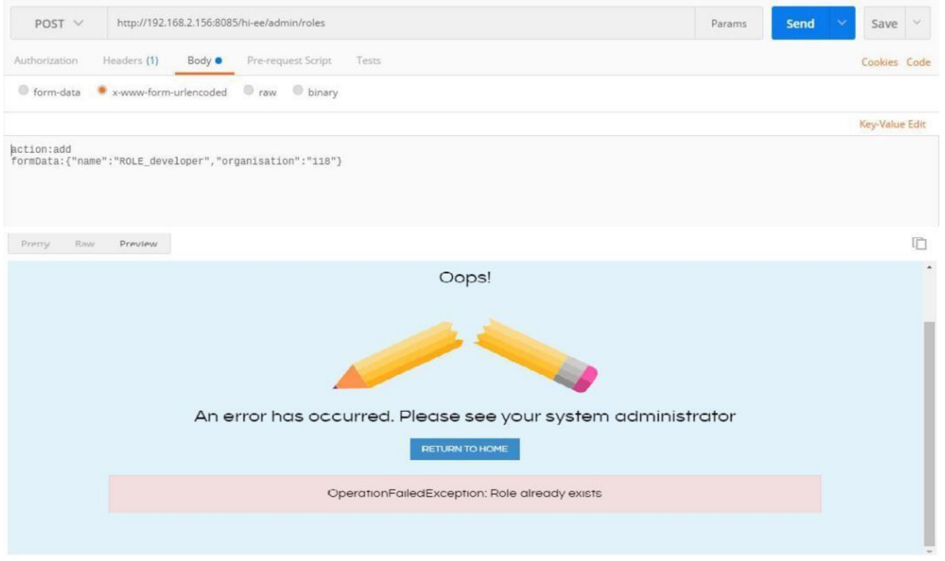
2.3.5 Add role under organization

URL	admin/roles
Description	It allows super admin to add role to an User. Moreover, • Superadmin/organization admin has the authority to add role for any user. • At user level, user admin has the authority to add roles for that respective user. • While passing organization name for particular role we need to pass the *Note : organization id which we get after creation of organization using POST Method. Refer Get OrganisationList .At a time single role can be added.*

Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/roles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'name':'ROLE_developer','organisation':'118'}" http://192.168.2.156:8085/hi-ee/admin/roles -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add roll
formData:	{"name": "ROLE_developer","organization": "118"}	JSON object containing role information. name : name of the ROLE. organization : id of the organization

Response Output (JSON format)	{
	    "status": 1,
	    "Response": {"message": "Role added successfully",
	        "id": "266",
	        "orgName": "HelicalInsight",
	    }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id for created role with assigned organization. New role is added to the respective organization and is saved in database.	
Service Status	200 OK	

Screenshot(Success)	 POST http://192.168.2.156:8085/hi-ee/admin/roles Body: x-www-form-urlencoded; action: add; formData: {"name":"ROLE_developer","organisation":"118"} Status: 200 OK Time: 78 ms Size: 408 B `{"status":1,"response":{"message":"Role added successfully","id":"266","orgName":"HelicalInsight"}}`
Possible Error	*Note: If the role name already exist then an error "OperationFailedException: Role already exists" is displayed*
Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/roles Body: x-www-form-urlencoded `action:add` `formData:{"name":"ROLE_developer","organisation":"118"}` Oops! An error has occurred. Please see your system administrator RETURN TO HOME OperationFailedException: Role already exists
Post-action	Can modify/delete role details and assign created role to particular user.

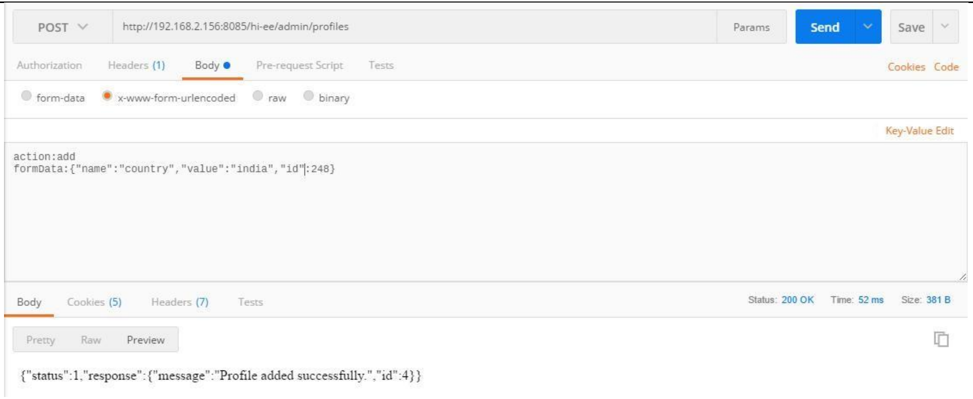
2.3.6 Add profile for selected user under Super organization

URL	admin/profiles	
Description	It allows super admin to add user's profile from the existing list. Moreover, • Superadmin can add all user's profile belongs to existing list of organization. • Organization admin can add user's profile belong to their respective organization. *Note: To add profile under Super organization for particular user you need to set the id which is assigned for selected user which you will get as response with api* ***Refer UserList API***	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/profiles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'name':'country','value':'india','id':'248'}" http://192.168.2.156:8085/hi-ee/admin/profiles -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add new user profile
formData:	{"name":"country","value":"india","id":248}	JSON object containing profile information name : Name of the profile value : Value of the profile id : id of the user having Super organisation

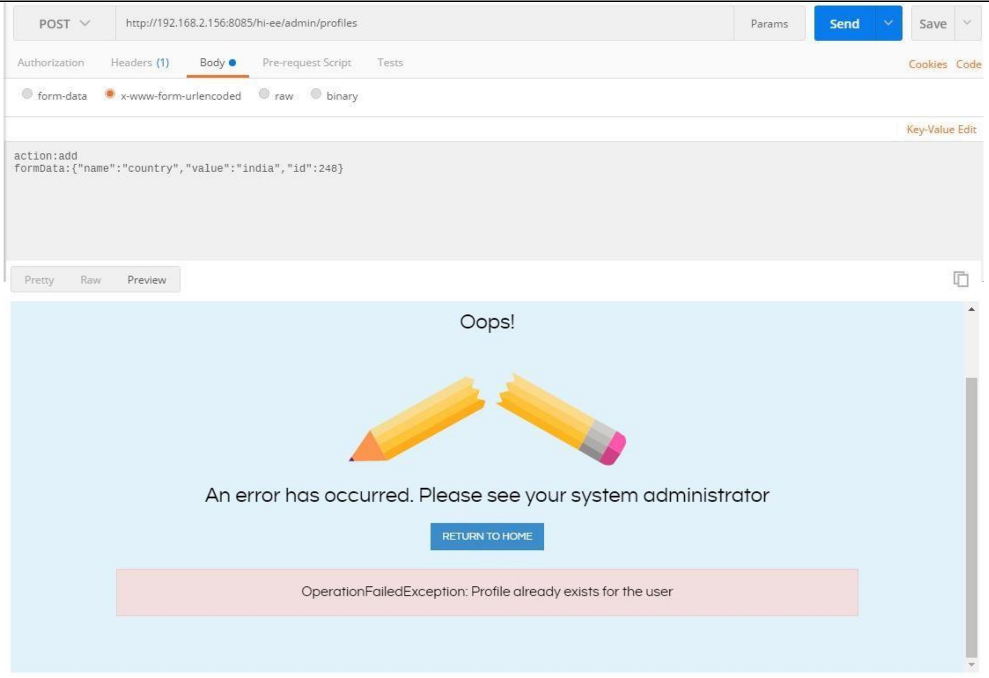
Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Profile added successfully.","id":4}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id for created profile.	

	A new profile is added with the respective user and the values are stored in database.
Service Status	200 OK

Screenshot(Success)	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	Body: x-www-form-urlencoded
	`action:add`
	`formData:{"name":"country","value":"india","id":248}`
	Status: 200 OK Time: 52 ms Size: 381 B
	`{"status":1,"response":{"message":"Profile added successfully.","id":4}}` Params
	Send
	Save
	Authorization
	Headers (1)
	Body
	Pre-request Script
	Tests
	Cookies
	Code
	form-data
	x-www-form-urlencoded
	raw
	binary
	Key-Value Edit
	Body
	Cookies (5)
	Headers (7)
	Tests
	Pretty
	Raw
	Preview

Possible Error	*Note: If the profile name already exist for the user then an error "OperationFailedException: Profile already exists for the user" is displayed*

Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	Body: x-www-form-urlencoded
	`action:add`
	`formData:{"name":"country","value":"india","id":248}`
	Oops!
	An error has occurred. Please see your system administrator
	OperationFailedException: Profile already exists for the user Params
	Send
	Save
	Authorization
	Headers (1)
	Body
	Pre-request Script
	Tests
	Cookies
	Code
	form-data
	x-www-form-urlencoded
	raw
	binary
	Key-Value Edit
	Pretty
	Raw
	Preview
	RETURN TO HOME

Post-action	User profile modification (edit / delete)

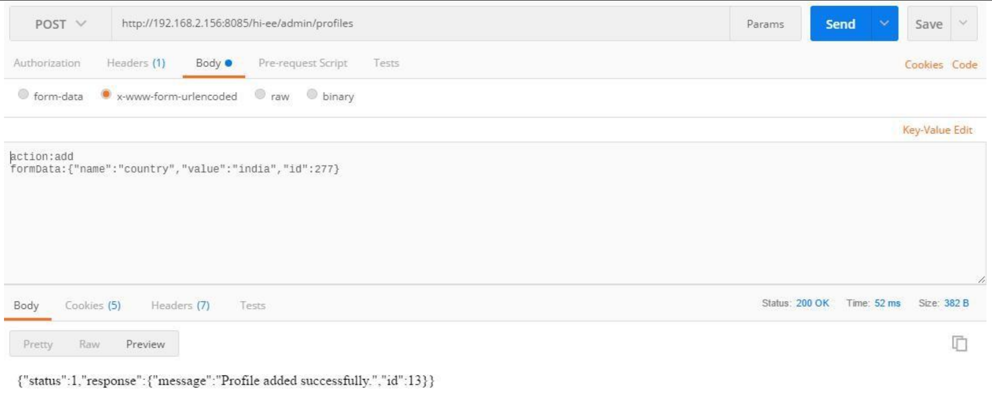
2.3.7 Add profile for selected user under organization

URL	admin/profiles	
Description	It allows super admin to add user's profile from the existing list. Moreover, • Superadmin can add all user's profile belongs to existing list of organization. • Organization admin can add user's profile belong to their respective organization. *Note: To add profile under Super organization for particular user you need to set the id which is assigned for selected user which you will get as response with api* ***Refer UserList API***	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/profiles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&formData:={'name':'country','value':'india','id':'277'}" http://192.168.2.156:8085/hi-ee/admin/profiles -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	add	Operation to add new user profile
formData:	{"name":"country", "value":"india", id":277}	JSON object containing profile information name : Name of the profile value : Value of the profile id : id of the selected user with organisation

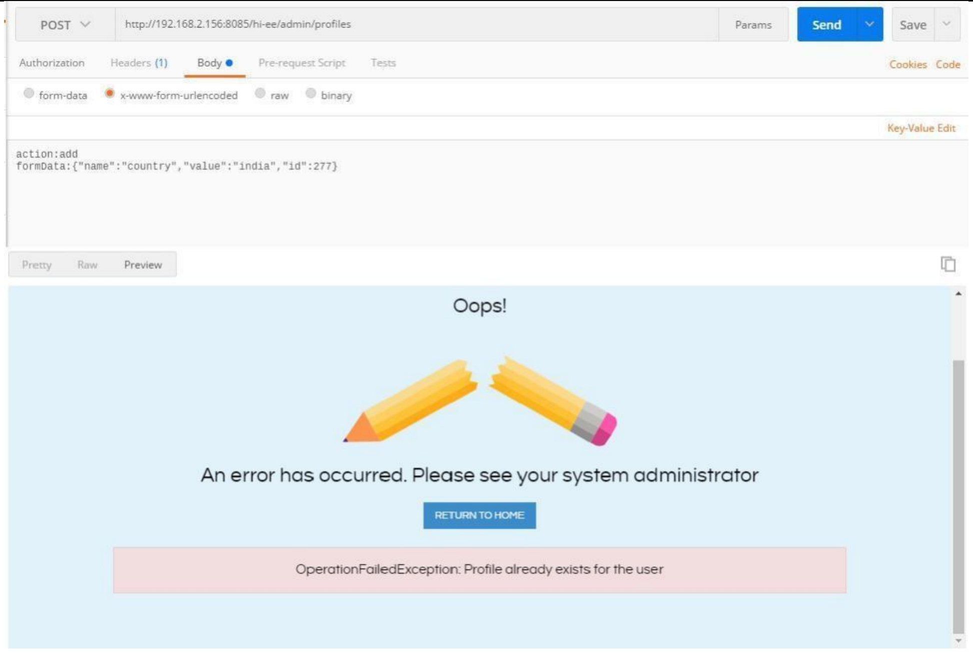
Response Output (JSON format)	{
	"status": 1,
	"response": {
	"message": "Profile added successfully.",
	"id": 13
	}
	}	

Description of Response output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message with assigned id for created profile.	

	A new profile is added with the respective user and organisation , the values are stored in database.
Service Status	200 OK

Screenshot(success)	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	Body: x-www-form-urlencoded
	`action:add`
	`formData:{"name":"country","value":"india","id":277}`
	Status: 200 OK Time: 52 ms Size: 382 B
	{"status":1,"response":{"message":"Profile added successfully.","id":13}}

Possible Error	*Note: If the profile name already exist for the user then an error "OperationFailedException: Profile already exists for the user" is displayed*

Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	Body: x-www-form-urlencoded
	`action:add`
	`formData:{"name":"country","value":"india","id":277}`
	Oops!
	An error has occurred. Please see your system administrator
	RETURN TO HOME
	OperationFailedException: Profile already exists for the user

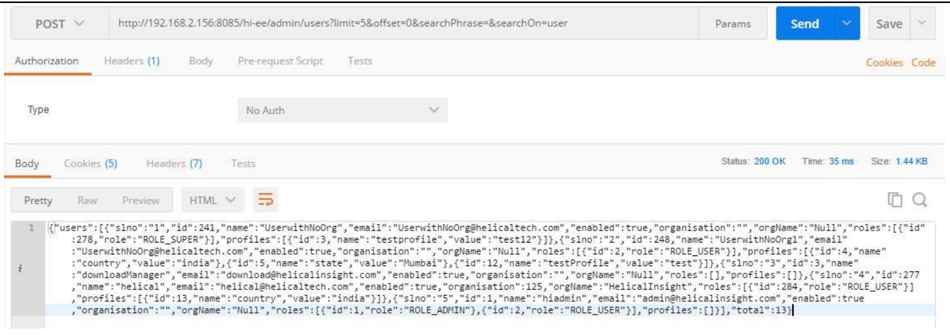
Post-action	User profile modification (edit / delete)

2.3.8 Get /Refresh/Pagination(Next,Prev) User-list

Note : To Get userlist/Refresh UserList/Pagination-Next/Prev we are using same API service.

URL	admin/users?limit=5&offset=0&searchPhrase=&searchOn=user
Description	It allows to shows/refresh the list of existing users and user details.
Pre-requisite	User should have logged in before accessing the service.[Refer login

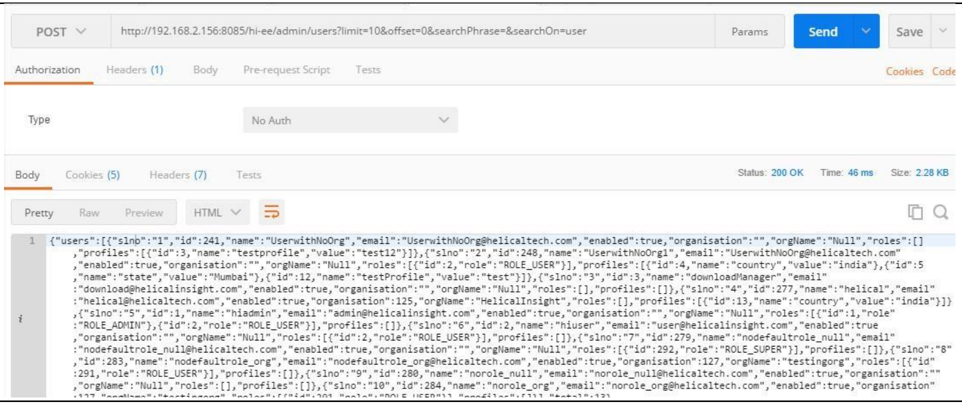
	module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET , POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?limit=5&offset=0&searchPhrase=&searchOn=user **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&offset=0&searchPhrase=&searchOn=user " http://192.168.2.156:8085/hi-ee/admin/users	
HTTP Request Key	**HTTP Request Value**	**Description**
v **(optional)**	1440151442591	Preferably timestamp to identify url uniquely
limit**(optional)**	5	Set the number of records
offset**(optional)**	0	Sets the starting record
searchPhrase **(optional)**		Search for a type phrase from the list Note : If you want to search for all users then keep it as blank.
searchOn **(optional)**	user	Search list by user name. Similarly by organization/ email/ roles
Response Output (JSON format)	{"users":[{"slno":"1","id":241,"name":"UserwithNoOrg","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":278,"role":"ROLE_SUPER"}],"profiles":[{"id":3,"name":"testprofile","value":"test12"}]},{"slno":"2","id":248,"name":"UserwithNoOrg1","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[{"id":4,"name":"country","value":"india"},{"id":5,"name":"state","value":"Mumbai"},{"id":12,"name":"testProfile","value":"test"}]},{"slno":"3","id":3,"name":"downloadManager","email":"download@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[],"profiles":[]},{"slno":"4","id":277,"name":"helical","email":"helical@helicaltech.com","enabled":true,"organisation":125,"orgName":"Helical Insight","roles":[{"id":284,"role":"ROLE_USER"}],"profiles":[{"id":13,"name":"country","value":"india"}]},{"slno":"5","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role":"ROLE_ADMIN"},{"id":2,"role":"	

	ROLE_USER"}],"profiles":[]}],"total":13}
Description of Response Output:	users json array which includes users information such as: **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc details get returned as response.
Service Status	200 OK
Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/users?limit=5&offset=0&searchPhrase=&searchOn=user Params Send Save Authorization Headers (1) Body Pre-request Script Tests Cookies Code Type No Auth Body Cookies (5) Headers (7) Tests Status: 200 OK Time: 35 ms Size: 1.44 KB Pretty Raw Preview HTML `{"users":[{"slno":"1","id":241,"name":"UserwithNoOrg","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":278,"role":"ROLE_SUPER"}],"profiles":[{"id":3,"name":"testprofile","value":"test12"}]},{"slno":"2","id":248,"name":"UserwithNoOrg1","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[{"id":4,"name":"country","value":"india"},{"id":5,"name":"state","value":"Mumbai"},{"id":12,"name":"testProfile","value":"test"}]},{"slno":"3","id":3,"name":"downloadManager","email":"download@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[],"profiles":[]},{"slno":"4","id":277,"name":"helical","email":"helical@helicaltech.com","enabled":true,"organisation":125,"orgName":"HelicalInsight","roles":[{"id":284,"role":"ROLE_USER"}],"profiles":[{"id":13,"name":"country","value":"india"}]},{"slno":"5","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role":"ROLE_ADMIN"},{"id":2,"role":"ROLE_USER"}],"profiles":[]}],"total":13}`
Post-action	Displays user details pagewise, all records, add/edit user details.

2.3.9 Show number of User entries

URL	admin/users?limit=10&offset=0&searchPhrase=&searchOn=user
Description	It shows the requested number of existing users list with user details.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**GET , POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?limit=10&offset=0&searchPhrase=&searchOn=user **Access through Curl command :** curl --data

	"j_username=hiadmin&j_password=hiadmin&limit=10&offset=0&searchPhrase=&searchOn=user" http://192.168.2.156:8085/hi-ee/admin/users	
HTTP Request Key	**HTTP Request Value**	**Description**
v **(optional)**	1440151442591	Preferably timestamp to identify url uniquely
limit**(optional)**	10	Set the number of records
offset**(optional)**	0	Sets the starting record
searchPhrase **(optional)**		Search for a type phrase from the list
searchOn **(optional)**	user	Search list by user name. Similarly by organization/ email/ roles
Response Output (JSON format)	{"users":[{"slno":"1","id":241,"name":"UserwithNoOrg","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":278,"role":"ROLE_SUPER"}],"profiles":[{"id":3,"name":"testprofile","value":"test12"}]},{"slno":"2","id":248,"name":"UserwithNoOrg1","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[{"id":4,"name":"country","value":"india"},{"id":5,"name":"state","value":"Mumbai"},{"id":12,"name":"testProfile","value":"test"}]},{"slno":"3","id":3,"name":"downloadManager","email":"download@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[],"profiles":[]},{"slno":"4","id":277,"name":"helical","email":"helical@helicaltech.com","enabled":true,"organisation":125,"orgName":"Helical Insight","roles":[{"id":284,"role":"ROLE_USER"}],"profiles":[{"id":13,"name":"country","value":"india"}]},{"slno":"5","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role":"ROLE_ADMIN"},{"id":2,"role":"ROLE_USER"}],"profiles":[]}],"total":13}	
Description of Response Output:	"users" is json array which includes users information such as: **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc shows total 10 user entries as response.	
Service Status	200 OK	

Screenshot	
Post-action	Displays user details pagewise, all records, add/edit user details.

POST | http://192.168.2.156:8085/hi-ee/admin/users?limit=10&offset=0&searchPhrase=&searchOn=user

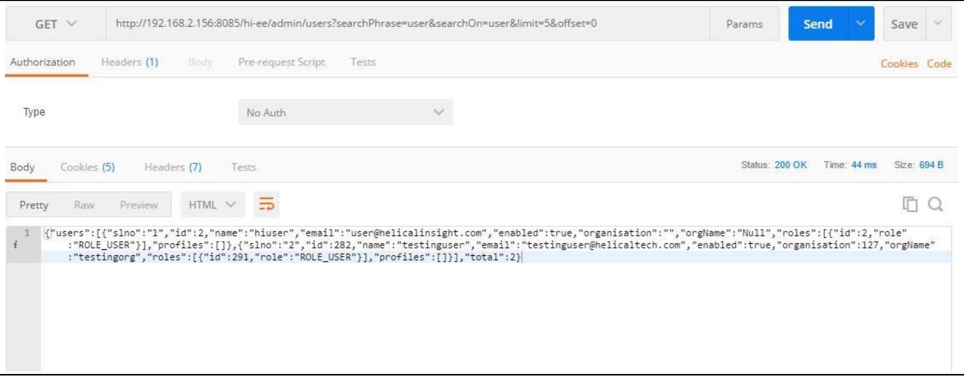
Type: No Auth

Status: 200 OK Time: 46 ms Size: 2.28 KB

```
{"users":[{"slnp":"1","id":241,"name":"UserwithNoOrg","email":"UserwithNoOrg@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[]
    ,"profiles":[{"id":3,"name":"testprofile","value":"test12"}]},{"slno":"2","id":248,"name":"UserwithNoOrg1","email":"UserwithNoOrg@helicaltech.com"
    ,"enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[{"id":4,"name":"country","value":"india"},{"id":5
    ,"name":"state","value":"Mumbai"},{"id":12,"name":"testProfile","value":"test"}]},{"slno":"3","id":3,"name":"downloadManager","email"
    :"download@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[],"profiles":[]},{"slno":"4","id":277,"name":"helical","email"
    :"helical@helicaltech.com","enabled":true,"organisation":125,"orgName":"HelicalInsight","roles":[],"profiles":[{"id":13,"name":"country","value":"india"}]}
    ,{"slno":"5","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role"
    :"ROLE_ADMIN"},{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"6","id":2,"name":"hiuser","email":"user@helicalinsight.com","enabled":true
    ,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"7","id":279,"name":"nodefaultrole_null","email"
    :"nodefaultrole_null@helicaltech.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":292,"role":"ROLE_SUPER"}],"profiles":[]},{"slno":"8"
    ,"id":283,"name":"nodefaultrole_org","email":"nodefaultrole_org@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id"
    :291,"role":"ROLE_USER"}],"profiles":[]},{"slno":"9","id":280,"name":"norole_null","email":"norole_null@helicaltech.com","enabled":true,"organisation":""
    ,"orgName":"Null","roles":[],"profiles":[]},{"slno":"10","id":284,"name":"norole_org","email":"norole_org@helicaltech.com","enabled":true,"organisation"
```

2.3.10 Search based on User

URL	admin/users?searchPhrase=user&searchOn=user&limit=5&offset=0	
Description	It filters the user list by user-name. Here we are checking for the user having keyword 'user'	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET , POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=user&searchOn=user&limit=5&offset=0 **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&searchPhrase=user&searchOn=user&limit=5&offset=0" http://192.168.2.156:8085/hi-ee/admin/users	
HTTP Request Key	**HTTP Request Value**	**Description**
searchPhrase	user	Search for a type phrase from the list
searchOn	user	Search list by user name.
limit**(optional)**	5	Set the number of records
offset **(optional)**	0	Sets the starting record
Response		

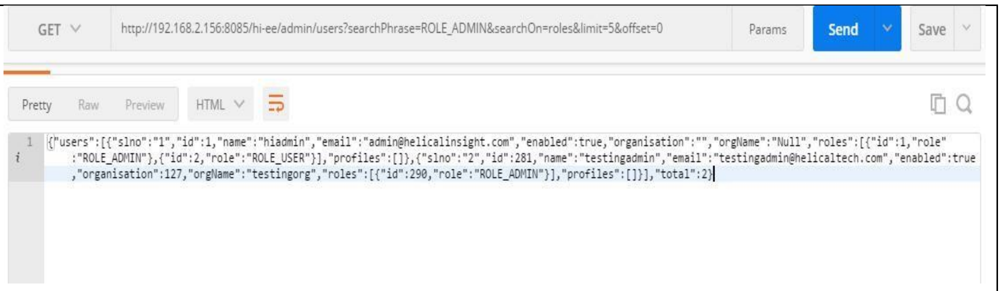
Output: **(JSON format)**	{"users":[{"slno":"1","id":2,"name":"hiuser","email":"user@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":282,"name":"testinguser","email":"testinguser@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id":291,"role":"ROLE_USER"}],"profiles":[]}],"total":2}
Description of Repsonse Output:	The response will get as the "users" is a json array with respective search criteria having all details related to search keyword for user. **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc
Service Status	200 OK
Screenshot	 GET http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=user&searchOn=user&limit=5&offset=0 Type: No Auth Status: 200 OK Time: 44 ms Size: 694 B {"users":[{"slno":"1","id":2,"name":"hiuser","email":"user@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":282,"name":"testinguser","email":"testinguser@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id":291,"role":"ROLE_USER"}],"profiles":[]}],"total":2}

Possible Errors :	When the search criteria do not match then will get response as :
	
	{
	   "users": [],
	   "total": 0
	}
	"users" array is blank

2.3.11 Search based on Roles

URL	admin/users?searchPhrase=ROLE_ADMIN&searchOn=roles&limit=5&offset=0
Description	User can filter the user list role wise. Here we are checking for the user having ROLE as 'ROLE_ADMIN'

Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTPP Method	**GET , POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=ROLE_ADMIN&searchOn=roles&limit=5&offset=0 **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&searchPhrase=ROLE_ADMIN&searchOn=roles&limit=5&offset=0" http://192.168.2.156:8085/hi-ee/admin/users	
Parameters	**Values**	**Parameters**
searchPhrase	ROLE_ADMIN	Search for a type phrase from the list
searchOn	roles	Search list by Roles.
limit**(optional)**	5	Set the number of records
offset **(optional)**	0	Sets the starting record
Response Output (JSON format)	{ "users":[{"slno":"1","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role":"ROLE_ADMIN"},{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":281,"name":"testingadmin","email":"testingadmin@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id":290,"role":"ROLE_ADMIN"}],"profiles":[]}],"total":2 }	
Description of the Response Output:	The response returned as the “users” array with respective all user details having ROLE as ROLE_ADMIN. We will get user array list according to search ROLE as follow: **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc	
Service Status	200 OK	

Screenshot	 GET `http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=ROLE_ADMIN&searchOn=roles&limit=5&offset=0` Params Send Save Pretty Raw Preview HTML `{"users":[{"slno":"1","id":1,"name":"hiadmin","email":"admin@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":1,"role":"ROLE_ADMIN"},{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":281,"name":"testingadmin","email":"testingadmin@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id":290,"role":"ROLE_ADMIN"}],"profiles":[]}],"total":2}`

Possible Errors	When the search criteria do not match then will get response as :
	{
	"users": [],
	"total": 0
	}
	“users” array is blank

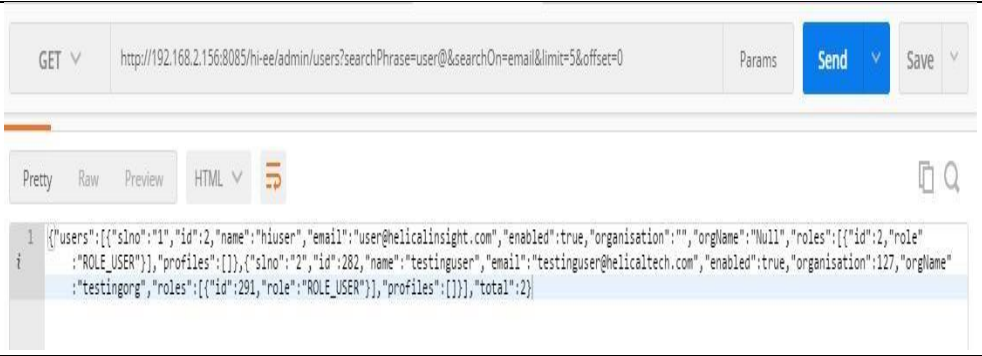
2.3.12 Search based on Email

URL	admin/users?searchPhrase=user@&searchOn=email&limit=5&offset=0
Description	User list get filter by user e-mail address. We need to pass the mail is as searchPhrase to filter userlist by email address.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**GET , POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=user@&searchOn=email&limit=5&offset=0 **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&searchPhrase=user@&searchOn=email&limit=5&offset=0" http://192.168.2.156:8085/hi-ee/admin/users

Parameters	**Values**	**Parameters**
searchPhrase	user@	Search for a type phrase from the list Which will check for mail id as user@
searchOn	email	Search list by email.

limit**(optional)**	5	Set the number of records
offset **(optional)**	0	Sets the starting record

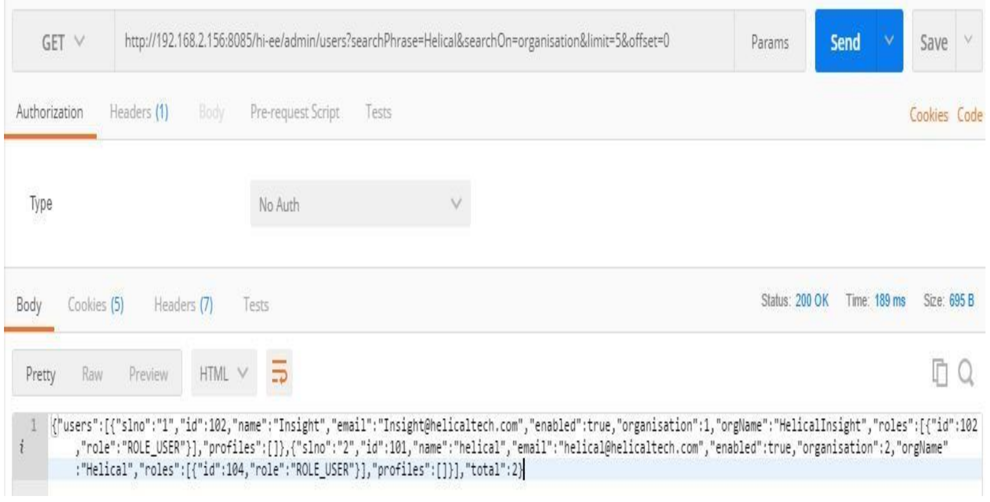
Response Output (JSON format)	{
	  "users": {
	    "slno": "1",
	    "id": 2,
	    "name": "hiuser",
	    "email": "user@helicalinsight.com",
	    "enabled": true,
	    "organisation": "",
	    "orgName": "Null",
	    "roles": [
	      {
	        "id": 2,
	        "role": "ROLE_USER"
	      }
	    ],
	    "profiles": []
	  },
	  "total": 1
	}	

Description of Response Output	The response will get as the user array with user details having the email address like ‘user@’. User details are : **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc	
Service Status	200 OK	
Screenshot	 GET http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=user@&searchOn=email&limit=5&offset=0 {"users":[{"slno":"1","id":2,"name":"hiuser","email":"user@helicalinsight.com","enabled":true,"organisation":"","orgName":"Null","roles":[{"id":2,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":282,"name":"testinguser","email":"testinguser@helicaltech.com","enabled":true,"organisation":127,"orgName":"testingorg","roles":[{"id":291,"role":"ROLE_USER"}],"profiles":[]}],"total":2}	
Possible Errors	When the search criteria do not match then will get the following response: {	

	"users": [], "total": 0 } “users” array is blank

2.3.13 Search based on Organization

URL	admin/users?searchPhrase=Helical&searchOn=organisation&limit=5&offset=0	
Description	It filters the User list organization wise.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET , POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=Helical&searchOn=organisation&limit=5&offset=0 **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&searchPhrase=Helical&searchOn=organisation&limit=5&offset=0 "http://192.168.2.156:8085/hi-ee/admin/users	
Parameters	**Values**	**Parameters**
searchPhrase	Helical	Search for a type phrase from the list
searchOn	organisation	Search list by organization.
limit**(optional)**	5	Set the number of records
offset **(optional)**	0	Sets the starting record
Response Output (JSON format)	{"users":[{"slno":"1","id":102,"name":"Insight","email":"Insight@helicaltech.com","enabled":true,"organisation":1,"orgName":"HelicalInsight","roles":[{"id":102,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":101,"name":"helical","email":"helical@helicaltech.com","enabled":true,"organisation":2,"orgName":"Helical","roles":[{"id":104,"role":"ROLE_USER"}],"profiles":[]}],"total":2}	
Description of	The response will get as the user array with user details having the	

Response Output:	organization as ‘Helical’ shown below : **slno:** serial number **id :** Id of the user **name:** Name of the user **email:** Email of user **enabled:** Enable status of the user **organization:** Name of organisation **roles:** Assigned roles for the user **profile:** Available profiles for the user etc
Service Status	200 OK
Screenshot	 GET http://192.168.2.156:8085/hi-ee/admin/users?searchPhrase=Helical&searchOn=organisation&limit=5&offset=0 Type: No Auth Status: 200 OK Time: 189 ms Size: 695 B `{"users":[{"slno":"1","id":102,"name":"Insight","email":"Insight@helicaltech.com","enabled":true,"organisation":1,"orgName":"HelicalInsight","roles":[{"id":102,"role":"ROLE_USER"}],"profiles":[]},{"slno":"2","id":101,"name":"helical","email":"helical@helicaltech.com","enabled":true,"organisation":2,"orgName":"Helical","roles":[{"id":104,"role":"ROLE_USER"}],"profiles":[]}],"total":2}`

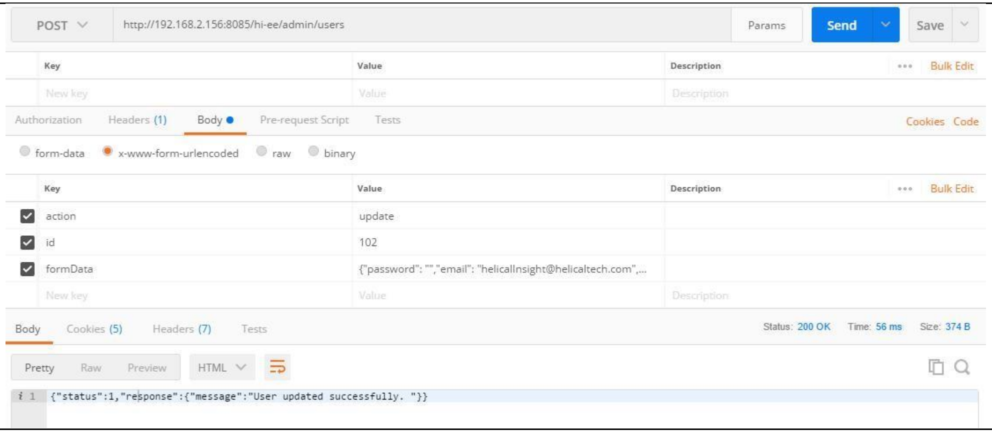
Possible Errors	When the search criteria do not match then you will get response as:
	{
	"users": [],
	"total": 0
	}
	“users” array is blank

2.3.14 Edit

2.3.14.1 Edit User

URL	admin/users
Description	It allows to update existing user details where as • Superadmin can update all the user details belongs to multiple organization. • Organization admin can update their respective organization user details. • Password , email ,enabled status can be editable.

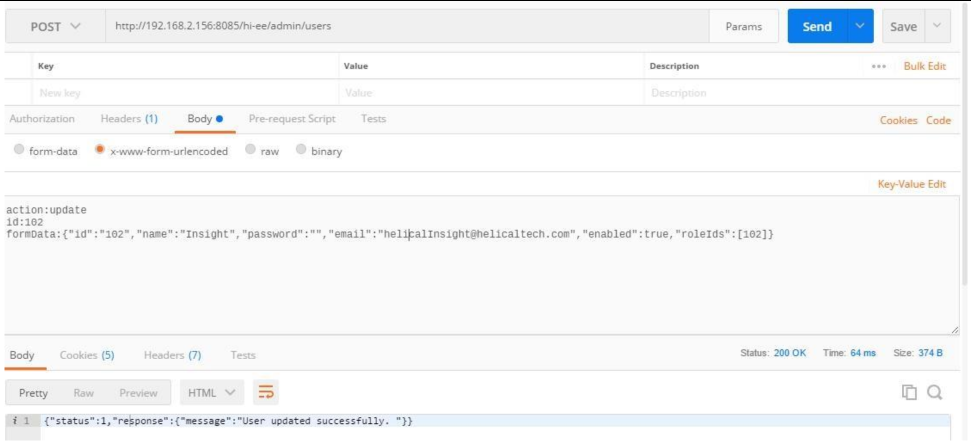
	Note : To edit user we requires the ID of the particular user which is assigned at the time of user creation to know the userID Refer UserList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=update&id=102&formData:={'password':'','email':'helicalInsight@helicaltech.com','enabled':true}" http://192.168.2.156:8085/hi-ee/admin/users -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	update	Operation to update user details
id:	102	ID of the user .
formData:	{"password": "","email": "helicalInsight@helicaltech.com", "enabled": true}	JSON object containing user information “Enabled” is a Boolean value having value “true” or “false”. If Enabled is false then respective user cannot login into the application. email : which you are going to update
Response Output (JSON format)	{ "status": 1, "Response": { "message": "User updated successfully. " } }	
Description of the Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. Respective user details are modified.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/users x-www-form-urlencoded action: update id: 102 formData: {"password": "","email": "helicalInsight@helicaltech.com",... Status: 200 OK Time: 56 ms Size: 374 B `{"status":1,"response":{"message":"User updated successfully. "}}`
Post-action	We Can edit / update user details.

2.3.14.2 Edit User Role

URL	admin/users	
Description	It allows admin to edit user's roles from the existing user list. Moreover, • Superadmin can edit all user's roles belongs to existing list of organization. • Organization admin can edit user's roles belong to their respective organization. Note : To edit user roles we requires the ID of the particular role which is assigned at the time of role creation to know the roleID Refer RoleList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=update&id=102&formData:={'id':102,'name':'Insight','password':'','email':'helicalInsight@helicaltech.com','enabled':true,'roleIds':[102]}" http://192.168.2.156:8085/hi-ee/admin/users -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	update	Operation to update information
id:	102	User-id
formData:	{"id": 102,"name":"Insight","password":"", "email": "helicalInsight@helicaltech.comEdits", "enabled": true, "roleIds": [102] }	JSON object containing user information We are updating the roleID for the user for that we need to pass the user id as id and roleIds as role id which are assigned at the time of role creation.

Response Output (JSON format)	{ "status": 1, "Response": { "message": "User updated successfully. " } }
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. Respective user role details are modified.
Service Status	200

Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/users
	Body: x-www-form-urlencoded
	`action:update`
	`id:102`
	`formData:{"id":"102","name":"Insight","password":"","email":"helicalInsight@helicaltech.com","enabled":true,"roleIds":[102]}`
	Status: 200 OK Time: 64 ms Size: 374 B
	`{"status":1,"response":{"message":"User updated successfully. "}}`

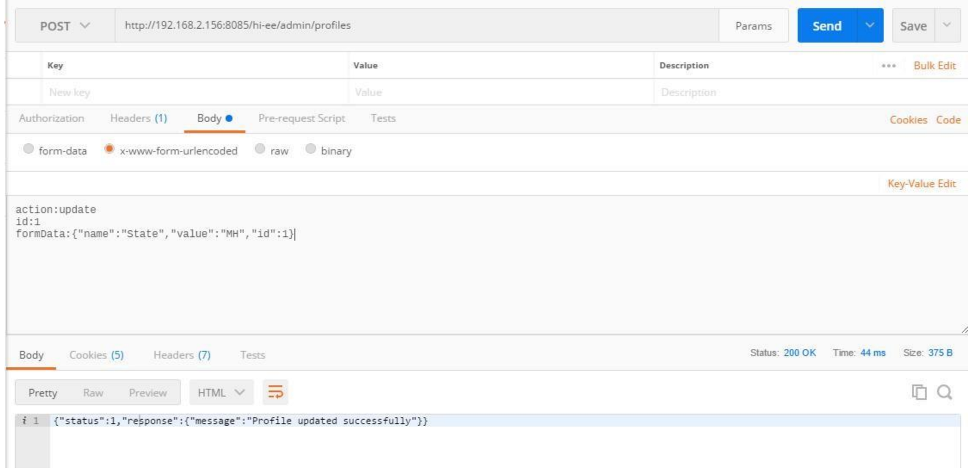
Post-action	Can edit / update user details.

2.3.14.3 Edit User Profile

URL	admin/profiles	
Description	It allows admin to edit user's profile from the existing list. Moreover, • Superadmin can edit all user's profile belongs to existing list of organization. • Organization admin can edit user's profile belong to their respective organization. Note : To edit user profile we requires the ID of the particular profile which is assigned at the time of profile creation to know the profileID Refer UserList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/profiles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=update&id=1&formData:={'id':1,'name':'State','value':'MH'}" http://192.168.2.156:8085/hi-ee/admin/profiles -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	update	Operation to add new user profile
id:	1	Profile id which needs to be edit
formData:	{"name":"State", "value":"MH", “id":1}	JSON object containing profile information name :name of the profile value : value of the profile id: ID of the profile

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Profile updated successfully"}
	}	

Description of response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. profile get updated with the respective user and the values is stored in database.	
Service Status	200 OK	

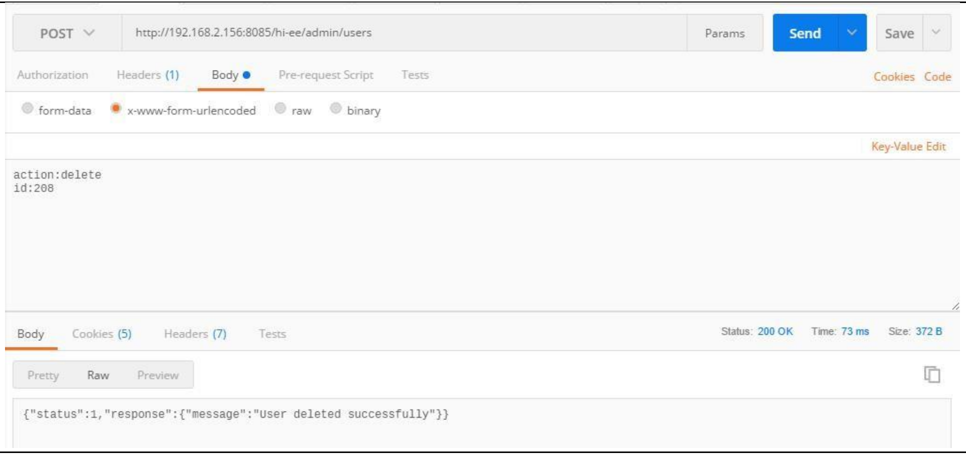
Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	Body: x-www-form-urlencoded
	`action:update`
	`id:1`
	`formData:{"name":"State","value":"MH","id":1}`
	Status: 200 OK Time: 44 ms Size: 375 B
	`{"status":1,"response":{"message":"Profile updated successfully"}}`

Post-action	User profile modification (edit / delete)

2.3.15 Delete

2.3.15.1 Delete User

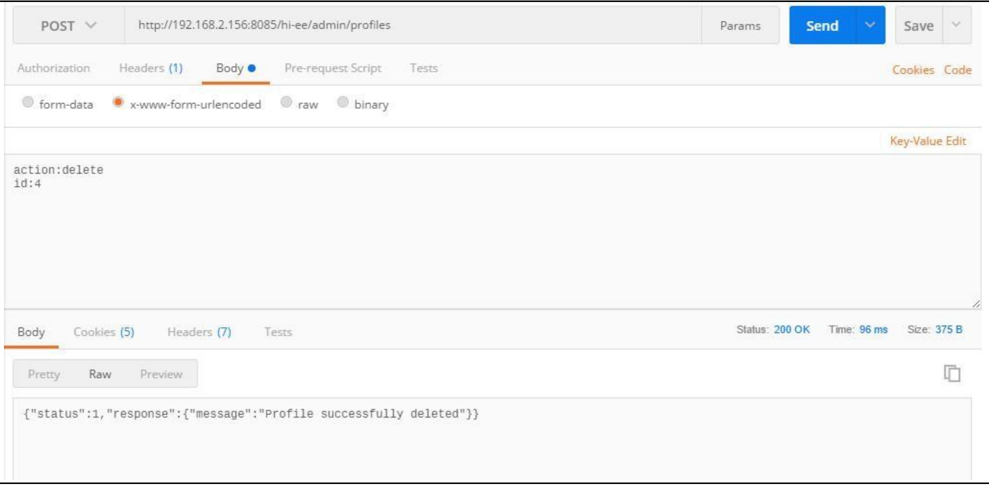
URL	admin/users	
Description	It allows admin to remove user’s profile from the existing list. Moreover, • Superadmin can remove all user’s profile belongs to existing list of organization. • Organization admin can remove user’s profile belong to their respective organization.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/users **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&id=208" http://192.168.2.156:8085/hi-ee/admin/users -v	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	delete	Operation to remove existing user
id:	208	Id of the user for delete
Response Output (JSON format)	{"status":1, "response": {"message":"User deleted successfully"}}	
Description of response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. Requested user get deleted	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/users
	x-www-form-urlencoded
	`action:delete`
	`id:208`
	Status: 200 OK Time: 73 ms Size: 372 B
	`{"status":1,"response":{"message":"User deleted successfully"}}`

2.3.15.2 Delete User Profile

URL	admin/profiles	
Description	It allows admin to remove user's profile from the existing list.Moreover, • Superadmin has the authority to remove all user's profile belongs to existing list of organization. • Organization admin has the authority to remove user's profile belong to their respective organization. Note : To delete user profile we requires the ID of the particular profile which is assigned at the time of profile creation to know the profileID Refer UserList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/profiles **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&id=4" http://192.168.2.156:8085/hi-ee/admin/profiles -v	
HTTP Request Key	**HTTP Request Value**	**Description**

action:	delete	Operation to remove existing user profile
id:	4	User profile ID.
Response Output (JSON format)	{ "status": 1, "Response": { "message": " Profile deleted successfully" } }	
Description of response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. Requested user profile get deleted	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/profiles
	x-www-form-urlencoded
	`action:delete`
	`id:4`
	Status: 200 OK Time: 96 ms Size: 375 B
	`{"status":1,"response":{"message":"Profile successfully deleted"}}`	

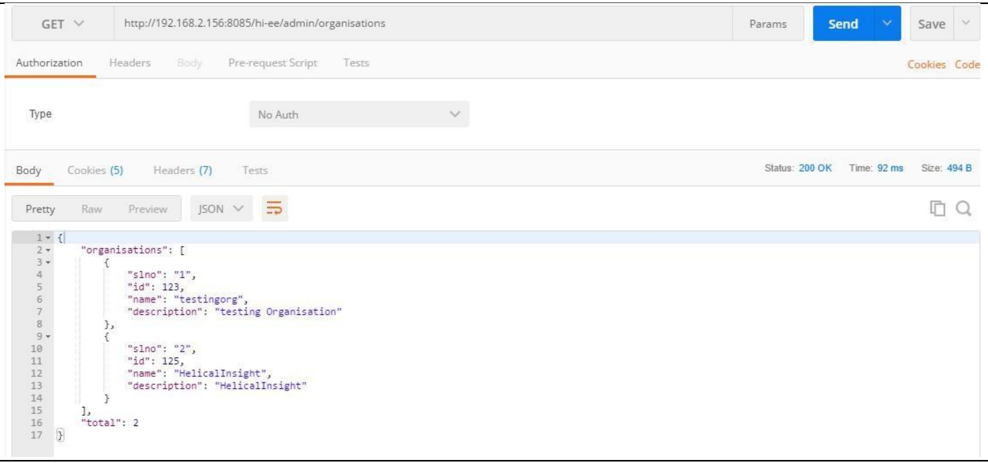
2.4 Organizations

2.4.1. Get Organization List/Refresh/Pagination-Next/Prev

URL	admin/organisations
Description	It displays the existing organizations list.User can refresh the organisation list and the user can use pagination -Next /Prev functionality as well.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**GET ,POST**

Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/organisations **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin" http://192.168.2.156:8085/hi-ee/admin/organisations -v

Response Output: (JSON format)	{
	    "organisations": [
	        {
	            "slno": "1",
	            "id": 123,
	            "name": "testingorg",
	            "description": "testing Organisation"
	        },
	        {
	            "slno": "2",
	            "id": 125,
	            "name": "HelicalInsight",
	            "description": "HelicalInsight"
	        }
	    ],
	    "total": 2
	}

Description of Response Output:	The response of the API is the array of existing organisations with serialNo , organisation ID , organisation name and the description of the organisation is returned with the total count of organisation. **slno** : Serial Number **id** : ID of the organisation **name** : Name of the organisation **description** : Description of organisation It displays all the non-Super organisations.
Service Status	200 OK
Screenshot	

GET http://192.168.2.156:8085/hi-ee/admin/organisations

Authorization — Type: No Auth

Status: 200 OK Time: 92 ms Size: 494 B

```
{
    "organisations": [
        {
            "slno": "1",
            "id": 123,
            "name": "testingorg",
            "description": "testing Organisation"
        },
        {
            "slno": "2",
            "id": 125,
            "name": "HelicalInsight",
            "description": "HelicalInsight"
        }
    ],
    "total": 2
}
```

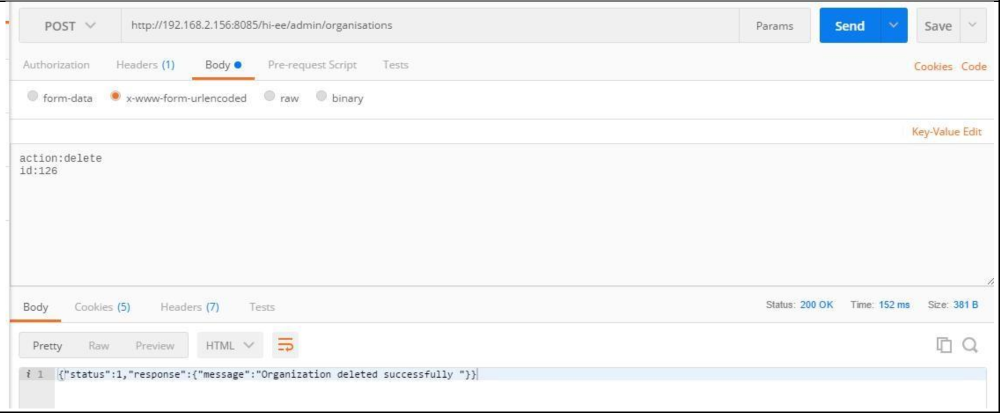
Post-action	Edit / Delete organization by superadmin.

2.4.2 Add Organization

2.4.3 Delete Organization

URL	admin/organisations	
Description	It allows super admin to remove organization from the existing organisation list. Organization will be deleted permanently from the database. This action is irreversible. Also, the respective users and roles are deleted.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN(super admin)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/organisations **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&id=126" http://192.168.2.156:8085/hi-ee/admin/organisations -v	
HTTP Request Key	**Values**	**Description**
action:	delete	Operation to remove organization
id:	126	Organization id assigned while organisation creation OR *Refer Get OrganisationList to get the id of the organisation you want to delete.*
Response Output: (JSON format)	{ "status": 1, "Response": { "message": "Organization deleted successfully " } }	
Description of	The response of the API is , it returns the success status value as 1 if it fails	

the Response Output :	returns 0 as the status. It returns response as success message and Organization is deleted permanently from the database. This action is irreversible. Also, the respective users and roles are deleted.
Service Status	200 OK

Screenshot	
	POST http://192.168.2.156:8085/hi-ee/admin/organisations
	Body: x-www-form-urlencoded
	`action:delete`
	`id:126`
	Status: 200 OK Time: 152 ms Size: 381 B
	`{"status":1,"response":{"message":"Organization deleted successfully "}}`

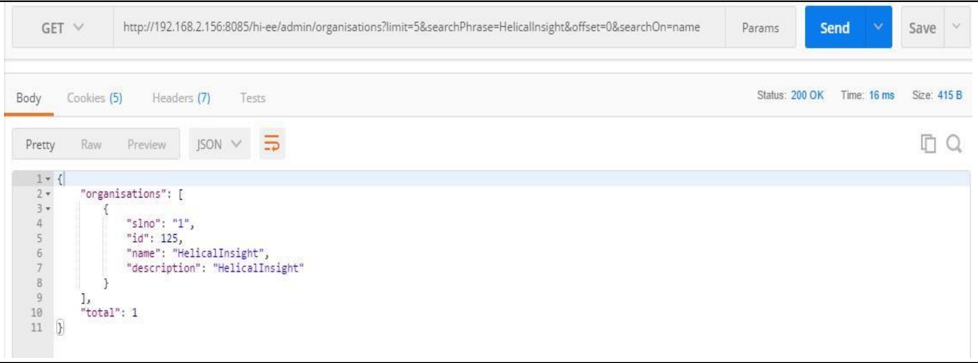
2.4.4 Search Organization

URL	admin/organisations?limit=5&searchPhrase=HelicalInsight&offset=0&searchOn=name
Description	It allows super admin to search the specific organization from the existing organization list by entering any phrase/keyword. In case of no records Found indicates organization not available in the list or typed phrase/keyword is wrong.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN(super admin)
HTTP Request Method	**GET ,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/organisations?limit=5&searchPhrase=HelicalInsight&offset=0&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&searchPhrase=HelicalInsight&offset=0&searchOn=name" http://192.168.2.156:8085/hi-ee/admin/organisations

HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	5	Set the number of records
searchPhrase	HelicalInsight	Search for a type phrase from the list Note : It should be the exact name passed while organisation creation.
offset **(optional)**	0	Sets the starting record
searchOn	name	Search list by user name.

Response Output: (JSON format)	{
	    "organisations": [
	        {
	            "slno": "1",
	            "id": 125,
	            "name": "HelicalInsight",
	            "description": "HelicalInsight"
	        }
	    ],
	    "total": 1
	}	

Description of Response Output :	The respone is the search organisation details which are the organisation name ,id , description etc with total count of organisations. **slno** : Serial Number **id** : ID of the organisation **name** : Name of the organisation **description** : Description of organisation	
Service Status	200 OK	

Screenshot	
	GET http://192.168.2.156:8085/hi-ee/admin/organisations?limit=5&searchPhrase=HelicalInsight&offset=0&searchOn=name
	Status: 200 OK Time: 16 ms Size: 415 B
	{
	    "organisations": [
	        {
	            "slno": "1",
	            "id": 125,
	            "name": "HelicalInsight",
	            "description": "HelicalInsight"
	        }
	    ],
	    "total": 1
	}
	 Params
	Send
	Save
	Body
	Cookies (5)
	Headers (7)
	Tests
	Pretty
	Raw
	Preview
	JSON	

Possible Error	If no records found
	{
	    "organisations": [],
	    "total": 0
	}	

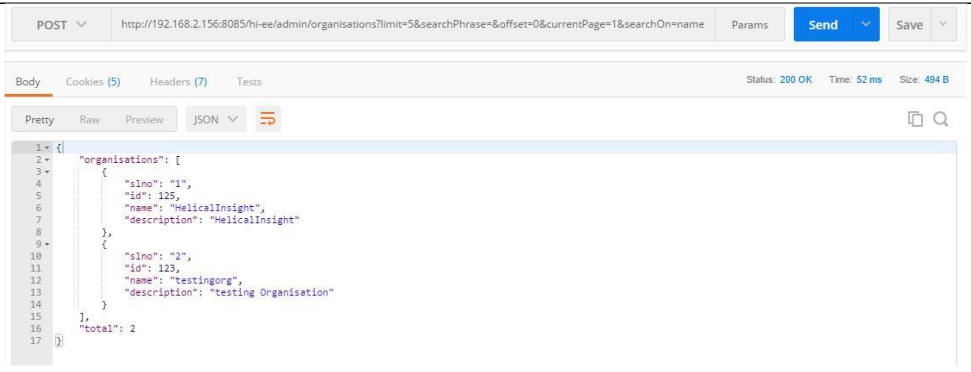
Post-action	Modify / Delete Organization	

2.4.5 Refresh Organization

URL	admin/organisations?limit=5&searchPhrase=&offset=0&currentPage=1&searchOn=name	
Description	It refreshes the list of organization.Shows you the list of organization after refresh if any new organisations are updated.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN(super admin)	
HTTP Request Method	**GET ,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/organisations?limit=5&searchPhrase=&offset=0&currentPage=1&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&searchPhrase=&offset=0&currentPage=1&searchOn=name" http://192.168.2.156:8085/hi-ee/admin/organisations	
HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	5	Set the number of records
searchPhrase		Search for a type phrase from the list Keep it as blank so that you will get all organisations list.
offset **(optional)**	0	Sets the starting record
currentPage	1	Set the number of page.
searchOn	name	Search list by user name.

Response Output: (JSON format)	{
	  "organisations": [
	    {
	      "slno": "1",
	      "id": 125,
	      "name": "HelicalInsight",
	      "description": "HelicalInsight"
	    },
	    {	

	"slno": "2", "id": 123, "name": "testingorg", "description": "testing Organisation" }], "total": 2 }
Description of the Response Output	The response of the API is the array of existing organisations with serialNo , organisation ID , organisation name and the description of the organisation is returned with the total count of organisation. **slno** : Serial Number **id** : ID of the organisation **name** : Name of the organisation **description** : Description of organisation It displays all the non-Super organisations.
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/organisations?limit=5&searchPhrase=&offset=0&currentPage=1&searchOn=name
	Status: 200 OK Time: 52 ms Size: 494 B
	{
	"organisations": [
	{
	"slno": "1",
	"id": 125,
	"name": "HelicalInsight",
	"description": "HelicalInsight"
	},
	{
	"slno": "2",
	"id": 123,
	"name": "testingorg",
	"description": "testing Organisation"
	}
	],
	"total": 2
	}

Error (expected)	If no records found
	{
	"organisations": [],
	"total": 0
	}

Post-action	Modify / Delete Organization

2.4.6 Show number of Organization entries

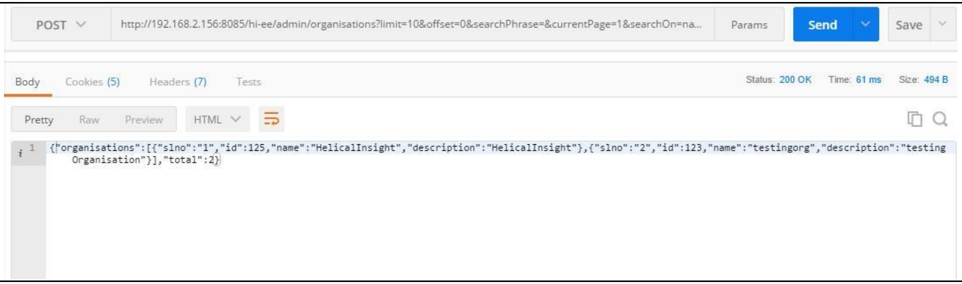
URL	admin/organisations?limit=10&offset=0&searchPhrase=&currentPage=1&searchOn=name
Description	It shows the number of entries in organization list.
Pre-requisite	User should have logged in before accessing the service.[Refer login

	module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN(super admin)
HTTP Request Method	**GET , POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/organisations?limit=10&offset=0&searchPhrase=&currentPage=1&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=10&offset=0&searchPhrase=&currentPage=1&searchOn=name "http://192.168.2.156:8085/hi-ee/admin/organisations

HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	10	Set the number of records /number of user entries
offset**(optional)**	0	Sets the starting record
currentPage	1	Number of the page
searchPhrase **(optional)**		Search for a type phrase from the list
searchOn **(optional)**	user	Search list by user name. Similarly by organization/ email/ roles

Response Output: (JSON format)

```
{
   "organisations": [
      {
         "slno": "1",
         "id": 125,
         "name": "HelicalInsight",
         "description": "HelicalInsight"
      },
      {
         "slno": "2",
         "id": 123,
         "name": "testingorg",
         "description": "testing Organisation"
      }
```

	], "total": 2 }
Description of Response Output:	"organizations" is json array which includes organization information such as slno, organization, id,and so on and shows total 10 user entries. **slno** : Serial Number **id** : ID of the organisation **name** : Name of the organisation **description** : Description of organisation
Service Status	200 OK
Screenshot	 POST http://192.168.2.156:8085/hi-ee/admin/organisations?limit=10&offset=0&searchPhrase=&currentPage=1&searchOn=na... Status: 200 OK Time: 61 ms Size: 494 B `{"organisations":[{"slno":"1","id":125,"name":"HelicalInsight","description":"HelicalInsight"},{"slno":"2","id":123,"name":"testingorg","description":"testing Organisation"}],"total":2}`

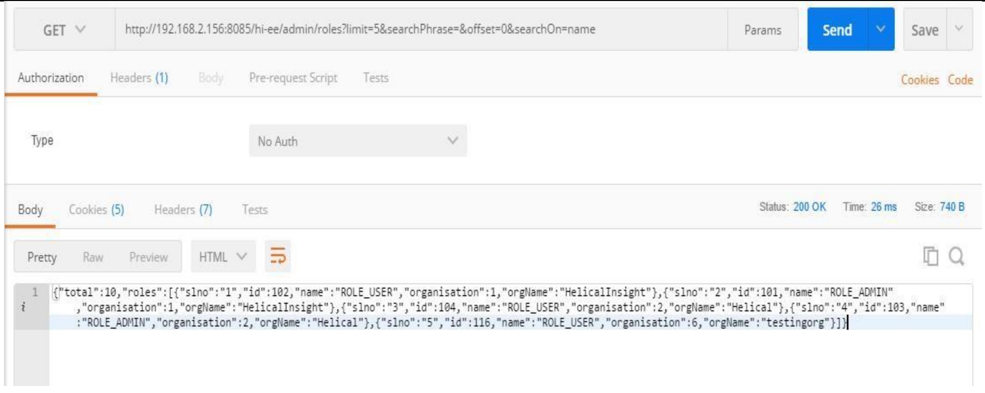
2.5 Roles

2.5.1 Add role under Super organization

2.5.2 Add role under organization

2.5.3 Get Role-List

URL	admin/roles?limit=5&searchPhrase=&offset=0&searchOn=name
Description	It allows super admin /organisation admin to get all roles from the existing list of roles.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**GET ,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-

	ee/admin/roles?limit=5&searchPhrase=&offset=0&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&searchPhrase=&offset=0&searchOn=name" http://192.168.2.156:8085/hi-ee/admin/roles	
HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	5	Set the number of records
searchPhrase		Search for a typed phrase from the list which role you want to search.
offset **(optional)**	0	Sets the starting record
searchOn**(optional)**	name	Search list by name
Response Output: (JSON format)	{"total":10,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":101,"name":"ROLE_ADMIN","organisation":1,"orgName":"HelicalInsight"},{"slno":"3","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"4","id":103,"name":"ROLE_ADMIN","organisation":2,"orgName":"Helical"},{"slno":"5","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"}]}	
Description of Response Output:	The response returned as the total number of records with roles array having role details as serialNo roleID,role name and associated organisation id and its name etc **slno** : Serial Number **id** : ID of the role **name** : Name of the role **organisation**: ID of the organisation. **orgName**: Name of organisation	
Service Status	200 OK	
Screenshot		

GET http://192.168.2.156:8085/hi-ee/admin/roles?limit=5&searchPhrase=&offset=0&searchOn=name

Type: No Auth

Status: 200 OK Time: 26 ms Size: 740 B

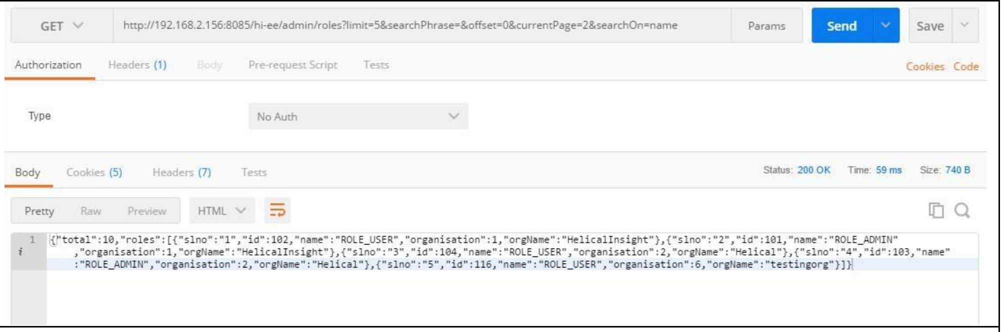
```
{"total":10,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":101,"name":"ROLE_ADMIN"
,"organisation":1,"orgName":"HelicalInsight"},{"slno":"3","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"4","id":103,"name"
:"ROLE_ADMIN","organisation":2,"orgName":"Helical"},{"slno":"5","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"}]}
```

Possible Error	If no record found then will get response as follow :
	{
	"total": 0,
	"roles": []
	}

Post-action	We can modify / delete roles.

2.5.4 Pagination-Next/Prev/Refresh Role-list/Show number of role entries

URL	admin/roles?limit=5&searchPhrase=&offset=0&currentPage=2&searchOn=name	
Description	It allows to refresh the role-list / show the number of entries in role page /the pagination operations. For that you can apply the search criteria according to your requirement. *Note: Required parameters are described below in HTTP Request Key-Value section.*	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET ,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/roles?limit=5&searchPhrase=&offset=0&currentPage=2&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&searchPhrase=&offset=0&currentPage=2&searchOn=name" http://192.168.2.156:8085/hi-ee/admin/roles	
HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	5	Set the number of records
searchPhrase	ROLE_USER	Search for a typed phrase from the list
offset **(optional)**	0	Sets the starting record

searchOn**(optional)**	name	Search list by user name
currentPage	2	Number of the page
Response Output: (JSON format)	{"total":10,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":101,"name":"ROLE_ADMIN","organisation":1,"orgName":"HelicalInsight"},{"slno":"3","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"4","id":103,"name":"ROLE_ADMIN","organisation":2,"orgName":"Helical"},{"slno":"5","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"}]}	
Description of Response Output:	The response returned as the total number of records with roles array having role details as serialNo roleID,role name and associated organisation id and its name etc. **slno** : Serial Number **id** : ID of the role **name** : Name of the role **organisation**: ID of the organisation. **orgName**: Name of organisation Displays the roles list as per the applied criteria	
Service Status	200 OK	
Screenshot	 GET http://192.168.2.156:8085/hi-ee/admin/roles?limit=5&searchPhrase=&offset=0&currentPage=2&searchOn=name Authorization: Type No Auth. Status: 200 OK Time: 59 ms Size: 740 B `{"total":10,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":101,"name":"ROLE_ADMIN","organisation":1,"orgName":"HelicalInsight"},{"slno":"3","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"4","id":103,"name":"ROLE_ADMIN","organisation":2,"orgName":"Helical"},{"slno":"5","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"}]}`	

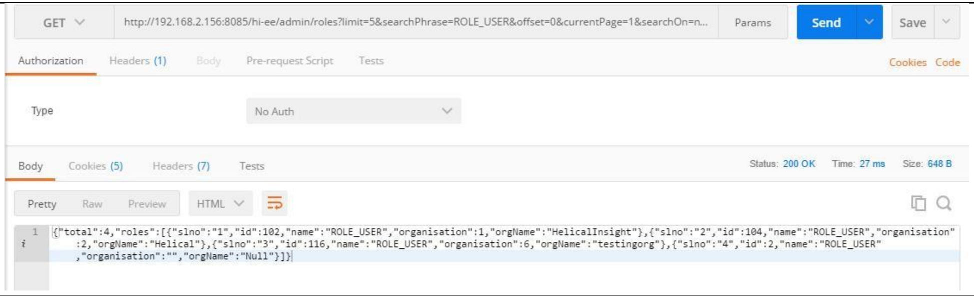
Possible Error	If no record found then will get response as below:
	{
	    "total": 0,
	    "roles": []
	}	

Post-action	We can modify / delete roles.	

2.5.5 Search for particular role

URL	admin/roles?limit=5&searchPhrase=ROLE_USER&offset=0&currentPage=1&searchOn=name

Description	It allows to search role from the existing list of roles. Moreover, • Superadmin can search role for any organization in the list • At organization level, organization admin can search any role belong to that respective organization. For that you can apply the search criteria according to your requirement. *Note: Required parameters are described below in HTTP Request Key-Value section*	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET ,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/roles?limit=5&searchPhrase=ROLE_USER&offset=0&currentPage=1&searchOn=name **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&limit=5&searchPhrase=ROLE_USER&offset=0&currentPage=1&searchOn=name" http://192.168.2.156:8085/hi-ee/admin/roles	
HTTP Request Key	**HTTP Request Value**	**Description**
limit**(optional)**	5	Set the number of records
searchPhrase	ROLE_USER	Search for a typed phrase from the list
offset **(optional)**	0	Sets the starting record
searchOn**(optional)**	name	Search list by user name
currentPage	1	Number of the page
ResponseOutput: (JSON format)	{"total":4,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"3","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"},{"slno":"4","id":2,"name":"ROLE_USER","organisation":"","orgName":"Null"}]}	
Description of the Response	The response returned as the total number of records with roles array having role details as serialNo roleID,role name and associated organisation id and	

Output:	its name etc. **slno** : Serial Number **id** : ID of the role **name** : Name of the role **organisation**: ID of the organisation. **orgName**: Name of organisation Displays the roles list as per the applied criteria
Service Status	200 OK
Screenshot	GET http://192.168.2.156:8085/hi-ee/admin/roles?limit=5&searchPhrase=ROLE_USER&offset=0&currentPage=1&searchOn=n... Params Send Save Authorization Headers (1) Body Pre-request Script Tests Cookies Code Type No Auth Body Cookies (5) Headers (7) Tests Status: 200 OK Time: 27 ms Size: 648 B Pretty Raw Preview HTML `{"total":4,"roles":[{"slno":"1","id":102,"name":"ROLE_USER","organisation":1,"orgName":"HelicalInsight"},{"slno":"2","id":104,"name":"ROLE_USER","organisation":2,"orgName":"Helical"},{"slno":"3","id":116,"name":"ROLE_USER","organisation":6,"orgName":"testingorg"},{"slno":"4","id":2,"name":"ROLE_USER","organisation":"","orgName":"Null"}]}`

Possible Errors:	If no record found then you will get response as:
	{
	   "total": 0,
	   "roles": []
	}

Post-action	We can modify / delete roles.

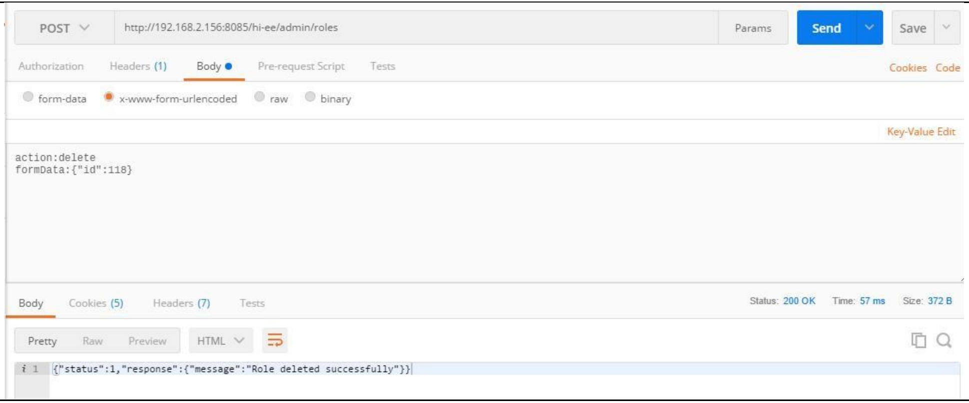
2.5.6 Delete role

URL	admin/roles
Description	It allows to remove role's from an organization. • Superadmin has authority to remove roles from any organization • At organization level, only organization admin can remove its respective roles for that organization. • Here to delete any role you need to pass the role ID which is assigned at time of role creation , so to get role id Refer GetRoleList
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/admin/roles

	Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&formData={'id':118}" http://192.168.2.156:8085/hi-ee/admin/roles	
HTTP Request Key	**HTTP Request Value**	**Description**
action:	delete	Operation to delete role
formData:	{"id":118}	JSON object containing user information id : id of the role which you want to delete.

Response Output (JSON Format)	{
	    "status": 1,
	    "Response": {
	        "message": "Role deleted successfully"
	    }
	}	

Description of Respone Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. Role is permanently deleted from the database. If the role was assigned to any user then it gets deallocated from that user. Note : Current login user's role deletion will not happen.	
Service Status	200 OK	

Screenshot	
	POST http://192.168.2.156:8085/hi-ee/admin/roles
	Body: x-www-form-urlencoded
	`action:delete`
	`formData:{"id":118}`
	Status: 200 OK Time: 57 ms Size: 372 B
	`{"status":1,"response":{"message":"Role deleted successfully"}}`	

2.6 Scheduling

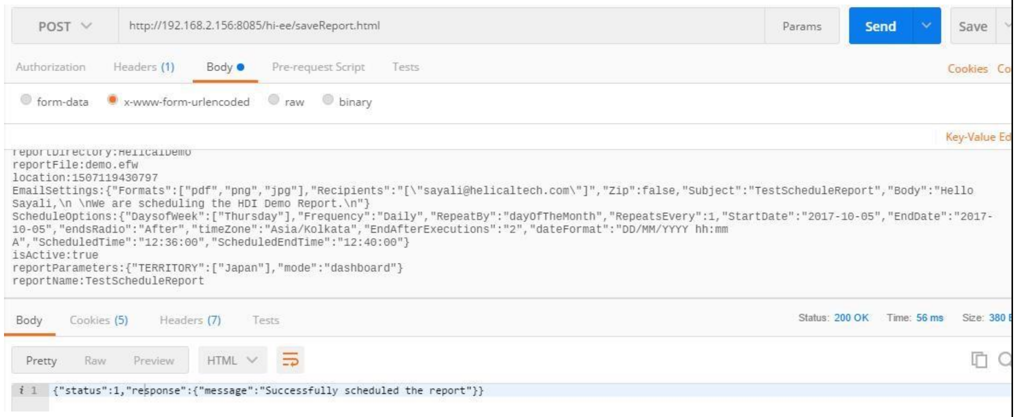
2.6.1 Schedule Report

URL	saveReport.html

Description	Any report efw/adhoc can be schedule and get emailed to provided Recipients. To schedule any report we need to pass some parameter values which are mentioned in HTTP Request Key-Value section. Scheduling of report can be done on daily,weekly,monthly,yearly basis.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/saveReport.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=add&command=add&reportDirectory=HelicalDemo&reportFile=demo.efw&location=1507119430797&EmailSettings={'Formats':['pdf','png','jpg'],'Recipients':'[\'sayali@helicaltech.com\']','Zip':false,'Subject':'TestScheduleReport','Body':'Hello Sayali,\n\nWe are scheduling the HDI Demo Report.\n'}&ScheduleOptions={'DaysofWeek':['Thursday'],'Frequency':'Daily','RepeatBy':'dayOfTheMonth','RepeatsEvery':1,'StartDate':'2017-10-05','EndDate':'2017-10-05','endsRadio':'After','timeZone':'Asia/Kolkata','EndAfterExecutions':'2','dateFormat':'DD/MM/YYYY hh:mm A','ScheduledTime':'12:21:00','ScheduledEndTime':'12:16:00'}&isActive=true&reportParameters={'TERRITORY':['Japan'],'mode':'dashboard'}&reportName=TestScheduleReport" http://192.168.2.156:8085/hi-ee/saveReport.html -v *Note : Make sure that ScheduledTime and ScheduledEndTime should not be the past datetime , otherwise you will get error message.*

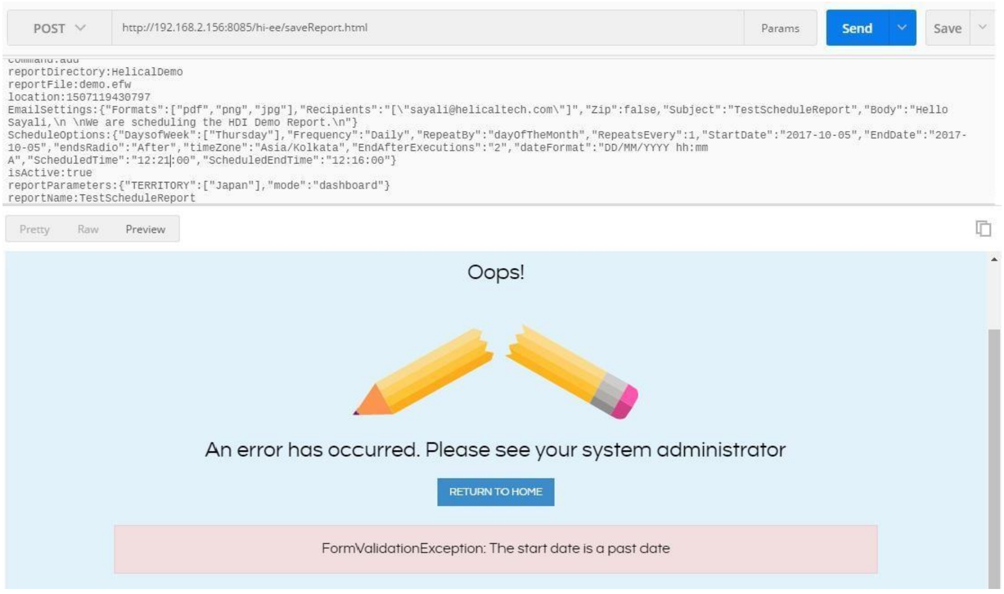
HTTP Request Key	**HTTP Request Value**	**Description**
command:	add	Command as add to schedule report
reportDirectory:	HelicalDemo	Directory of the report which you want to schedule.
reportFile:	demo.efw	The report file physical name
location:	1507119430797	Physical Location
EmailSettings:	{"Formats":["pdf","png","jpg"],"Recipients":"[\"sayali@helicaltech.co	The JSON object holding the email details

	m\"]","Zip":false,"Subject":"TestScheduleReport","Body":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n"}	
ScheduleOptions:	{"DaysofWeek":["Thursday"],"Frequency":"Daily","RepeatBy":"dayOfTheMonth","RepeatsEvery":1,"StartDate":"2017-10-05","EndDate":"2017-10-05","endsRadio":"After","timeZone":"Asia/Kolkata","EndAfterExecutions":"2","dateFormat":"DD/MM/YYYY hh:mm A","ScheduledTime":"12:21:00","ScheduledEndTime":"12:16:00"}	The JSON object holding the scheduling information. *Note : Make sure that ScheduledTime and ScheduledEndTime should not be the past datetime , otherwise you will get error message.*
isActive:	true	Schedule active status
reportParameters:	{"TERRITORY":["Japan"],"mode": "dashboard"}	Report parameters that is being considered for schedule.**(optional)**
reportName:	TestScheduleReport	Name of the report given at time of scheduling.**(optional)**
Response Output: (JSON Format)	{ "status": 1, "Response": {"message": "Successfully scheduled the report"} }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as success message. The respective file is scheduled for email to the recipients provided by the user	
Service Status	200 OK	

Screenshot(success)	
	POST http://192.168.2.156:8085/hi-ee/saveReport.html
	x-www-form-urlencoded
	`reportDirectory:HelicalDemo`
	`reportFile:demo.efw`
	`location:1507119430797`
	`EmailSettings:{"Formats":["pdf","png","jpg"],"Recipients":"[\"sayali@helicaltech.com\"]","Zip":false,"Subject":"TestScheduleReport","Body":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n"}`
	`ScheduleOptions:{"DaysofWeek":["Thursday"],"Frequency":"Daily","RepeatBy":"dayOfTheMonth","RepeatsEvery":1,"StartDate":"2017-10-05","EndDate":"2017-10-05","endsRadio":"After","timeZone":"Asia/Kolkata","EndAfterExecutions":"2","dateFormat":"DD/MM/YYYY hh:mm A","ScheduledTime":"12:36:00","ScheduledEndTime":"12:40:00"}`
	`isActive:true`
	`reportParameters:{"TERRITORY":["Japan"],"mode":"dashboard"}`
	`reportName:TestScheduleReport`
	Status: 200 OK Time: 56 ms Size: 380
	`{"status":1,"response":{"message":"Successfully scheduled the report"}}`	

Possible Error	While Scheduling if scheduled time and scheduled end time is past datetime	

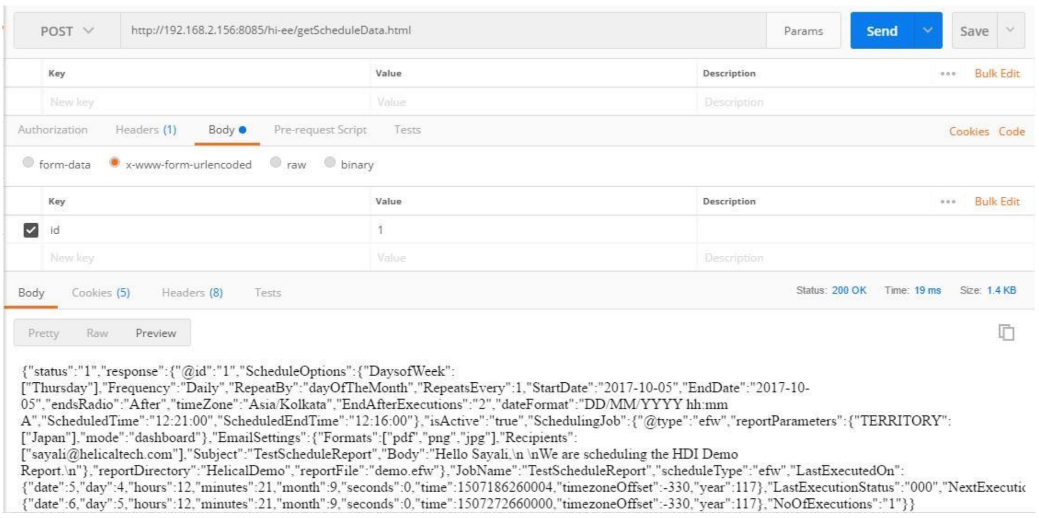
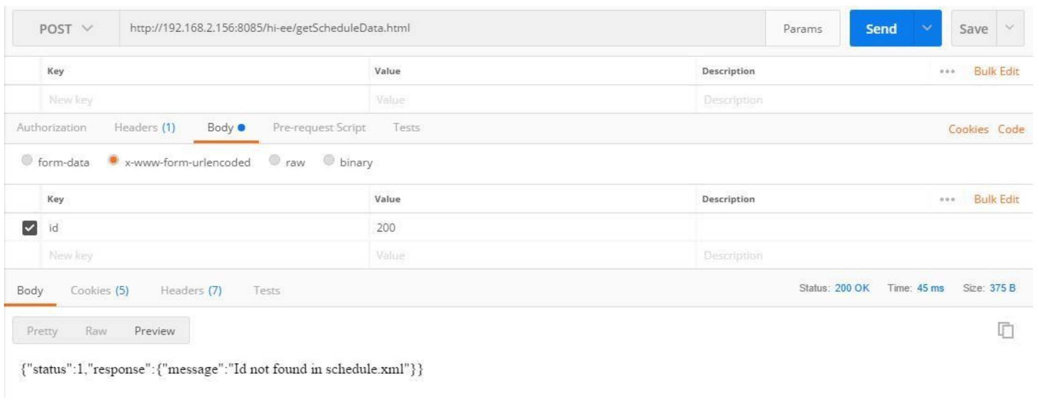
	then you will get error. *Note : Make sure that ScheduledTime and ScheduledEndTime should not be the past datetime , otherwise you will get error message.*

Screenshot(Error)	
	POST http://192.168.2.156:8085/hi-ee/saveReport.html
	`command:add`
	`reportDirectory:HelicalDemo`
	`reportFile:demo.efw`
	`location:1507119430797`
	`EmailSettings:{"Formats":["pdf","png","jpg"],"Recipients":"[\"sayali@helicaltech.com\"]","Zip":false,"Subject":"TestScheduleReport","Body":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n"}`
	`ScheduleOptions:{"DaysofWeek":["Thursday"],"Frequency":"Daily","RepeatBy":"dayOfTheMonth","RepeatsEvery":1,"StartDate":"2017-10-05","EndDate":"2017-10-05","endsRadio":"After","timeZone":"Asia/Kolkata","EndAfterExecutions":"2","dateFormat":"DD/MM/YYYY hh:mm A","ScheduledTime":"12:21:00","ScheduledEndTime":"12:16:00"}`
	`isActive:true`
	`reportParameters:{"TERRITORY":["Japan"],"mode":"dashboard"}`
	`reportName:TestScheduleReport`
	Oops!
	An error has occurred. Please see your system administrator
	FormValidationException: The start date is a past date

2.6.2 Get Schedule job information

URL	getScheduleData.html
Description	The user can obtain the already scheduled job information from the existing scheduled report by passing it the jobID , you will get scheduled job id Refer GetScheduleList
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getScheduleData.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&id=1" http://192.168.2.156:8085/hi-ee/getScheduleData.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
id:	1	JOBId of the scheduled report
Response Output (JSON Format)	JSON object containing the schedule report For Ex. {"status":"1","response":{"@id":"1","ScheduleOptions":{"DaysofWeek":["Wednesday"],"Frequency":"Daily","RepeatBy":"dayOfTheMonth","RepeatsEvery":1,"StartDate":"2017-09-20","EndDate":"2017-09-20","endsRadio":"After","timeZone":"Asia/Kolkata","EndAfterExecutions":"3","dateFormat":"DD/MM/YYYY hh:mm A","ScheduledTime":"18:50:00","ScheduledEndTime":"18:46:00"},"isActive":"true","SchedulingJob":{"@type":"report","reportParameters":{},"EmailSettings":{"Formats":["pdf","png","jpg"],"Recipients":["sayali@helicaltech.com"],"Subject":"Schedule","Body":"Hi"},"reportDirectory":"1463377807724\\1463983915686\\1463838054907","reportFile":"d1560c88-be0d-4380-8225-8a8df4eb53bf.report"},"JobName":"TestSchedule","scheduleType":"report","LastExecutedOn":{"date":20,"day":3,"hours":18,"minutes":50,"month":8,"seconds":0,"time":1505913600005,"timezoneOffset":-330,"year":117},"LastExecutionStatus":"000","NextExecutionOn":{"date":21,"day":4,"hours":18,"minutes":50,"month":8,"seconds":0,"time":1506000000000,"timezoneOffset":-330,"year":117},"NoOfExecutions":"1"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the scheduled report details for requested JobID. **@id:** Scheduled job id **ScheduleOptions:** **Frequency:** Schedule can be done in Daily/Weekly/Monthly/Yearly **RepeatBy/On:** Days of week, Day of the month, day of the week **RepeatsEvery:** Schedule can be repeat after given iterations ex: 2 days, Every 2 days schedule will repeat. **StartDate:** Start date of the scheduler **EndDate :** End date of the scheduler **endsRadio :**Never/ After no.of iterations/ End date of the scheduler **timeZone :** Can select different time-zones **ScheduledTime :** Start time of the scheduler **ScheduledEndTime:** End time of the scheduler **isActive:** provide working status of the scheduler	

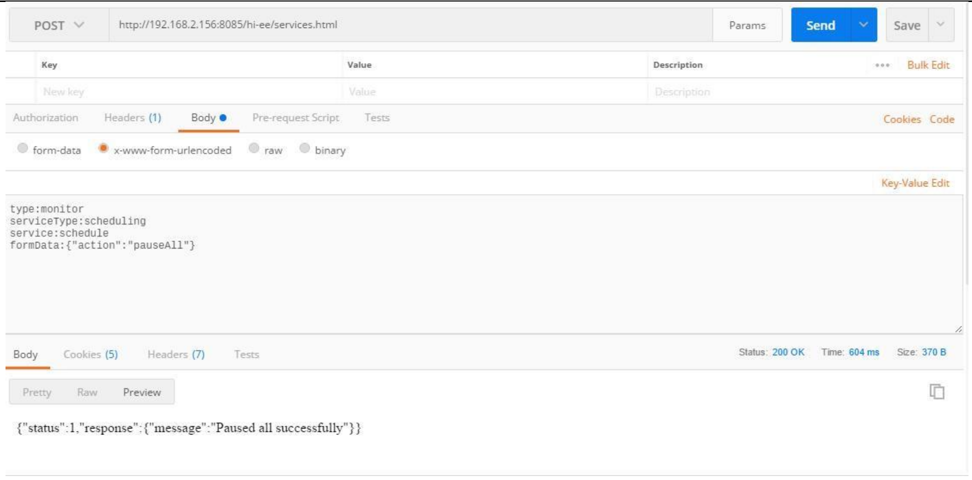
	SchedulingJob : @type **reportParameters :**It will show selected parameters on scheduled report **EmailSettings:** **Formats:** mail attachment formats of scheduled report **Recipients:**mail address **Subject:** can provide Subject of the scheduler **Body**:can provide Body of the scheduler **reportDirectory:** It provide the information where report is located. **ReportFile:**Name of the report along with extension **JobName:** Name of the scheduler job, user can provide at the time of saving **scheduleType:** type of report is scheduled **LastExecutedOn:** gives last execution date and time of the scheduler **LastExecutionStatus:** **NextExecutionOn:**gives next execution date and time of the scheduler
Service Status	200 OK
Screenshot(Success)	 POST http://192.168.2.156:8085/hi-ee/getScheduleData.html — Body: x-www-form-urlencoded — Key: id, Value: 1 — Status: 200 OK Time: 19 ms Size: 1.4 KB — `{"status":"1","response":{"@id":"1","ScheduleOptions":{"DaysofWeek": ["Thursday"],"Frequency":"Daily","RepeatBy":"dayOfTheMonth","RepeatsEvery":1,"StartDate":"2017-10-05","EndDate":"2017-10-05","endsRadio":"After","timeZone":"Asia/Kolkata","EndAfterExecutions":"2","dateFormat":"DD/MM/YYYY hh:mm A","ScheduledTime":"12:21:00","ScheduledEndTime":"12:16:00"},"isActive":"true","SchedulingJob":{"@type":"efw","reportParameters":{"TERRITORY": ["Japan"],"mode":"dashboard"},"EmailSettings":{"Formats":["pdf","png","jpg"],"Recipients": ["sayali@helicaltech.com"],"Subject":"TestScheduleReport","Body":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n"},"reportDirectory":"HelicalDemo","reportFile":"demo.efw"},"JobName":"TestScheduleReport","scheduleType":"efw","LastExecutedOn": {"date":5,"day":4,"hours":12,"minutes":21,"month":9,"seconds":0,"time":1507186260004,"timezoneOffset":-330,"year":117},"LastExecutionStatus":"000","NextExecutic {"date":6,"day":5,"hours":12,"minutes":21,"month":9,"seconds":0,"time":1507272660000,"timezoneOffset":-330,"year":117},"NoOfExecutions":"1"}}`
Possible Error	If the JobID doesnot exists , you will get an error message as “Id not found in schedule.xml”
ScreenShot(Error)	 POST http://192.168.2.156:8085/hi-ee/getScheduleData.html — Body: x-www-form-urlencoded — Key: id, Value: 200 — Status: 200 OK Time: 45 ms Size: 375 B — `{"status":1,"response":{"message":"Id not found in schedule.xml"}}`

2.6.3 Pause All Scheduled job

URL	services.html	
Description	It allows super admin to pause all the running scheduled jobs. All running scheduled jobs get paused.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'pauseAll'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"pauseAll"}	Action to pause all scheduled jobs

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Paused all successfully"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"pauseAll"}
```

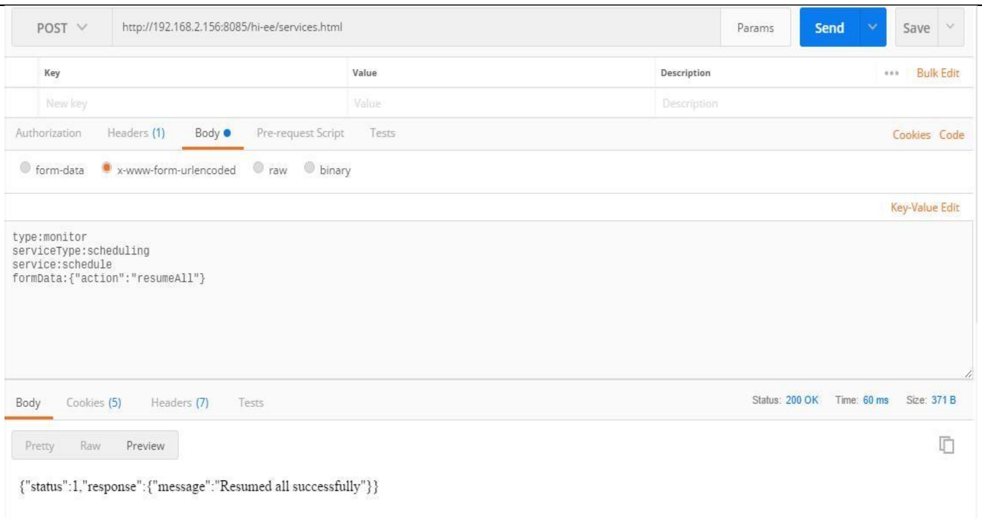
Status: 200 OK Time: 604 ms Size: 370 B

```
{"status":1,"response":{"message":"Paused all successfully"}}
```

2.6.4 Resume All Scheduled job

URL	services.html
Description	It allows super admin to resume all the paused scheduled jobs. All paused scheduled jobs get resumed.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'resumeAll'}" http://192.168.2.156:8085/hi-ee/services.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"resumeAll"}	Action to resume all scheduled jobs

Response Output (JSON format)	{ "status":1, "response":{"message":"Resumed all successfully"} }
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. All paused scheduled jobs get resumed. Note : After the resumeAll job, scheduled job list automatically get refreshed.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

x-www-form-urlencoded

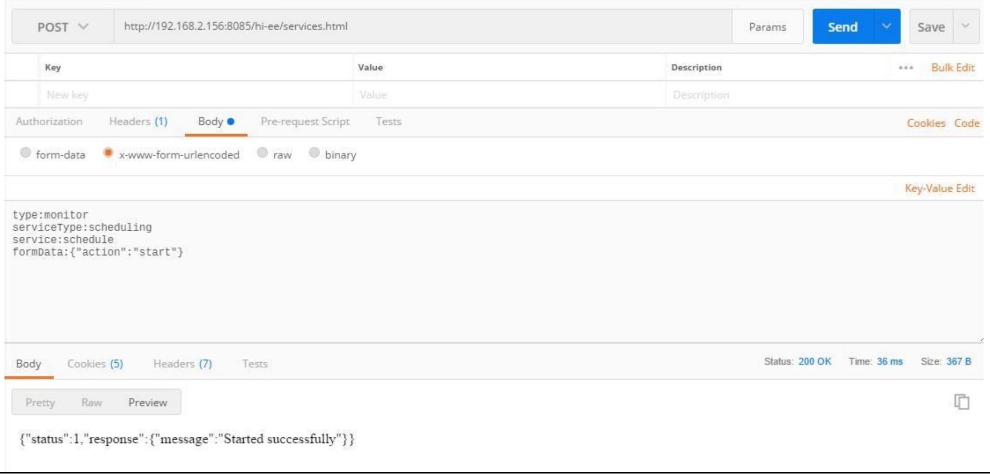
```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"resumeAll"}
```

Status: 200 OK Time: 60 ms Size: 371 B

```
{"status":1,"response":{"message":"Resumed all successfully"}}
```

2.6.5 Start Scheduler

URL	services.html	
Description	It allows super admin to start the scheduler.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** `curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'start'}" http://192.168.2.156:8085/hi-ee/services.html -v`	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"start"}	Action to start the scheduler.
Response Output (JSON format)	{ "status":1, "response":{"message":"Started successfully"} }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Scheduler get started. Note : After the scheduler get started scheduled job list automatically get refreshed.	
Service Status	200 OK	

Screenshot	

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"start"}
```

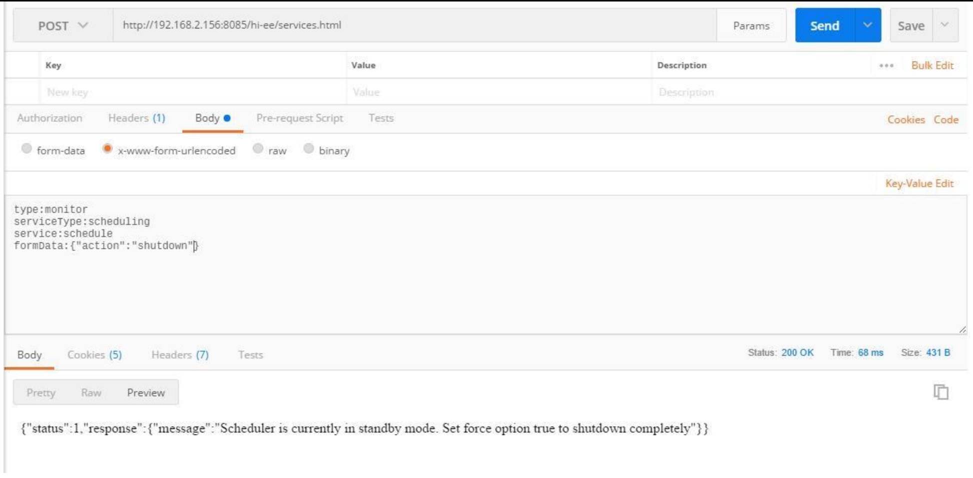
```
{"status":1,"response":{"message":"Started successfully"}}
```

2.6.6 Stop/Shutdown Scheduler

URL	services.html
Description	It allows super admin to stop/shutdown the scheduler.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'shutdown'}" http://192.168.2.156:8085/hi-ee/services.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"shutdown"}	Action to stop the scheduler.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"Scheduler is currently in standby mode. Set force option true to shutdown completely"}
	}

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Scheduler get stopped. Note : After the scheduler get started scheduled job list automatically get refreshed.
Service Status	200 OK
Screenshot	

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"shutdown"}
```

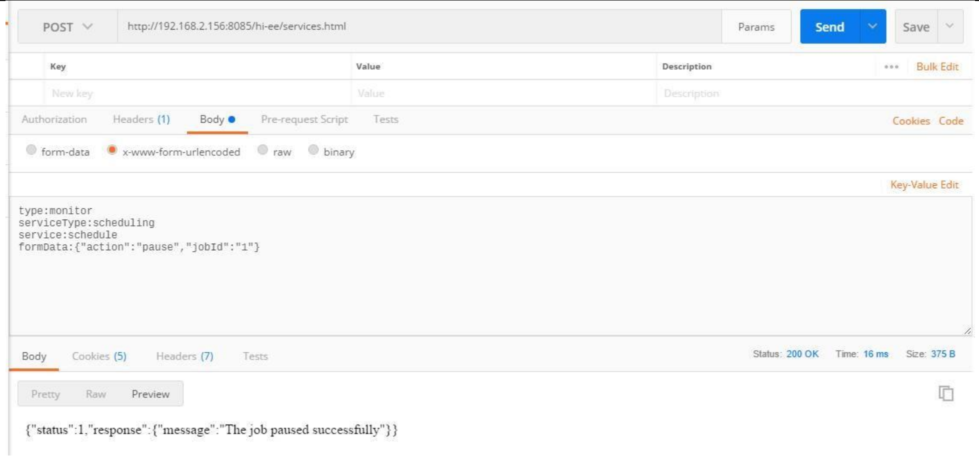
```
{"status":1,"response":{"message":"Scheduler is currently in standby mode. Set force option true to shutdown completely"}}
```

2.6.7 Pause the scheduled job

URL	services.html	
Description	It allows super admin to pause the selected running scheduled job. To pause the particular job we need the job id so to get the jobId of scheduled report Refer GetScheduledList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'pause','jobId':'1'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"pause","jobId":"1"}	Action to pause the provided scheduled job id.

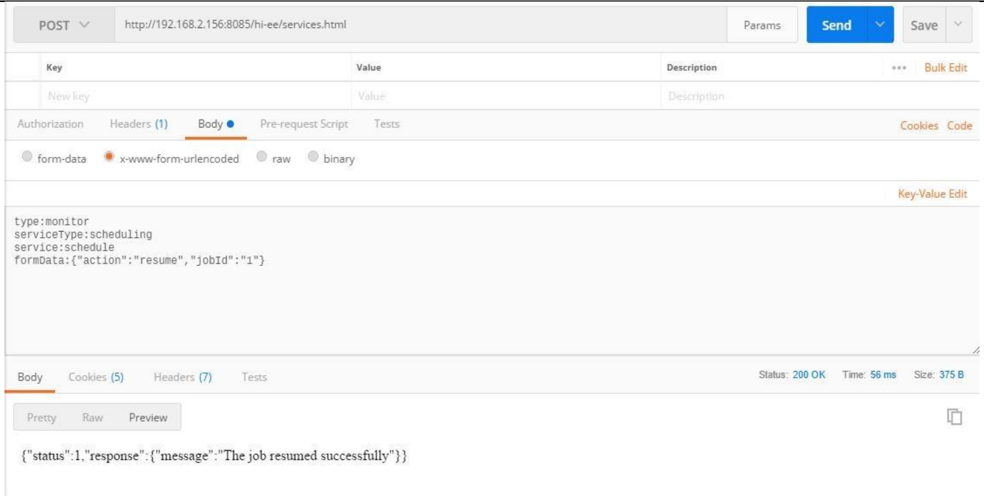
Response Output (JSON format)	{
	"status":1,
	"response":{"message":"The job paused successfully"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Selected scheduled job get paused. Note : After the paused scheduled job , scheduled job list get refreshed.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	x-www-form-urlencoded
	`type:monitor`
	`serviceType:scheduling`
	`service:schedule`
	`formData:{"action":"pause","jobId":"1"}`
	Status: 200 OK Time: 16 ms Size: 375 B
	`{"status":1,"response":{"message":"The job paused successfully"}}`

2.6.8 Resume the scheduled job

URL	services.html	
Description	It super allows admin to resume the selected scheduled job. To resume the particular job we need the job id so to get the jobId of scheduled report Refer GetScheduledList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'resume','jobId':'1'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"resume","jobId":"1"}	Action to resume the provided scheduled job id.

Response Output (JSON format)	{ "status":1 ,"response":{"message":"The job resumed successfully"} }
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Selected scheduled job get resumed. Note : After the resumed scheduled job , scheduled job list get refreshed.
Service Status	200 OK
Screenshot	

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"resume","jobId":"1"}
```

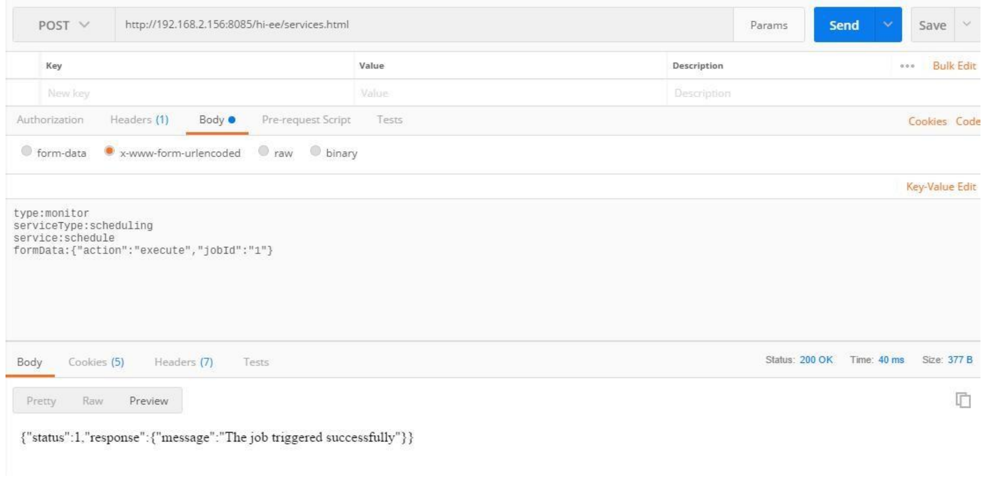
```
{"status":1,"response":{"message":"The job resumed successfully"}}
```

2.6.9 Execute the scheduled job

URL	services.html	
Description	It allows super admin to execute the selected scheduled job. To execute the particular job we need the job id so to get the jobId of scheduled report Refer GetScheduledList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'execute','jobId':'1'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"execute","jobId":"1"}	Action to execute the provided scheduled job id.

Response Output (JSON format)	{
	"status":1
	,"response":{"message":"The job triggered successfully"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Selected scheduled job get executed. Note : After the execution of scheduled job , scheduled job list get refreshed.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"execute","jobId":"1"}
```

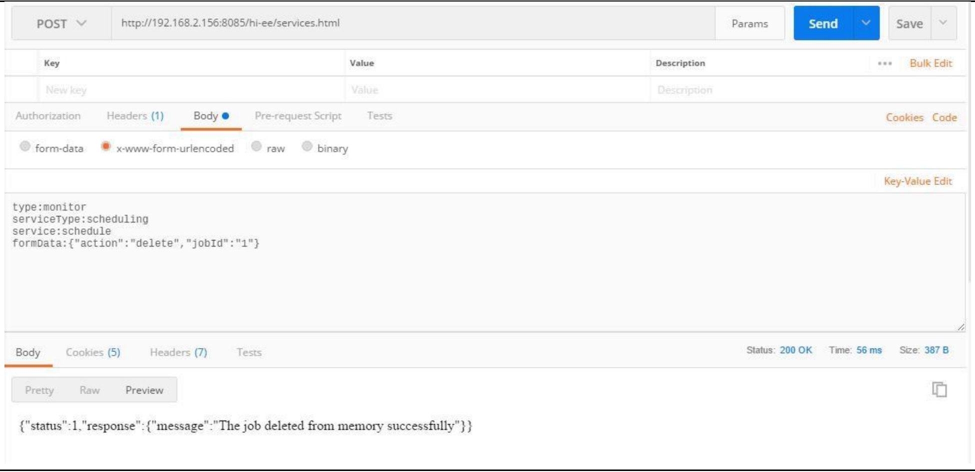
Status: 200 OK Time: 40 ms Size: 377 B

```
{"status":1,"response":{"message":"The job triggered successfully"}}
```

2.6.10 Delete the scheduled job

URL	services.html	
Description	It allows super admin to delete the selected scheduled job. To delete the particular job we need the job id so to get the jobId of scheduled report Refer GetScheduledList	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'delete','jobId':'1'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"delete","jobId":"1"}	Action to delete the provided scheduled job id.

Response Output (JSON format)	{
	"status":1,
	"response":{"message":"The job deleted from memory successfully"}
	}

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. Selected scheduled job get deleted. Note : After the deletion of scheduled job , scheduled job list get refreshed.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:monitor
serviceType:scheduling
service:schedule
formData:{"action":"delete","jobId":"1"}
```

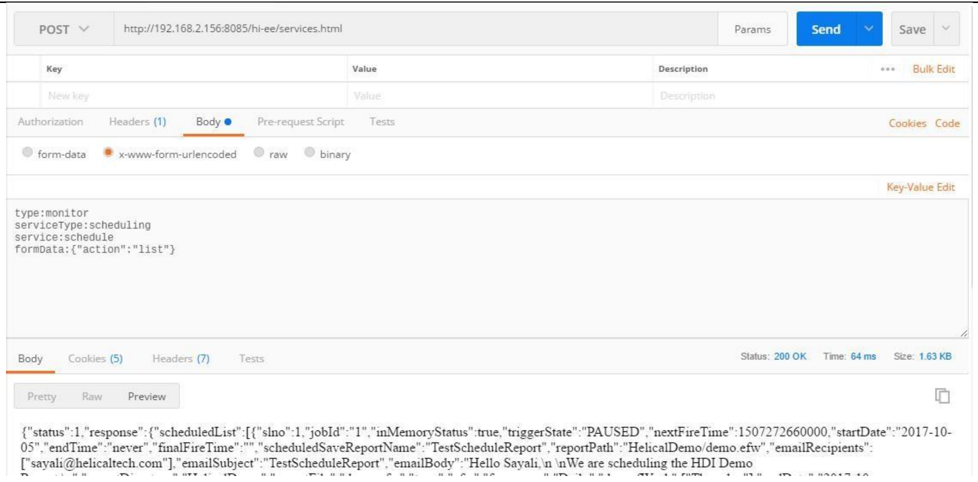
Status: 200 OK Time: 56 ms Size: 387 B

```
{"status":1,"response":{"message":"The job deleted from memory successfully"}}
```

2.6.11 Get scheduled job List

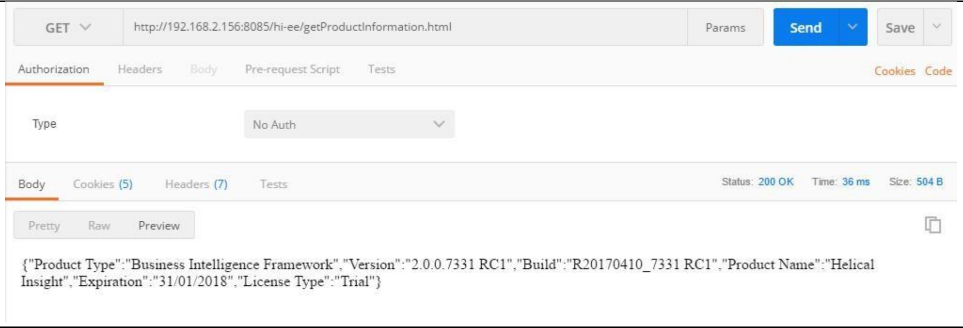
URL	services.html	
Description	It allows super admin to get all the scheduled jobs information	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=scheduling&service=schedule&formData={'action':'list'}" http://192.168.2.156:8085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	monitor	Type of the Operation
serviceType:	scheduling	serviceType as scheduling
service:	schedule	Service name as schedule
formData:	{"action":"list"}	Action to get all scheduled jobs
Response Output (JSON format)	{"status":1,"response":{"scheduledList":[{"slno":1,"jobId":"1","inMemoryStatus":true,"triggerState":"PAUSED","nextFireTime":1507272660000,"startDate":"2017-10-05","endTime":"never","finalFireTime":"","scheduledSaveReportName":"TestScheduleReport","reportPath":"HelicalDemo/demo.efw","emailRecipients":["sayali@helicaltech.com"],"emailSubject":"TestScheduleReport","emailBody":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n","reportDirectory":"HelicalDemo","reportFile":"demo.efw","type":"efw","frequency":"Daily","daysofWeek":["Thursday"],"endDate":"2017-10-05","scheduledTime":"12:21:00","lastExecutedOn":"1507186260004","reportParameters":{"TERRITORY":["Japan"],"mode":"dashboard"}},{"slno":2,"jobId":"2","inMemoryStatus":true,"triggerState":"PAUSED","nextFireTime":1507273560000,"startDate":"2017-10-05","endTime":"never","finalFireTime":"","scheduledSaveReportName":"TestScheduleReport","reportPath":"HelicalDemo/demo.efw","emailRecipients":["sayali@helicaltech.com"],"emailSubject":"TestScheduleReport","emailBody":"Hello Sayali,\n \nWe are scheduling the HDI Demo Report.\n","reportDirectory":"HelicalDemo","reportFile":"demo.efw","type":"efw","frequency":"Daily","daysofWeek":["Thursday"],"endDate":"2017-10-	

	05","scheduledTime":"12:36:00","lastExecutedOn":"1507187160002","reportParameters":{"TERRITORY":["Japan"],"mode":"dashboard"}}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the all scheduled job information.
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:monitor`
	`serviceType:scheduling`
	`service:schedule`
	`formData:{"action":"list"}`
	Status: 200 OK Time: 64 ms Size: 1.63 KB
	{"status":1,"response":{"scheduledList":[{"slno":1,"jobId":"1","inMemoryStatus":true,"triggerState":"PAUSED","nextFireTime":1507272660000,"startDate":"2017-10-05","endTime":"never","finalFireTime":"","scheduledSaveReportName":"TestScheduleReport","reportPath":"HelicalDemo/demo.efw","emailRecipients":["sayali@helicaltech.com"],"emailSubject":"TestScheduleReport","emailBody":"Hello Sayali,\n \nWe are scheduling the HDI Demo

2.7 About Page

URL	getProductInformation.html
Description	This page includes information about product such as product type, product name, expiry date, license type, build and version
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**GET, POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getProductInformation.html **Access through Curl command :** curl http://192.168.2.156:8085/hi-ee/getProductInformation.html -v
Response Output(JSON Format)	{ "Product Type": "Business Intelligence Framework", "Version": "2.0.0.7331 RC1", "Build": "R20170410_7331 RC1",

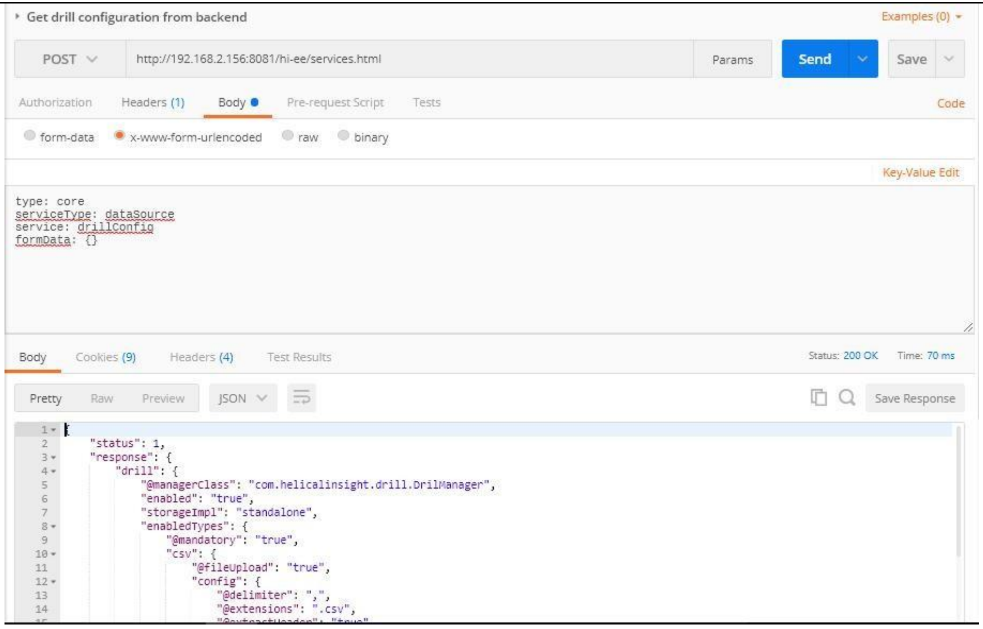
	"Product Name": "Helical Insight", "Expiration": "31/01/2018", "License Type": "Trial" }
Service Status	200 OK
Screenshot	 GET Params Send Save Authorization Headers Body Pre-request Script Tests Cookies Code Body Cookies (5) Headers (7) Tests Pretty Raw Preview http://192.168.2.156:8085/hi-ee/getProductInformation.html Type No Auth Status: 200 OK Time: 36 ms Size: 504 B {"Product Type":"Business Intelligence Framework","Version":"2.0.0.7331 RC1","Build":"R20170410_7331 RC1","Product Name":"Helical Insight","Expiration":"31/01/2018","License Type":"Trial"}

2.8 Management

2.8.1 Middleware

2.8.1.1 Get the drill configuration from backend

URL	services.html	
Description	User will get the drill configuration from backend which is saved in drillconfig.xml	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=drillConfig&formData={}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	core	Type of the Operation
serviceType:	dataSource	serviceType as dataSource
service:	drillConfig	Service name as drillConfig
formData:	{}	Action to get drill config info
Response Output (JSON format)	{"status":1,"response":{"drill":{"@managerClass":"com.helicalinsight.drill.DrilManager","enabled":"true","storageImpl":"standalone","enabledTypes":{"@mandatory":"true","csv":{"@fileUpload":"true","config":{"@delimiter":",","@extensions":".csv","@extractHeader":"true","@type":"text"}},"json":{"@fileUpload":"true","config":{"@extensions":".json","@type":"json"}},"parquet":{"@fileUpload":"true","config":{"@type":"parquet"}},"pcap":{"@fileUpload":"true","config":{"@type":"pcap"}},"tsv":{"@fileUpload":"true","config":{"@delimiter":" ","@extensions":".tsv","@extractHeader":"true","@type":"text"}}},"extractHeaders":"csv,csvh,tsv,psv","fileSystemConfiguration":{"hdfs":{"description":" Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running. Incase drill/middleware is installed in the Windows machine, please use linux sytle path in Datawarehouse path. Example /C:/Users/Helical/your/path/to/datawarehouse","host":[],"password":[],"port":"22","username":[]},"standalone":{"description":" Use standalone when middleware and helical insight are in the same machine. The dataware house path will be created inside the System Directory of the hi-repository folder. All the files uploaded will be saved in that location","subDescription":[]}},"url":"{{https}}://{{host}}:{{port}}","endPointsDetails":{"@endPointManager":"com.helicalinsight.adhoc.services.DrillEndPointManager","query":{"endpoint":"/query.json","actions":"select","method":"POST","output":"application/json"},"storage":{"endpoint":"/storage.json","actions":"create,edit,delete","method":"GET,POS	

	T","output":"application/json"},"threads":{"endpoint":"/thread.json","actions":"read","method":"GET,POST","output":"application/json"},"options":{"endpoint":"/option.json","actions":"read","method":"GET","output":"application/json"}},"urlConfig":{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":[],"securityEnabled":"true","securityMode":"plain","securityCheckType":"/j_security_check","username":"helical","password":"helical","httpsState":"false","https":"http","distributedMode":"false","zookeeperPort":"2181"},"drillStorageLocation":[{"@path":"/home/helical/testdrill"}]}}}05","scheduledTime":"12:36:00","lastExecutedOn":"1507187160002","reportParameters":{"TERRITORY":["Japan"],"mode":"dashboard"}}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK
Screenshot	

Get drill configuration from backend

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

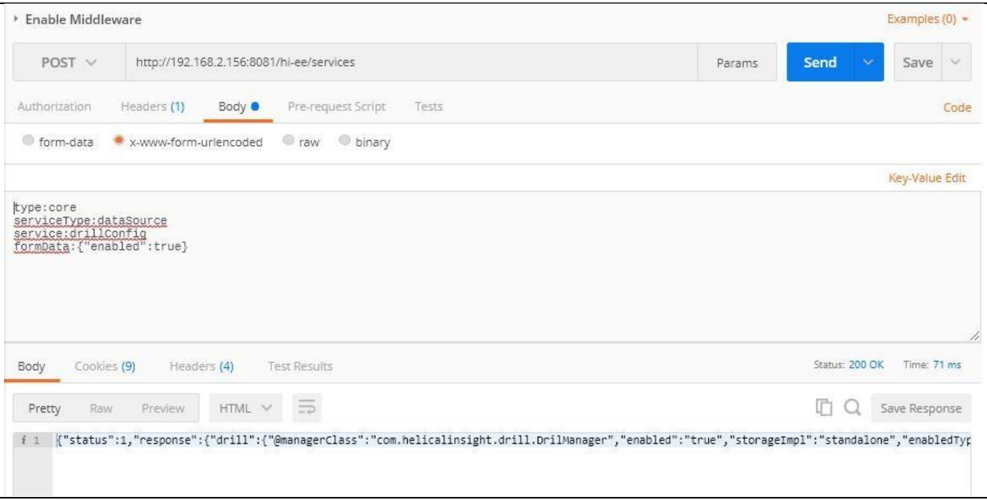
```
type: core
serviceType: dataSource
service: drillConfig
formData: {}
```

Status: 200 OK Time: 70 ms

```
{
    "status": 1,
    "response": {
        "drill": {
            "@managerClass": "com.helicalinsight.drill.DrilManager",
            "enabled": "true",
            "storageImpl": "standalone",
            "enabledTypes": {
                "@mandatory": "true",
                "csv": {
                    "@fileUpload": "true",
                    "config": {
                        "@delimiter": ",",
                        "@extensions": ".csv",
```

2.8.1.2 Enable Middleware

URL	services.html	
Description	User can enable middleware settings in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=drillConfig&formData={"enabled":true}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	core	Type of the Operation
serviceType:	dataSource	serviceType as dataSource
service:	drillConfig	Service name as drillConfig
formData:	{"enabled":true}	Action to enable drill config
Response Output (JSON format)	{"status":1,"response":{"drill":{"@managerClass":"com.helicalinsight.drill.DrilManager","enabled":"true","storageImpl":"standalone","enabledTypes":{"@mandatory":"true","csv":{"@fileUpload":"true","config":{"@delimiter":",","@extensions":".csv","@extractHeader":"true","@type":"text"}},"json":{"@fileUpload":"true","config":{"@extensions":".json","@type":"json"}},"parquet":{"@fileUpload":"true","config":{"@type":"parquet"}},"pcap":{"@fileUpload":"true","config":{"@type":"pcap"}},"tsv":{"@fileUpload":"true","config":{"@delimiter":" ","@extensions":".tsv","@extractHeader":"true","@type":"text"}}},"extractHeaders":"csv,csvh,tsv,psv","fileSystemConfiguration":{"hdfs":{"description":" Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running. Incase drill/middleware is installed in the Windows machine, please use linux sytle path in Datawarehouse path. Example /C:/Users/Helical/your/path/to/datawarehouse","host":[],"password":[],"port":"22","username":[]},"standalone":{"description":" Use standalone when middleware and helical insight are in the same machine. The dataware house path will be created inside the System Directory of the hi-repository folder. All the files uploaded will be saved in that	

	location","subDescription":[]}},"url":"{{https}}://{{host}}:{{port}}","endPointsDetails":{"@endPointManager":"com.helicalinsight.adhoc.services.DrillEndPointManager","query":{"endpoint":"/query.json","actions":"select","method":"POST","output":"application/json"},"storage":{"endpoint":"/storage.json","actions":"create,edit,delete","method":"GET,POST","output":"a* Connection #0 to host 192.168.2.156 left intact pplication/json"},"threads":{"endpoint":"/thread.json","actions":"read","method":"GET,POST","output":"application/json"},"options":{"endpoint":"/option.json","actions":"read","method":"GET","output":"application/json"}},"urlConfig":{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":[],"securityEnabled":"true","securityMode":"plain","securityCheckType":"/j_security_check","username":"helical","password":"helical","httpsState":"false","https":"http","distributedMode":"false","zookeeperPort":"2181"},"drillStorageLocation":[{"@path":"/home/helical/testdrill"}]}}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK
Screenshot	

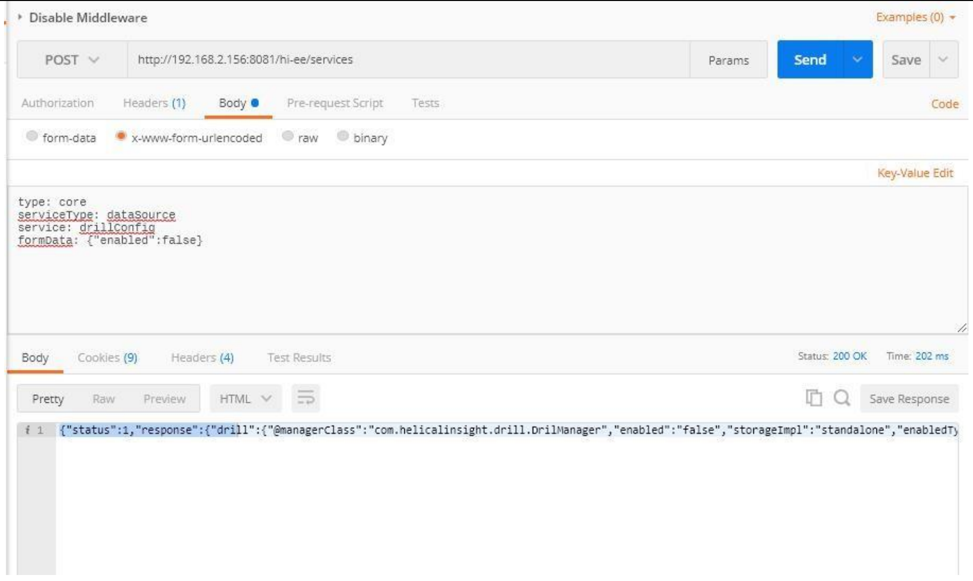
```
type:core
serviceType:dataSource
service:drillConfig
formData:{"enabled":true}
```

```
{"status":1,"response":{"drill":{"@managerClass":"com.helicalinsight.drill.DrilManager","enabled":"true","storageImpl":"standalone","enabledTyp
```

2.8.1.3 Disable Middleware

URL	services.html	
Description	User can disable middleware settings in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=drillConfig&formData={"enabled":false}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	core	Type of the Operation
serviceType:	dataSource	serviceType as dataSource
service:	drillConfig	Service name as drillConfig
formData:	{"enabled":false}	Action to disable drill config
Response Output (JSON format)	{"status":1,"response":{"drill":{"@managerClass":"com.helicalinsight.drill.DrilManager","enabled":"false","storageImpl":"standalone","enabledTypes":{"@mandatory":"true","csv":{"@fileUpload":"true","config":{"@delimiter":",","@extensions":".csv","@extractHeader":"true","@type":"text"}},"json":{"@fileUpload":"true","config":{"@extensions":".json","@type":"json"}},"parquet":{"@fileUpload":"true","config":{"@type":"parquet"}},"pcap":{"@fileUpload":"true","config":{"@type":"pcap"}},"tsv":{"@fileUpload":"true","config":{"@delimiter":" ","@extensions":".tsv","@extractHeader":"true","@type":"text"}}},"extractHeaders":"csv, csvh,tsv,psv","fileSystemConfiguration":{"hdfs":{"description":" Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running. Incase drill/middleware is installed in the Windows machine, please use linux sytle path in Datawarehouse path. Example /C:/Users/Helical/your/path/to/datawarehouse","host":[],"password":[],"port":"22","username":[]},"standalone":{"description":" Use standalone when middleware and helical insight are in the same machine. The dataware house path will be created inside the System Directory of the hi-repository folder. All the files uploaded will be saved in that location","subDescription":[]}},"url":"{{https}}://{{host}}:{{port}}","endPointsDetails":{"@endPointManager":"com.helicalinsight.adhoc.services.DrillEndPointManager","query":{"endpoint":"/query.json","actions":"select","method":"POST","output":"application/json"},"storage":{"endpoint":"/storage.json","actions":"create,edit,delete","method":"GET,POS	

	T","output":"* Connection #0 to host 192.168.2.156 left intact application/json"},"threads":{"endpoint":"/thread.json","actions":"read","method":"GET,POST","output":"application/json"},"options":{"endpoint":"/option.json","actions":"read","method":"GET","output":"application/json"}},"urlConfig":{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":[],"securityEnabled":"true","securityMode":"plain","securityCheckType":"/j_security_check","username":"helical","password":"helical","httpsState":"false","https":"http","distributedMode":"false","zookeeperPort":"2181"},"drillStorageLocation":[{"@path":"/home/helical/testdrill"}]}}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK

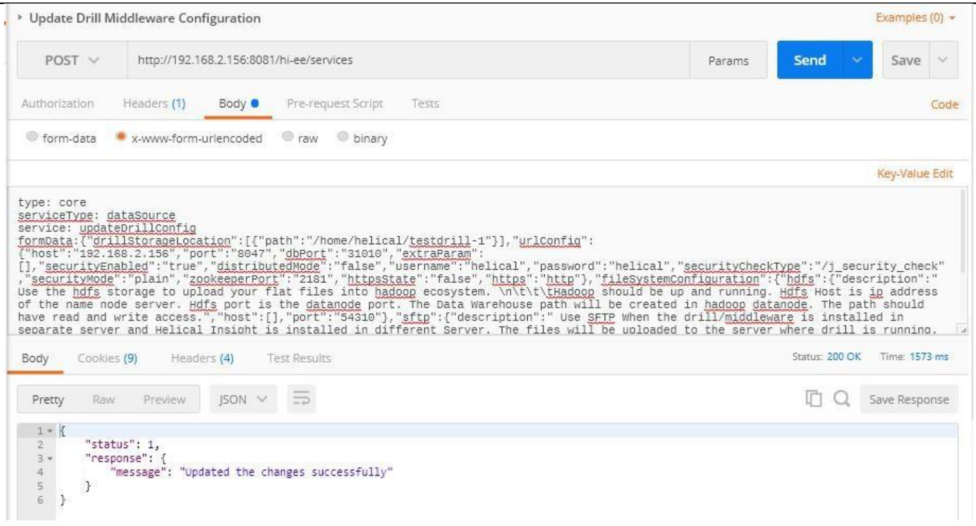
Screenshot	 Disable Middleware
	POST http://192.168.2.156:8081/hi-ee/services
	Body: x-www-form-urlencoded
	`type: core`
	`serviceType: dataSource`
	`service: drillConfig`
	`formData: {"enabled":false}`
	Status: 200 OK Time: 202 ms
	`{"status":1,"response":{"drill":{"@managerClass":"com.helicalinsight.drill.DrilManager","enabled":"false","storageImpl":"standalone","enabledTy` Examples (0)
	Params
	Send
	Save
	Authorization
	Headers (1)
	Body
	Pre-request Script
	Tests
	Code
	form-data
	x-www-form-urlencoded
	raw
	binary
	Key-Value Edit
	Body
	Cookies (9)
	Headers (4)
	Test Results
	Pretty
	Raw
	Preview
	HTML
	Save Response

2.8.1.4 Update Drill Middleware Configuration

URL	services.html
Description	User can update drill middleware settings in management page.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=updateDrillConfig&formData={"drillStorageLocation":[{"path":"/home/helical/testdrill-1"}],"urlConfig":{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":[],"securityEnabled":"true","distributedMode":"false","username":"helical","password":"helical","securityCheckType":"/j_security_check","securityMode":"plain","zookeeperPort":"2181","httpsState":"false","https":"http"},"fileSystemConfiguration":{"hdfs":{"description":" Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running. Incase drill/middleware is installed in the Windows machine, please use linux sytle path in Datawarehouse path. Example /C:/Users/Helical/your/path/to/datawarehouse","host":[],"password":[],"port":"22","username":[]},"standalone":{"description":" Use standalone when middleware and helical insight are in the same machine. The dataware house path will be created inside the System Directory of the hi-repository folder. All the files uploaded will be saved in that location","subDescription":[]}},"enabled":true,"storageImpl":"standalone"}" http://192.168.2.156:8081/hi-ee/services.html -v

HTTP Request Key	HTTP Request Value	Description
type:	core	Type of the Operation
serviceType:	dataSource	serviceType as dataSource
service:	updateDrillConfig	Service name as updateDrillConfig
formData:	{"drillStorageLocation":[{"path":"/home/heli	Action to update drill config

	cal/testdrill-1"}],"urlConfig":{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":[],"securityEnabled":"true","distributedMode":"false","username":"helical","password":"helical","securityCheckType":"/j_security_check","securityMode":"plain","zookeeperPort":"2181","httpsState":"false","https":"http"},"fileSystemConfiguration":{"hdfs":{"description":" Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running. Incase drill/middleware is installed in the Windows machine, please use linux sytle path in Datawarehouse path. Example /C:/Users/Helical/your/path/to/datawarehouse","host":[],"password":[],"port":"22","username":[]},"standalone":{"description":" Use standalone when middleware and helical insight are in the same machine. The dataware house path will be created inside the System Directory of the hi-repository folder. All the files uploaded will be saved in that location","subDescription":[]}},"enabled":true,"storageImpl":"standalone"}	according to available storage implementation.
Response Output (JSON format)	{ "status": 1, "response": { "message": "Updated the changes successfully" } }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Update Drill Middleware Configuration

POST http://192.168.2.156:8081/hi-ee/services

Body: x-www-form-urlencoded

```
type: core
serviceType: dataSource
service: updateDrillConfig
formData:{"drillStorageLocation":[{"path":"/home/helical/testdrill-1"}],"urlConfig":
{"host":"192.168.2.156","port":"8047","dbPort":"31010","extraParam":
[],"securityEnabled":"true","distributedMode":"false","username":"helical","password":"helical","securityCheckType":"/j_security_check"
,"securityMode":"plain","zookeeperPort":"2181","httpsState":"false","https":"http"},"fileSystemConfiguration":{"hdfs":{"description":"
Use the hdfs storage to upload your flat files into hadoop ecosystem. \n\t\t\tHadoop should be up and running. Hdfs Host is ip address
of the name node server. Hdfs port is the datanode port. The Data Warehouse path will be created in hadoop datanode. The path should
have read and write access.","host":[],"port":"54310"},"sftp":{"description":" Use SFTP When the drill/middleware is installed in
separate server and Helical Insight is installed in different Server. The files will be uploaded to the server where drill is running.
```

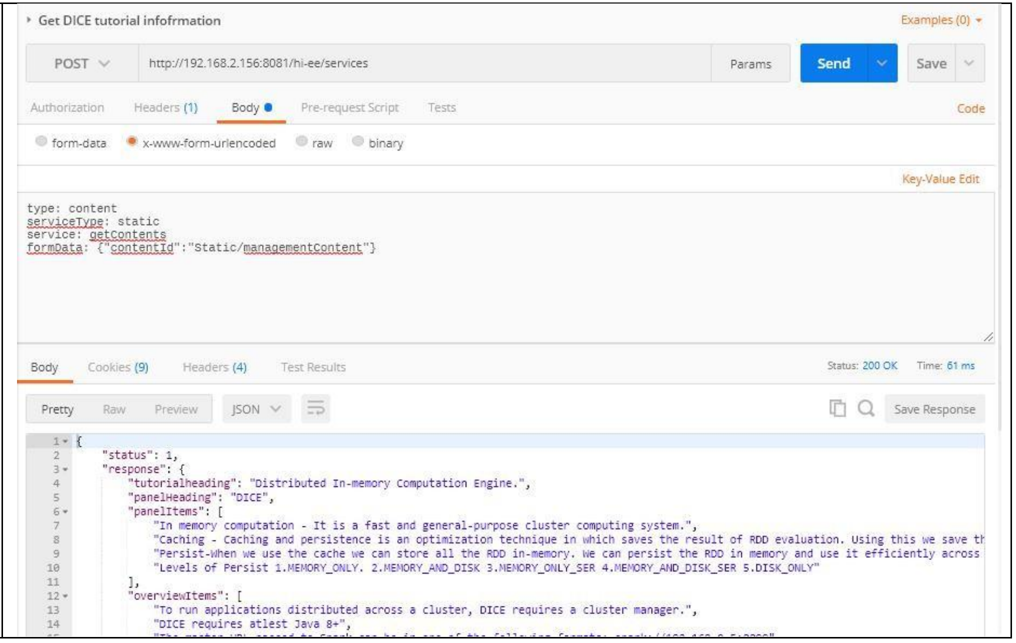
Status: 200 OK Time: 1573 ms

```
{
    "status": 1,
    "response": {
        "message": "Updated the changes successfully"
    }
}
```

2.8.2 DICE

2.8.2.1 Get DICE tutorial information

URL	services.html	
Description	User will get DICE tutorial information in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=content&serviceType=static&service=getcontents&formData={'contentId':'Static/managementContent'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	content	Type of the Operation
serviceType:	static	serviceType as static
service:	getContents	Service name as getcontents
formData:	{"contentId":"Static/managementContent"}	Action to get DICE tutorial information.
Response Output (JSON format)	{"status":1,"response":{"tutorialheading":"Distributed In-memory Computation Engine.","panelHeading":"DICE","panelItems":["In memory computation - It is a fast and general-purpose cluster computing system.","Caching - Caching and persistence is an optimization technique in which saves the result of RDD evaluation. Using this we save the intermediate result so that we can use it further if required. It reduces the computation overhead.","Persist-When we use the cache we can store all the RDD in-memory. We can persist the RDD in memory and use it efficiently across parallel operations.","Levels of Persist 1.MEMORY_ONLY. 2.MEMORY_AND_DISK 3.MEMORY_ONLY_SER 4.MEMORY_AND_DISK_SER 5.DISK_ONLY"],"overviewItems":["To run applications distributed across a cluster, DICE requires a cluster manager.","DICE requires atlest Java 8+","The master URL passed to Spark can be in one of the following formats: spark://192.168.0.5:8899","Configuration tab: Helps you to add configuration. You may also override an existing configuration","Advanced tab- This tab helps you to start stop services related to Master, Worker, Application and JDBC driver"],"enableMessage":"Enabling in memory computation may use excessive memory. Excessive RAM is required. This may hamper the application drastically in case you have less memory space. Are you sure?","disableMessage":"Disabling in memory computation may affect some of the datasources and reports as some of the reports may use in memory computation. Are you sure?"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Get DICE tutorial infofrmation

POST http://192.168.2.156:8081/hi-ee/services

Body: x-www-form-urlencoded

```
type: content
serviceType: static
service: getContents
formData: {"contentId":"Static/managementContent"}
```

Status: 200 OK Time: 61 ms

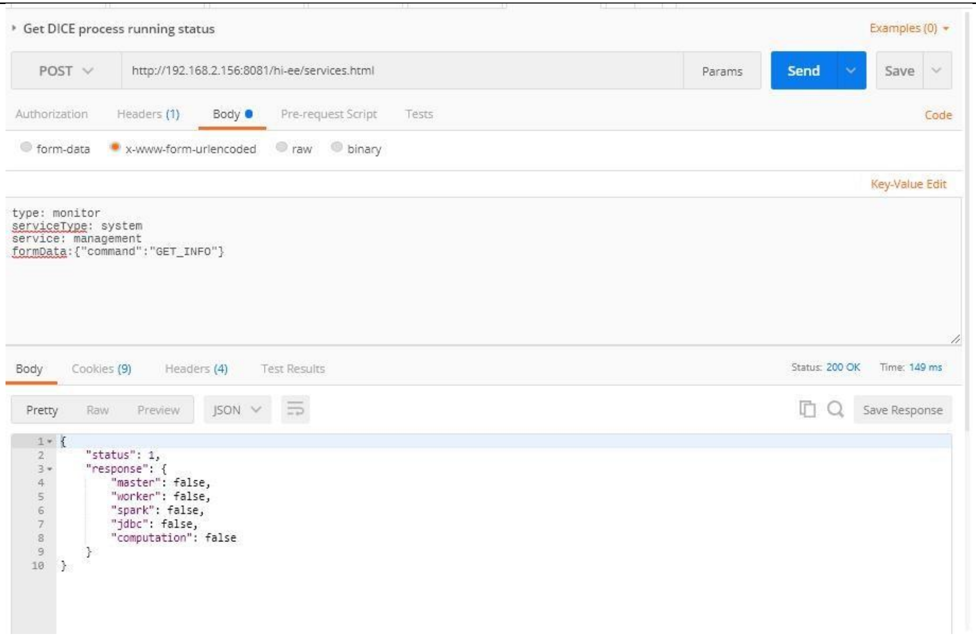
```
{
    "status": 1,
    "response": {
        "tutorialheading": "Distributed In-memory Computation Engine.",
        "panelHeading": "DICE",
        "panelItems": [
            "In memory computation - It is a fast and general-purpose cluster computing system.",
            "Caching - Caching and persistence is an optimization technique in which saves the result of RDD evaluation. Using this we save th
            "Persist-When we use the cache we can store all the RDD in-memory. We can persist the RDD in memory and use it efficiently across
            "Levels of Persist 1.MEMORY_ONLY. 2.MEMORY_AND_DISK 3.MEMORY_ONLY_SER 4.MEMORY_AND_DISK_SER 5.DISK_ONLY"
        ],
        "overviewItems": [
            "To run applications distributed across a cluster, DICE requires a cluster manager.",
            "DICE requires atlest Java 8+",
```

2.8.2.2 Get DICE process running status

URL	services.html	
Description	User will get the running status of DICE in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'GET_INFO'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"GET_INFO"}	Action to get DICE process running status.

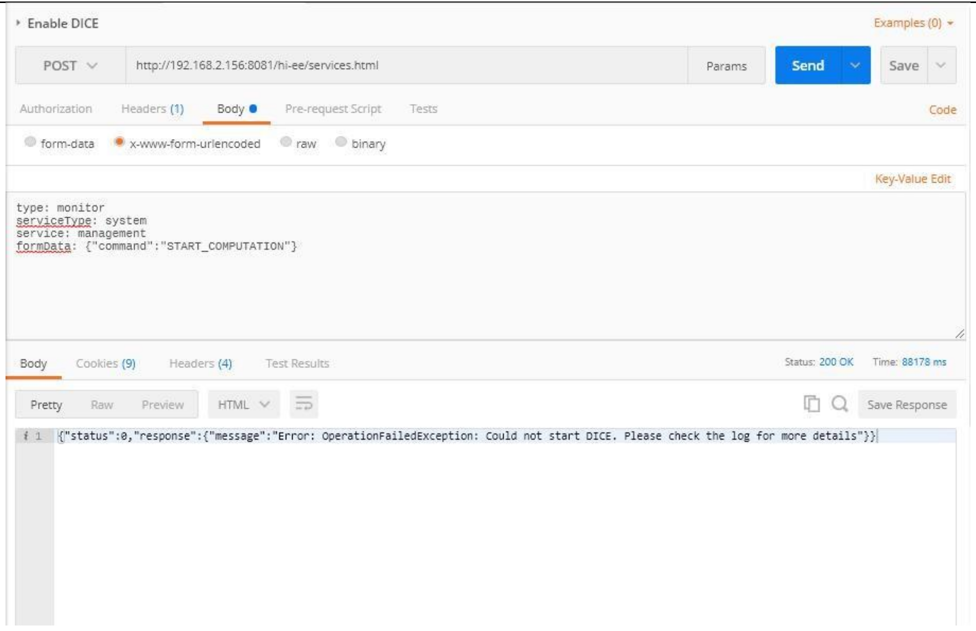
Response Output (JSON format)	{
	  "status": 1,
	  "response": {
	    "master": false,
	    "worker": false,
	    "spark": false,
	    "jdbc": false,
	    "computation": false
	  }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Get DICE process running status
	Examples (0)
	POST http://192.168.2.156:8081/hi-ee/services.html
	Params Send Save
	Authorization Headers (1) Body Pre-request Script Tests Code
	form-data x-www-form-urlencoded raw binary
	Key-Value Edit
	`type: monitor`
	`serviceType: system`
	`service: management`
	`formData:{"command":"GET_INFO"}`
	Body Cookies (9) Headers (4) Test Results Status: 200 OK Time: 149 ms
	Pretty Raw Preview JSON Save Response
	`{`
	`"status": 1,`
	`"response": {`
	`"master": false,`
	`"worker": false,`
	`"spark": false,`
	`"jdbc": false,`
	`"computation": false`
	`}`
	`}`

2.8.2.3 Enable DICE

URL	services.html	
Description	User can enable the DICE in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'START_COMPUTATION'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"START_COMPUTATION"}	Action to enable DICE
Response Output (JSON format)	{"status":0,"response":{"message":"Error: OperationFailedException: Could not start DICE. Please check the log for more details"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

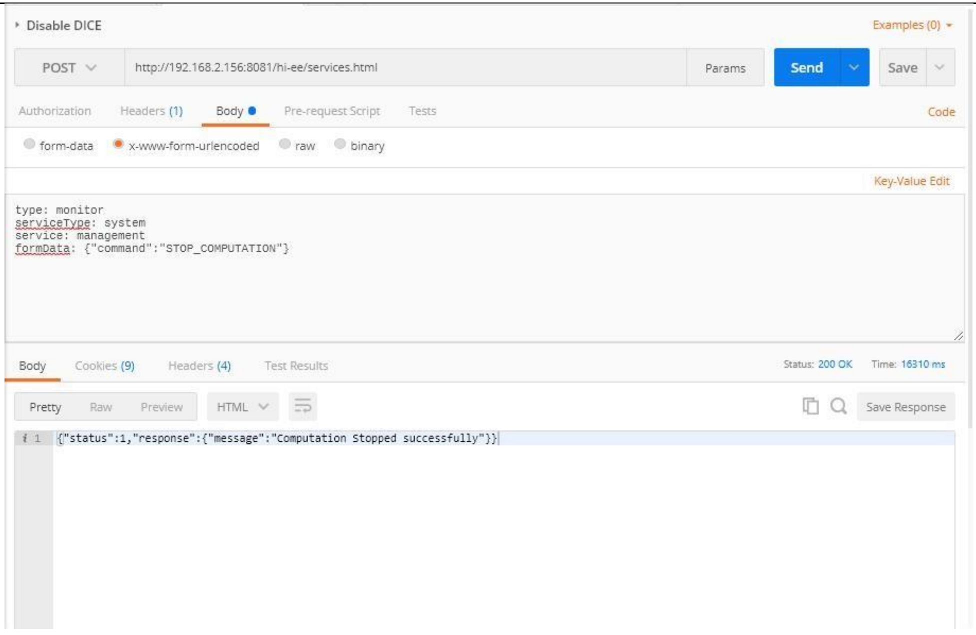
```
http://192.168.2.156:8081/hi-ee/services.html
```

```
type: monitor
serviceType: system
service: management
formData: {"command":"START_COMPUTATION"}
```

```
{"status":0,"response":{"message":"Error: OperationFailedException: Could not start DICE. Please check the log for more details"}}
```

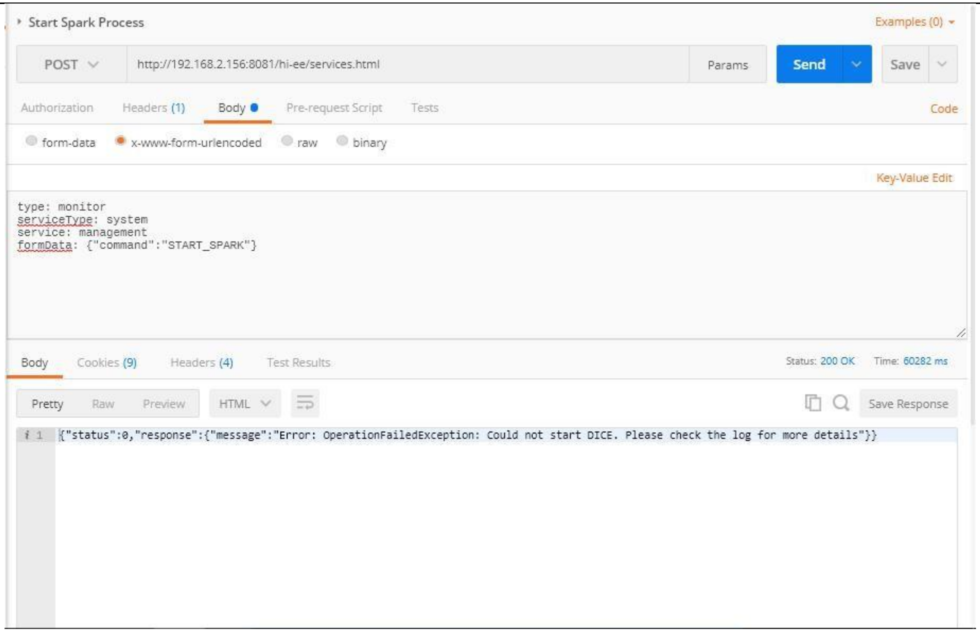
2.8.2.4 Disable DICE

URL	services.html	
Description	User can disable the DICE in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'STOP_COMPUTATION'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"STOP_COMPUTATION"}	Action to disable DICE
Response Output (JSON format)	{"status":1,"response":{"message":"Computation Stopped successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Disable DICE POST http://192.168.2.156:8081/hi-ee/services.html Body: x-www-form-urlencoded
	`type: monitor`
	`serviceType: system`
	`service: management`
	`formData: {"command":"STOP_COMPUTATION"}`
	Status: 200 OK Time: 16310 ms
	`{"status":1,"response":{"message":"Computation Stopped successfully"}}`

2.8.2.5 Start SPARK

URL	services.html	
Description	User can start spark process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'START_SPARK'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"START_SPARK"}	Action to start spark process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: OperationFailedException: Could not start DICE. Please check the log for more details"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

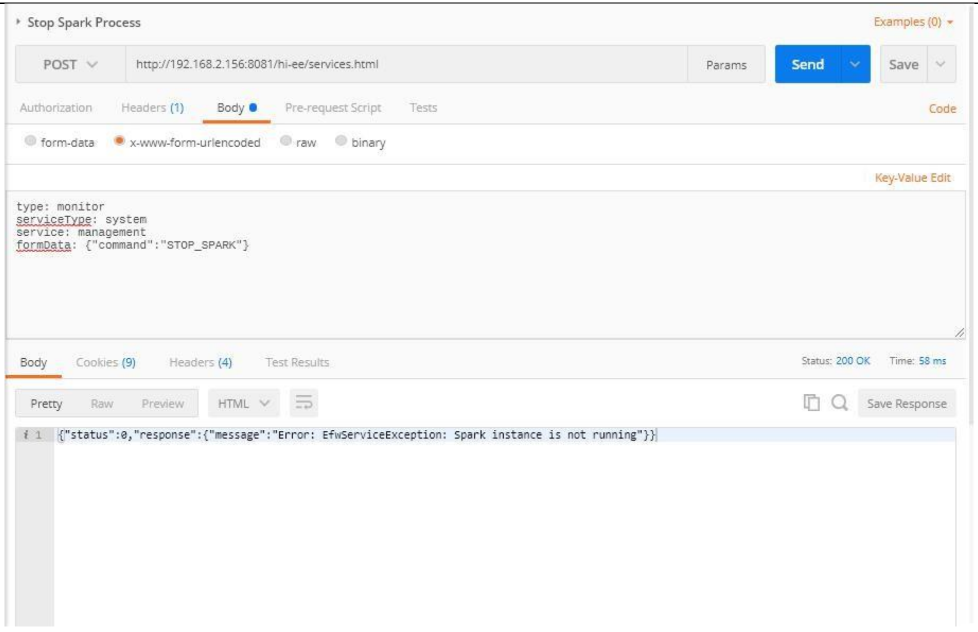
Screenshot	

```
type: monitor
serviceType: system
service: management
formData: {"command":"START_SPARK"}
```

```
{"status":0,"response":{"message":"Error: OperationFailedException: Could not start DICE. Please check the log for more details"}}
```

2.8.2.6 Stop SPARK

URL	services.html	
Description	User can stop spark process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'STOP_SPARK'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"STOP_SPARK"}	Action to stop spark process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: EfwServiceException: Spark instance is not running"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

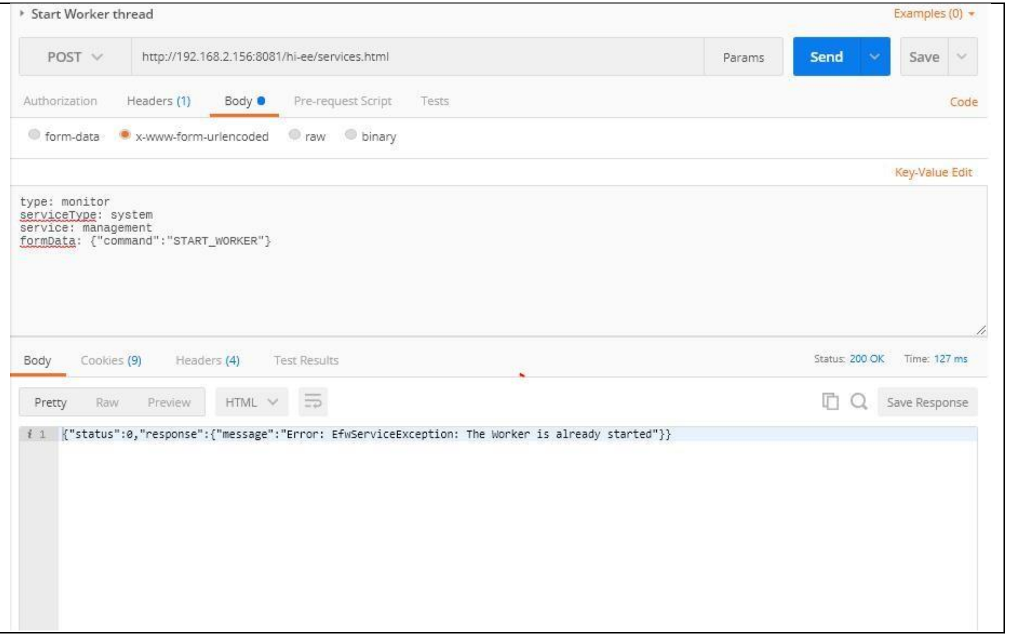
http://192.168.2.156:8081/hi-ee/services.html

```
type: monitor
serviceType: system
service: management
formData: {"command":"STOP_SPARK"}
```

```
{"status":0,"response":{"message":"Error: EfwServiceException: Spark instance is not running"}}
```

2.8.2.7 Start Worker thread

URL	services.html	
Description	User can start worker thread process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'START_WORKER'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"START_WORKER"}	Action to start worker thread process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: EfwServiceException: The Worker is already started"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

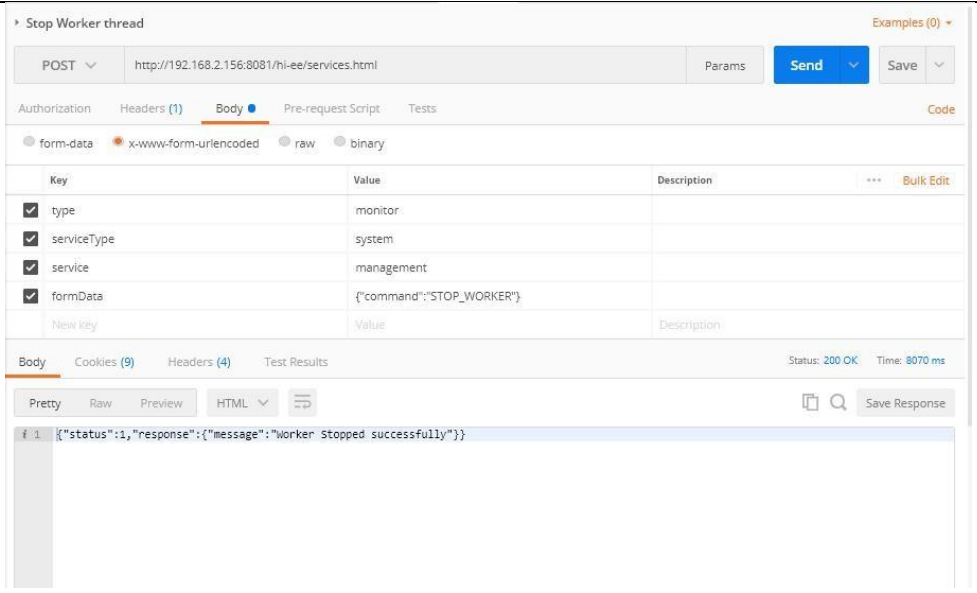
http://192.168.2.156:8081/hi-ee/services.html

```
type: monitor
serviceType: system
service: management
formData: {"command":"START_WORKER"}
```

```
{"status":0,"response":{"message":"Error: EfwServiceException: The Worker is already started"}}
```

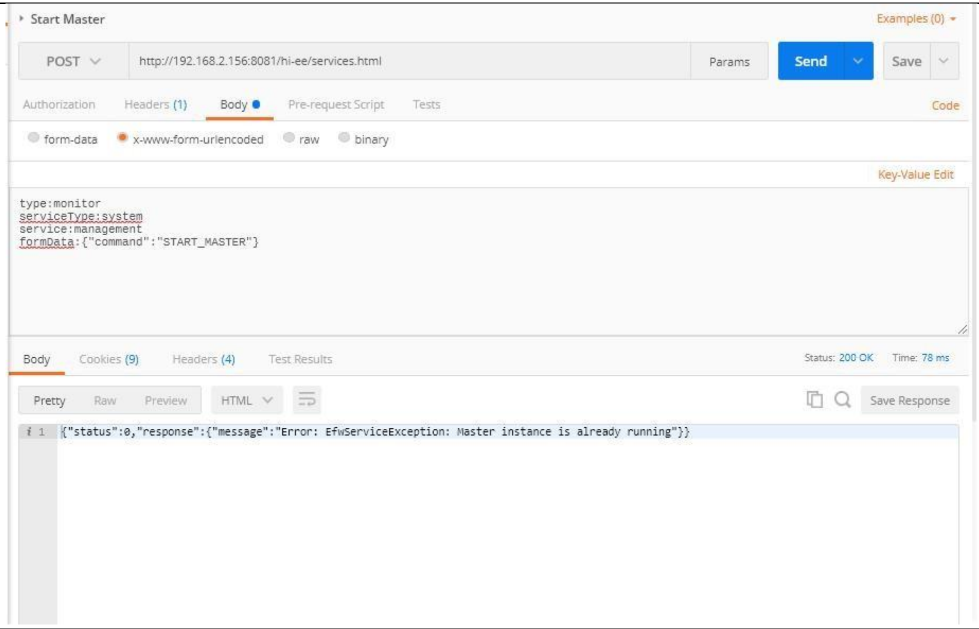
2.8.2.8 Stop Worker thread

URL	services.html	
Description	User can stop worker thread process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'STOP_WORKER’}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"STOP_WORKER"}	Action to stop worker process
Response Output (JSON format)	{"status":1,"response":{"message":"Worker Stopped successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Stop Worker thread POST http://192.168.2.156:8081/hi-ee/services.html Body: x-www-form-urlencoded Key / Value: type: monitor serviceType: system service: management formData: {"command":"STOP_WORKER"} Status: 200 OK Time: 8070 ms `{"status":1,"response":{"message":"Worker Stopped successfully"}}`

2.8.2.9 Start Master

URL	services.html	
Description	User can start master process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'START_MASTER'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"START_MASTER"}	Action to start master process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: EfwServiceException: Master instance is already running"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8081/hi-ee/services.html

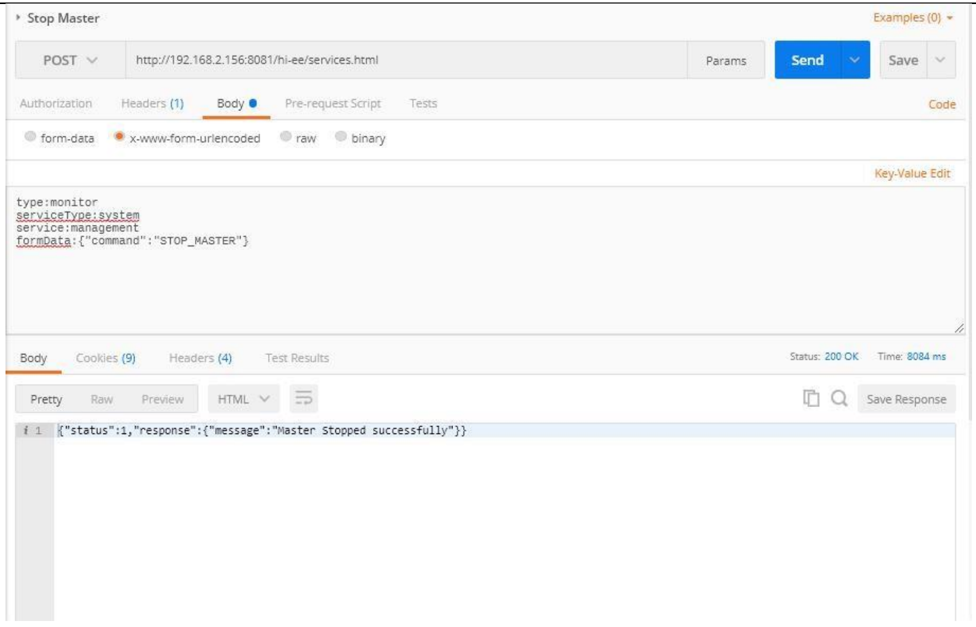
```
type:monitor
serviceType:system
service:management
formData:{"command":"START_MASTER"}
```

Status: 200 OK Time: 78 ms

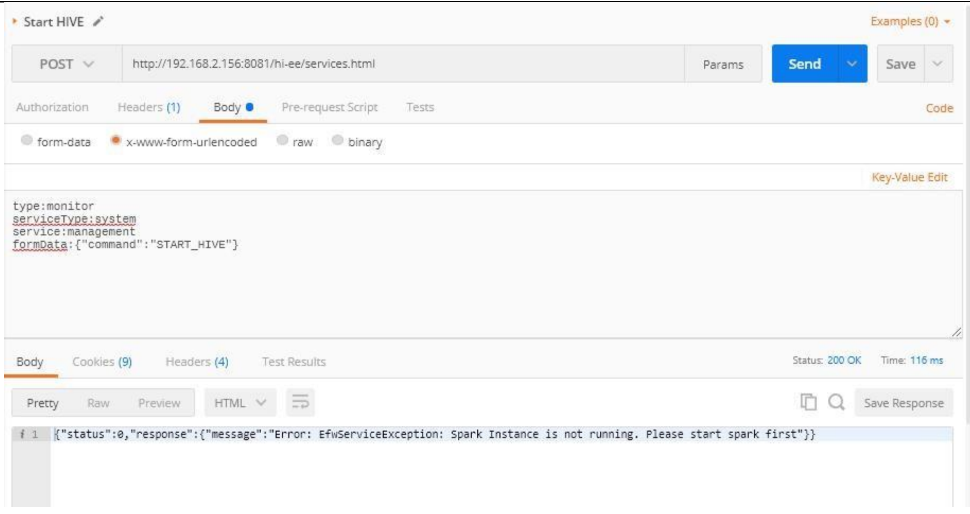
```
{"status":0,"response":{"message":"Error: EfwServiceException: Master instance is already running"}}
```

2.8.2.10 Stop Master

URL	services.html	
Description	User can stop worker thread process under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'STOP_MASTER'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"STOP_MASTER"}	Action to stop master process
Response Output (JSON format)	{"status":1,"response":{"message":"Worker Stopped successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Stop Master
	Examples (0)
	POST
	Params
	Send
	Save
	Authorization
	Headers (1)
	Body
	Pre-request Script
	Tests
	Code
	form-data
	x-www-form-urlencoded
	raw
	binary
	Key-Value Edit
	Body
	Cookies (9)
	Headers (4)
	Test Results
	Pretty
	Raw
	Preview
	HTML
	Save Response http://192.168.2.156:8081/hi-ee/services.html
	`type:monitor`
	`serviceType:system`
	`service:management`
	`formData:{"command":"STOP_MASTER"}`
	Status: 200 OK Time: 8084 ms
	`{"status":1,"response":{"message":"Master Stopped successfully"}}`

2.8.2.11 Start HIVE

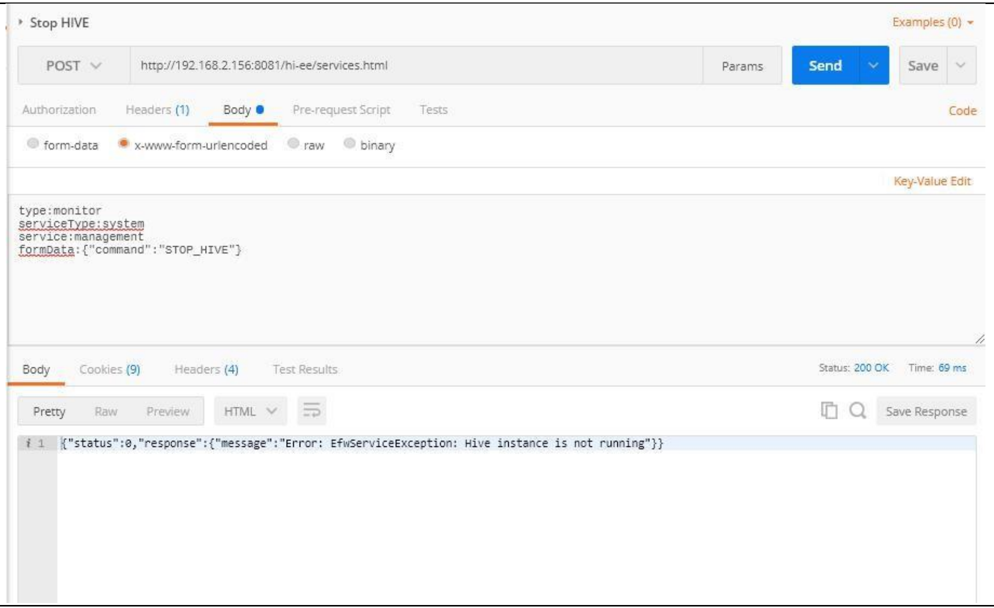
URL	services.html	
Description	User can start HIVE under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'START_HIVE'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"START_HIVE"}	Action to start HIVE process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: EfwServiceException: Spark Instance is not running. Please start spark first"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

```
type:monitor
serviceType:system
service:management
formData:{"command":"START_HIVE"}
```

```
{"status":0,"response":{"message":"Error: EfwServiceException: Spark Instance is not running. Please start spark first"}}
```

2.8.2.12 Stop HIVE

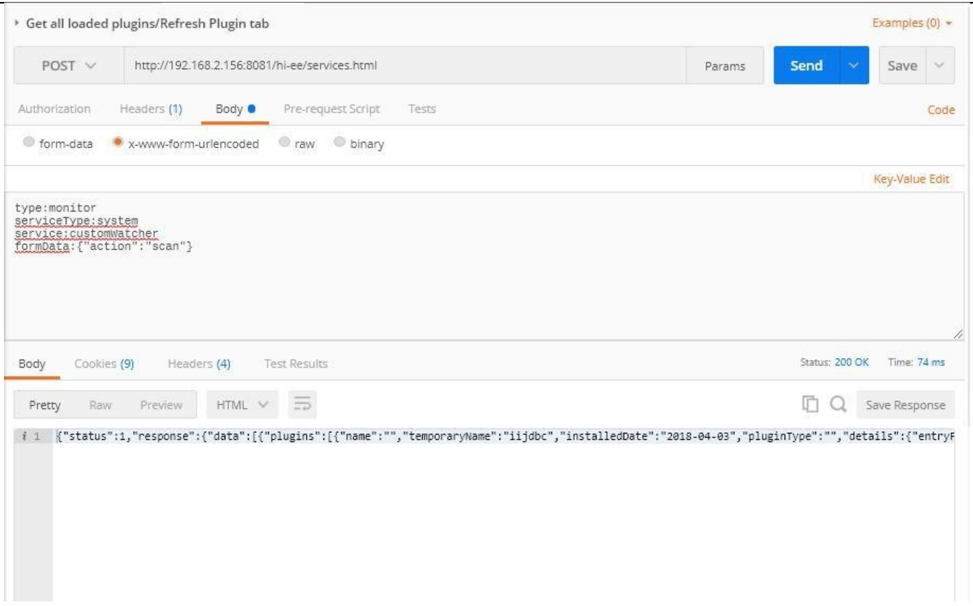
URL	services.html	
Description	User can stop HIVE under the DICE->Advanced in management page.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=management&formData={'command':'STOP_HIVE'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	management	Service name as management
formData:	{"command":"STOP_HIVE"}	Action to stop HIVE process
Response Output (JSON format)	{"status":0,"response":{"message":"Error: EfwServiceException: Spark Instance is not running. Please start spark first"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Stop HIVE
	POST http://192.168.2.156:8081/hi-ee/services.html
	x-www-form-urlencoded
	`type:monitor`
	`serviceType:system`
	`service:management`
	`formData:{"command":"STOP_HIVE"}`
	Status: 200 OK Time: 69 ms
	`{"status":0,"response":{"message":"Error: EfwServiceException: Hive instance is not running"}}`

2.9 Plugins

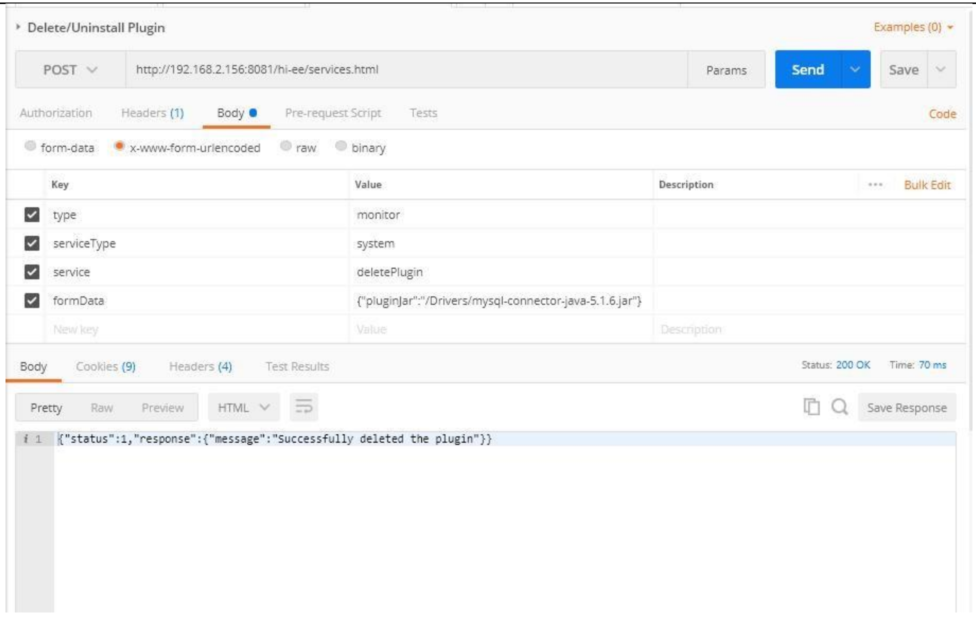
2.9.1 Get all loaded Plugins/Refresh Plugins

URL	services.html	
Description	User will get all loaded plugins from Driver/Plugins folder.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=customWatcher&formData={'action':'scan’}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	customWatcher	Service name as customWatcher
formData:	{"action":"scan"}	Action to scan/refresh the plugins
Response Output (JSON format)	{"status":1,"response":{"data":[{"plugins":[{"name":"","temporaryName":"iijdbc","installedDate":"2018-04-03","pluginType":"","details":{"entryPoint":[],"isDriver":"No","actualPath":"/Drivers/iijdbc.jar","classLoaderInstance":"","jarName":"iijdbc.jar"},"status":"Disabled"},{"name":"","temporaryName":"mysql-connector-java-5.1.42-bin","installedDate":"2018-12-05","pluginType":"","details":{"entryPoint":[],"isDriver":"No","actualPath":"/Drivers/mysql-connector-java-5.1.42-bin.jar","classLoaderInstance":"","jarName":"mysql-connector-java-5.1.42-bin.jar"},"status":"Disabled"},{"name":"","temporaryName":"sqlite-jdbc-3.20.0","installedDate":"2018-04-03","pluginType":"","details":{"entryPoint":[],"isDriver":"No","actualPath":"/Drivers/sqlite-jdbc-3.20.0.jar","classLoaderInstance":"","jarName":"sqlite-jdbc-3.20.0.jar"},"status":"Disabled"},{"name":"drill","temporaryName":"drill-jdbc-all-1.13.0","installedDate":"2018-06-25","pluginType":"DatbaseDriver","details":{"entryPoint":["org.apache.drill.jdbc.Driver"],"isDriver":"Yes","actualPath":"/Drivers/drill-jdbc-all-1.13.0.jar","classLoaderInstance":"com.helicalinsight.efw.framework.ParentLastClassLoader@1cf93585","jarName":"drill-jdbc-all-1.13.0.jar"},"status":"Enabled"},{"name":"oracle","temporaryName":"ojdbc6-11","installedDate":"2018-12-03","pluginType":"DatbaseDriver","details":{"entryPoint":["oracle.jdbc.OracleDriver"],"isDriver":"Yes","actualPath":"/Drivers/ojdbc6-11.jar","classLoaderInstance":"com.helicalinsight.efw.framework.ParentLastClassLoader@7b2c79c6","jarName":"ojdbc6-11.jar"},"status":"Enabled"},{"name":"","temporaryName":"mariadb-java-client-1.1.7","installedDate":"2018-04-03","pluginType":"","details":{"entryPoint":[],"isDriver":"No","actualPath":"/Drivers/mari	

	adb-java-client-1.1.7.jar","classLoaderInstance":"","jarName":"mariadb-java-client-1.1.7.jar"},"status":"Disabled"},{"name":"postgresql","temporaryName":"postgresql-9.3-1103.jdbc4","installedDate":"2018-04-03","pluginType":"DatbaseDriver","details":{"entryPoint":["org.postgresql.Driver"],"isDriver":"Yes","actualPath":"/Drivers/postgresql-9.3-1103.jdbc4.jar","classLoaderInstance":"com.helicalinsight.efw.framework.ParentLastClassLoader@7ffcfb60","jarName":"postgresql-9.3-1103.jdbc4.jar"},"status":"Enabled"},{"name":"sqlserver","temporaryName":"sybasejtds-1.3.1","installedDate":"2018-04-03","pluginType":"DatbaseDriver","details":{"entryPoint":["net.sourceforge.jtds.jdbc.Driver"],"isDriver":"Yes","actualPath":"/Drivers/sybasejtds-1.3.1.jar","classLoaderInstance":"com.helicalinsight.efw.framework.ParentLastClassLoader@541ccc24","jarName":"sybasejtds-1.3.1.jar"},"status":"Enabled"},{"name":"mysql","temporaryName":"mysql-connector-java-8.0.11","installedDate":"2018-12-05","pluginType":"DatbaseDriver","details":{"entryPoint":["com.mysql.cj.jdbc.Driver"],"isDriver":"Yes","actualPath":"/Drivers/mysql-connector-java-8.0.11.jar","classLoaderInstance":"com.helicalinsight.efw.framework.ParentLastClassLoader@3d0d786c","jarName":"mysql-connector-java-8.0.11.jar"},"status":"Enabled"},{"name":"","temporaryName":"hive","installedDate":"201 8-12-03","pluginType":"","details":{"entryPoint":["com.mysql.jdbc.Driver","com.mysql.cj.jdbc.Driver","com.mysql.cj.jdbc.Driver","com.mysql.cj.jdbc.Driver","com.mysql.cj.jdbc.Driver","com.mysql.cj.jdbc.Driver","com.mysql.cj.jdbc.Driver"],"isDriver":"No","actualPath":"/Plugins/hive","classLoaderInstance":"","jarName":"hive"},"status":"Enabled"}]}],"message":"Plugin Refresh Successfully"}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK
Screenshot	 Get all loaded plugins/Refresh Plugin tab POST http://192.168.2.156:8081/hi-ee/services.html Body: x-www-form-urlencoded type:monitor serviceType:system service:customWatcher formData:{"action":"scan"} Status: 200 OK Time: 74 ms {"status":1,"response":{"data":[{"plugins":[{"name":"","temporaryName":"iijdbc","installedDate":"2018-04-03","pluginType":"","details":{"entryF

2.9.2 Uninstall/Delete Plugins

URL	services.html	
Description	User can uninstall/delete plugin from Driver/Plugins folder.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=deletePlugin&formData={'pluginJar':'/Drivers/mysql-connector-java-5.1.6.jar'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	deletePlugin	Service name as deletePlugin
formData:	{"pluginJar":"/Drivers/mysql-connector-java-5.1.6.jar"}	Action to uninstall/delete the plugins
Response Output (JSON format)	{"status":1,"response":{"message":"Successfully deleted the plugin"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Delete/Uninstall Plugin

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

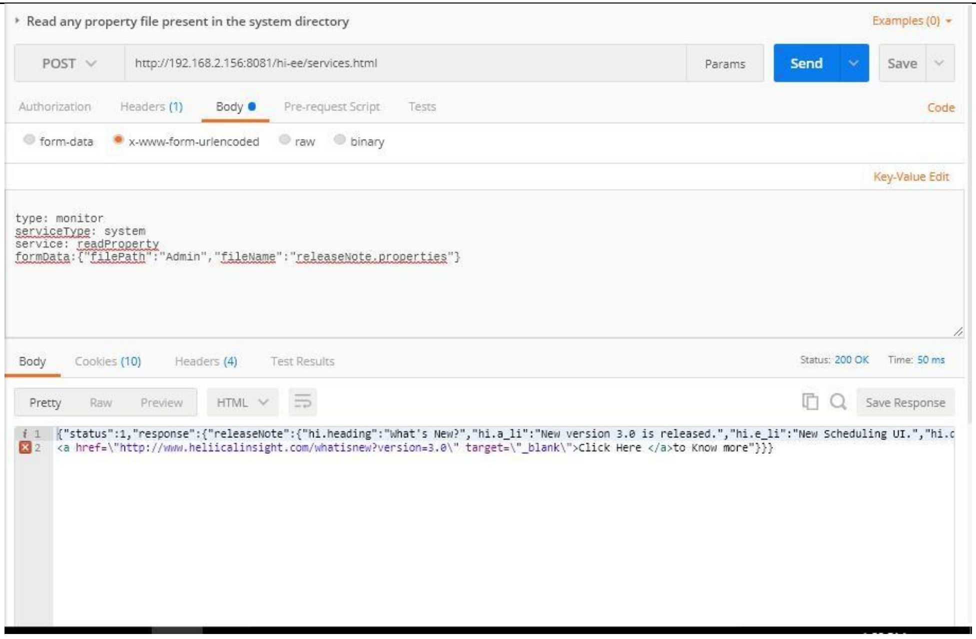
Key	Value	Description
type	monitor	
serviceType	system	
service	deletePlugin	
formData	{"pluginJar":"/Drivers/mysql-connector-java-5.1.6.jar"}	

Status: 200 OK Time: 70 ms

```
{"status":1,"response":{"message":"Successfully deleted the plugin"}}
```

2.10 Read any property file present in System directory

URL	services.html	
Description	User can read any property file which is present in System directory(hi-repository/System/Admin)	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	Access through browser : http://192.168.2.156:8081/hi-ee/services.html Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=monitor&serviceType=system&service=readProperty&formData={'filePath':'Admin','fileName':'releaseNote.properties'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	HTTP Request Value	Description
type:	monitor	Type of the Operation
serviceType:	system	serviceType as system
service:	readProperty	Service name as readProperty
formData:	{"filePath":"Admin","fileName":"releaseNote.properties"}	Provide name of file which is present under hi-repository/System/Admin
Response Output (JSON format)	{"status":1,"response":{"releaseNote":{"hi.heading":"What's New?","hi.a_li":"New version 3.0 is released.","hi.e_li":"New Scheduling UI.","hi.d_li":"Direct links to tutorials.","hi.b_li":"New generation UI with one click access.","hi.c_li":"Exporting and printing bugs fixed.","hi.f_li":"<a href=\"http://www.heliicalinsight.com/whatisnew?version=3.0\" target=\"_blank\">Click Here </a>to Know more"}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Read any property file present in the system directory

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
type: monitor
serviceType: system
service: readProperty
formData:{"filePath":"Admin","fileName":"releaseNote.properties"}
```

Status: 200 OK Time: 50 ms

```
{"status":1,"response":{"releaseNote":{"hi.heading":"What's New?","hi.a_li":"New version 3.0 is released.","hi.e_li":"New Scheduling UI.","hi.c
<a href=\"http://www.heliicalinsight.com/whatisnew?version=3.0\" target=\"_blank\">Click Here </a>to Know more"}}}
```

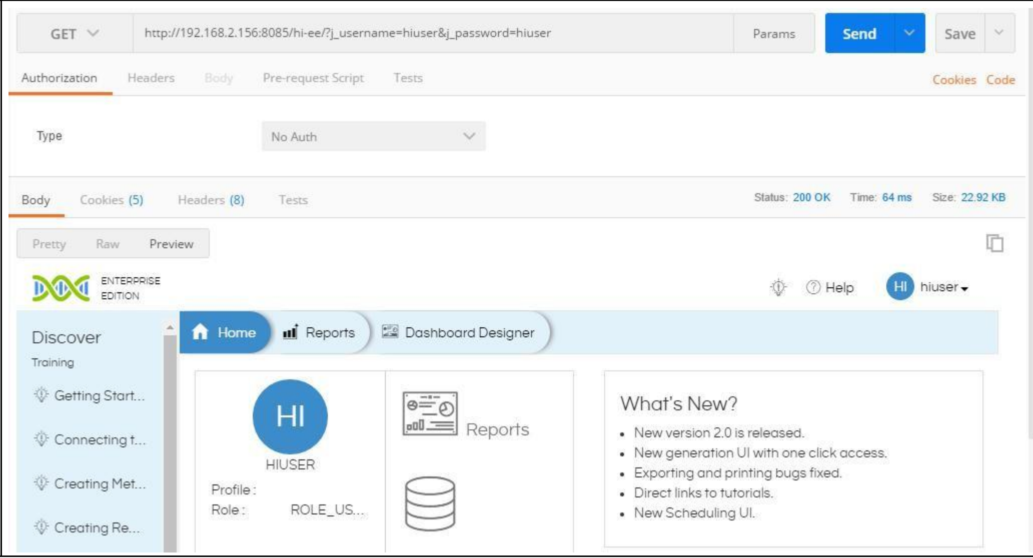
3.HI Module

HI module allows user to do file browser operations.HI module allows you to open or rename or export/import the different types of report.It allows you to take export of report in different file formats.

User can filter reports in file browser.

3.1 User Login

URL	?j_username=hiuser&j_password=hiuser	
Description	Any user can log into the application.	
Pre-requisite	Helical Insight application should be up/running.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/?j_username=hiuser&j_password=hiuser **Access through Curl command :** curl --data "j_username=hiuser&j_password=hiuser" http://192.168.2.156:8085/hi-ee/ -v	
HTTP Request Key	**HTTP Request Value**	**Description**
j_username	hiuser	Username for login
j_password	hiuser	Password for login
j_organization **(Optional)**		Organization name /blank if no organization
Service Status	200 OK	

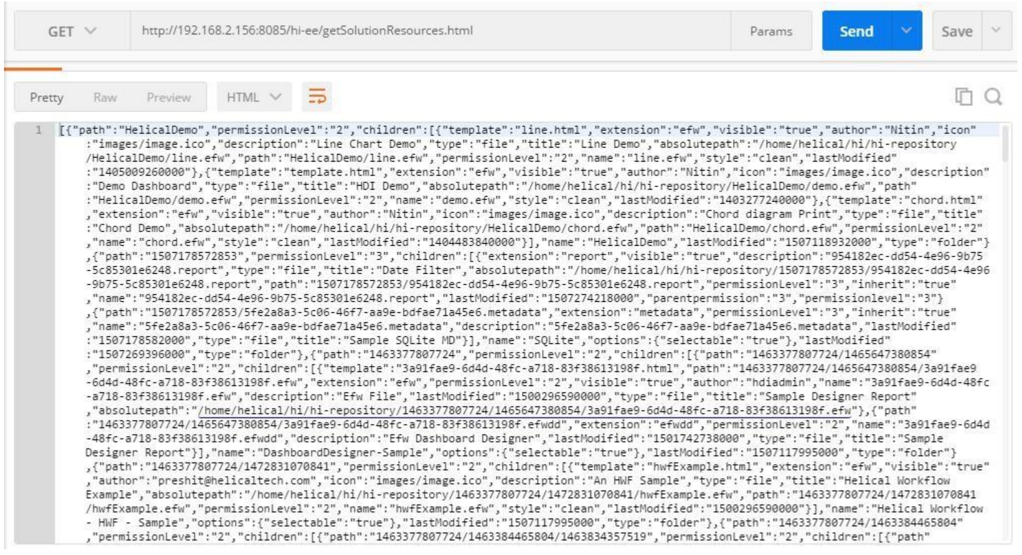
Screenshot	 GET http://192.168.2.156:8085/hi-ee/?j_username=hiuser&j_password=hiuser Type: No Auth Status: 200 OK Time: 64 ms Size: 22.92 KB What's New? New version 2.0 is released. New generation UI with one click access. Exporting and printing bugs fixed. Direct links to tutorials. New Scheduling UI.

3.2 File Browser Operations

3.2.1 Open/Refresh File Browser

URL	getSolutionResources.html
Description	Loads all the resources i.e file system whichever is accessible to the logged in user. You can see the repository details with associated files and folders. It loads folders,files with permission levels.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN,ROLE_USER
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getSolutionResources.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin" http://192.168.2.156:8085/hi-ee/getSolutionResources.html -v
Response Output (JSON Format)	[{ "path": "HelicalDemo", "permissionLevel": "2", "children": [

	{ "template": "line.html", "extension": "efw", "visible": "true", "author": "Nitin", "icon": "images/image.ico", "description": "Line Chart Demo", "type": "file", "title": "Line Demo", "absolutepath": "/home/helical/hi/hi-repository/HelicalDemo/line.efw", "path": "HelicalDemo/line.efw", "permissionLevel": "2", "name": "line.efw", "style": "clean", "lastModified": "1405009260000" }], "name": "HelicalDemo", "lastModified": "1507118932000", "type": "folder" }]
Description of Response Output:	The response returned is the JSON array having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file **lastModified** : lastModified timestamp of file. Etc.
Service Status	200 OK

Screenshot		

```
http://192.168.2.156:8085/hi-ee/getSolutionResources.html
```

```
[{"path":"HelicalDemo","permissionLevel":"2","children":[{"template":"line.html","extension":"efw","visible":"true","author":"Nitin","icon"
    :"images/image.ico","description":"Line Chart Demo","type":"file","title":"Line Demo","absolutepath":"/home/helical/hi/hi-repository
    /HelicalDemo/line.efw","path":"HelicalDemo/line.efw","permissionLevel":"2","name":"line.efw","style":"clean","lastModified"
    :"1405009260000"},{"template":"template.html","extension":"efw","visible":"true","author":"Nitin","icon":"images/image.ico","description"
    :"Demo Dashboard","type":"file","title":"HDI Demo","absolutepath":"/home/helical/hi/hi-repository/HelicalDemo/demo.efw","path"
    :"HelicalDemo/demo.efw","permissionLevel":"2","name":"demo.efw","style":"clean","lastModified":"1403277240000"},{"template":"chord.html"
    ,"extension":"efw","visible":"true","author":"Nitin","icon":"images/image.ico","description":"Chord diagram Print","type":"file","title"
    :"Chord Demo","absolutepath":"/home/helical/hi/hi-repository/HelicalDemo/chord.efw","path":"HelicalDemo/chord.efw","permissionLevel":"2"
    ,"name":"chord.efw","style":"clean","lastModified":"1404483840000"}],"name":"HelicalDemo","lastModified":"1507118932000","type":"folder"}
    ,{"path":"1507178572853","permissionLevel":"3","children":[{"extension":"report","visible":"true","description":"954182ec-dd54-4e96-9b75
    -5c85301e6248.report","type":"file","title":"Date Filter","absolutepath":"/home/helical/hi/hi-repository/1507178572853/954182ec-dd54-4e96
    -9b75-5c85301e6248.report","path":"1507178572853/954182ec-dd54-4e96-9b75-5c85301e6248.report","permissionLevel":"3","inherit":"true"
    ,"name":"954182ec-dd54-4e96-9b75-5c85301e6248.report","lastModified":"1507274218000","parentpermission":"3","permissionlevel":"3"}
    ,{"path":"1507178572853/5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6.metadata","extension":"metadata","permissionLevel":"3","inherit":"true"
    ,"name":"5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6.metadata","description":"5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6.metadata","lastModified"
    :"1507178582000","type":"file","title":"Sample SQLite MD"}],"name":"SQLite","options":{"selectable":"true"},"lastModified"
    :"1507269396000","type":"folder"},{"path":"1463377807724","permissionLevel":"2","children":[{"path":"1463377807724/1465647380854"
    ,"permissionLevel":"2","children":[{"template":"3a91fae9-6d4d-48fc-a718-83f38613198f.html","path":"1463377807724/1465647380854/3a91fae9
    -6d4d-48fc-a718-83f38613198f.efw","extension":"efw","permissionLevel":"2","visible":"true","author":"hdiadmin","name":"3a91fae9-6d4d-48fc
    -a718-83f38613198f.efw","description":"Efw File","lastModified":"1500296590000","type":"file","title":"Sample Designer Report"
    ,"absolutepath":"/home/helical/hi/hi-repository/1463377807724/1465647380854/3a91fae9-6d4d-48fc-a718-83f38613198f.efw"},{"path"
    :"1463377807724/1465647380854/3a91fae9-6d4d-48fc-a718-83f38613198f.efwdd","extension":"efwdd","permissionLevel":"2","name":"3a91fae9-6d4d
    -48fc-a718-83f38613198f.efwdd","description":"Efw Dashboard Designer","lastModified":"1501742738000","type":"file","title":"Sample
    Designer Report"}],"name":"DashboardDesigner-Sample","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"}
    ,{"path":"1463377807724/1472831070841","permissionLevel":"2","children":[{"template":"hwfExample.html","extension":"efw","visible":"true"
    ,"author":"preshit@helicaltech.com","icon":"images/image.ico","description":"An HWF Sample","type":"file","title":"Helical Workflow
    Example","absolutepath":"/home/helical/hi/hi-repository/1463377807724/1472831070841/hwfExample.efw","path":"1463377807724/1472831070841
    /hwfExample.efw","permissionLevel":"2","name":"hwfExample.efw","style":"clean","lastModified":"1500296590000"}],"name":"Helical Workflow
    - HWF - Sample","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"},{"path":"1463377807724/1463384465804"
    ,"permissionLevel":"2","children":[{"path":"1463377807724/1463384465804/1463834357519","permissionLevel":"2","children":[{"path"
```

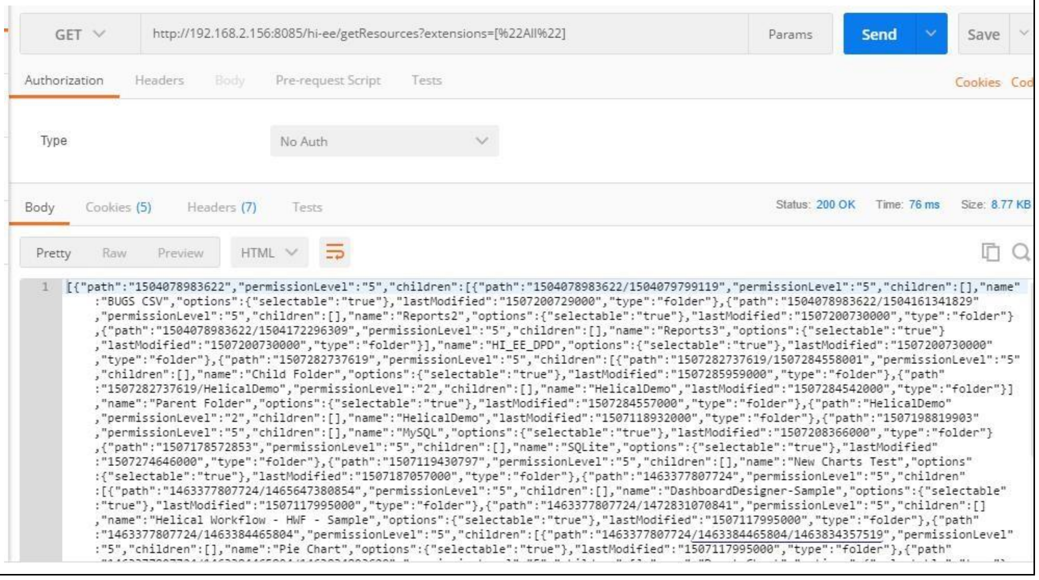
Post-action		Display the file structure in tree format . The user can perform file system based operations like cut- paste, adding, saving file etc.
Permission Level Interpretation	**0**	No access
	1	Execute Only
	2	Read only (Context menu just show open, open in new window and properties option)
	3	Read + Write (Context menu should show open, open in new window, edit (if any))
	4	Read + Write + Delete (Context menu should show open, open in new window, edit and delete)
	5	Read + Write + Delete + share (Context menu should show open, open innew window, edit, delete and share)
		### Creating a new folder and import can only happen where user has 3+ privileges"

3.2.2 Filter by type

3.2.2.1 All

URL	getResources?extensions=[%22All%22]
Description	Loads all the resources i.e file system whichever is accessible to the logged in user. It will show you the all resources related details like its sub-folder , name , fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request	**GET,POST**

Method	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22All%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22All%22]" http://192.168.2.156:8085/hi-ee/getResources
Response Output (JSON Format)	[{"path":"1501585888507","permissionLevel":"5","children":[{"path":"1501585888507/1502447762911","permissionLevel":"5","children":[{"path":"1501585888507/1502447762911/1504244039446","permissionLevel":"5","children":[],"name":"Dashboard Test","options":{"selectable":"true"},"lastModified":"1506093251000","type":"folder"}],"name":"Tested Reports","options":{"selectable":"true"},"lastModified":"1506093253000","type":"folder"},{"path":"1501585888507/1501670047455","permissionLevel":"5","children":[],"name":"HI Module Testing PreCondition","options":{"selectable":"true"},"lastModified":"1506317911000","type":"folder"}],"name":"AutomationTesting","options":{"selectable":"true"},"lastModified":"1506350827000","type":"folder"},{"path":"1506336430114","permissionLevel":"5","children":[],"name":"Test Created Folder","options":{"selectable":"true"},"lastModified":"1506336430000","type":"folder"},{"path":"HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1506093051000","type":"folder"},{"path":"1506344570988","permissionLevel":"5","children":[],"name":"SQL Server","options":{"selectable":"true"},"lastModified":"1506404658000","type":"folder"}]
Description of Response Output:	The response returned is the JSON array of all resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file

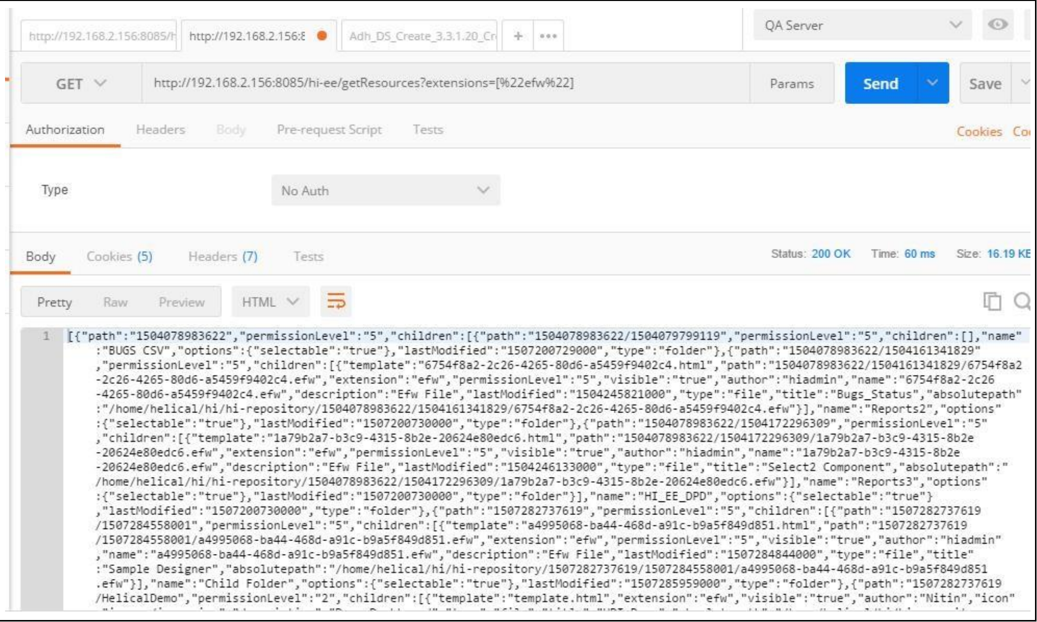
	lastModified : lastModified timestamp of file.
Service Status	200 OK
Screenshot	

```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}
    ,{"path":"1504078983622/1504172296309","permissionLevel":"5","children":[],"name":"Reports3","options":{"selectable":"true"}
    ,"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"},"lastModified":"1507200730000"
    ,"type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001","permissionLevel":"5"
    ,"children":[],"name":"Child Folder","options":{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path"
    :"1507282737619/HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507284542000","type":"folder"}]
    ,"name":"Parent Folder","options":{"selectable":"true"},"lastModified":"1507284557000","type":"folder"},{"path":"HelicalDemo"
    ,"permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507118932000","type":"folder"},{"path":"1507198819903"
    ,"permissionLevel":"5","children":[],"name":"MySQL","options":{"selectable":"true"},"lastModified":"1507208366000","type":"folder"}
    ,{"path":"1507178572853","permissionLevel":"5","children":[],"name":"SQLite","options":{"selectable":"true"},"lastModified"
    :"1507274646000","type":"folder"},{"path":"1507119430797","permissionLevel":"5","children":[],"name":"New Charts Test","options"
    :{"selectable":"true"},"lastModified":"1507187057000","type":"folder"},{"path":"1463377807724","permissionLevel":"5","children"
    :[{"path":"1463377807724/1465647380854","permissionLevel":"5","children":[],"name":"DashboardDesigner-Sample","options":{"selectable"
    :"true"},"lastModified":"1507117995000","type":"folder"},{"path":"1463377807724/1472831070841","permissionLevel":"5","children":[]
    ,"name":"Helical Workflow - HWF - Sample","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"},{"path"
    :"1463377807724/1463384465804","permissionLevel":"5","children":[{"path":"1463377807724/1463384465804/1463834357519","permissionLevel"
    :"5","children":[],"name":"Pie Chart","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"},{"path"
```

3.2.2.2 Report

URL	getResources?extensions=[%22efw%22]
Description	Loads all the efw report resources i.e file system whichever is accessible to the logged in user. It will show you the all efw report resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22efw%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22efw%22]" http://192.168.2.156:8085/hi-ee/getResources
Response Output(JSON	[{

Format)	`"path": "1504078983622", "permissionLevel": "5", "children": [ { "path": "1504078983622/1504079799119", "permissionLevel": "5", "children": [], "name": "BUGS CSV", "options": { "selectable": "true" }, "lastModified": "1507200729000", "type": "folder" } ] } ]`
Description of Response Output:	The response returned is the JSON array of all efw report resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file
Service Status	200 OK

Screenshot	

http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22efw%22]

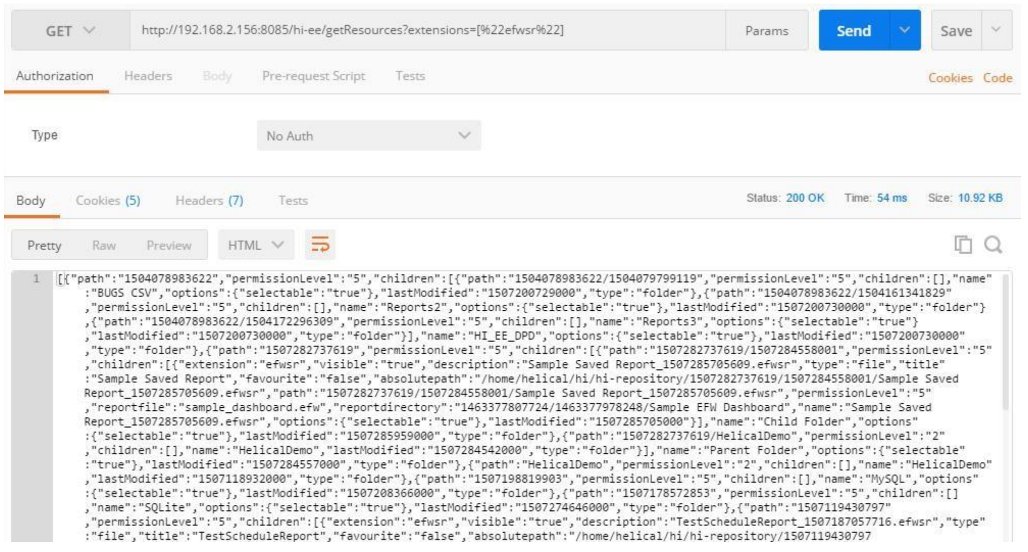
Type: No Auth

Status: 200 OK Time: 60 ms Size: 16.19 KB

```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[{"template":"6754f8a2-2c26-4265-80d6-a5459f9402c4.html","path":"1504078983622/1504161341829/6754f8a2
    -2c26-4265-80d6-a5459f9402c4.efw","extension":"efw","permissionLevel":"5","visible":"true","author":"hiadmin","name":"6754f8a2-2c26
    -4265-80d6-a5459f9402c4.efw","description":"Efw File","lastModified":"1504245821000","type":"file","title":"Bugs_Status","absolutepath"
    :"/home/helical/hi/hi-repository/1504078983622/1504161341829/6754f8a2-2c26-4265-80d6-a5459f9402c4.efw"}],"name":"Reports2","options"
    :{"selectable":"true"},"lastModified":"1507200730000","type":"folder"},{"path":"1504078983622/1504172296309","permissionLevel":"5"
    ,"children":[{"template":"1a79b2a7-b3c9-4315-8b2e-20624e80edc6.html","path":"1504078983622/1504172296309/1a79b2a7-b3c9-4315-8b2e
    -20624e80edc6.efw","extension":"efw","permissionLevel":"5","visible":"true","author":"hiadmin","name":"1a79b2a7-b3c9-4315-8b2e
    -20624e80edc6.efw","description":"Efw File","lastModified":"1504246133000","type":"file","title":"Select2 Component","absolutepath":"
    /home/helical/hi/hi-repository/1504078983622/1504172296309/1a79b2a7-b3c9-4315-8b2e-20624e80edc6.efw"}],"name":"Reports3","options"
    :{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"}
    ,"lastModified":"1507200730000","type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619
    /1507284558001","permissionLevel":"5","children":[{"template":"a4995068-ba44-468d-a91c-b9a5f849d851.html","path":"1507282737619
    /1507284558001/a4995068-ba44-468d-a91c-b9a5f849d851.efw","extension":"efw","permissionLevel":"5","visible":"true","author":"hiadmin"
    ,"name":"a4995068-ba44-468d-a91c-b9a5f849d851.efw","description":"Efw File","lastModified":"1507284844000","type":"file","title"
    :"Sample Designer","absolutepath":"/home/helical/hi/hi-repository/1507282737619/1507284558001/a4995068-ba44-468d-a91c-b9a5f849d851
    .efw"}],"name":"Child Folder","options":{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path":"1507282737619
    /HelicalDemo","permissionLevel":"2","children":[{"template":"template.html","extension":"efw","visible":"true","author":"Nitin","icon"
```

3.2.2.3 Saved Report

URL	getResources?extensions=[%22efwsr%22]
Description	Loads all the saved report resources i.e file system whichever is accessible to the logged in user. It will show you the all efwsr/saved report resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22efwsr%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22efwsr%22]" http://192.168.2.156:8085/hi-ee/getResources
Response Output(JSON Format)	[{"path":"1501585888507","permissionLevel":"5","children":[{"path":"1501585888507/1502447762911","permissionLevel":"5","children":[{"extension":"efwsr","visible":"true","description":"alwaysAnewPage_1502457346899.efwsr",

	"type":"file","title":"alwaysAnewPage","favourite":"false","absolutepath":"/home/helical/hi/hi-repository/1501585888507/1502447762911/alwaysAnewPage_1502457346899.efwsr","path":"1501585888507/1502447762911/alwaysAnewPage_1502457346899.efwsr","permissionLevel":"5","schedulingreference":"2","reportfile":"7ae144b2-0361-4640-b985-b3d15da1f5b2.efw","reportdirectory":"1463377807724/1473687108420","name":"alwaysAnewPage_1502457346899.efwsr","options":{"selectable":"true"},"lastModified":"1502457346000"}]
Description of Response Output:	The response returned is the JSON array of all efwsr report resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file Etc.
Service Status	200 OK
Screenshot	

GET http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22efwsr%22]

Type: No Auth

Status: 200 OK Time: 54 ms Size: 10.92 KB

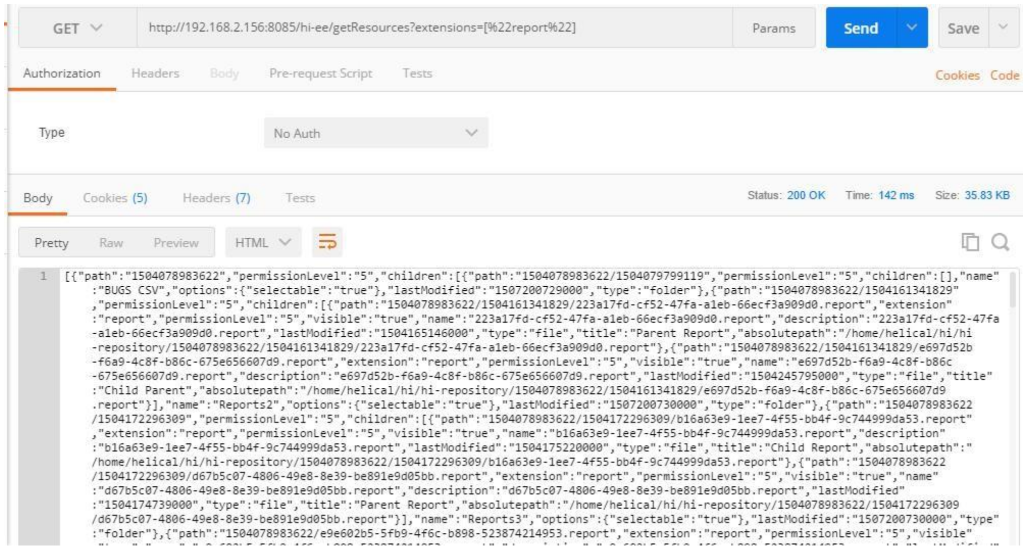
```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}
    ,{"path":"1504078983622/1504172296309","permissionLevel":"5","children":[],"name":"Reports3","options":{"selectable":"true"}
    ,"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"},"lastModified":"1507200730000"
    ,"type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001","permissionLevel":"5"
    ,"children":[{"extension":"efwsr","visible":"true","description":"Sample Saved Report_1507285705609.efwsr","type":"file","title"
    :"Sample Saved Report","favourite":"false","absolutepath":"/home/helical/hi/hi-repository/1507282737619/1507284558001/Sample Saved
    Report_1507285705609.efwsr","path":"1507282737619/1507284558001/Sample Saved Report_1507285705609.efwsr","permissionLevel":"5"
    ,"reportfile":"sample_dashboard.efw","reportdirectory":"1463377807724/1463377978248/Sample EFW Dashboard","name":"Sample Saved
    Report_1507285705609.efwsr","options":{"selectable":"true"},"lastModified":"1507285705000"}],"name":"Child Folder","options"
    :{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path":"1507282737619/HelicalDemo","permissionLevel":"2"
    ,"children":[],"name":"HelicalDemo","lastModified":"1507284542000","type":"folder"}],"name":"Parent Folder","options":{"selectable"
    :"true"},"lastModified":"1507284557000","type":"folder"},{"path":"HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo"
    ,"lastModified":"1507118932000","type":"folder"},{"path":"1507198819903","permissionLevel":"5","children":[],"name":"MySQL","options"
    :{"selectable":"true"},"lastModified":"1507208366000","type":"folder"},{"path":"1507178572853","permissionLevel":"5","children":[]
    ,"name":"SQLite","options":{"selectable":"true"},"lastModified":"1507274646000","type":"folder"},{"path":"1507119430797"
    ,"permissionLevel":"5","children":[{"extension":"efwsr","visible":"true","description":"TestScheduleReport_1507187057716.efwsr","type"
    :"file","title":"TestScheduleReport","favourite":"false","absolutepath":"/home/helical/hi/hi-repository/1507119430797
```

3.2.2.4 Adhoc Report

URL	getResources?extensions=[%22report%22]

Description	Loads all adhoc report resources i.e file system whichever is accessible to the logged in user. It will show you the all adhoc report resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22report%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22report%22]" http://192.168.2.156:8085/hi-ee/getResources

Response Output(JSON Format)	[
	  {
	    "path": "1504078983622",
	    "permissionLevel": "5",
	    "children": [
	      {
	        "path": "1504078983622/e9e602b5-5fb9-4f6c-b898-523874214953.report",
	        "extension": "report",
	        "permissionLevel": "5",
	        "visible": "true",
	        "name": "e9e602b5-5fb9-4f6c-b898-523874214953.report",
	        "description": "e9e602b5-5fb9-4f6c-b898-523874214953.report",
	        "lastModified": "1504251908000",
	        "type": "file",
	        "title": "CustomFilter",
	        "absolutepath": "/home/helical/hi/hi-repository/1504078983622/e9e602b5-5fb9-4f6c-b898-523874214953.report"
	      }
	    ],
	    "name": "HI_EE_DPD",
	    "options": {
	      "selectable": "true"
	    },
	    "lastModified": "1507200730000",
	    "type": "folder"

	}]
Description of Response Output:	The response returned is the JSON array of all adhoc report resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file **lastModified** : lastModified timestamp of file. Etc.
Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22report%22]

Status: 200 OK Time: 142 ms Size: 35.83 KB

```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[{"path":"1504078983622/1504161341829/223a17fd-cf52-47fa-a1eb-66ecf3a909d0.report","extension"
    :"report","permissionLevel":"5","visible":"true","name":"223a17fd-cf52-47fa-a1eb-66ecf3a909d0.report","description":"223a17fd-cf52-47fa
    -a1eb-66ecf3a909d0.report","lastModified":"1504165146000","type":"file","title":"Parent Report","absolutepath":"/home/helical/hi/hi
    -repository/1504078983622/1504161341829/223a17fd-cf52-47fa-a1eb-66ecf3a909d0.report"},{"path":"1504078983622/1504161341829/e697d52b
    -f6a9-4c8f-b86c-675e656607d9.report","extension":"report","permissionLevel":"5","visible":"true","name":"e697d52b-f6a9-4c8f-b86c
    -675e656607d9.report","description":"e697d52b-f6a9-4c8f-b86c-675e656607d9.report","lastModified":"1504245795000","type":"file","title"
    :"Child Parent","absolutepath":"/home/helical/hi/hi-repository/1504078983622/1504161341829/e697d52b-f6a9-4c8f-b86c-675e656607d9
    .report"}],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"},{"path":"1504078983622
    /1504172296309","permissionLevel":"5","children":[{"path":"1504078983622/1504172296309/b16a63e9-1ee7-4f55-bb4f-9c744999da53.report"
    ,"extension":"report","permissionLevel":"5","visible":"true","name":"b16a63e9-1ee7-4f55-bb4f-9c744999da53.report","description"
    :"b16a63e9-1ee7-4f55-bb4f-9c744999da53.report","lastModified":"1504175220000","type":"file","title":"Child Report","absolutepath":"
    /home/helical/hi/hi-repository/1504078983622/1504172296309/b16a63e9-1ee7-4f55-bb4f-9c744999da53.report"},{"path":"1504078983622
    /1504172296309/d67b5c07-4806-49e8-8e39-be891e9d05bb.report","extension":"report","permissionLevel":"5","visible":"true","name"
    :"d67b5c07-4806-49e8-8e39-be891e9d05bb.report","description":"d67b5c07-4806-49e8-8e39-be891e9d05bb.report","lastModified"
    :"1504174739000","type":"file","title":"Parent Report","absolutepath":"/home/helical/hi/hi-repository/1504078983622/1504172296309
    /d67b5c07-4806-49e8-8e39-be891e9d05bb.report"}],"name":"Reports3","options":{"selectable":"true"},"lastModified":"1507200730000","type"
    :"folder"},{"path":"1504078983622/e9e602b5-5fb9-4f6c-b898-523874214953.report","extension":"report","permissionLevel":"5","visible"
```

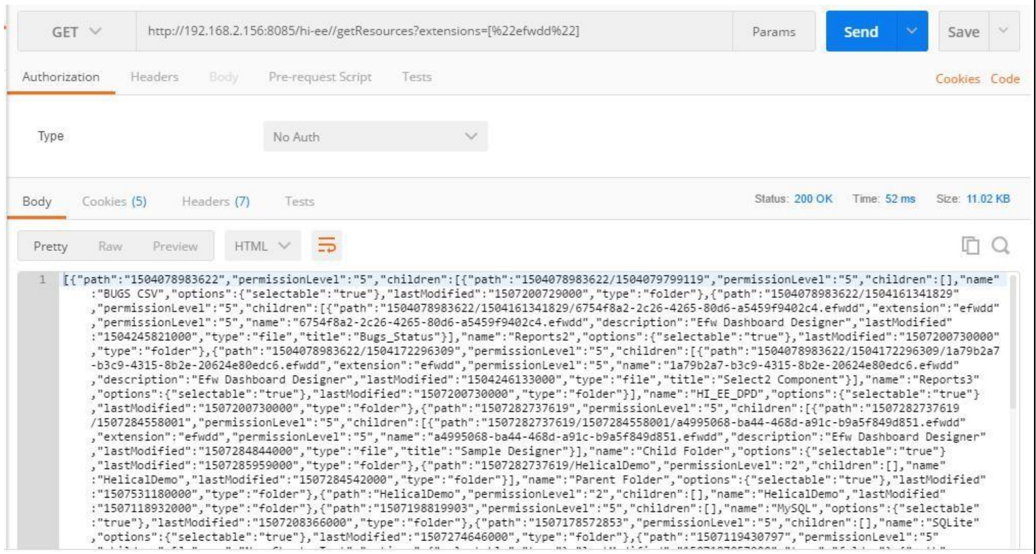
3.2.2.5 Dashboard Designer

URL	getResources?extensions=[%22efwdd%22]
Description	Loads all the dashboard resources i.e file system whichever is accessible to the logged in user. It will show you the all dashboard resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module]

	If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22efwdd%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22efwdd%22]" http://192.168.2.156:8085/hi-ee/getResources

Response Output(JSON Format)	[{
	"path": "1507282737619",
	"permissionLevel": "5",
	"children": [
	{
	"path": "1507282737619/1507284558001",
	"permissionLevel": "5",
	"children": [
	{
	"path": "1507282737619/1507284558001/a4995068-ba44-468d-a91c-b9a5f849d851.efwdd",
	"extension": "efwdd",
	"permissionLevel": "5",
	"name": "a4995068-ba44-468d-a91c-b9a5f849d851.efwdd",
	"description": "Efw Dashboard Designer",
	"lastModified": "1507284844000",
	"type": "file",
	"title": "Sample Designer"
	}],
	"name": "Child Folder",
	"options": {
	"selectable": "true"
	},
	"lastModified": "1507285959000",
	"type": "folder"
	},
	{
	"path": "1507282737619/HelicalDemo",
	"permissionLevel": "2",
	"children": [],
	"name": "HelicalDemo",
	"lastModified": "1507284542000",
	"type": "folder"

	}
	      ],
	      "name": "Parent Folder",
	      "options": {
	        "selectable": "true"
	      },
	      "lastModified": "1507531180000",
	      "type": "folder"
	   }
	]

Description of Response Output:	The response returned is the JSON array of all dashboard resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file etc.
Service Status	200 OK
Screenshot	 GET http://192.168.2.156:8085/hi-ee//getResources?extensions=[%22efwdd%22] Params Send Save Authorization Headers Body Pre-request Script Tests Cookies Code Type No Auth Body Cookies (5) Headers (7) Tests Pretty Raw Preview HTML Status: 200 OK Time: 52 ms Size: 11.02 KB `[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name":"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829","permissionLevel":"5","children":[{"path":"1504078983622/1504161341829/6754f8a2-2c26-4265-80d6-a5459f9402c4.efwdd","extension":"efwdd","permissionLevel":"5","name":"6754f8a2-2c26-4265-80d6-a5459f9402c4.efwdd","description":"Efw Dashboard Designer","lastModified":"1504245821000","type":"file","title":"Bugs_Status"}],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"},{"path":"1504078983622/1504172296309","permissionLevel":"5","children":[{"path":"1504078983622/1504172296309/1a79b2a7-b3c9-4315-8b2e-20624e80edc6.efwdd","extension":"efwdd","permissionLevel":"5","name":"1a79b2a7-b3c9-4315-8b2e-20624e80edc6.efwdd","description":"Efw Dashboard Designer","lastModified":"1504246133000","type":"file","title":"Select2 Component"}],"name":"Reports3","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001/a4995068-ba44-468d-a91c-b9a5f849d851.efwdd","extension":"efwdd","permissionLevel":"5","name":"a4995068-ba44-468d-a91c-b9a5f849d851.efwdd","description":"Efw Dashboard Designer","lastModified":"1507284844000","type":"file","title":"Sample Designer"}],"name":"Child Folder","options":{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path":"1507282737619/HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507284542000","type":"folder"}],"name":"Parent Folder","options":{"selectable":"true"},"lastModified":"1507531180000","type":"folder"},{"path":"HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507118932000","type":"folder"},{"path":"1507198819903","permissionLevel":"5","children":[],"name":"MySQL","options":{"selectable":"true"},"lastModified":"1507208366000","type":"folder"},{"path":"1507178572853","permissionLevel":"5","children":[],"name":"SQLite","options":{"selectable":"true"},"lastModified":"1507274646000","type":"folder"},{"path":"1507119430797","permissionLevel":"5"`

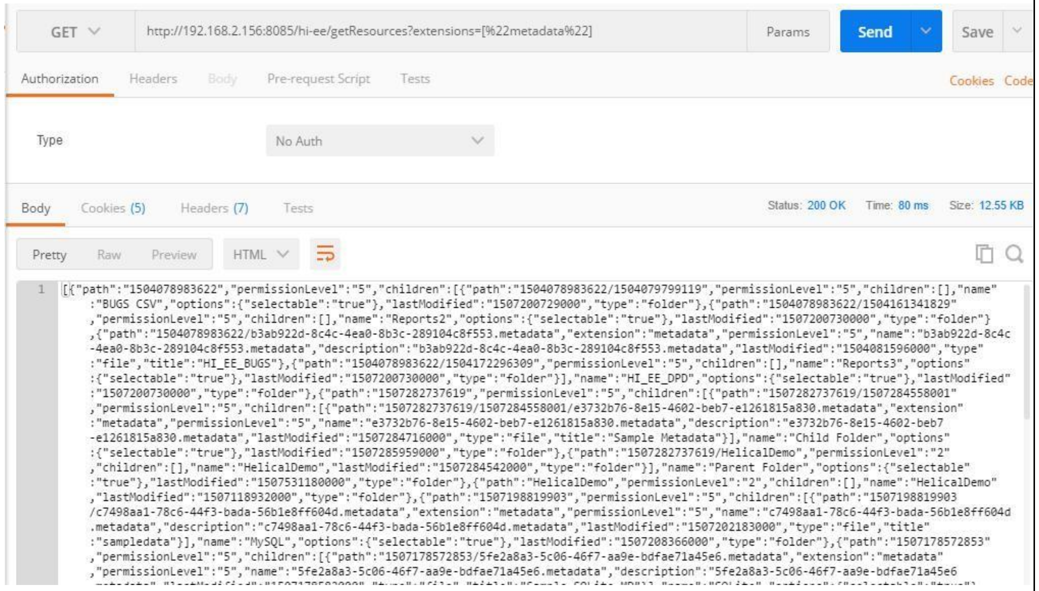
3.2.2.6 Metadata

URL	getResources?extensions=[%22metadata%22]

Description	Loads all the metadata resources i.e file system whichever is accessible to the logged in user. It will show you the all metadata resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22metadata%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22metadata%22]" http://192.168.2.156:8085/hi-ee/getResources

Response Output(JSON Output)	`[{`
	`"path": "1504078983622",`
	`"permissionLevel": "5",`
	`"children": [`
	`{`
	`"path": "1504078983622/b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata",`
	`"extension": "metadata",`
	`"permissionLevel": "5",`
	`"name": "b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata",`
	`"description": "b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata",`
	`"lastModified": "1504081596000",`
	`"type": "file",`
	`"title": "HI_EE_BUGS"`
	`}`
	`],`
	`"name": "HI_EE_DPD",`
	`"options": {`
	`"selectable": "true"`
	`},`
	`"lastModified": "1507200730000",`
	`"type": "folder"`
	`}]`

Description of Response Output:	The response returned is the JSON array of all metadata resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc.

	It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file Etc.
Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22metadata%22]

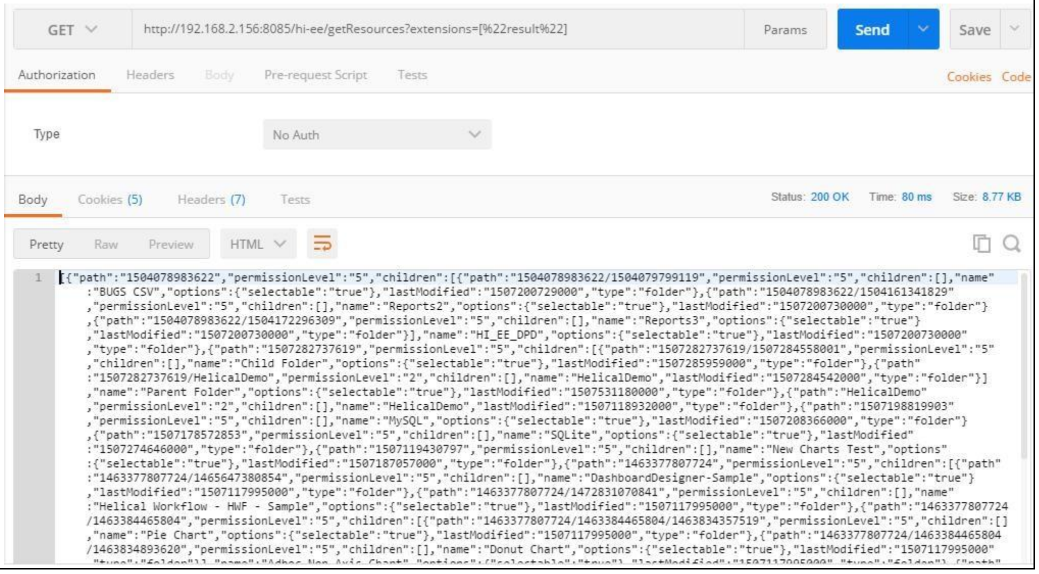
Type: No Auth

Status: 200 OK Time: 80 ms Size: 12.55 KB

```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}
    ,{"path":"1504078983622/b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata","extension":"metadata","permissionLevel":"5","name":"b3ab922d-8c4c
    -4ea0-8b3c-289104c8f553.metadata","description":"b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata","lastModified":"1504081596000","type"
    :"file","title":"HI_EE_BUGS"},{"path":"1504078983622/1504172296309","permissionLevel":"5","children":[],"name":"Reports3","options"
    :{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"},"lastModified"
    :"150720073000","type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001"
    ,"permissionLevel":"5","children":[{"path":"1507282737619/1507284558001/e3732b76-8e15-4602-beb7-e1261815a830.metadata","extension"
    :"metadata","permissionLevel":"5","name":"e3732b76-8e15-4602-beb7-e1261815a830.metadata","description":"e3732b76-8e15-4602-beb7
    -e1261815a830.metadata","lastModified":"1507284716000","type":"file","title":"Sample Metadata"}],"name":"Child Folder","options"
    :{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path":"1507282737619/HelicalDemo","permissionLevel":"2"
    ,"children":[],"name":"HelicalDemo","lastModified":"1507284542000","type":"folder"}],"name":"Parent Folder","options":{"selectable"
    :"true"},"lastModified":"1507531180000","type":"folder"},{"path":"HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo"
    ,"lastModified":"1507118932000","type":"folder"},{"path":"1507198819903","permissionLevel":"5","children":[{"path":"1507198819903
    /c7498aa1-78c6-44f3-bada-56b1e8ff604d.metadata","extension":"metadata","permissionLevel":"5","name":"c7498aa1-78c6-44f3-bada-56b1e8ff604d
    .metadata","description":"c7498aa1-78c6-44f3-bada-56b1e8ff604d.metadata","lastModified":"1507202183000","type":"file","title"
    :"sampledata"}],"name":"MySQL","options":{"selectable":"true"},"lastModified":"1507208366000","type":"folder"},{"path":"1507178572853"
    ,"permissionLevel":"5","children":[{"path":"1507178572853/5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6.metadata","extension":"metadata"
    ,"permissionLevel":"5","name":"5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6.metadata","description":"5fe2a8a3-5c06-46f7-aa9e-bdfae71a45e6
```

3.2.2.7 Result

URL	getResources?extensions=[%22result%22]
Description	Loads all the result resources i.e file system whichever is accessible to the logged in user. It will show you the all result resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET,POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getResources?extensions=[%22result%22] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=[%22result%22]" http://192.168.2.156:8085/hi-ee/getResources
Response Output(JSON Format)	[{"path":"1501585888507","permissionLevel":"5","children":[{"path":"1501585888507/1502447762911","permissionLevel":"5","children":[{"path":"1501585888507/1502447762911/1504244039446","permissionLevel":"5","children":[],"name":"Dashboard Test","options":{"selectable":"true"},"lastModified":"1506093251000","type":"folder"}],"name":"Tested Reports","options":{"selectable":"true"},"lastModified":"1506093253000","type":"folder"},{"path":"1501585888507/1501670047455","permissionLevel":"5","children":[],"name":"HI Module Testing PreCondition","options":{"selectable":"true"},"lastModified":"1506317911000","type":"folder"}],"name":"AutomationTesting","options":{"selectable":"true"},"lastModified":"1506350827000","type":"folder"},{"path":"1506336430114","permissionLevel":"5","children":[],"name":"Test Created Folder","options":{"selectable":"true"},"lastModified":"1506336430000","type":"folder"},{"path":"HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1506093051000","type":"folder"}]
Description of Response Output:	The response returned is the JSON array of all result resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder.

	• PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. **path** : Holds the physical name of the file/folder. **permissionLevel**: Permission level of the folder Click for more details **children** : Children array of directory **extension** : Extension of the file **title**: Title of the file **type** : type as folder/file Etc.
Service Status	200 OK
Screenshot	

```
[{"path":"1504078983622","permissionLevel":"5","children":[{"path":"1504078983622/1504079799119","permissionLevel":"5","children":[],"name"
    :"BUGS CSV","options":{"selectable":"true"},"lastModified":"1507200729000","type":"folder"},{"path":"1504078983622/1504161341829"
    ,"permissionLevel":"5","children":[],"name":"Reports2","options":{"selectable":"true"},"lastModified":"1507200730000","type":"folder"}
    ,{"path":"1504078983622/1504172296309","permissionLevel":"5","children":[],"name":"Reports3","options":{"selectable":"true"}
    ,"lastModified":"1507200730000","type":"folder"}],"name":"HI_EE_DPD","options":{"selectable":"true"},"lastModified":"1507200730000"
    ,"type":"folder"},{"path":"1507282737619","permissionLevel":"5","children":[{"path":"1507282737619/1507284558001","permissionLevel":"5"
    ,"children":[],"name":"Child Folder","options":{"selectable":"true"},"lastModified":"1507285959000","type":"folder"},{"path"
    :"1507282737619/HelicalDemo","permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507284542000","type":"folder"}]
    ,"name":"Parent Folder","options":{"selectable":"true"},"lastModified":"1507531180000","type":"folder"},{"path":"HelicalDemo"
    ,"permissionLevel":"2","children":[],"name":"HelicalDemo","lastModified":"1507118932000","type":"folder"},{"path":"1507198819903"
    ,"permissionLevel":"5","children":[],"name":"MySQL","options":{"selectable":"true"},"lastModified":"1507208366000","type":"folder"}
    ,{"path":"1507178572853","permissionLevel":"5","children":[],"name":"SQLite","options":{"selectable":"true"},"lastModified"
    :"1507274646000","type":"folder"},{"path":"1507119430797","permissionLevel":"5","children":[],"name":"New Charts Test","options"
    :{"selectable":"true"},"lastModified":"1507187057000","type":"folder"},{"path":"1463377807724","permissionLevel":"5","children":[{"path"
    :"1463377807724/1465647380854","permissionLevel":"5","children":[],"name":"DashboardDesigner-Sample","options":{"selectable":"true"}
    ,"lastModified":"1507117995000","type":"folder"},{"path":"1463377807724/1472831070841","permissionLevel":"5","children":[],"name"
    :"Helical Workflow - HWF - Sample","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"},{"path":"1463377807724
    /1463384465804","permissionLevel":"5","children":[{"path":"1463377807724/1463384465804/1463834357519","permissionLevel":"5","children":[]
    ,"name":"Pie Chart","options":{"selectable":"true"},"lastModified":"1507117995000","type":"folder"},{"path":"1463377807724/1463384465804
    /1463834893620","permissionLevel":"5","children":[],"name":"Donut Chart","options":{"selectable":"true"},"lastModified":"1507117995000"
```

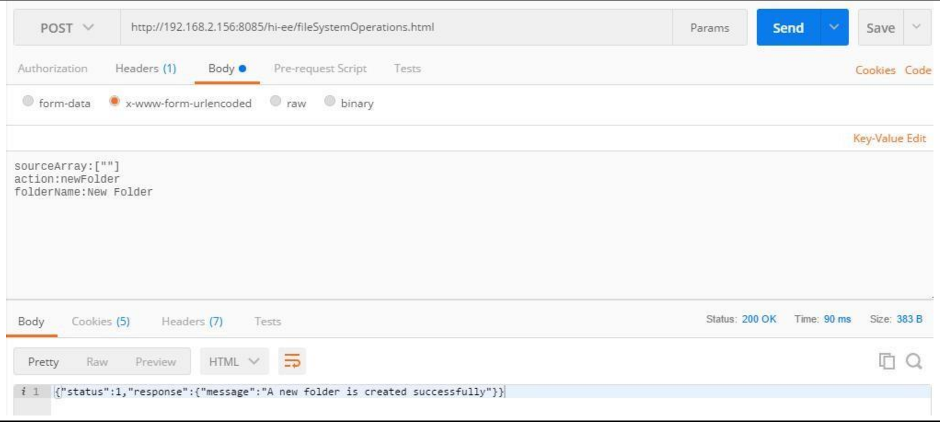
3.2.3 File Browser Empty Space

3.2.3.1 Create Empty space Folder

URL	fileSystemOperations.html
Description	Creates the empty space folder in repository. By default the name of empty space folder is New Folder.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN,ROLE_USER(Note: User should have write permission)
HTTP Request Method	**POST**
Example	**Access through browser :**

	http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=newFolder&folderName=New Folder&sourceArray=["]" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[""]	The directory/file where the add folder action is performed
action:	newFolder	Action as newFolder to create new folder.
folderName:	New Folder	By default for empty space folder New Folder as folderName is taken.

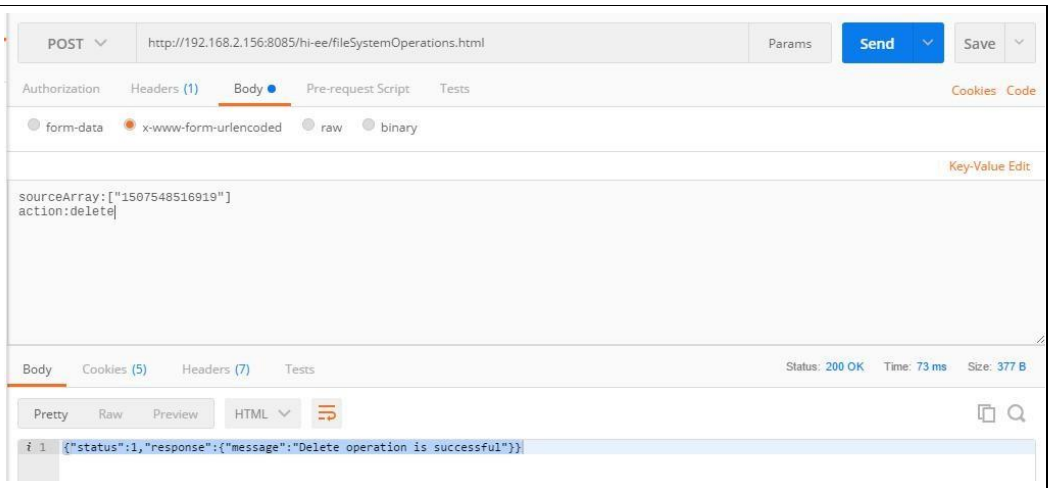
Response Output (JSON format)	{
	"status":1,"response":{"message":"A new folder is created successfully"}
	}	

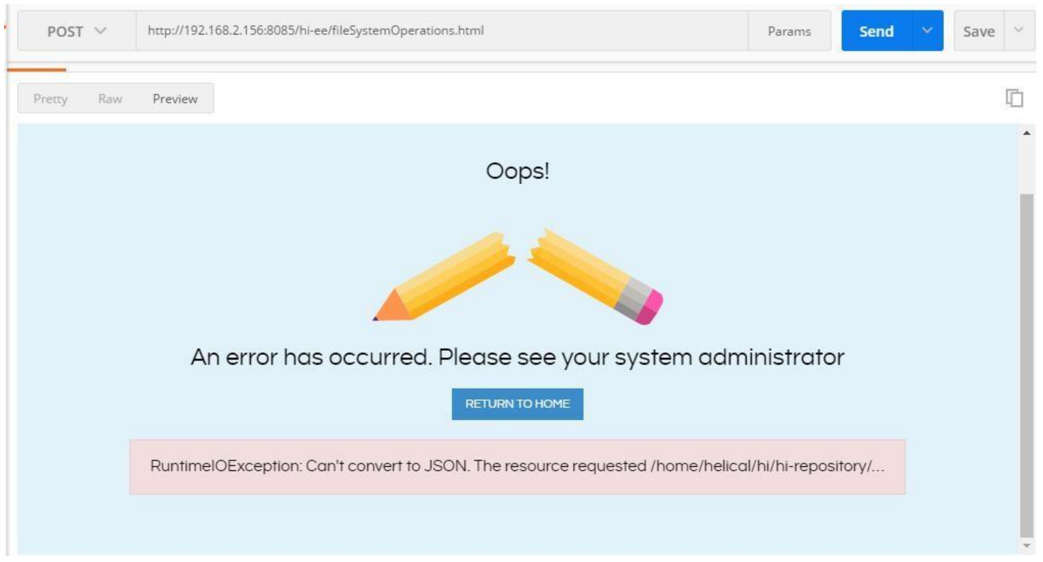
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	
Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html Body: x-www-form-urlencoded sourceArray:[""] action:newFolder folderName:New Folder Status: 200 OK Time: 90 ms Size: 383 B {"status":1,"response":{"message":"A new folder is created successfully"}}	
Post-action	After creation of empty space folder we can do folder rename, cut,paste,delete operations.	

3.3 File Browser :: Folder/File operations

3.3.1 Delete Folder/File

URL	fileSystemOperations.html	
Description	It allows user to delete the file/ folder if the user is permitted.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have delete permission)	
HTTP Request Method	POST	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&sourceArray=['1507548659227']" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507548516919"]]	The directory/file where the delete action is performed
action:	delete	File operation action type is delete
Response Output(JSON Format)	{"status":1,"response":{"message":"Delete operation is successful"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. The selected file /folder is permanently deleted form the system. This action is irreversible.	
Service Status	200 OK	

Screenshot(Success)	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html
	x-www-form-urlencoded
	`sourceArray:["1507548516919"]`
	`action:delete`
	Status: 200 OK Time: 73 ms Size: 377 B
	`{"status":1,"response":{"message":"Delete operation is successful"}}`

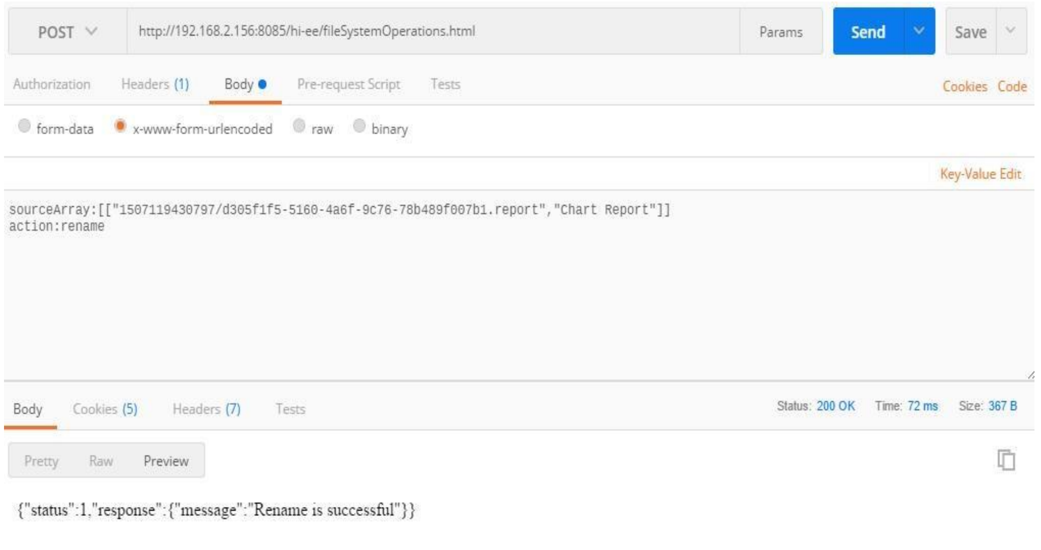
Possible Error	If the file/folder which you want to delete doesnot exists in that case you will get an error.
Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html Oops! An error has occurred. Please see your system administrator RETURN TO HOME RuntimeIOException: Can't convert to JSON. The resource requested /home/helical/hi/hi-repository/...

3.3.2 Rename Folder/File

URL	fileSystemOperations.html	
Description	It allows user to rename the file/folder if the user is permitted.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=rename&sourceArray=[['1507551052264','ReportList']]" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507119430797/d305f1f5-5160-4a6f-9c76-78b489f007b1.report","Chart Report"]]	The directory/file where the rename is performed We need to set the path of file and the to which name you want to rename. Note : report path you will get it from rightclick->file/folder ->properties
action:	rename	File operation action type is rename

Response Output(JSON Format)	{"status": 1,
	"Response":
	{ "message": "Rename is successful" }
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

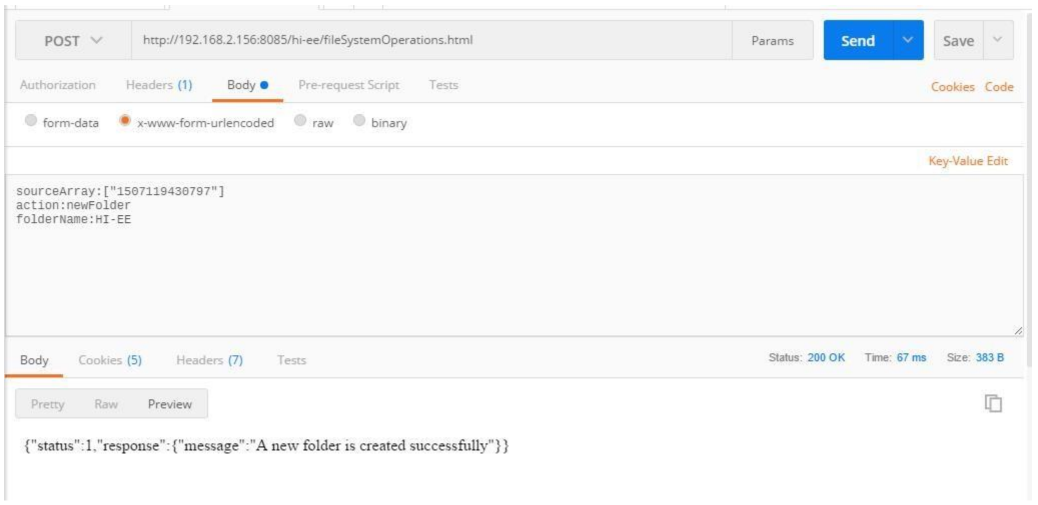
Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html x-www-form-urlencoded `sourceArray:[["1507119430797/d305f1f5-5160-4a6f-9c76-78b489f007b1.report","Chart Report"]]` `action:rename` Status: 200 OK Time: 72 ms Size: 367 B `{"status":1,"response":{"message":"Rename is successful"}}`

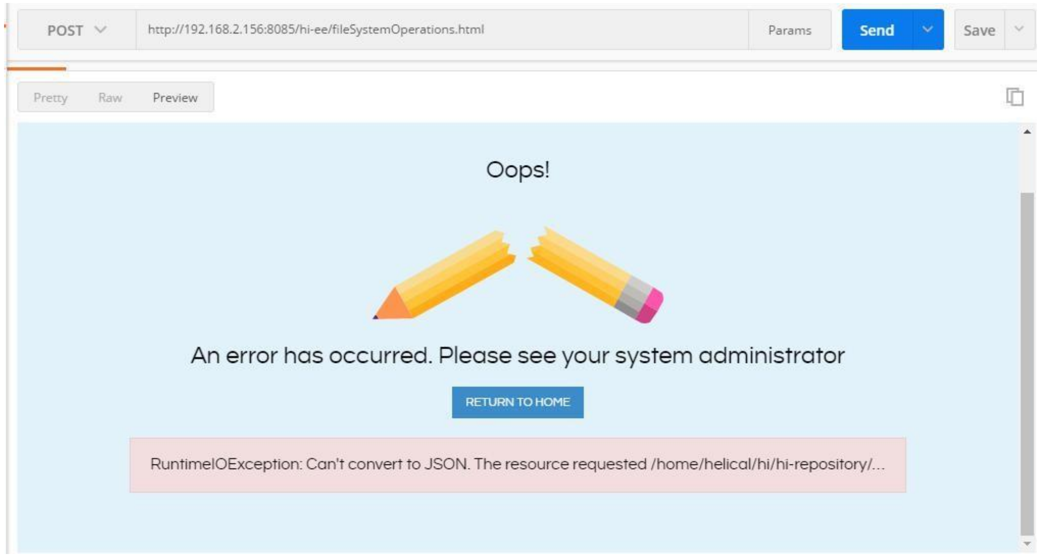
3.3.3 Create Folder

URL	fileSystemOperations.html	
Description	It allows user to create a new folder as per the requirement.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=rename&sourceArray=[['1507119430797','HI-EE']]" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	["1507119430797"]	The directory/file where new folder action is performed
action:	newFolder	File operation action type is newFolder, ie a new folder is created
folderName:	HI-EE	The folder name

Response Output(JSON Format)	{
	"status": 1,
	"Response": {
	"message": "A new folder is created successfully"
	}
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. The new folder is created. In the backend a file with extension .efwfolder is created which has ownership or security information.
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html
	x-www-form-urlencoded
	`sourceArray:["1507119430797"]`
	`action:newFolder`
	`folderName:HI-EE`
	Status: 200 OK Time: 67 ms Size: 383 B
	{"status":1,"response":{"message":"A new folder is created successfully"}}

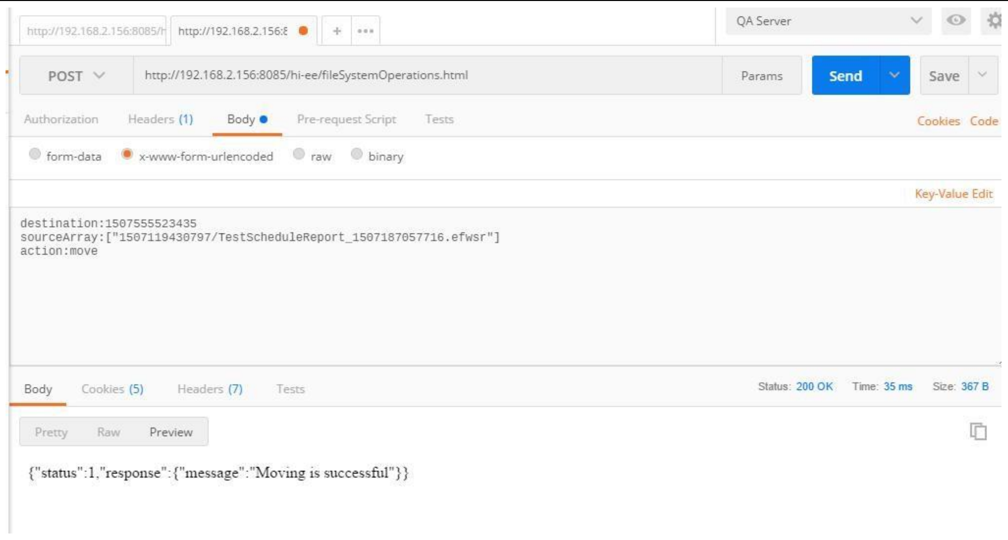
Possible Error	If the file/folder which you want to delete doesnot exists in that case you will get an error.
Screenshot(Error)	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html Oops! An error has occurred. Please see your system administrator RETURN TO HOME RuntimeIOException: Can't convert to JSON. The resource requested /home/helical/hi/hi-repository/...
Post-action	Rename, delete, share, save any report file etc operations can be performed.

3.3.4 Paste operation

Note : The report having extension as .efwsr/saved reports and result files have the cut-paste options to move files.

URL	fileSystemOperations.html	
Description	It allows user to perform moving a file operations. We can cut/paste the file.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=move&destination=1507555523435&sourceArray=['1507119430797/TestScheduleReport_1507187057716.efwsr']" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
HTTP Request Key	**HTTP Request Value**	**Description**
destination:	1507555523435	The directory where paste operation is to be performed
sourceArray:	["1507119430797/TestScheduleReport_1507187057716.efwsr"]	The file where to which cut action is performed
action:	move	File operation action type is move

Response Output(JSON Format)	{
	    "status": 1,
	    "Response": {
	        "message": "Moving is successful"
	    }
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. In the backend the physical file is moved to different location as requested by the user
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html

x-www-form-urlencoded

```
destination:1507555523435
sourceArray:["1507119430797/TestScheduleReport_1507187057716.efwsr"]
action:move
```

Status: 200 OK Time: 35 ms Size: 367 B

```
{"status":1,"response":{"message":"Moving is successful"}}
```

3.3.5 Import .crt file

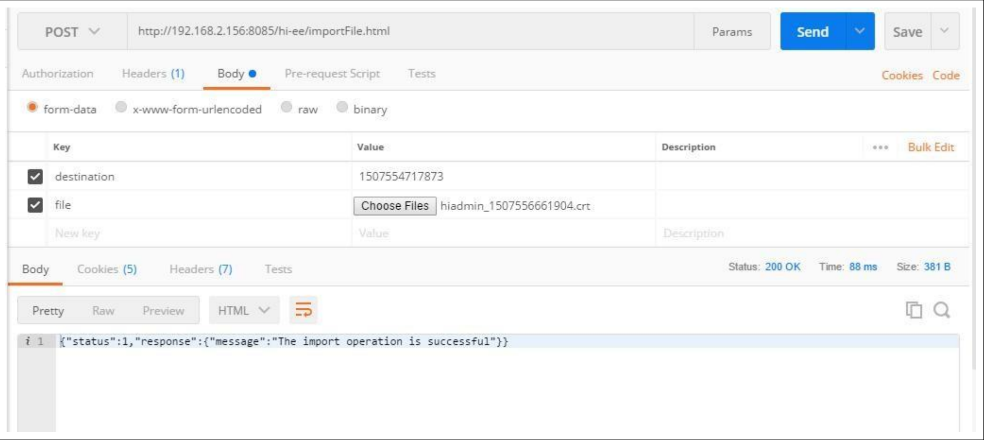
Note : To take export of any report we need to open that report in new window then takes its export.

Need discussion regarding curl command

URL	importFile.html	
Description	It allows user to import .crt file in selected folder. .crt file is the export of report file which gt saved with .crt extension.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST** **Note : For post method we need to select the form-data as body.**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/importFile.html **Access through Curl command :**	
HTTP Request Key	**HTTP Request Value**	**Description**
destination:	1507554717873	Destination to import the file
file:	Select the file to be import from your system.	Select file for import and make sure the the type of file key should be the file only.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"The import operation is successful"}
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. In the backend the physical file is moved to different location as requested by the user	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/importFile.html

Body: form-data [✓ form-data] x-www-form-urlencoded raw binary

Key	Value	Description
destination	1507554717873	
file	hiadmin_1507556661904.crt	

Status: 200 OK Time: 88 ms Size: 381 B

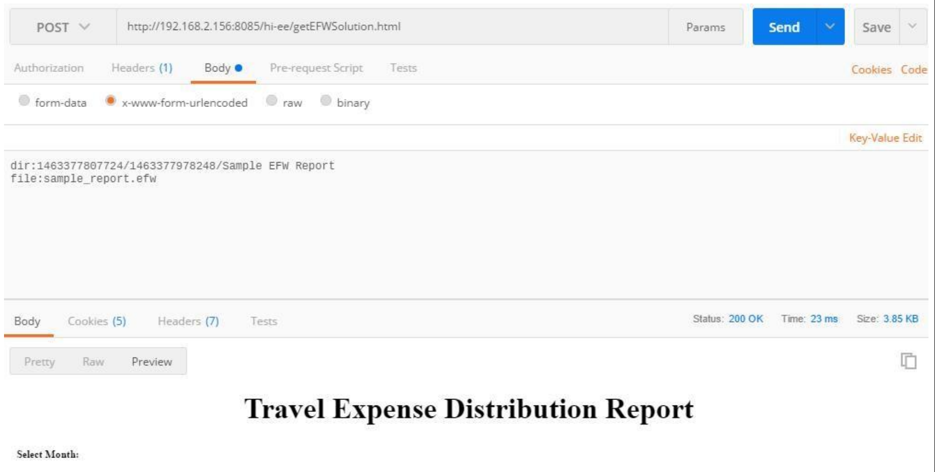
```
{"status":1,"response":{"message":"The import operation is successful"}}
```

3.4 EFW Report Operations

3.4.1 Open EFW Report

URL	getEFWSolution.html	
Description	It allows user to open the EFW report. To open the EFW report we required two parameters to be provided that is directory and file(which exists).If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST,GET**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getEFWSolution.html?dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw" http://192.168.2.156:8085/hi-ee/getEFWSolution.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377978248/Sample EFW Report	The directory where the .efw file is present.
file:	sample_report.efw	The efw file which we want to open.

Response Output(html contents)	Here response is nothing but the report html contents. The EFW report will get opened.

Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/getEFWSolution.html

form-data [✓ x-www-form-urlencoded] raw binary

```
dir:1463377807724/1463377978248/Sample EFW Report
file:sample_report.efw
```

Status: 200 OK Time: 23 ms Size: 3.85 KB

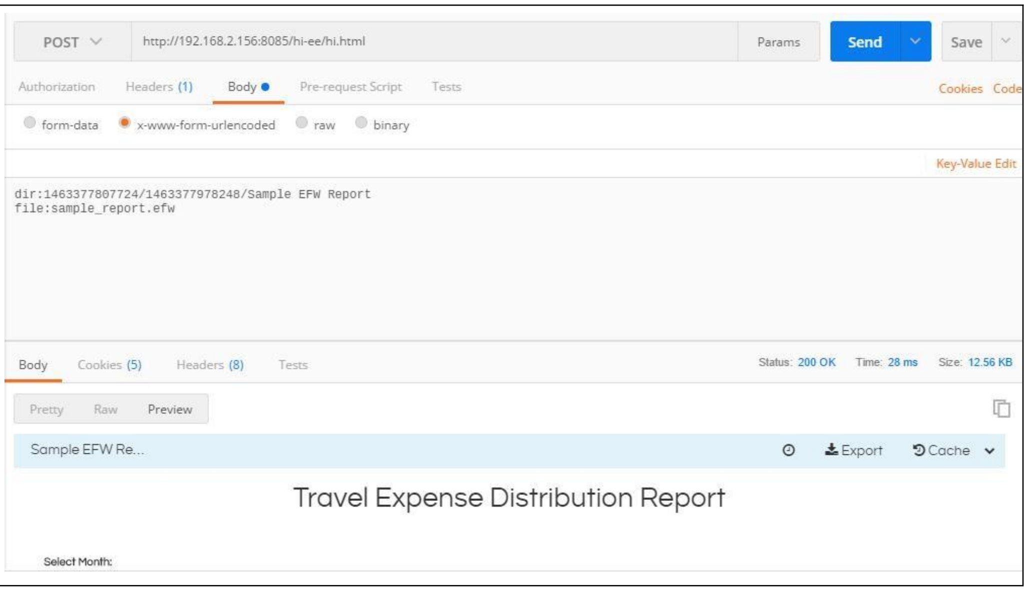
Travel Expense Distribution Report

Select Month:

3.4.2 Open EFW Report in new window

URL	hi.html?dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw&mode=open	
Description	It allows user to open the EFW report in new window , it requires directory and filename(which exists).If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw&mode=open **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw&mode=open" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377978248/Sample EFW Report	The directory where the .efw file is present.
file:	sample_report.efw	The efw file which we want to open in new window.
mode:	open	Mode to open report in new window.
Response Output(JSON Format)	Here response is nothing but the report html contents.	

Service Status	200 OK

Screenshot	

POST http://192.168.2.156:8085/hi-ee/hi.html

Body: x-www-form-urlencoded

```
dir:1463377807724/1463377978248/Sample EFW Report
file:sample_report.efw
```

Status: 200 OK Time: 28 ms Size: 12.56 KB

Sample EFW Re...

Travel Expense Distribution Report

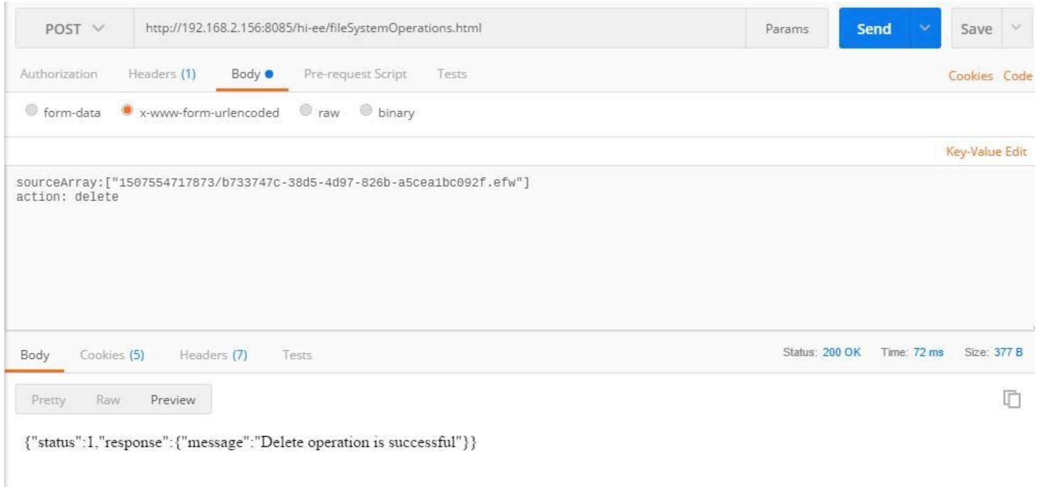
Select Month:

3.4.3 Delete EFW Report

URL	fileSystemOperations.html	
Description	It allows user to delete the EFW report.We need to pass the filename which you want to delete(which exists).If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=['1507554717873/b733747c-38d5-4d97-826b-a5cea1bc092f.efw']&action=delete" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	["1507554717873/b733747c-38d5-4d97-826b-a5cea1bc092f.efw"]	sourceArray having the directory with file name which we want to delete.
action:	delete	Action name to perform the operation.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"Delete operation is successful"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

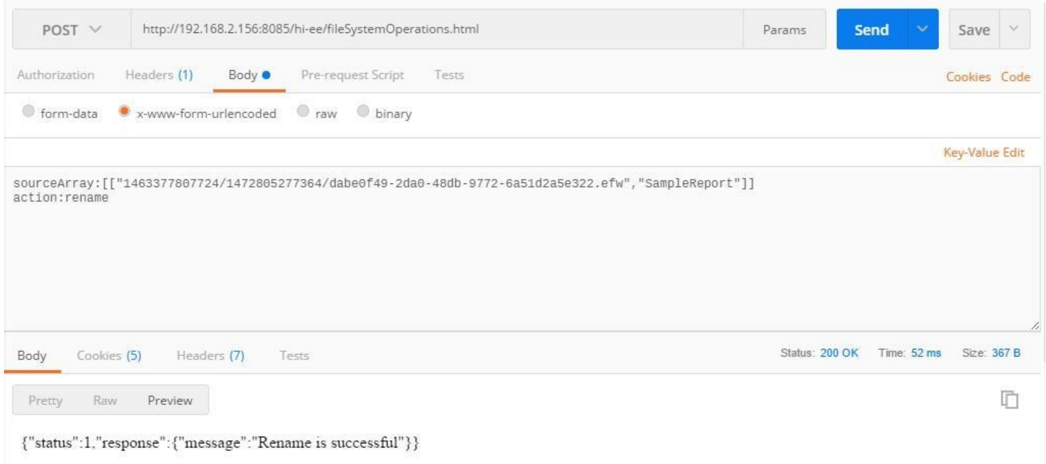
Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html
	Body: x-www-form-urlencoded
	`sourceArray:["1507554717873/b733747c-38d5-4d97-826b-a5cea1bc092f.efw"]`
	`action: delete`
	Status: 200 OK Time: 72 ms Size: 377 B
	`{"status":1,"response":{"message":"Delete operation is successful"}}`

3.4.4 Rename EFW Report

URL	fileSystemOperations.html	
Description	It allows user to rename the .efw file if the user is permitted.To rename .efw file we need to pass the filename with the name by which you want to rename.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1463377807724/1472805277364/dabe0f49-2da0-48db-9772-6a51d2a5e322.efw',',SampleReport']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**

sourceArray:	[["1463377807724/1472805277364/dabe0f49-2da0-48db-9772-6a51d2a5e322.efw","SampleReport"]]	The directory/file where the rename is performed
action:	rename	File operation action type is rename

Response Output(JSON Format)	{"status": 1,
	"Response":
	{ "message": "Rename is successful" }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective efw file.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html

Body: x-www-form-urlencoded

```
sourceArray:[["1463377807724/1472805277364/dabe0f49-2da0-48db-9772-6a51d2a5e322.efw","SampleReport"]]
action:rename
```

Status: 200 OK Time: 52 ms Size: 367 B

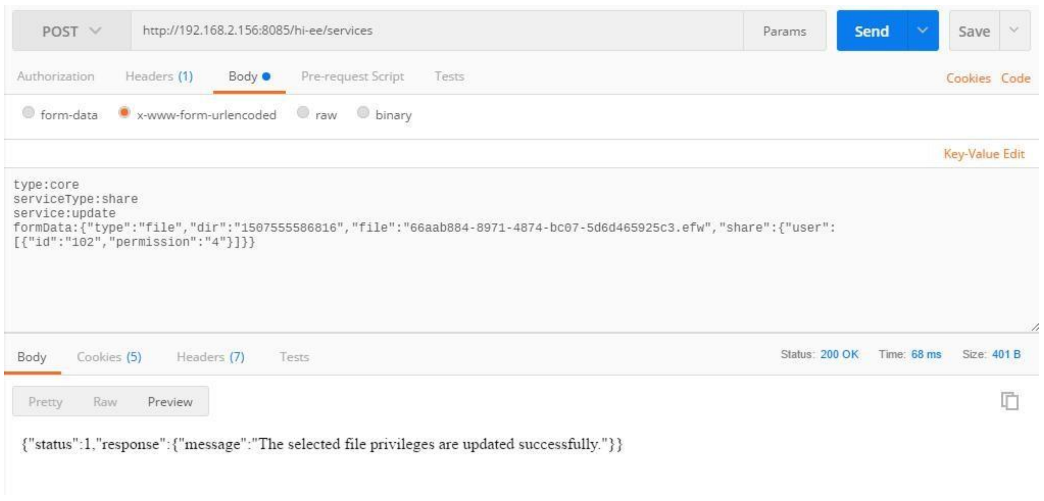
```
{"status":1,"response":{"message":"Rename is successful"}}
```

3.4.5 Share EFW Report

URL	services
Description	It allows user to share the EFW report with any user/organisation/role. The EFW report will get share with provided user with permission.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services

	Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507555586816','file':'66aab884-8971-4874-bc07-5d6d465925c3.efw','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507555586816","file":"66aab884-8971-4874-bc07-5d6d465925c3.efw","share":{"user":[{"id":"109","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The selected file privileges are updated successfully."}
	}	

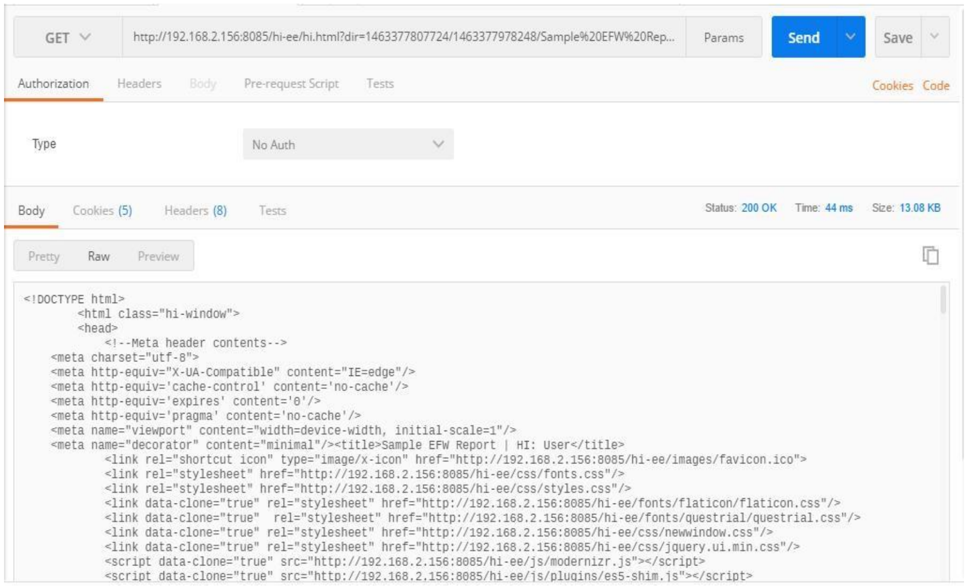
Service Status	200 OK	
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services Body: x-www-form-urlencoded type:core serviceType:share service:update formData:{"type":"file","dir":"1507555586816","file":"66aab884-8971-4874-bc07-5d6d465925c3.efw","share":{"user":[{"id":"102","permission":"4"}]}} Status: 200 OK Time: 68 ms Size: 401 B {"status":1,"response":{"message":"The selected file privileges are updated successfully."}}	

3.4.6 URL printing of EFW Report

Note : Printing of URL can be done in different file formats like pdf, png, jpeg, xls.

Below we are showing example for URL printing in pdf format.

URL	hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&print=pdf	
Description	It allows user to print the URL of EFW report in different printing formats.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&print=pdf **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&print=pdf" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377978248/Sample%20EFW%20Report	The directory where the efw report file is present.
file:	sample_report.efw	The efw report file.
print:	pdf	The print parameter as pdf for printing. Note: Along with pdf printing png, jpeg,xls printing format is possible.
Response Output(JSON Format)	Here response html contents of the requested report export .	

Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Rep...

Type: No Auth

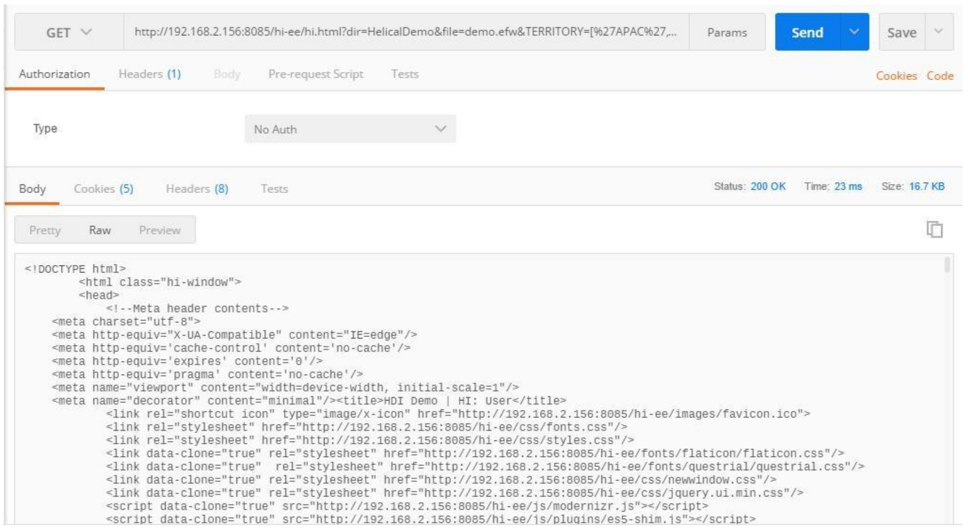
Status: 200 OK Time: 44 ms Size: 13.08 KB

```
<!DOCTYPE html>
        <html class="hi-window">
        <head>
            <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><title>Sample EFW Report | HI: User</title>
            <link rel="shortcut icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico">
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
            <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/modernizr.js"></script>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/plugins/es5-shim.js"></script>
```

3.4.7 Change Report parameters through URL for EFW Report

Note : This API's allows user to change the EFW report parameters through URL.Parameters differs according to report.

URL	hi.html?dir=HelicalDemo&file=demo.efw&TERRITORY=['APAC','Japan','EMEA']	
Description	It allows user to change the report parameters through URL.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=HelicalDemo&file=demo.efw&TERRITORY=['APAC','Japan','EMEA'] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=HelicalDemo&file=demo.efw&TERRITORY=['APAC','Japan','EMEA']" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	HelicalDemo	The directory where the EFW report file is present.
file:	demo.efw	The efw report file.
TERRITORY:	['APAC','Japan','EMEA']	Report parameters which differs from report to report. Here , TERRITORY is the reporr parameter name and ['APAC','Japan','EMEA'] are values of the parameter.
Response		

Output(JSON Format)	Here response is the requested report html contents for requested parameters.
Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/hi.html?dir=HelicalDemo&file=demo.efw&TERRITORY=[%27APAC%27,...

Type: No Auth

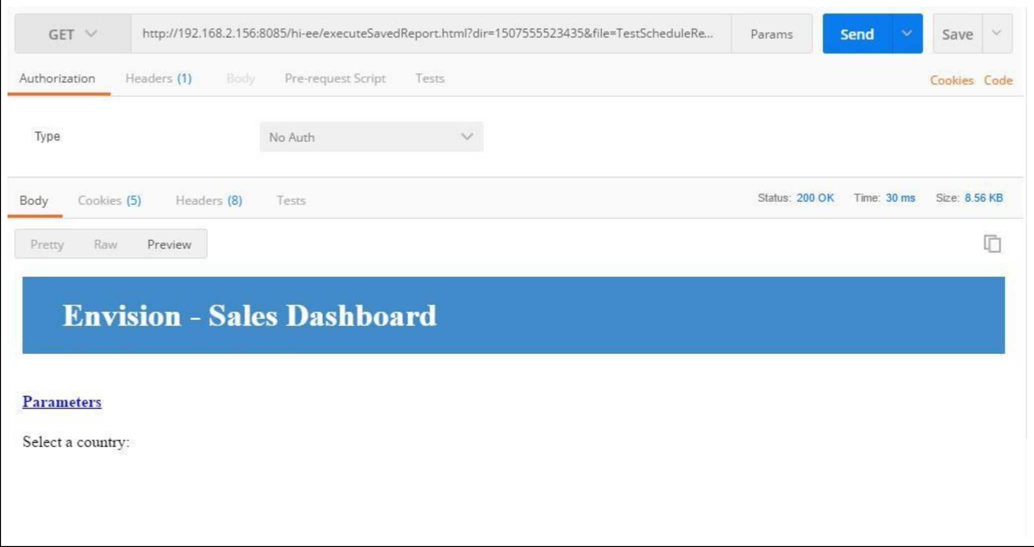
Status: 200 OK Time: 23 ms Size: 16.7 KB

```
<!DOCTYPE html>
        <html class="hi-window">
        <head>
            <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><title>HDI Demo | HI: User</title>
            <link rel="shortcut icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico">
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
            <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/modernizr.js"></script>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/plugins/es5-shim.js"></script>
```

3.5 EFWSR Report Operations

3.5.1 Open EFWSR Report

URL	executeSavedReport.html	
Description	It allows user to open the EFWSR report which requires directory and name of the file(which exists).If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accesible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/executeSavedReport.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1507555523435&file=TestScheduleReport_1507187057716.efwsr" http://192.168.2.156:8085/hi-ee/executeSavedReport.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1507555523435	The directory where the .efwsr file is present.
file:	TestScheduleReport_1507187057716.efwsr	The efwsr file which we want to open.
Response Output(html)	Here response is nothing but the report html contents.Report get opened in new window.	
Service Status	200 OK	

Screenshot	

GET http://192.168.2.156:8085/hi-ee/executeSavedReport.html?dir=1507555234358&file=TestScheduleRe...

Authorization Headers (1) Body Pre-request Script Tests

Type: No Auth

Body Cookies (5) Headers (8) Tests

Status: 200 OK Time: 30 ms Size: 8.56 KB

Pretty Raw Preview

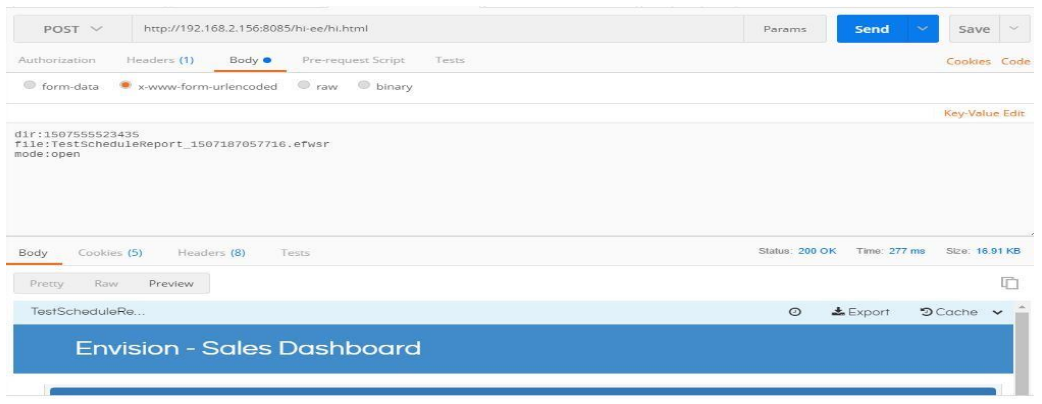
Envision - Sales Dashboard

Parameters

Select a country:

3.5.2 Open EFWSR Report in new window

URL	hi.html?dir=1507555523435&file=TestScheduleReport_1507187057716.efwsr &mode=open	
Description	It allows user to open the EFWSR report in new window. The EFWSR report will get opened in new window.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1507555523435&file=TestScheduleReport_1507187057716.efwsr&mode=open **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1507555523435&file=TestScheduleReport_1507187057716.efwsr&mode=open" http://192.168.2.156:8085/hi-ee/executeSavedReport.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1507555523435	The directory where the .efwsr file is present.
file:	TestScheduleReport_1507187057716.efwsr	The efwsr file which we want to open in new window.
mode:	open	Set mode as open to open the report in new window.
Response Output(HTML)	Here response is nothing but the report html contents.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/hi.html
	x-www-form-urlencoded
	`dir:1507555523435`
	`file:TestScheduleReport_1507187057716.efwsr`
	`mode:open`
	Status: 200 OK Time: 277 ms Size: 16.91 KB
	TestScheduleRe...
	Envision - Sales Dashboard

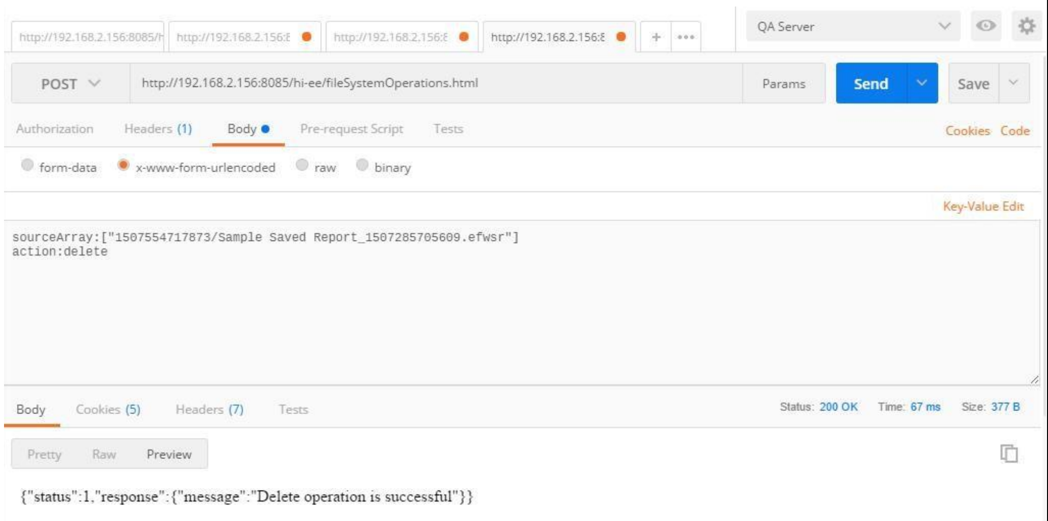
3.5.3 Delete EFWSR Report

URL	fileSystemOperations.html	
Description	It allows user to delete the EFWSR report.If the file/directory doesnot exists you will get an error	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=['1507554717873/Sample Saved Report_1507285705609.efwsr']&action=delete" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	["1507554717873/Sample Saved Report_1507285705609.efwsr"]	sourceArray having the directory with file name which we want to delete.
action:	delete	Action name to perform the operation.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"Delete operation is successful"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	

Service Status	200 OK

Screenshot	

POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html

```
sourceArray:["1507554717873/Sample Saved Report_1507285705609.efwsr"]
action:delete
```

Status: 200 OK Time: 67 ms Size: 377 B

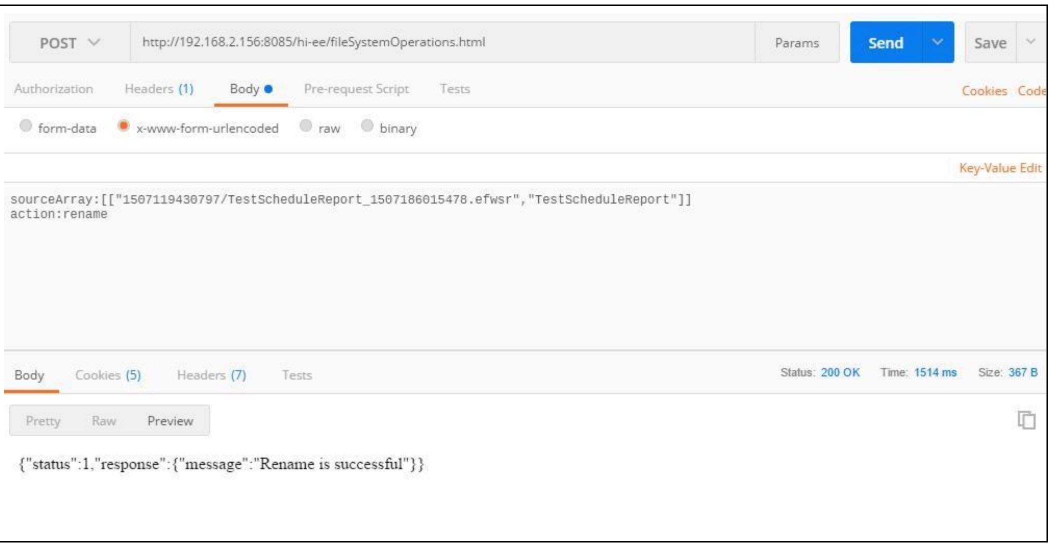
```
{"status":1,"response":{"message":"Delete operation is successful"}}
```

3.5.4 Rename EFWSR Report

URL	fileSystemOperations.html	
Description	It allows user to rename the .efwsr file if the user is permitted.If the file/directory doesnot exists you will get an error	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1507119430797/TestScheduleReport_1507186015478.efwsr',',TestScheduleReport']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507119430797/TestScheduleReport_1507186015478.efwsr","TestScheduleReport1"]]	The directory/file where the rename is performed
action:	rename	File operation action type is rename

Response Output(JSON Format)	{"status": 1,
	"Response":
	{ "message": "Rename is successful" }
	}	

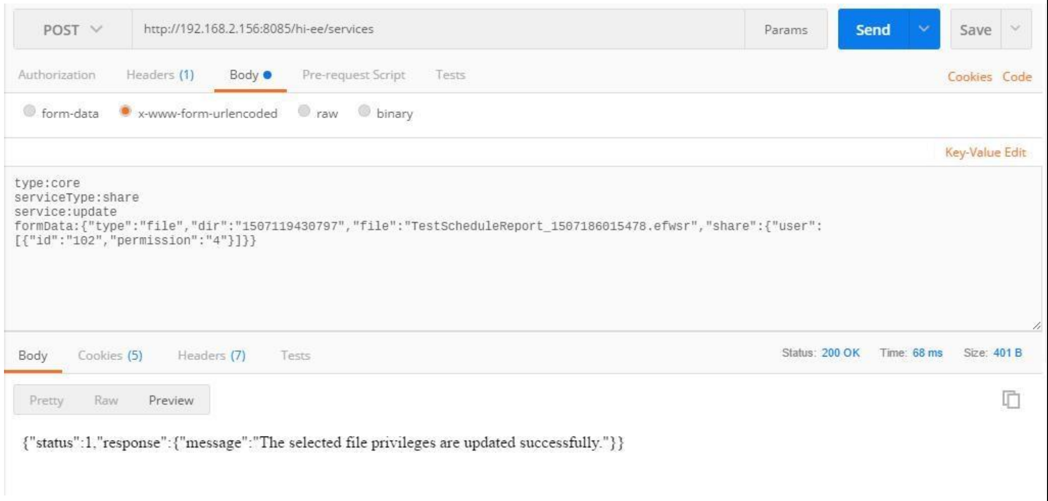
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective file.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html
	Body: x-www-form-urlencoded
	`sourceArray:[["1507119430797/TestScheduleReport_1507186015478.efwsr","TestScheduleReport"]]`
	`action:rename`
	Status: 200 OK Time: 1514 ms Size: 367 B
	`{"status":1,"response":{"message":"Rename is successful"}}`

3.5.5 Share EFWSR Report

URL	services	
Description	It allows user to share the EFWSR report with any user/organisation/role. The EFWSR report will get share with provided user with permission.If the file/directory doesnot exists you will get an error	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507119430797','file':'TestScheduleReport_1507186015478.efwsr','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507119430797","file":"TestScheduleReport_1507186015478.efwsr","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The selected file privileges are updated successfully."}
	}

Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/services

```
type:core
serviceType:share
service:update
formData:{"type":"file","dir":"1507119430797","file":"TestScheduleReport_1507186015478.efwsr","share":{"user":
[{"id":"102","permission":"4"}]}}
```

Status: 200 OK Time: 68 ms Size: 401 B

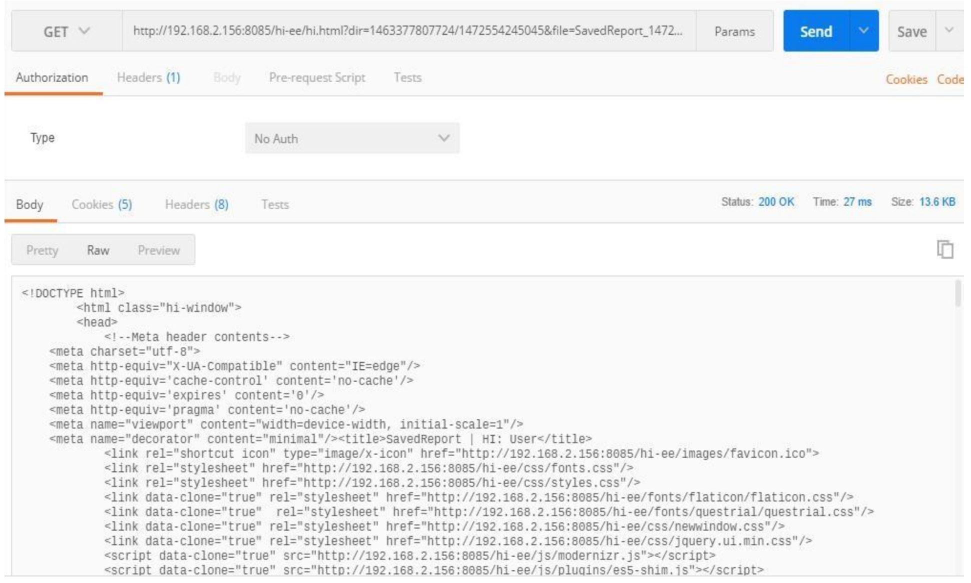
```
{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}
```

3.5.6 URL printing of EFWSR Report

Note : Printing of URL can be done in different file formats like pdf, png, jpeg, xls.

Below we are showing example for URL printing in pdf format.

URL	hi.html?dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&mode=open&print=pdf	
Description	It allows user to print the URL of EFWSR report in different printing formats.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&mode=open&print=pdf **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&mode=open&print=pdf" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1472554245045	The directory where the efwsr report file is present.
file:	SavedReport_1472554274862.efwsr	The efwsr report file.
print:	pdf	The print parameter as pdf for printing. Note: Along with pdf printing png, jpeg,xls printing format is possible.
Response		

Output(JSON Format)	Here response html contents of the requested report export .
Service Status	200 OK
Screenshot	

GET http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1472554245045&file=SavedReport_1472...

Authorization | Headers (1) | Body | Pre-request Script | Tests

Type: No Auth

Body | Cookies (5) | Headers (8) | Tests

Status: 200 OK Time: 27 ms Size: 13.6 KB

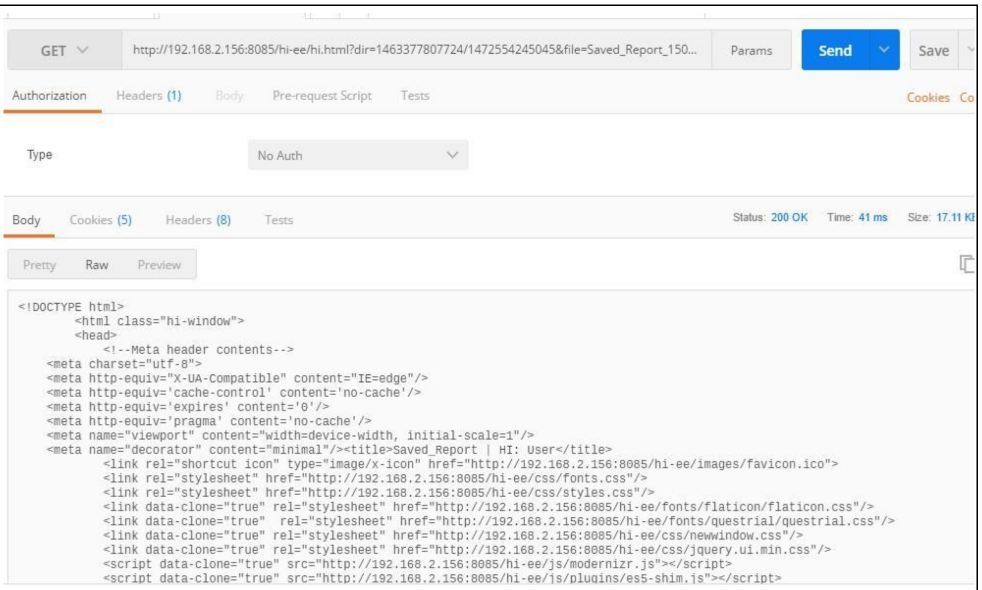
```
<!DOCTYPE html>
        <html class="hi-window">
        <head>
            <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><title>SavedReport | HI: User</title>
            <link rel="shortcut icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico">
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
            <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/modernizr.js"></script>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/plugins/es5-shim.js"></script>
```

3.5.7 Change Report parameters through URL for EFWSR Report

Note : This API's allows user to change the EFWSR report parameters through URL.Parameters differs according to report.

URL	hi.html?dir=1463377807724/1472554245045&file=Saved_Report_1509531808014.efwsr&mode=open&TERRITORY=['APAC','Japan','EMEA']	
Description	It allows user to change the report parameters through URL.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1472554245045&file=Saved_Report_1509531808014.efwsr&mode=open&TERRITORY=['APAC','Japan','EMEA'] **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1472554245045&file=Saved_Report_1509531808014.efwsr&mode=open&TERRITORY=['APAC','Japan','EMEA']" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1472554245045	The directory where the EFWSR report file is present.
file:	Saved_Report_1509531808014.efwsr	The efwsr report file.
TERRITORY:	['APAC','Japan','EMEA']	Report parameters which differs from report to report. Here , TERRITORY is the report parameter name and ['APAC','Japan','EMEA'] are values of the parameter.
Response Output(JSON Format)	Here response is the requested report html contents for requested parameters.	

Service Status	200 OK

Screenshot	

http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1472554245045&file=Saved_Report_150...

Type: No Auth

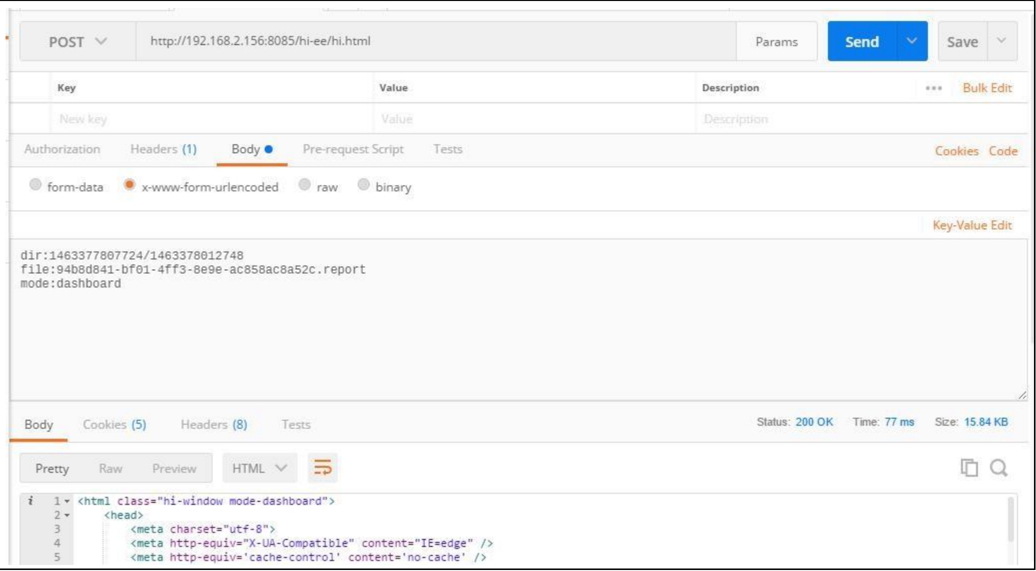
Status: 200 OK Time: 41 ms Size: 17.11 KI

```
<!DOCTYPE html>
        <html class="hi-window">
        <head>
            <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><title>Saved_Report | HI: User</title>
            <link rel="shortcut icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico">
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
            <link rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
            <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
            <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/modernizr.js"></script>
            <script data-clone="true" src="http://192.168.2.156:8085/hi-ee/js/plugins/es5-shim.js"></script>
```

3.6 Adhoc Report Operations

3.6.1 Open Adhoc Report

URL	hi.html	
Description	It allows user to open the adhoc report which requires directory and name of the file(which exists) with mode of file.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accesible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST,GET**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory where the adhoc report file is present.
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	The adhoc report file which we want to open.
mode:	dashboard	Mode of the file to open adhoc report .
Response Output(html)	Here response is nothing but the report html contents.Report get opened in new window.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/hi.html

Body: x-www-form-urlencoded

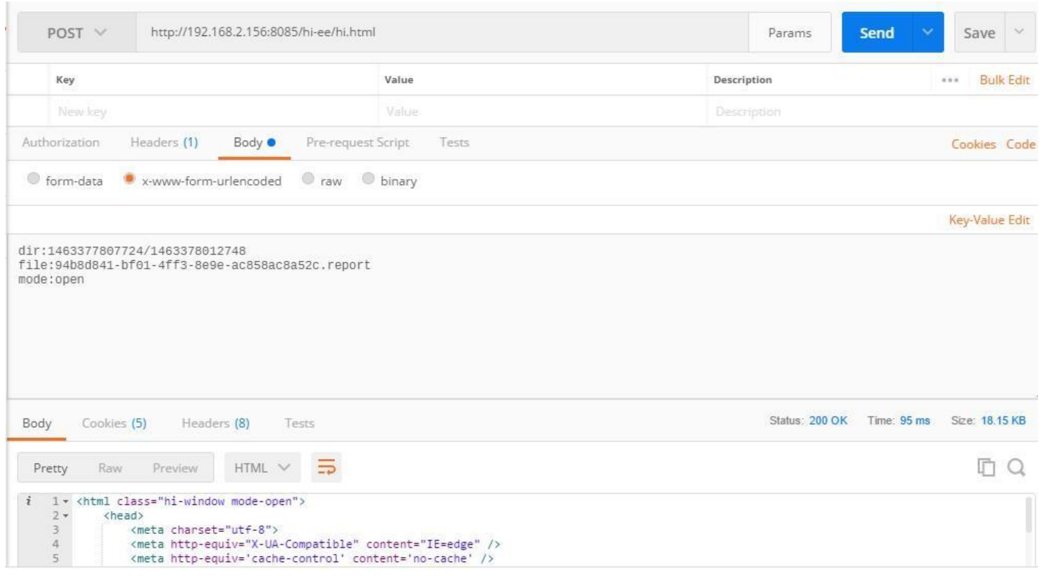
```
dir:1463377807724/1463378012748
file:94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report
mode:dashboard
```

Status: 200 OK Time: 77 ms Size: 15.84 KB

```
<html class="hi-window mode-dashboard">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv='cache-control' content='no-cache' />
```

3.6.2 Open Adhoc Report in new window

URL	/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open	
Description	It allows user to open the adhoc report in new window which requires directory , file name and mode of file as parameters.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory where the .efwsr file is present.
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	The efwsr file which we want to open in new window.
mode:	open	Set mode as open to open the report in new window.
Response Output(HTML)	Here response is nothing but the report html contents.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/hi.html

Body: x-www-form-urlencoded

```
dir:1463377807724/1463378012748
file:94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report
mode:open
```

Status: 200 OK Time: 95 ms Size: 18.15 KB

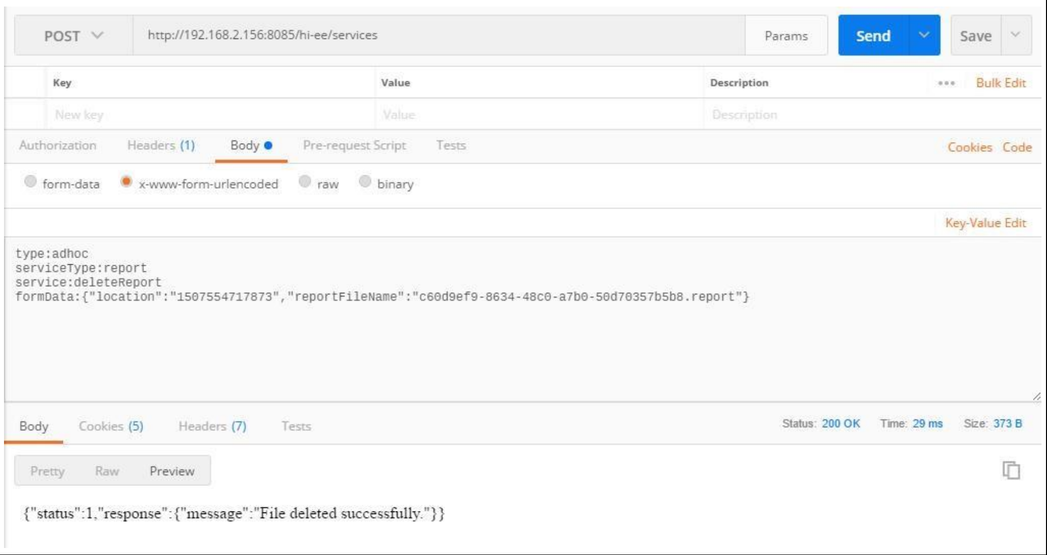
```
<html class="hi-window mode-open">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv='cache-control' content='no-cache' />
```

3.6.3 Delete Adhoc Report

URL	services	
Description	It allows user to delete the adhoc report, we need to pass file name which you want to delete.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=deleteReport&formData={'location':'1507554717873','reportFileName':'c60d9ef9-8634-48c0-a7b0-50d70357b5b8.report'}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc report type.
serviceType:	report	Servicetype as report
service:	deleteReport	Service to delete report.
formData:	{"location":"1507554717873","reportFileName":"c60d9ef9-8634-48c0-a7b0-50d70357b5b8.report"}	Formdata having location of file , file to delete.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"File deleted successfully."}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services

Body: x-www-form-urlencoded

```
type:adhoc
serviceType:report
service:deleteReport
formData:{"location":"1507554717873","reportFileName":"c60d9ef9-8634-48c0-a7b0-50d70357b5b8.report"}
```

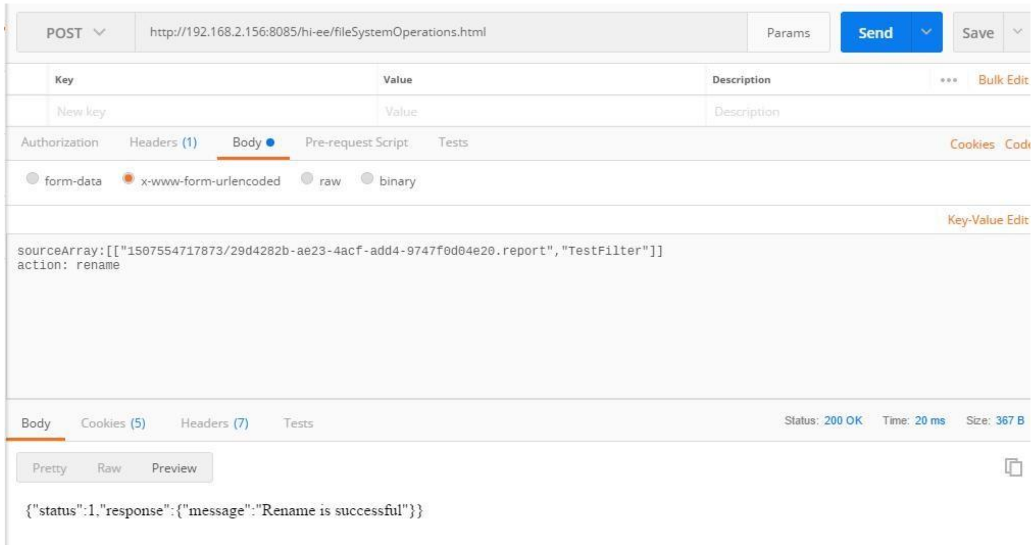
Status: 200 OK Time: 29 ms Size: 373 B

```
{"status":1,"response":{"message":"File deleted successfully."}}
```

3.6.4 Rename adhoc Report

URL	fileSystemOperations.html
Description	It allows user to rename the adhoc report file if the user is permitted.We need to pass the filename with the name by which you wan to rename.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1507554717873/29d4282b-ae23-4acf-add4-9747f0d04e20.report',',TestFilter']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507554717873/29d4282b-ae23-	The directory/file where the rename is

	4acf-add4-9747f0d04e20.report","TestFilter"]]	performed
action:	rename	File operation action type is rename
Response Output(JSON Format)	{"status": 1, "Response": { "message": "Rename is successful" } }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective file.	
Service Status	200 OK	
Screenshot		

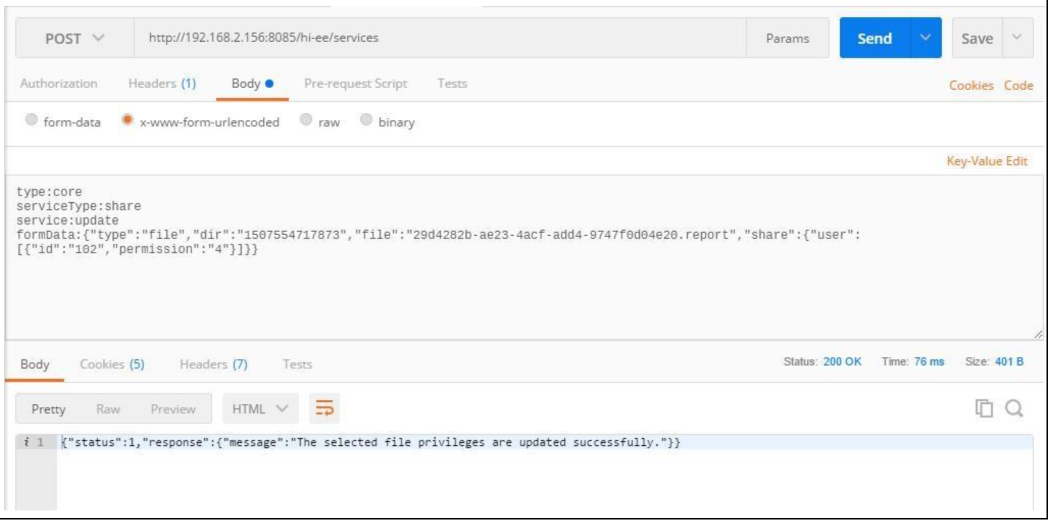
```
sourceArray:[["1507554717873/29d4282b-ae23-4acf-add4-9747f0d04e20.report","TestFilter"]]
action: rename
```

```
{"status":1,"response":{"message":"Rename is successful"}}
```

3.6.5 Share adhoc Report

URL	services	
Description	It allows user to share the adhoc report with any user/organisation/role. The adhoc report will get share with provided user with permission.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507554717873','file':'29d4282b-ae23-4acf-add4-9747f0d04e20.report','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON	{ "status":1,

Format)	"response":{"message":"The selected file privileges are updated successfully."} }
Service Status	200 OK
Screenshot	

```
type:core
serviceType:share
service:update
formData:{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","share":{"user":
[{"id":"102","permission":"4"}]}}
```

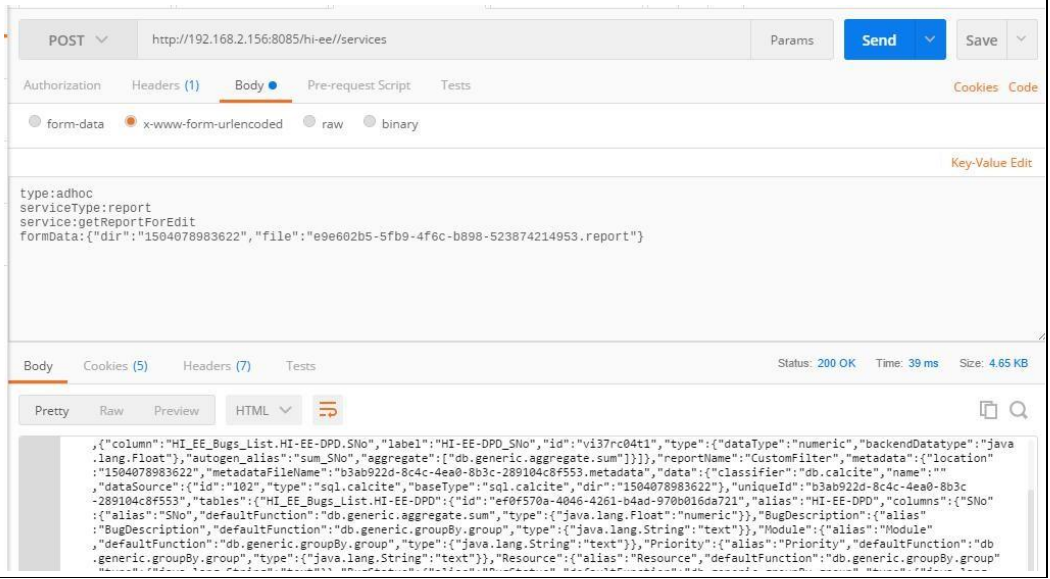
```
{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}
```

URL	/services
Description	It allows user to edit the adhoc report where we can do the changes to existing report .If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data

3.6.6 Edit Adhoc Report

	"j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=getReportForEdit&formData={'dir':'1504078983622','file':'e9e602b5-5fb9-4f6c-b898-523874214953.report'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc report type.
serviceType:	report	Servicetype as report
service:	getReportForEdit	Service to edit report.
formData:	{"dir":"1504078983622","file":"e9e602b5-5fb9-4f6c-b898-523874214953.report"}	Formdata having location of file and filename.

Response Output(JSON Format)	{"status":1,"response":
	{
	Response data with requested report details like columns , filters,metadata used etc.
	}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the requested report details.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

```
type:adhoc
serviceType:report
service:getReportForEdit
formData:{"dir":"1504078983622","file":"e9e602b5-5fb9-4f6c-b898-523874214953.report"}
```

Status: 200 OK Time: 39 ms Size: 4.65 KB

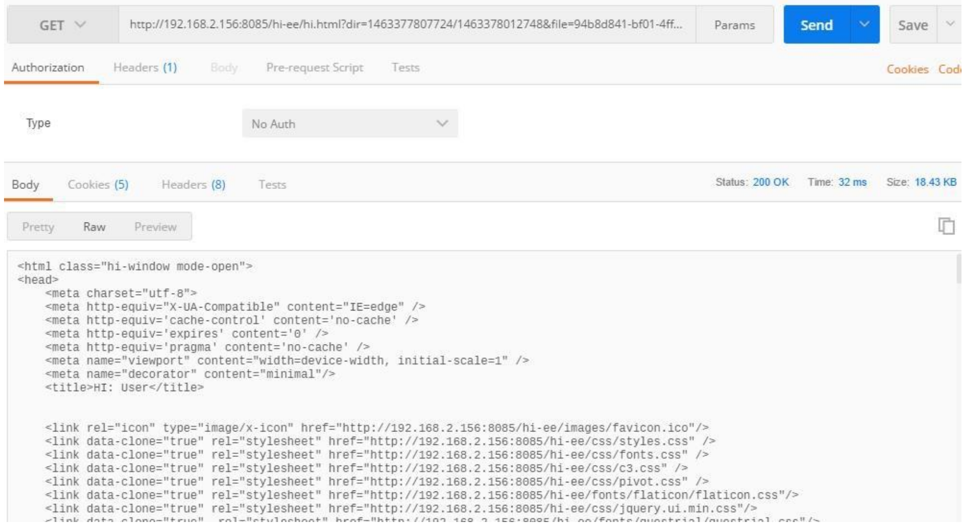
```
,{"column":"HI_EE_Bugs_List.HI-EE-DPD.SNo","label":"HI-EE-DPD_SNo","id":"vi37rc04t1","type":{"dataType":"numeric","backendDatatype":"java
.lang.Float"},"autogen_alias":"sum_SNo","aggregate":["db.generic.aggregate.sum"]}]},"reportName":"CustomFilter","metadata":{"location"
:"1504078983622","metadataFileName":"b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata","data":{"classifier":"db.calcite","name":""
,"dataSource":{"id":"102","type":"sql.calcite","baseType":"sql.calcite","dir":"1504078983622"},"uniqueId":"b3ab922d-8c4c-4ea0-8b3c
-289104c8f553","tables":{"HI_EE_Bugs_List.HI-EE-DPD":{"id":"ef0f570a-4046-4261-b4ad-970b016da721","alias":"HI-EE-DPD","columns":{"SNo"
:{"alias":"SNo","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Float":"numeric"}},"BugDescription":{"alias"
:"BugDescription","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Module":{"alias":"Module"
,"defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Priority":{"alias":"Priority","defaultFunction":"db
.generic.groupBy.group","type":{"java.lang.String":"text"}},"Resource":{"alias":"Resource","defaultFunction":"db.generic.groupBy.group"
```

3.6.7 URL printing of Adhoc Report

Note : Printing of URL can be done in different file formats like pdf, png, jpeg, xls.

Below we are showing example for URL printing in pdf format.

URL	hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open&print=pdf	
Description	It allows user to print the URL of adhoc report in different printing formats.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open&print=pdf **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=open&print=pdf" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory where the adhoc report file is present.
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	The adhoc report file.
print:	pdf	The print parameter as pdf for printing. Note: Along with pdf printing png, jpeg,xls printing format is possible.
Response Output(JSON Format)	Here response html contents of the requested report export .	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff...

Type: No Auth

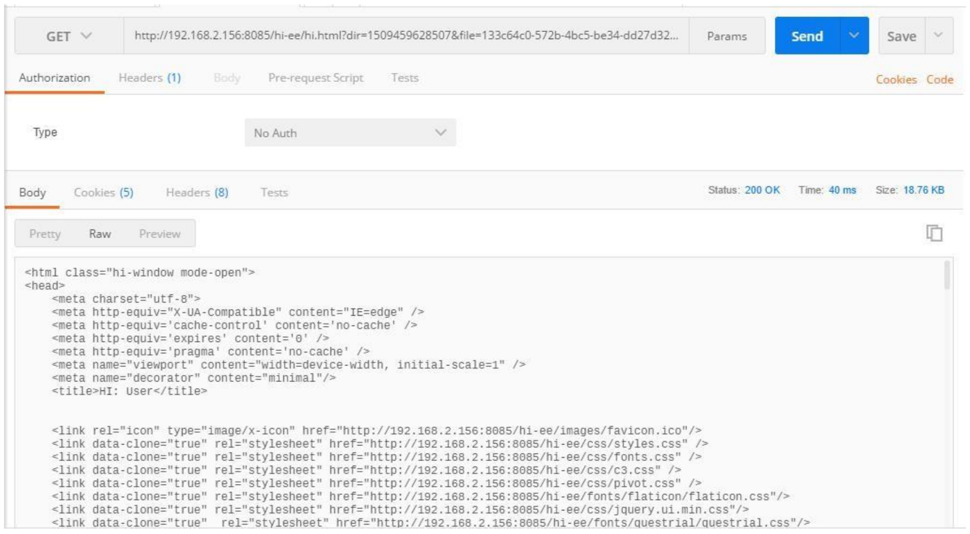
Status: 200 OK Time: 32 ms Size: 18.43 KB

```
<html class="hi-window mode-open">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta http-equiv='cache-control' content='no-cache' />
    <meta http-equiv='expires' content='0' />
    <meta http-equiv='pragma' content='no-cache' />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="decorator" content="minimal"/>
    <title>HI: User</title>


    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/c3.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/pivot.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
```

3.6.8 Change Report parameters through URL for adhoc Report

Note : This API's allows user to change the adhoc report parameters through URL.Parameters differs according to report.

URL	hi.html?dir=1509459628507&file=133c64c0-572b-4bc5-be34-dd27d324cfb2.report&mode=open&EMPLOYEE_DETAILS_ADDRESS=Delhi	
Description	It allows user to change the report parameters through URL.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1509459628507&file=133c64c0-572b-4bc5-be34-dd27d324cfb2.report&mode=open&EMPLOYEE_DETAILS_ADDRESS=Delhi **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1509459628507&file=133c64c0-572b-4bc5-be34-dd27d324cfb2.report&mode=open&EMPLOYEE_DETAILS_ADDRESS=Delhi" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1509459628507	The directory where the adhoc report file is present.
file:	133c64c0-572b-4bc5-be34-dd27d324cfb2.report	The adhoc report file.

EMPLOYEE_DETAILS_ADDRESS:	Delhi	Report parameters which differs from report to report. Here , EMPLOYEE_DETAILS_ADDRESS is the report parameter name and Delhi is the value of the parameter.
Response Output(JSON Format)	Here response is the requested report html contents for requested parameters.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8085/hi-ee/hi.html?dir=1509459628507&file=133c64c0-572b-4bc5-be34-dd27d32...

Type: No Auth

Status: 200 OK Time: 40 ms Size: 18.76 KB

```
<html class="hi-window mode-open">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta http-equiv='cache-control' content='no-cache' />
    <meta http-equiv='expires' content='0' />
    <meta http-equiv='pragma' content='no-cache' />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="decorator" content="minimal"/>
    <title>HI: User</title>


    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/c3.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/pivot.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
```

3.6.9 Apply Filter on Adhoc Report

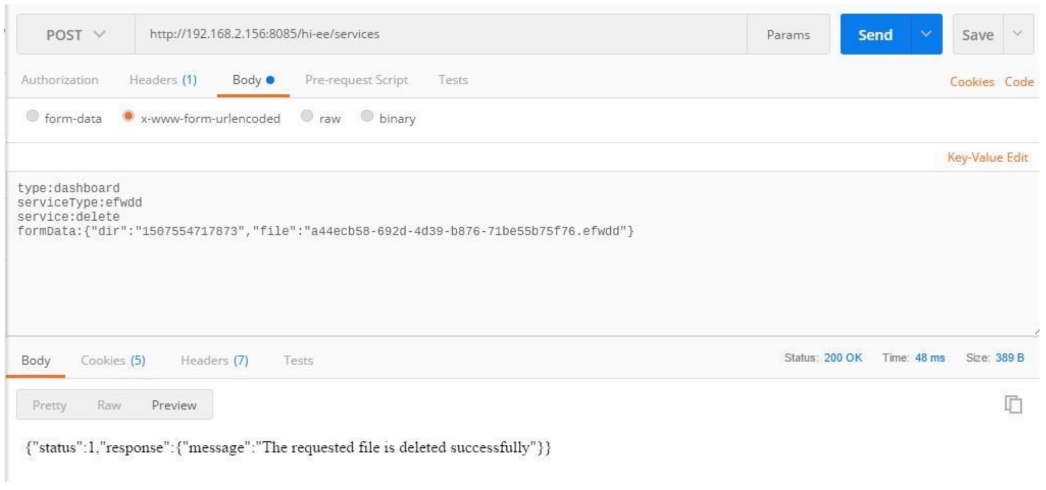
3.7 EFWDD Report Operations

3.7.1 Delete EFWDD Report

URL	services	
Description	It allows user to delete the EFWDD report, we need to pass file name which you want to delete.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwdd&service=delete&formData={'dir':'1507554717873','file':'a44ecb58-692d-4d39-b876-71be55b75f76.efwdd'}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as dashboard type.
serviceType:	efwdd	Servicetype as efwdd
service:	delete	Service to delete report.
formData:	{"dir":"1507554717873","file":"a44ecb58-692d-4d39-b876-71be55b75f76.efwdd"}	Formdata having location of file , file to delete.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"The requested file is deleted successfully."}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services
	Body: x-www-form-urlencoded
	`type:dashboard`
	`serviceType:efwdd`
	`service:delete`
	`formData:{"dir":"1507554717873","file":"a44ecb58-692d-4d39-b876-71be55b75f76.efwdd"}`
	Status: 200 OK Time: 48 ms Size: 389 B
	`{"status":1,"response":{"message":"The requested file is deleted successfully"}}`

3.7.2 Share EFWDD Report

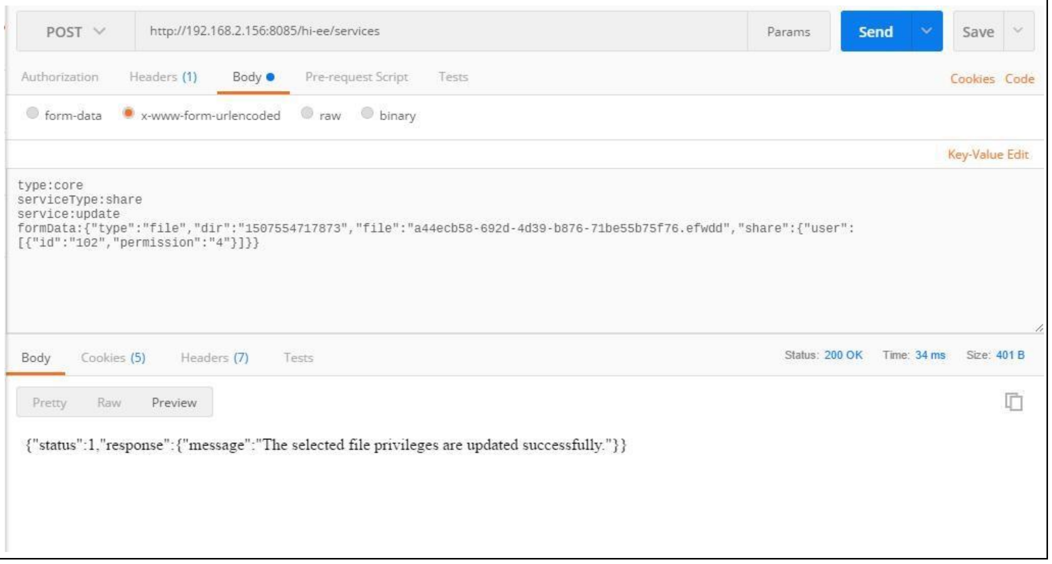
URL	services
Description	It allows user to share the EFWDD report with any user/organisation/role. The EFWDD report will get share with provided user with permission.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)
HTTP Request Method	**POST**

Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507554717873','file':'a44ecb58-692d-4d39-b876-

	71be55b75f76.efwdd','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507554717873","file":"a44ecb58-692d-4d39-b876-71be55b75f76.efwdd","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The selected file privileges are updated successfully."}
	}	

Service Status	200 OK	

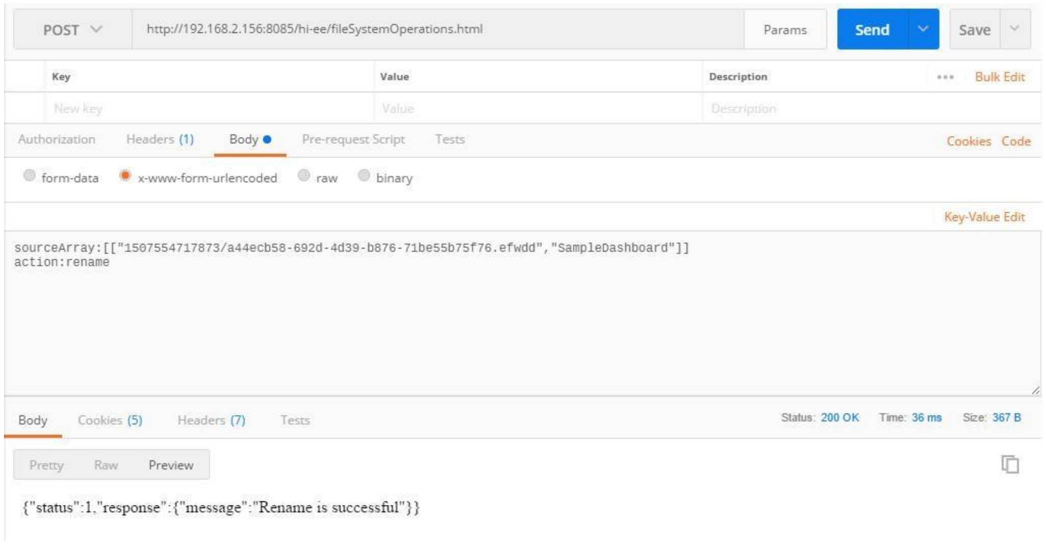
Screenshot	 POST http://192.168.2.156:8085/hi-ee/services
	x-www-form-urlencoded
	`type:core`
	`serviceType:share`
	`service:update`
	`formData:{"type":"file","dir":"1507554717873","file":"a44ecb58-692d-4d39-b876-71be55b75f76.efwdd","share":{"user":[{"id":"102","permission":"4"}]}}`
	Status: 200 OK Time: 34 ms Size: 401 B
	{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}	

3.7.3 Rename EFWDD Report

URL	fileSystemOperations.html	
Description	It allows user to rename the EFWDD report file if the user is permitted.We need to pass the filename with the name by which you wan to rename.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1507554717873/a44ecb58-692d-4d39-b876-71be55b75f76.efwdd',',SampleDashboard']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507554717873/a44ecb58-692d-4d39-b876-71be55b75f76.efwdd","SampleDashboard"]]	The directory/file where the rename is performed
action:	rename	File operation action type is rename

Response Output(JSON Format)	{"status": 1,
	"Response":
	{ "message": "Rename is successful" }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective file.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/fileSystemOperations.html

```
sourceArray:[["1507554717873/a44ecb58-692d-4d39-b876-71be55b75f76.efwdd","SampleDashboard"]]
action:rename
```

Status: 200 OK Time: 36 ms Size: 367 B

```
{"status":1,"response":{"message":"Rename is successful"}}
```

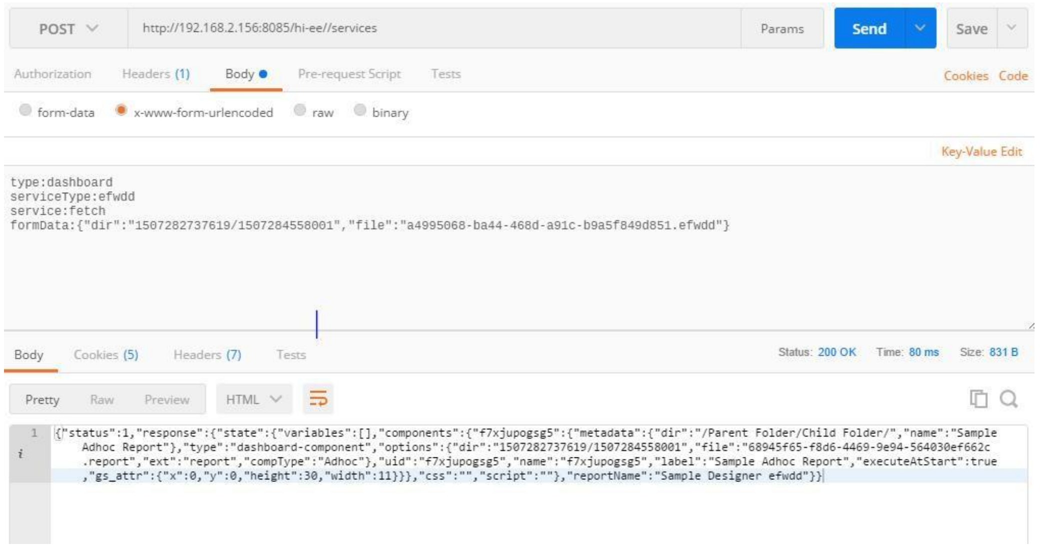
3.7.4 Edit EFWDD Report

URL	/services	
Description	It allows user to edit the dashboard/EFWDD report where we can do the changes to existing dashboard/efwdd report .If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwdd&service=fetch&formData={'dir':'1507282737619/1507284558001','file':'a4995068-ba44-468d-a91c-b9a5f849d851.efwdd'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as dashboard report type.
serviceType:	efwdd	Servicetype as efwdd
service:	fetch	Service to edit report.
formData:	{"dir":"1507282737619/1507284558001","file":"a4995068-ba44-468d-a91c-b9a5f849d851.efwdd"}	Formdata having location of file and filename.

Response Output(JSON Format)	{"status":1,"response":
	{
	Response data with dashboard details like variables, components ,metadata used etc.
	}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the dashboard details.	

Service Status	200 OK

Screenshot	

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

```
type:dashboard
serviceType:efwdd
service:fetch
formData:{"dir":"1507282737619/1507284558001","file":"a4995068-ba44-468d-a91c-b9a5f849d851.efwdd"}
```

Status: 200 OK Time: 80 ms Size: 831 B

```
{"status":1,"response":{"state":{"variables":[],"components":{"f7xjupogsg5":{"metadata":{"dir":"/Parent Folder/Child Folder/","name":"Sample
    Adhoc Report"},"type":"dashboard-component","options":{"dir":"1507282737619/1507284558001","file":"68945f65-f8d6-4469-9e94-564030ef662c
    .report","ext":"report","compType":"Adhoc"},"uid":"f7xjupogsg5","name":"f7xjupogsg5","label":"Sample Adhoc Report","executeAtStart":true
    ,"gs_attr":{"x":0,"y":0,"height":30,"width":11}}},"css":"","script":""},"reportName":"Sample Designer efwdd"}}
```

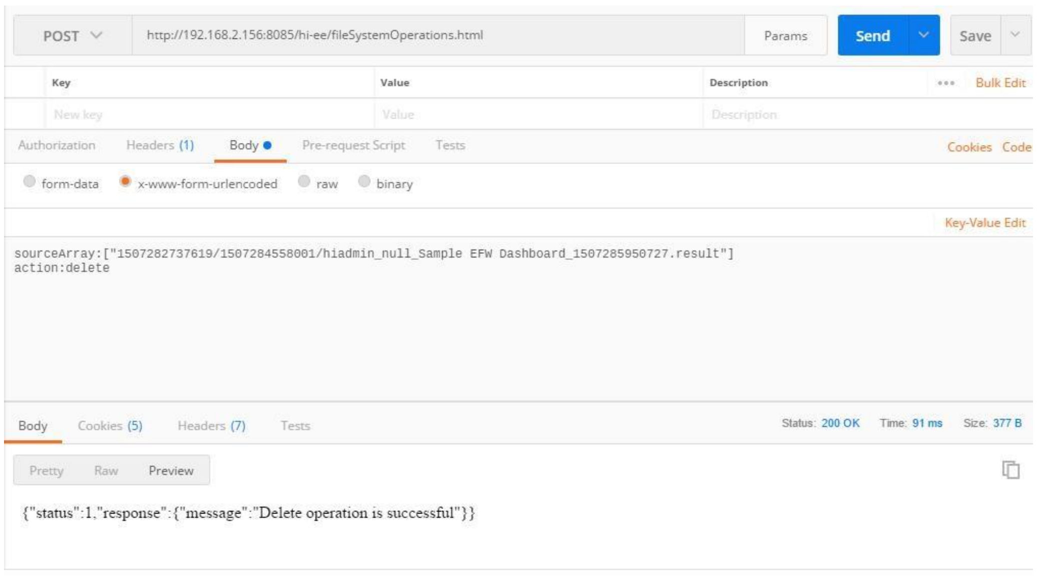
3.8 Result Operations

3.8.1 Delete Result Report

URL	fileSystemOperations.html	
Description	It allows user to delete the result report, we need to pass file name which you want to delete.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&action=delete&sourceArray=['1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result']" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	["1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result"]	sourceArray of result file
action:	delete	Service to delete report.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"Delete operation is successful."}
	}	

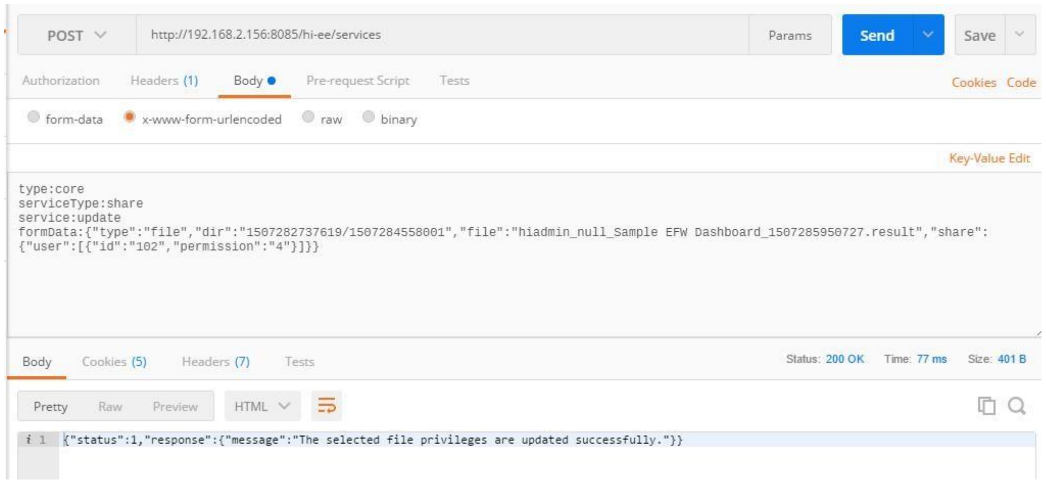
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html x-www-form-urlencoded
	`sourceArray:["1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result"]`
	`action:delete`
	Status: 200 OK Time: 91 ms Size: 377 B
	`{"status":1,"response":{"message":"Delete operation is successful"}}`

3.8.2 Share Result Report

URL	services
Description	It allows user to share the result report with any user/organisation/role. The esult report will get share with provided user with permission.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)
HTTP Request Method	**POST**

Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507282737619/1507284558001','fil

	e':'hiadmin_null_Sample EFW Dashboard_1507285950727.result','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507282737619/1507284558001","file":"hiadmin_null_Sample EFW Dashboard_1507285950727.result","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.
Response Output(JSON Format)	{ "status":1, "response":{"message":"The selected file privileges are updated successfully."} }	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee/services

Body: x-www-form-urlencoded

```
type:core
serviceType:share
service:update
formData:{"type":"file","dir":"1507282737619/1507284558001","file":"hiadmin_null_Sample EFW Dashboard_1507285950727.result","share":
{"user":[{"id":"102","permission":"4"}]}}
```

Status: 200 OK Time: 77 ms Size: 401 B

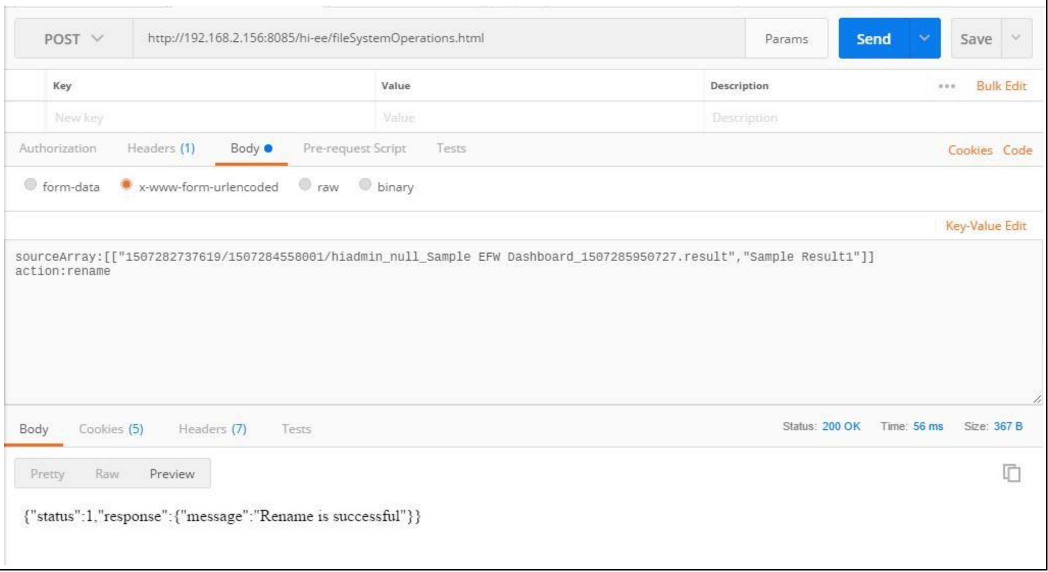
```
{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}
```

3.8.3 Rename Result Report

URL	fileSystemOperations.html	
Description	It allows user to rename the result report file if the user is permitted.We need to pass the filename with the name by which you wan to rename.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result',',Sample Result1']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result","Sample Result1"]]	The directory/file where the rename is performed
action:	rename	File operation action type is rename

Response Output(JSON Format)	{"status": 1,
	"Response":
	{ "message": "Rename is successful" }
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective file.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/fileSystemOperations.html
	Body: x-www-form-urlencoded
	`sourceArray:[["1507282737619/1507284558001/hiadmin_null_Sample EFW Dashboard_1507285950727.result","Sample Result1"]]`
	`action:rename`
	Status: 200 OK Time: 56 ms Size: 367 B
	`{"status":1,"response":{"message":"Rename is successful"}}`

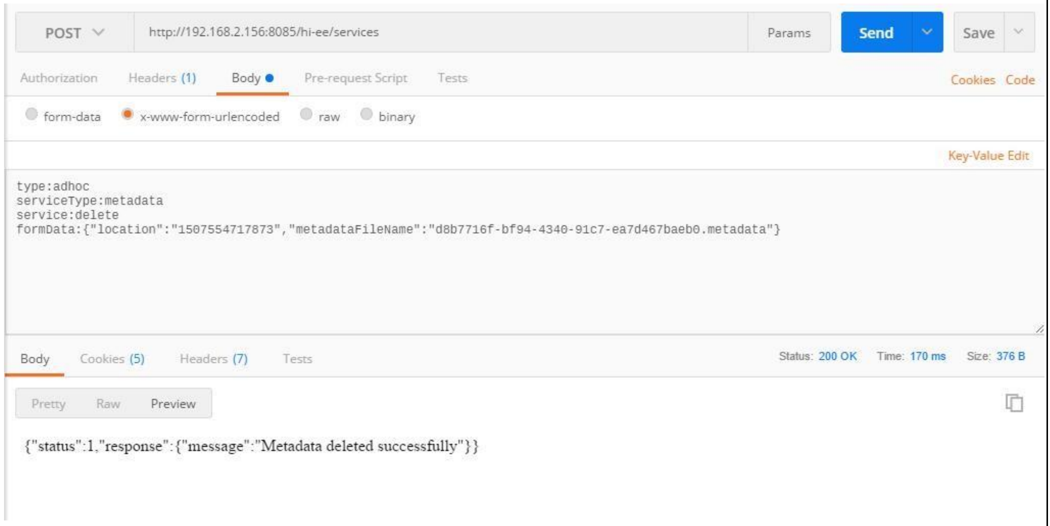
3.9 MetaData Operations

URL	services
Description	It allows user to delete the metdata, we need to pass file name which you want to delete.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=delete&formData={'dir':'1507554717873','file':'d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata'}" http://192.168.2.156:8085/hi-ee/services -

3.9.1 Delete Metadata

	v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc report type.
serviceType:	metadata	Servicetype as metdata
service:	delete	Service to delete report.
formData:	{"location":"1507554717873","metadataFileName":"d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata"}	Formdata having location of file , file to delete.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"Metadata deleted successfully"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8085/hi-ee/services

Body: [✓ x-www-form-urlencoded] form-data raw binary

```
type:adhoc
serviceType:metadata
service:delete
formData:{"location":"1507554717873","metadataFileName":"d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata"}
```

Status: 200 OK Time: 170 ms Size: 376 B

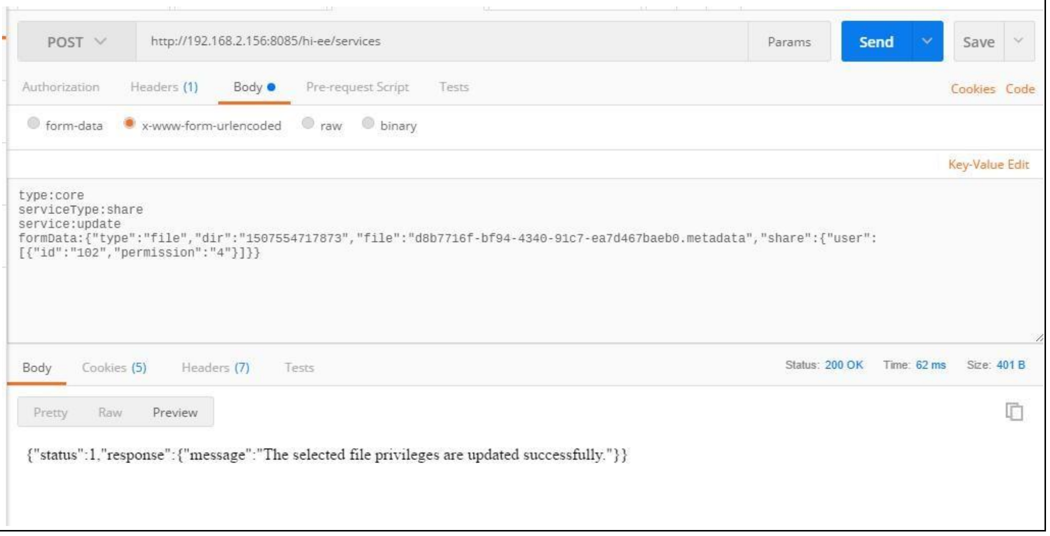
```
{"status":1,"response":{"message":"Metadata deleted successfully"}}
```

3.9.2 Share Metadata

URL	services	
Description	It allows user to share the metadata with any user/organisation/role. The metadata will get share with provided user with permission.If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507554717873','file':'d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"file","dir":"1507554717873","file":"d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

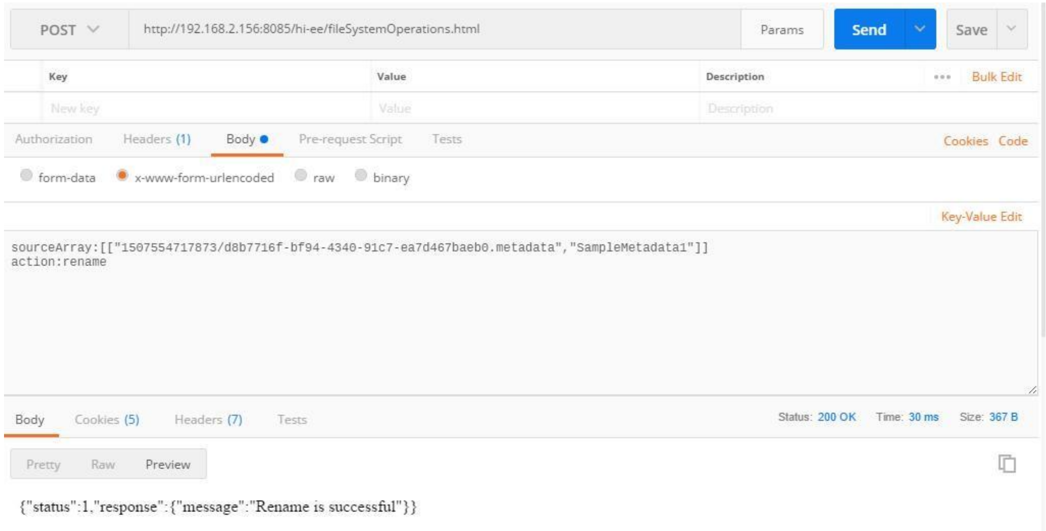
Response	{

Output(JSON Format)	"status":1, "response":{"message":"The selected file privileges are updated successfully."} }
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services
	Body: x-www-form-urlencoded
	`type:core`
	`serviceType:share`
	`service:update`
	`formData:{"type":"file","dir":"1507554717873","file":"d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata","share":{"user":[{"id":"102","permission":"4"}]}}`
	Status: 200 OK Time: 62 ms Size: 401 B
	{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}

3.9.3 Rename Metadata

URL	fileSystemOperations.html
Description	It allows user to rename the metadata file if the user is permitted.We need to pass the filename with the name by which you wan to rename.If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note: User should have write permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/fileSystemOperations.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&sourceArray=[['1507554717873/d8b7716f-bf94-4340-91c7-

	ea7d467baeb0.metadata',',SampleMetadata1']]&action=rename" http://192.168.2.156:8085/hi-ee/fileSystemOperations.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
sourceArray:	[["1507554717873/d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata","SampleMetadata1"]]	The directory/file where the rename is performed
action:	rename	File operation action type is rename
Response Output(JSON Format)	{"status": 1, "Response": { "message": "Rename is successful" } }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.Renaming action takes place for the respective file.	
Service Status	200 OK	
Screenshot		

```
sourceArray:[["1507554717873/d8b7716f-bf94-4340-91c7-ea7d467baeb0.metadata","SampleMetadata1"]]
action:rename
```

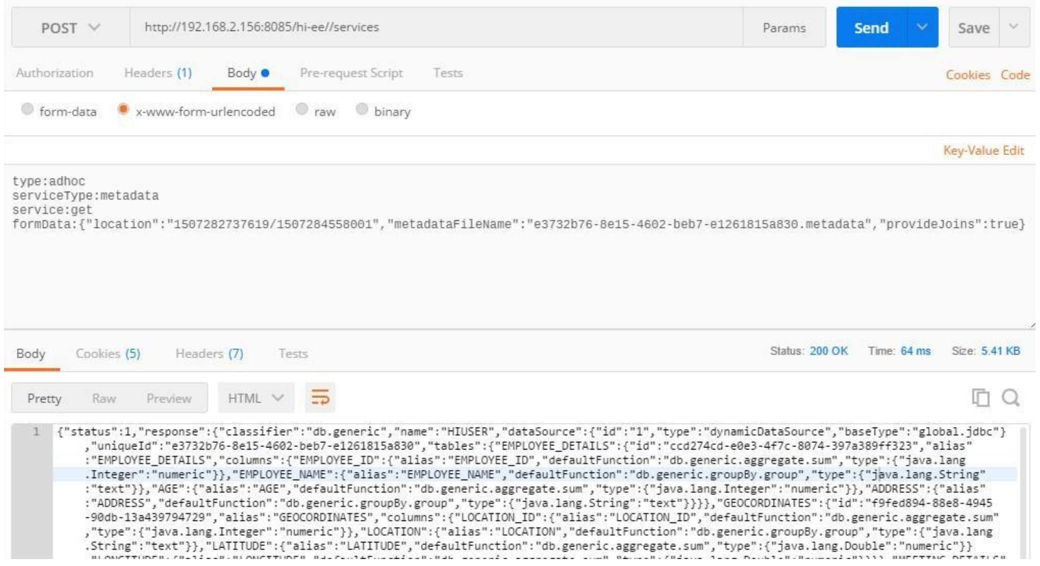
```
{"status":1,"response":{"message":"Rename is successful"}}
```

3.9.4 Edit Metadata

URL	/services
Description	It allows user to edit the metadata where we can do the changes to existing metadata .If the file/directory doesnot exists you will get an error.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have delete permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metad

	ata&service=get&formData={'location':'1507282737619/1507284558001','metadataFileName':'e3732b76-8e15-4602-beb7-e1261815a830.metadata','provideJoins':true}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc report type.
serviceType:	metadata	Servicetype as metadata
service:	get	Service to edit report.
formData:	{"location":"1507282737619/1507284558001","metadataFileName":"e3732b76-8e15-4602-beb7-e1261815a830.metadata","provideJoins":true}	Formdata having location of file and filename.

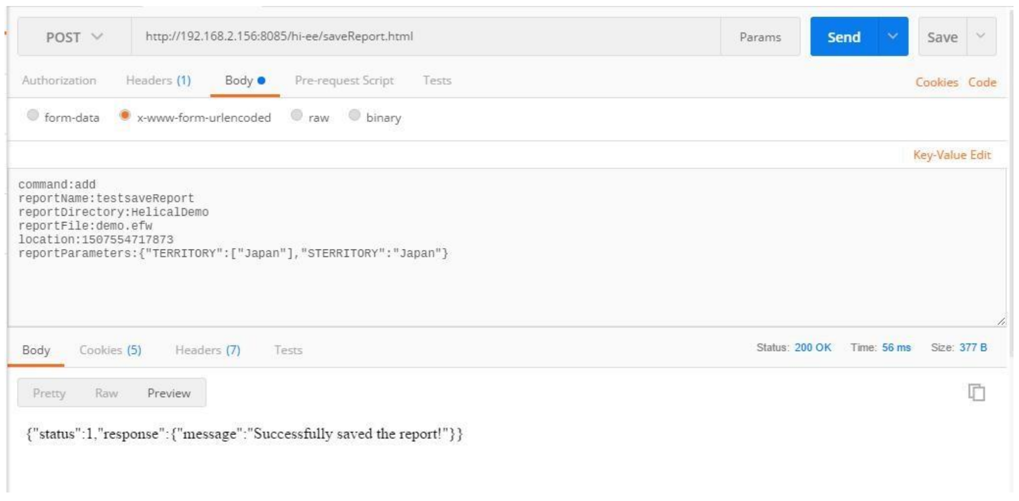
Response Output(JSON Format)	{"status":1,"response":
	{
	Response data with metadata details like name,db details etc.
	}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the metadata details.	
Service Status	200 OK	
Screenshot		

```
POST  http://192.168.2.156:8085/hi-ee//services
Body: x-www-form-urlencoded

type:adhoc
serviceType:metadata
service:get
formData:{"location":"1507282737619/1507284558001","metadataFileName":"e3732b76-8e15-4602-beb7-e1261815a830.metadata","provideJoins":true}

Status: 200 OK  Time: 64 ms  Size: 5.41 KB

{"status":1,"response":{"classifier":"db.generic","name":"HIUSER","dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc"}
    ,"uniqueId":"e3732b76-8e15-4602-beb7-e1261815a830","tables":{"EMPLOYEE_DETAILS":{"id":"ccd274cd-e0e3-4f7c-8074-397a389ff323","alias"
    :"EMPLOYEE_DETAILS","columns":{"EMPLOYEE_ID":{"alias":"EMPLOYEE_ID","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang
    .Integer":"numeric"}},"EMPLOYEE_NAME":{"alias":"EMPLOYEE_NAME","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String"
    :"text"}},"AGE":{"alias":"AGE","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"ADDRESS":{"alias"
    :"ADDRESS","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}}},"GEOCORDINATES":{"id":"f9fed894-88e8-4945
    -90db-13a439794729","alias":"GEOCORDINATES","columns":{"LOCATION_ID":{"alias":"LOCATION_ID","defaultFunction":"db.generic.aggregate.sum"
    ,"type":{"java.lang.Integer":"numeric"}},"LOCATION":{"alias":"LOCATION","defaultFunction":"db.generic.groupBy.group","type":{"java.lang
    .String":"text"}},"LATITUDE":{"alias":"LATITUDE","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Double":"numeric"}}
```

3.10 Save Report

URL	saveReport.html	
Description	To save the .efw report we can use this api service.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_USER, ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/saveReport.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&command=add&reportName=testsaveReport&reportDirectory=HelicalDemo&reportFile=demo.efw&location=1507554717873&reportParameters={'TERRITORY':['Japan'],'STERRITORY':'Japan'}" http://192.168.2.156:8085/hi-ee/saveReport.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
command:	add	Command to save the report
reportDirectory:	HelicalDemo	Directory of the report which we are going to save.
reportFile:	demo.efw	The report file physical name.
location:	1507554717873	Physical Location (Where you want to save report)
reportParameters: **(optional)**	{"TERRITORY":["Japan"],"STERRITORY":"Japan"}	Report parameters if report have.
reportName:	testsaveReport	Name of the report to be saved
Response Output(JSON Format)	{"status":1,"response":{"message":"Successfully saved the report!"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	

	It returns response as the success message and the report get saved in the respective directory having extension .efw
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/saveReport.html

Body: x-www-form-urlencoded

```
command:add
reportName:testsaveReport
reportDirectory:HelicalDemo
reportFile:demo.efw
location:1507554717873
reportParameters:{"TERRITORY":["Japan"],"STERRITORY":"Japan"}
```

Status: 200 OK Time: 56 ms Size: 377 B

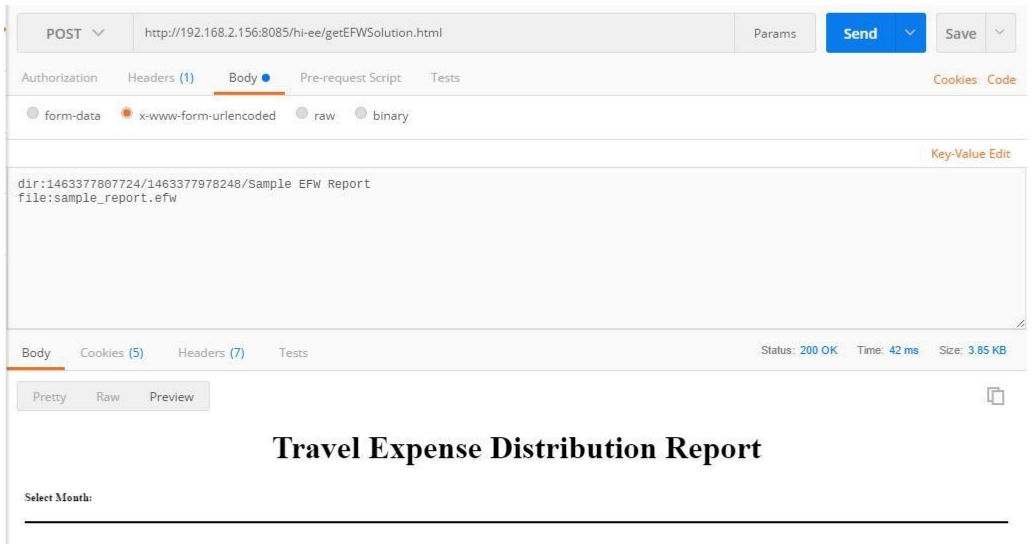
```
{"status":1,"response":{"message":"Successfully saved the report!"}}
```

3.11 Refresh cache/Report Operations

3.11.1 Refresh cache/Report Operations :: EFW Report

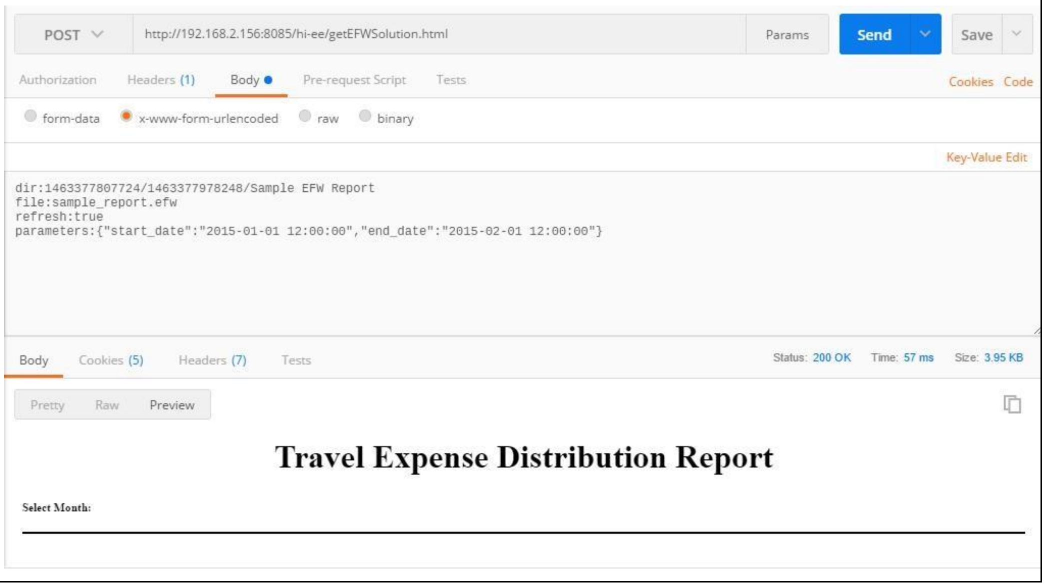
3.11.1.1 Refresh EFW Report

URL	getEFWSolution.html
Description	The current efw report will get refreshed. If we made any changes to report you can refresh same report using this API.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getEFWSolution.html **Access through Curl command :** curl --data

	"j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw" http://192.168.2.156:8085/hi-ee/getEFWSolution.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377978248 /Sample EFW Report	The directory for the report file
file:	sample_report.efw	Report file name
Response Output(JSON Fomat)	Response is the report which get refreshed	
Service Status	200 OK	
Screenshot	 POST http://192.168.2.156:8085/hi-ee/getEFWSolution.html x-www-form-urlencoded `dir:1463377807724/1463377978248/Sample EFW Report` `file:sample_report.efw` Status: 200 OK Time: 42 ms Size: 3.85 KB **Travel Expense Distribution Report** Select Month:	

3.11.1.2 Refresh EFW Report Cache

URL	getEFWSolution.html
Description	The current report cache get refreshed. Deletes the old cache of report and refresh the report cache.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/getEFWSolution.html

	Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample EFW Report&file=sample_report.efw&refresh=true&parameters={'start_date':'2015-01-01 12:00:00','end_date':'2015-02-01 12:00:00'}" http://192.168.2.156:8085/hi-ee/getEFWSolution.html -v	
HTTP Request Key	**HTTP Request value**	**Description**
dir:	1463377807724/1463377978248 /Sample EFW Report	The directory for the report file
file:	sample_report.efw	Report file name
refresh:	TRUE	Deletes the old cache and refresh the report
parameters:	{"start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00"}	Parameters of the report if any.
Response Output(JSON Fomat)	Response is the report cache get refreshed and returns the report contents.	
Service Status	200 OK	
Screenshot		

```
dir:1463377807724/1463377978248/Sample EFW Report
file:sample_report.efw
refresh:true
parameters:{"start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00"}
```

Travel Expense Distribution Report

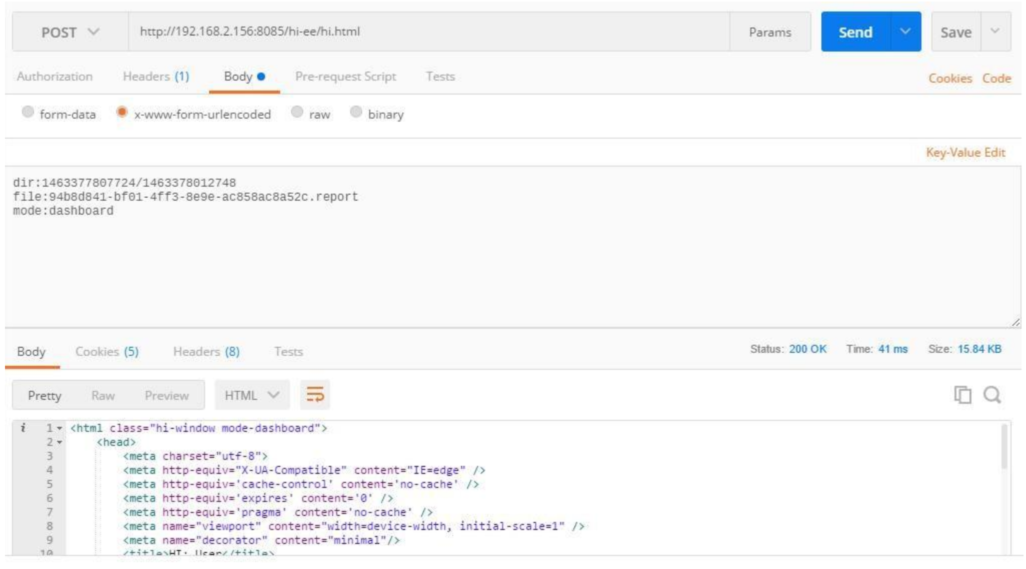
Select Month:

3.11.2 Refresh cache/Report Operations :: Adhoc Report

3.11.2.1 Refresh Adhoc Report

URL	hi.html

Description	The current adhoc report will get refreshed.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST,GET**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory for the report file
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	Report file name
mode:	dashboard	Mode of the report
Response Output(JSON Format)	Response is the report which get refreshed and returns the report contents.	
Service Status	200 OK	

Screenshot	

```
POST    http://192.168.2.156:8085/hi-ee/hi.html
```

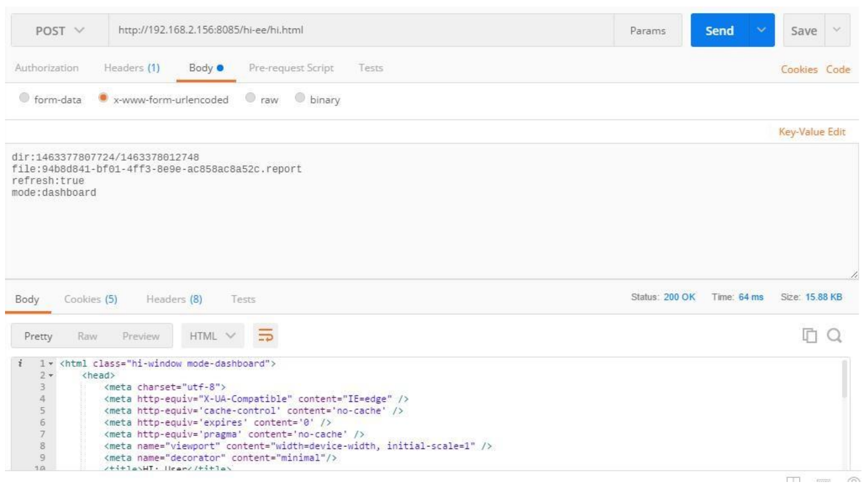
```
dir:1463377807724/1463378012748
file:94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report
mode:dashboard
```

Status: 200 OK Time: 41 ms Size: 15.84 KB

```
<html class="hi-window mode-dashboard">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv='cache-control' content='no-cache' />
        <meta http-equiv='expires' content='0' />
        <meta http-equiv='pragma' content='no-cache' />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <meta name="decorator" content="minimal"/>
```

3.11.2.2 Refresh Adhoc Report Cache

URL	hi.html	
Description	The current report cache get refreshed. Deletes the old cache of report and refresh the report cache.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard&refresh=true **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&mode=dashboard&refresh=true" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory for the report file

file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	Report file name
refresh:	true	Deletes the old cache and refresh the report
mode:	dashboard	Mode of the report
Response Output(JSON Format)	Response is the report cache which get refreshed and returns the report contents.	
Service Status	200 OK	
Screenshot		

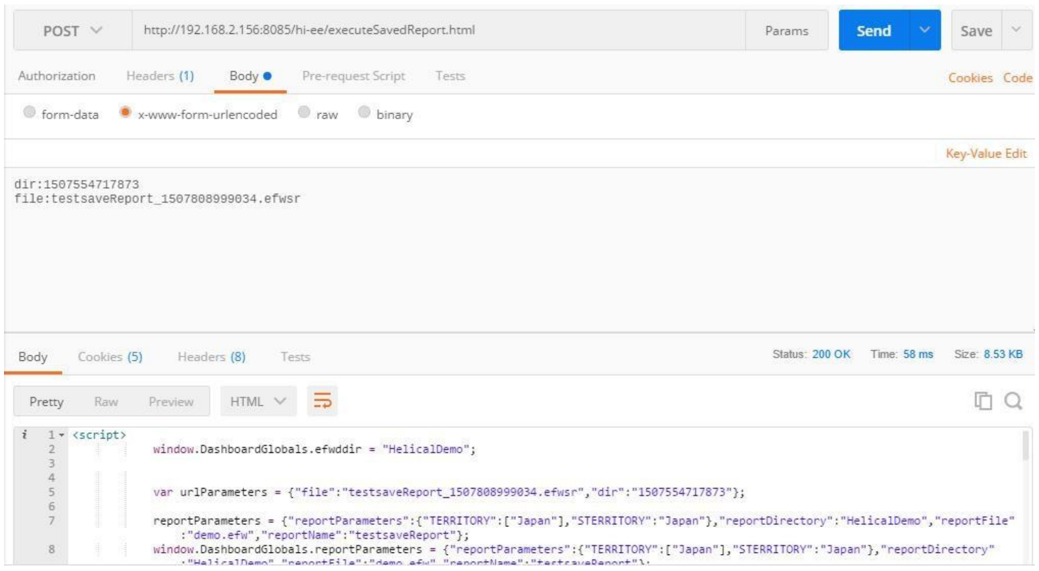
```
dir:1463377807724/1463378012748
file:94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report
refresh:true
mode:dashboard
```

```
<html class="hi-window mode-dashboard">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv='cache-control' content='no-cache' />
        <meta http-equiv='expires' content='0' />
        <meta http-equiv='pragma' content='no-cache' />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <meta name="decorator" content="minimal"/>
```

3.11.3 Refresh cache/Report Operations :: Saved/efwsr Report

3.11.3.1 Refresh Saved Report

URL	executeSavedReport.html
Description	The current saved report will get refreshed.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/executeSavedReport.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1507554717873&file=testsaveReport_1507808999034.efwsr" http://192.168.2.156:8085/hi-

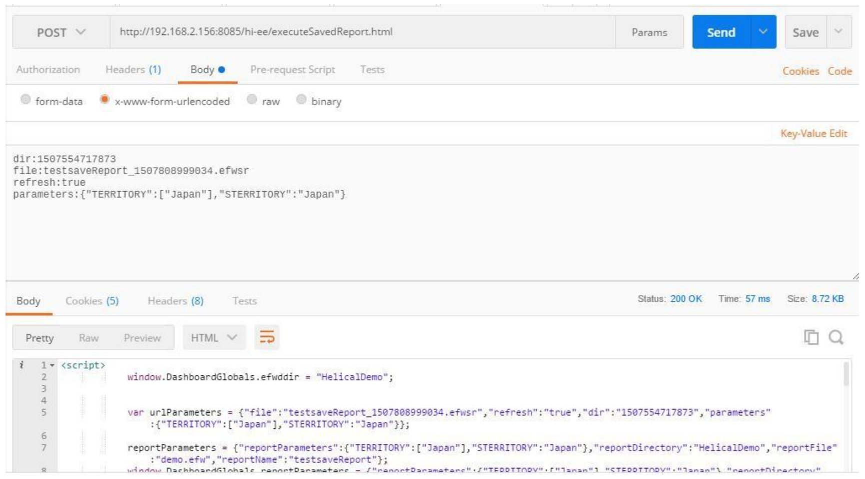
	ee/executeSavedReport.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1507554717873	The directory for the report file
file:	testsaveReport_1507808999034.efwsr	Report file name
Response Output(JSON Format)	Response is the report which get refreshed and returns the report contents.	
Service Status	200 OK	
Screenshot		

```
dir:1507554717873
file:testsaveReport_1507808999034.efwsr
```

```
<script>
        window.DashboardGlobals.efwddir = "HelicalDemo";


        var urlParameters = {"file":"testsaveReport_1507808999034.efwsr","dir":"1507554717873"};

        reportParameters = {"reportParameters":{"TERRITORY":["Japan"],"STERRITORY":"Japan"},"reportDirectory":"HelicalDemo","reportFile"
            :"demo.efw","reportName":"testsaveReport"};
        window.DashboardGlobals.reportParameters = {"reportParameters":{"TERRITORY":["Japan"],"STERRITORY":"Japan"},"reportDirectory"
            :"HelicalDemo","reportFile":"demo.efw","reportName":"testsaveReport"};
```

3.11.3.2 Refresh Saved Report Cache

URL	executeSavedReport.html
Description	The cache in the server side is refreshed.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/executeSavedReport.html **Access through Curl command :**

	curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&refresh=true&parameters={'start_date':'2015-01-01 12:00:00','end_date':'2015-02-01 12:00:00'}" http://192.168.2.156:8085/hi-ee/executeSavedReport.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1472554245045	The directory for the report file
file:	SavedReport_1472554274862.efwsr	Report file name
refresh:	true	Deletes the old cache and refresh the report
parameters:	{"start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00"}	Parameters of the report if any
Response Output(JSON Format)	Response is the report cache which get refreshed and returns the report contents.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee/executeSavedReport.html

Body: x-www-form-urlencoded

```
dir:1507554717873
file:testsaveReport_1507808999034.efwsr
refresh:true
parameters:{"TERRITORY":["Japan"],"STERRITORY":"Japan"}
```

Status: 200 OK Time: 57 ms Size: 8.72 KB

```
<script>
        window.DashboardGlobals.efwddir = "HelicalDemo";


        var urlParameters = {"file":"testsaveReport_1507808999034.efwsr","refresh":"true","dir":"1507554717873","parameters"
            :{"TERRITORY":["Japan"],"STERRITORY":"Japan"}};

        reportParameters = {"reportParameters":{"TERRITORY":["Japan"],"STERRITORY":"Japan"},"reportDirectory":"HelicalDemo","reportFile"
            :"demo.efw","reportName":"testsaveReport"};
```

3.12 Scheduling a Report

3.13 Emailing a Report

URL	sendMail.html	
Description	It allows user to mail reports(EFW,EFWSR,Adhoc reoprt) to any receipts . To mail report we need to pass required information covered on HTTP Request Key-Value section.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/sendMail.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463983915686/1463838054907&reportFile=d1560c88-be0d-4380-8225-8a8df4eb53bf.report&reportType=report&formats=[%22pdf%22,%22png%22,%22jpg%22]&recipients=[%22sayali@helicaltech.com%22]&reportSourceType=url&reportParameters={}&subject=testmail&body=Hello&reportName=Simple%20Bubble%20Chart" http://192.168.2.156:8085/hi-ee/sendMail.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**

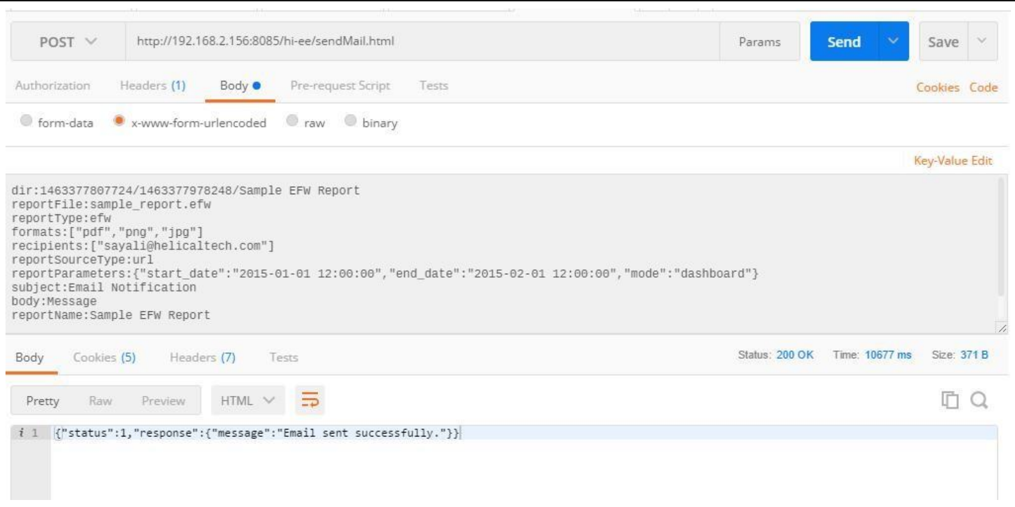
dir:	1463377807724/1463377978248/Sample EFW Report	Location of the file for email
reportFile:	sample_report.efw	File name which we are going to email
reportType:	efw	Type of the report
formats:	["pdf","png","jpg"]	The format in which the report is to be received via email. User can select one or more export format to mail.
recipients:	["sayali@helicaltech.com"]	Recipients email address
reportSourceType:	url	Report Source Type

reportParameters:	{
	"start_date":"2015-01-01 12:00:00",
	"end_date":"2015-02-01 12:00:00"
	}
		Report parameters for email if any

subject:	Email Notification	Email Subject
body: **(optional)**	Message	Message to be included in the body of email
reportName: **(optional)**	Sample EFW Report	Name of the report in attachment

Response Output(JSON Format)	{
	"status": 1,
	"Response": {
	"message": "Email sent successfully."
	}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/sendMail.html

```
dir:1463377807724/1463377978248/Sample EFW Report
reportFile:sample_report.efw
reportType:efw
formats:["pdf","png","jpg"]
recipients:["sayali@helicaltech.com"]
reportSourceType:url
reportParameters:{"start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00","mode":"dashboard"}
subject:Email Notification
body:Message
reportName:Sample EFW Report
```

Status: 200 OK Time: 10677 ms Size: 371 B

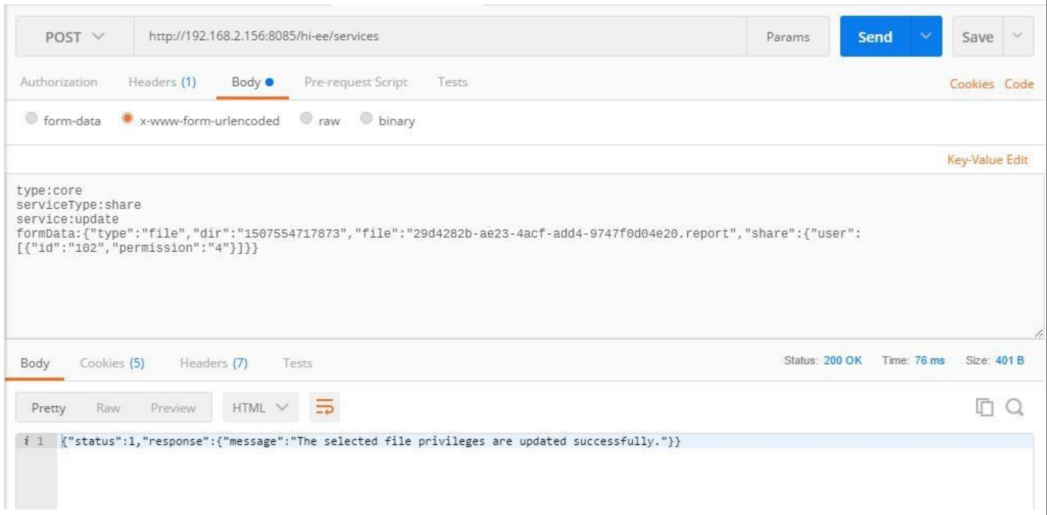
```
{"status":1,"response":{"message":"Email sent successfully."}}
```

3.14 Share Module

3.14.1 Share/Revoke Report

3.14.1.1 Share Report

URL	services	
Description	It allows user to share the report(EFW,EFWSR,Adhoc,EFWDD,Metdata,Result) with any user/organisation/role.We can set different permissions while sharing report , while sharing file with particular user/role/organisation we need to set the permission using permissionID(Refer to get PermissionID) If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Note : Here , we are taking example of Adhoc report for sharing and sharing it with user same way we can share report with organisation/role.** **Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507554717873','file':'29d4282b-ae23-4acf-add4-9747f0d04e20.report','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.

formData:	{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.
Response Output(JSON Format)	{ "status":1, "response":{"message":"The selected file privileges are updated successfully."} }	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee/services

Body: x-www-form-urlencoded

```
type:core
serviceType:share
service:update
formData:{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","share":{"user":
[{"id":"102","permission":"4"}]}}
```

Status: 200 OK Time: 76 ms Size: 401 B

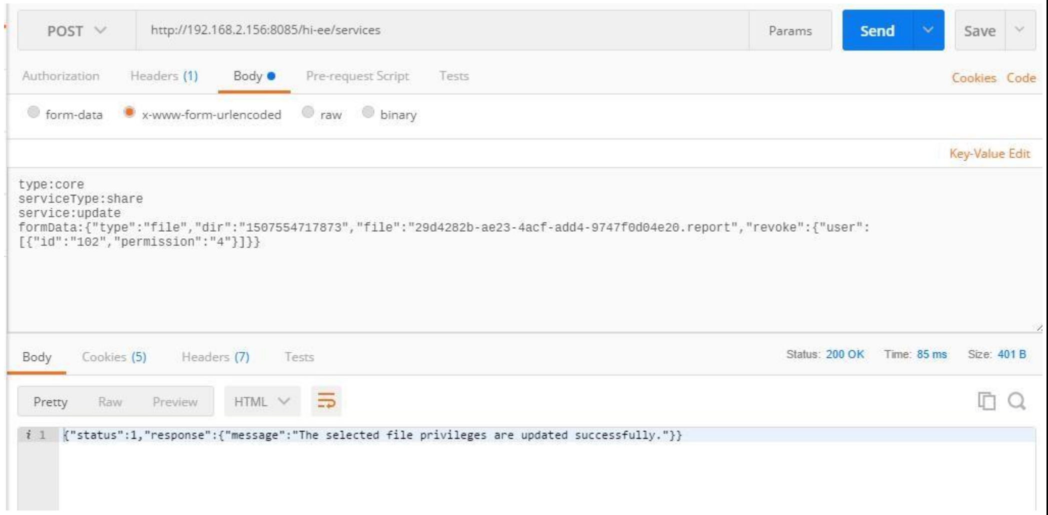
```
{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}
```

3.14.1.2 Revoke Report

URL	services	
Description	It allows user to revoke the already shared report(EFW,EFWSR,Adhoc,EFWDD,Metdata,Result) with any user/organisation/role. If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Note : Here , we are taking example of Adhoc report for revoking of already shared report.** **Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'file','dir':'1507554717873','file':'29d4282b-ae23-4acf-add4-9747f0d04e20.report','revoke':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.

formData:	{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","revoke":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type of the file , its dir where the file is present and the file name and the revoke info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The selected file privileges are updated successfully."}
	}	

Service Status	200 OK	
Screenshot		

```
POST http://192.168.2.156:8085/hi-ee/services
```

```
type:core
serviceType:share
service:update
formData:{"type":"file","dir":"1507554717873","file":"29d4282b-ae23-4acf-add4-9747f0d04e20.report","revoke":{"user":
[{"id":"102","permission":"4"}]}}
```

Status: 200 OK Time: 85 ms Size: 401 B

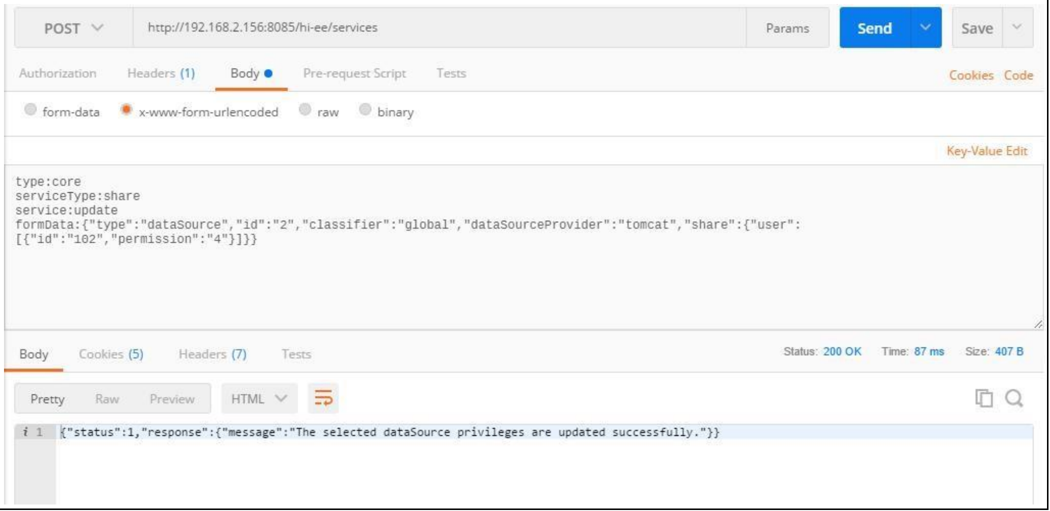
```
{"status":1,"response":{"message":"The selected file privileges are updated successfully."}}
```

3.14.2 Share/Revoke Datasource

3.14.2.1 Share Datasource

URL	services	
Description	It allows user to share the datasource with any user/organisation/role. The datasource will get share with provided user/organisation/role with permission.While sharing Datasource we need to update the connectionID ,datasource classifier,dataSourceProvider of datasource. If the file/directory doesnot exists you will get an error.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Note : We can share datasource with user/organisation/role.** **Here , we are taking example to share datasource with user.** **Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'dataSource','id':'2','classifier':'global','dataSourceProvider':'tomcat','share':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.

formData:	{"type":"dataSource","id":"2","classifier":"global","dataSourceProvider":"tomcat","share":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type as datasource , its connectionID ,datasource provider and the share info which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{ "status":1,"response":{"message":"The selected dataSource privileges are updated successfully."} }
Service Status	200 OK
Screenshot	

http://192.168.2.156:8085/hi-ee/services

```
type:core
serviceType:share
service:update
formData:{"type":"dataSource","id":"2","classifier":"global","dataSourceProvider":"tomcat","share":{"user":
[{"id":"102","permission":"4"}]}}
```

Status: 200 OK Time: 87 ms Size: 407 B

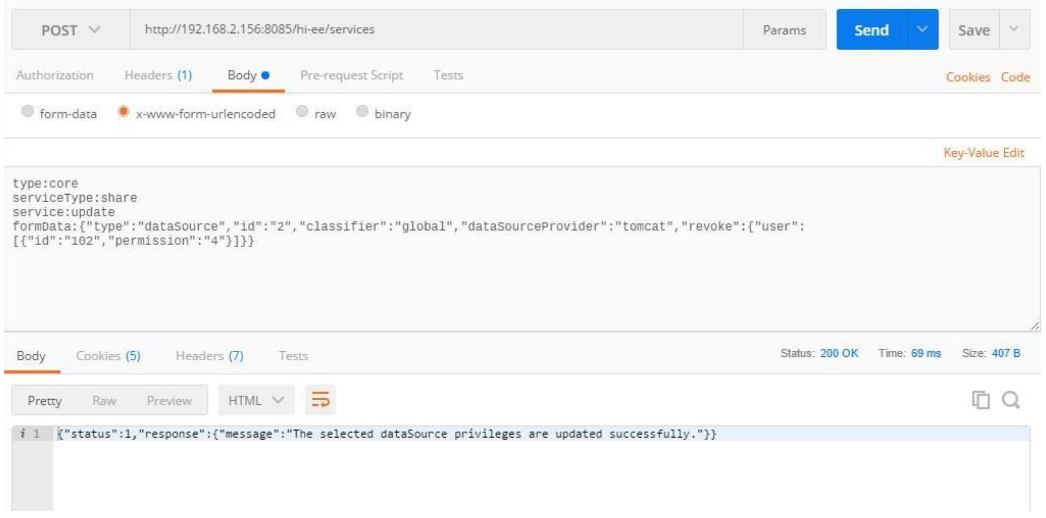
```
{"status":1,"response":{"message":"The selected dataSource privileges are updated successfully."}}
```

3.14.2.2 Revoke Datasource

URL	services	
Description	It allows user to revoke the already shared datasource with any user/organisation/role.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=share&service=update&formData={'type':'dataSource','id':'2','classifier':'global','dataSourceProvider':'tomcat','revoke':{'user':[{'id':'102','permission':'4'}]}}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request values**	**Description**
type:	core	Type of the operation.
serviceType:	share	ServiceType as share .
service:	update	Service to update the share information.
formData:	{"type":"dataSource","id":"2","classifier":"global","dataSourceProvider":"tomcat","revoke":{"user":[{"id":"102","permission":"4"}]}}	formData: getting pass to service tells the type as datasource ,datasource connection id,provider of the datasource etc and the revoke array with id which is nothing but the user ID(To know ID of the user) and the permission id (Click here to check permissionID) which we are going to set while sharing.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"The selected dataSource privileges are updated successfully."}
	}	

Service Status	200 OK

Screenshot	

```
type:core
serviceType:share
service:update
formData:{"type":"dataSource","id":"2","classifier":"global","dataSourceProvider":"tomcat","revoke":{"user":
[{"id":"102","permission":"4"}]}}
```

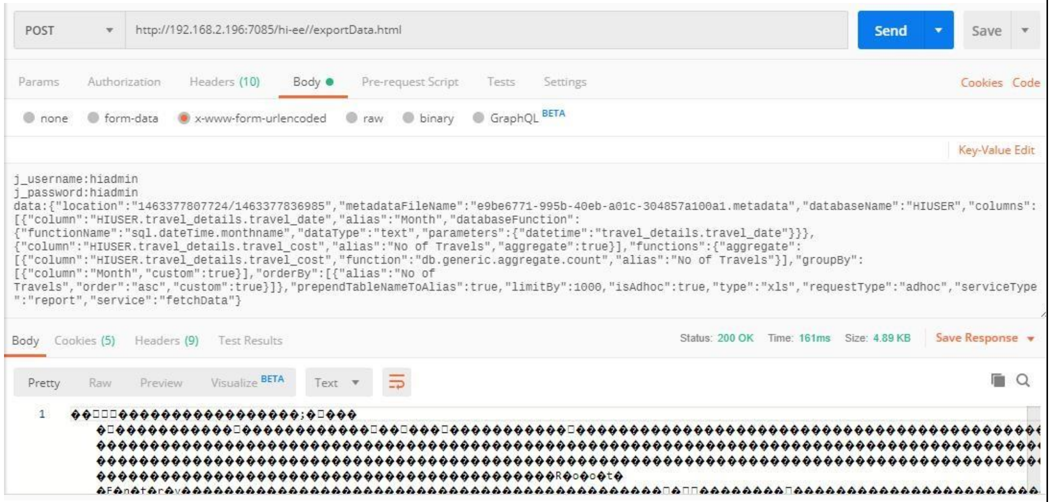
```
{"status":1,"response":{"message":"The selected dataSource privileges are updated successfully."}}
```

3.15 Export Excel / CSV

URL	//exportData.html
Description	It allows user to export the report in xls/csv format.User need to mention the type in data as(xls/csv) fo specific export.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER(Note : User should have share permission)
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&data={"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","databaseName":"HIUSER","columns":[{"column":"HIUSER.travel_details.travel_date","alias":"Month","databaseFunction":{"functionName":"sql.dateTime.monthname","dataType":"text","parameters":{"datetime":"travel_details.travel_date"}}},{"column":"HIUSER.travel_details.travel_cost","alias":"No of Travels","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.travel_details.travel_cost","function":"db.generic.aggregate.count","alias":"No of Travels"}],"groupBy":[{"column":"Month","custom":true}],"orderBy":[{"alias":"No of Travels","order":"asc","custom":true}]},"prependTableNameToAlias":true,"limitBy":1000,"isAdhoc":true,"type":"xls","requestType":"adhoc","serviceType":"report","service":"fetchData"}' http://localhost:7085/hi-ee//exportData.html -v

HTTP Request Key	**HTTP Request values**	**Description**
j_username:	hiadmin	Username for helical insight
j_password:	hiadmin	Password for helical insight

data:	{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","databaseName":"HIUSER","columns":[{"column":"HIUSER.travel_details.travel_date","alias":"Month","databaseFunction":{"functionName":"sql.dateTime.monthname","dataTy	For data we need to provide the location of metadata file, its name, database name , column, if any db functions applied, alias etc. User need to set the type of data as xls/csv for xls and csv export.

	pe":"text","parameters":{"datetime":"travel_details.travel_date"}}},{"column":"HIUSER.travel_details.travel_cost","alias":"No of Travels","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.travel_details.travel_cost","function":"db.generic.aggregate.count","alias":"No of Travels"}],"groupBy":[{"column":"Month","custom":true}],"orderBy":[{"alias":"No of Travels","order":"asc","custom":true}]},"prependTableNameToAlias":true,"limitBy":1000,"isAdhoc":true,"type":"xls","requestType":"adhoc","serviceType":"report","service":"fetchData"}	
Response Output(JSON Format)	Reponse will be in text format which will be some contents of CSV and XLS.Those contents will not be in user readable format.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.196:7085/hi-ee//exportData.html

Body: x-www-form-urlencoded

```
j_username:hiadmin
j_password:hiadmin
data:{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","databaseName":"HIUSER","columns":
[{"column":"HIUSER.travel_details.travel_date","alias":"Month","databaseFunction":
{"functionName":"sql.dateTime.monthname","dataType":"text","parameters":{"datetime":"travel_details.travel_date"}}},
{"column":"HIUSER.travel_details.travel_cost","alias":"No of Travels","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.travel_details.travel_cost","function":"db.generic.aggregate.count","alias":"No of Travels"}],"groupBy":
[{"column":"Month","custom":true}],"orderBy":[{"alias":"No of
Travels","order":"asc","custom":true}]},"prependTableNameToAlias":true,"limitBy":1000,"isAdhoc":true,"type":"xls","requestType":"adhoc","serviceType
":"report","service":"fetchData"}
```

Status: 200 OK Time: 161ms Size: 4.89 KB

```
1  ��□□□��������������������;�□���
     �□������������□�������������□��□���□������������□����������������������������������������������
     ����������������������������������������������������������������������������������������������
     ����������������������������������������������������������������������������������������������
     ������������������������������������������R�o�o�t�
```

4. Adhoc Module

Adhoc module contains 3 major components: Datasource, Metadata and Report. Datasource allows to create,edit and Share Datasource, then create Metadata using datasource and edit metadata. Then finally create adhoc report using metadata by drag and drop functionality.

4.1 Create Datasource

4.1.1 Create Managed Datasource

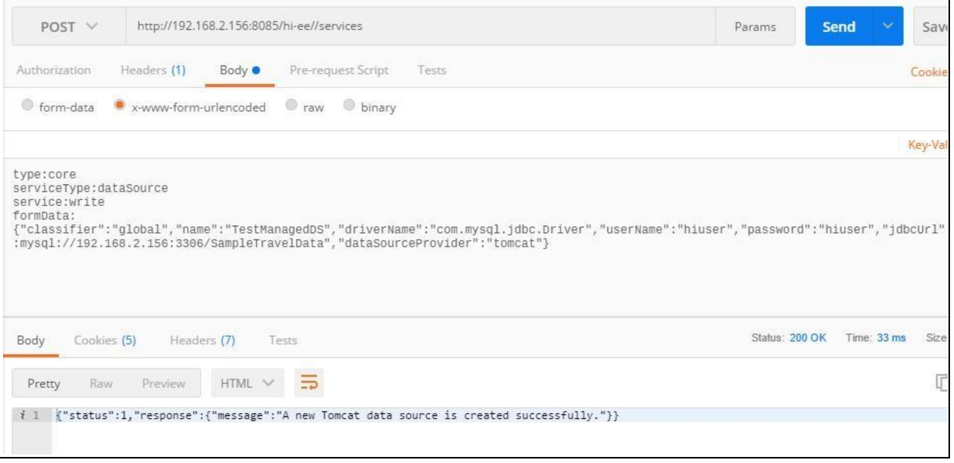
URL	/services
Description	It allows user to create managed datasources which supports different databases. Managed datasource is the global datasource connection which get saved to globalConnections.xml file in backend. Note : While datasource connection different datasource providers are available we can use same as well(Find in Advanced Option section).

Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Note : Here we are creating the MySQL db connection using managed datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=write&formData={'classifier':'global','name':'Test ManagedDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/SampleTravelData','dataSourceProvider':'tomcat'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	write	The service is write to add new datasource connection
formData:	{"classifier":"global","name":"TestManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/SampleTravelData","dataSourceProvider":"tomcat"}	name : name of the datasource Classifer as global All datasource connection details like drivername,username,password, connecion string and the dataSource provider name.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"A new Tomcat data source is created successfully."}
	}	

Description of Response Output	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. The managed datasource get saved in the backend. Managed datasource is the global connection ,all connection details get saved in globalConnections.xml in backend.	

Service Status	200 OK

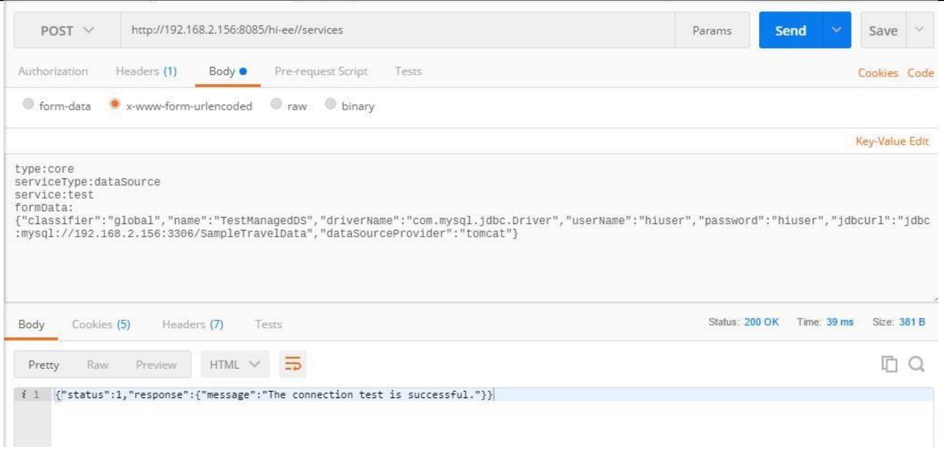
Screenshot	 POST http://192.168.2.156:8085/hi-ee//services
	Body: x-www-form-urlencoded
	`type:core`
	`serviceType:dataSource`
	`service:write`
	`formData:`
	`{"classifier":"global","name":"TestManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl"`
	`:mysql://192.168.2.156:3306/SampleTravelData","dataSourceProvider":"tomcat"}`
	Status: 200 OK Time: 33 ms
	`{"status":1,"response":{"message":"A new Tomcat data source is created successfully."}}`

4.1.2 Test Managed Datasource

URL	/services
Description	It allows user to test the provided connection details while creating the managed datasource connection , so that before saving the datasource we can test the provided connection. Note : If the connection details(host,username,password,dbname) are not correct , then you will get an Exception.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Note : Here we are creating and testing the MySQL db connection using managed datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=test&formData={'classifier':'global','name':'TestManagedDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','pass

	word':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/SampleTravelData','dataSourceProvider':'tomcat'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	test	The service is to test the provided connection details while creating managed datasource connection.
formData:	{"classifier":"global","name":"TestManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/SampleTravelData","dataSourceProvider":"tomcat"}"	name : name of the datasource Classifer as global All datasource connection details like drivername,username,password, connecion string and the dataSource provider name.

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The connection test is successful"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. If all the credential and jdbc url is correct the user will get success message.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

```
type:core
serviceType:dataSource
service:test
formData:
{"classifier":"global","name":"TestManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc
:mysql://192.168.2.156:3306/SampleTravelData","dataSourceProvider":"tomcat"}
```

Status: 200 OK Time: 39 ms Size: 381 B

```
{"status":1,"response":{"message":"The connection test is successful."}}
```

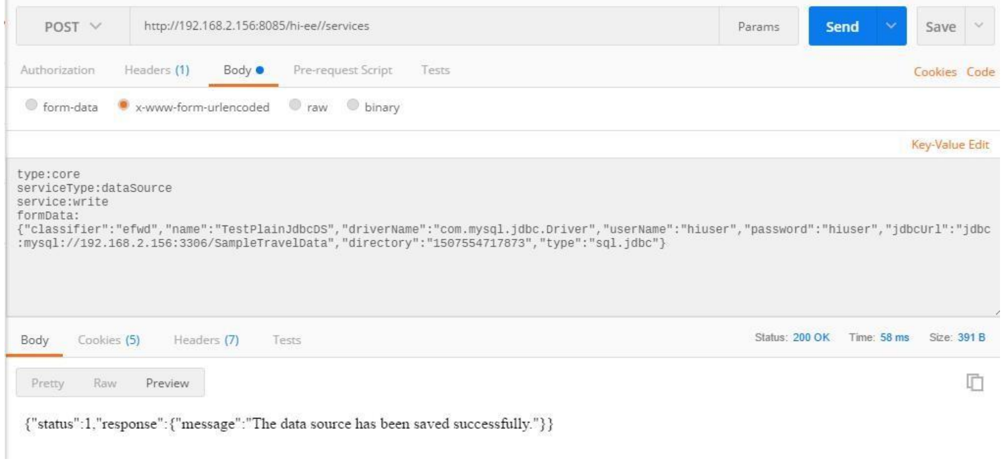
4.1.3 Create Plain JDBC Datasource

URL	/services

Description	It allows user to create plain datasources.After creation of plain jdbc datasource it get save at provided directory with .efwd extension.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Note : Here we are creating the MySQL db connection using plain Jdbc datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=write&formData={'classifier':'efwd','name':'TestPlainJdbcDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/SampleTravelData','directory':1507554717873','type':'sql.jdbc'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	write	The service is write to add new plain jdbc datasource connection
formData:	{"classifier":"efwd","name":"TestPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/SampleTravelData","directory":"1507554717873","type":"sql.jdbc"}	name : name of the datasource Classifer as efwd All datasource connection details like drivername,username,password, connecion string and the type of datasource with directory name where the datasource will get save.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"The data source has been saved successfully."}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message.	

	The plain jdbc datasource get created and get save at provided directory with .efwd extension.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

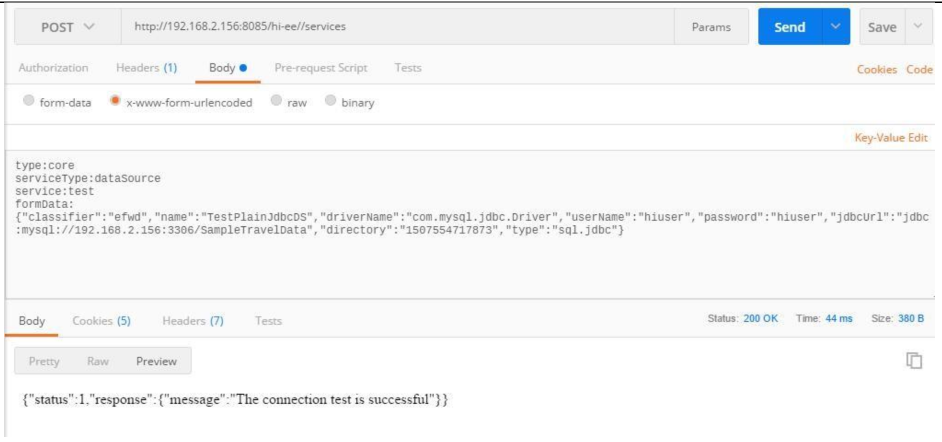
```
type:core
serviceType:dataSource
service:write
formData:
{"classifier":"efwd","name":"TestPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc
:mysql://192.168.2.156:3306/SampleTravelData","directory":"1507554717873","type":"sql.jdbc"}
```

Status: 200 OK Time: 58 ms Size: 391 B

```
{"status":1,"response":{"message":"The data source has been saved successfully."}}
```

4.1.4 Test Plain JDBC Datasource

URL	/services
Description	It allows user to test the provided connection details while creating the plain jdbc datasource connection , so that before saving the datasource we can test the provided connection. Note : If the connection details(host,username,password,dbname) are not correct , then you will get an Exception.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Note : Here we are creating the MySQL db connection using plain Jdbc datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :**

	curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=test&formData={'classifier':'efwd','name':'TestPlainJdbcDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/SampleTravelData','directory':1507554717873','type':'sql.jdbc'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	test	The service is to test the provided connection details while creating plain jdbc datasource connection.
formData:	{"classifier":"efwd","name":"TestPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/SampleTravelData","directory":"1507554717873","type":"sql.jdbc"}	name : name of the datasource Classifer as efwd All datasource connection details like drivername,username,password, connecion string and the type of datasource with directory name where the datasource will get save.
Response Output(JSON Format):	{ "status":1,"response":{"message":"The connection test is successful"} }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. If all the credential and jdbc url is correct the user will get success message.	
Service Status	200 OK	
Screenshot		

```
type:core
serviceType:dataSource
service:test
formData:
{"classifier":"efwd","name":"TestPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc
:mysql://192.168.2.156:3306/SampleTravelData","directory":"1507554717873","type":"sql.jdbc"}
```

POST http://192.168.2.156:8085/hi-ee//services — Body: x-www-form-urlencoded

Status: 200 OK Time: 44 ms Size: 380 B

```
{"status":1,"response":{"message":"The connection test is successful"}}
```

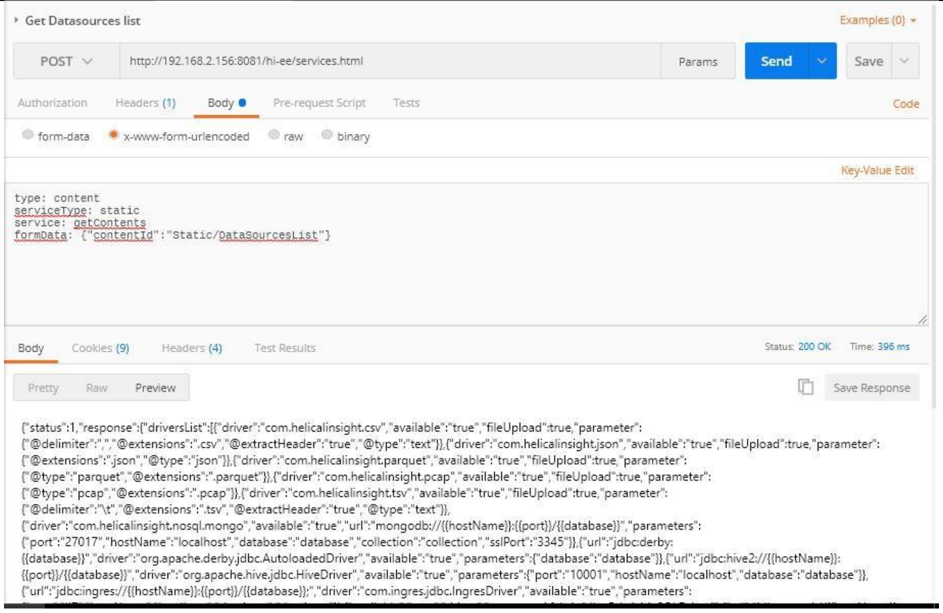
4.1.5 Get Data sources list

URL	/services.html	
Description	User will get the list of datasources in detail	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=content&serviceType=static&service=getContents&formData={'contentId':'Static/DataSourcesList'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	static	serviceType as static
service:	getContents	Service as getContents
formData:	{"contentId":"Static/DataSourcesList"}	Action to get the data source list
Response Output(JSON Format):	{"status":1,"response":{"driversList":[{"driver":"com.helicalinsight.csv","available":"true","fileUpload":true,"parameter":{"@delimiter":",","@extensions":".csv","@extractHeader":"true","@type":"text"}},{"driver":"com.helicalinsight.json","available":"true","fileUpload":true,"parameter":{"@extensions":".json","@type":"json"}},{"driver":"com.helicalinsight.parquet","available":"true","fileUpload":true,"parameter":{"@type":"parquet","@extensions":".parquet"}},{"driver":"com.helicalinsight.pcap","available":"true","fileUpload":true,"parameter":{"@type":"pcap","@extensions":".pcap"}},{"driver":"com.helicalinsight.tsv","available":"true","fileUpload":true,"parameter":{"@delimiter":"\t","@extensions":".tsv","@extractHeader":"true","@type":"text"}},{"driver":"com.helicalinsight.nosql.mongo","available":"true","url":"mongodb://{{hostName}}:{{port}}/{{database}}","parameters":{"port":"27017","hostName":"localhost","database":"database","collection":"collection","sslPort":"3345"}},{"url":"jdbc:derby:{{database}}","driver":"org.apache.derby.jdbc.AutoloadedDriver","available":"true","parameters":{"database":"database"}},{"url":"jdbc:hive2://{{hostName}}:{{port}}/{{database}}","driver":"org.apache.hive.jdbc.HiveDriver","available":"true","parameters":{"port":"10001","hostName":"localhost","database":"database"}},{"url":"jdbc:ingres://{{host	

	Name}}:{{port}}/{{database}};","driver":"com.ingres.jdbc.IngresDriver","available":"true","parameters":{"port":"II7","hostName":"localhost","database":"database"}},{"available":"true","driver":"com.mysql.fabric.jdbc.FabricMySQLDriver"},{"url":"jdbc:mysql://{{hostName}}:{{port}}/{{database}}","driver":"com.mysql.jdbc.Driver","available":"true","parameters":{"port":"3306","hostName":"localhost","database":"database"}},{"url":"jdbc:drill:{{hostName}}:{{port}}","driver":"org.apache.drill.jdbc.Driver","available":"true","parameters":{"port":"31010","hostName":"drillbit=localhost"}},{"url":"jdbc:oracle:thin:@{{hostName}}:{{port}}:{{database}}","driver":"oracle.jdbc.OracleDriver","available":"true","parameters":{"port":"1521","hostName":"localhost","database":"database"}},{"url":"jdbc:mariadb://{{hostName}}:{{port}}/{{database}}","driver":"org.mariadb.jdbc.Driver","available":"true","parameters":{"port":"3306","hostName":"localhost","database":"database"}},{"url":"jdbc:postgresql://{{hostName}}:{{port}}/{{database}}","driver":"org.postgresql.Driver","available":"true","parameters":{"port":"5433","hostName":"localhost","database":"database"}},{"url":"jdbc:jtds:sqlserver://{{hostName}}:{{port}}/{{database}}","driver":"net.sourceforge.jtds.jdbc.Driver","available":"true","parameters":{"port":"1433","hostName":"localhost","database":"database"}},{"url":"jdbc:hive2://{{hostName}}:{{port}}/{{database}}","driver":"org.apache.hive.jdbc.HiveDriver","available":"true","parameters":{"port":"10001","hostName":"localhost","database":"database"}},{"url":"jdbc:sqlite:{{database}}","driver":"org.sqlite.JDBC","available":"true","parameters":{"database":"database"}}],"dataSourceTypes":[{"type":"global.jdbc","name":"Managed DataSource","classifier":"global","categoryName":"advanced","categoryType":"advanced"},{"type":"sql.jdbc","name":"Plain Jdbc DataSource","classifier":"efwd","categoryName":"advanced","categoryType":"advanced"},{"type":"sql.jdbc.groovy","name":"Groovy Plain Jdbc DataSource","classifier":"efwd","categoryName":"advanced","categoryType":"advanced"}],"dataSources":[{"driver":"org.apache.drill.jdbc.Driver","databaseDialect":"drill","enabledTypes":true,"name":"Apache Drill","categoryName":"Big Data","categoryType":"big_data","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:drill:{{hostName}}:{{port}}","parameters":{"port":"31010","hostName":"drillbit=localhost"}},{"driver":"org.postgresql.Driver","databaseDialect":"postgresql","name":"Postgresql","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:postgresql://{{hostName}}:{{port}}/{{database}}","parameters":{"port":"5433","hostName":"localhost","database":"database"}},{"driver":"net.sourceforge.jtds.jdbc.Driver","databaseDialect":"sqlserver","name":"Microsoft Sqlserver(sourceforge)","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:jtds:sqlserver://{{hostName}}:{{port}}/{{database}}","parameters":{

	"port":"1433","hostName":"localhost","database":"database"}},{"driver":"com.helicalinsight.tsv","databaseDialect":"","name":"Tsv","categoryName":"Flat Files","categoryType":"flat_files","type":"global.jdbc","dataSourceProvider":"tomcat","fileUpload":true,"classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"driver":"oracle.jdbc.OracleDriver","databaseDialect":"oracle","name":"Oracle","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:oracle:thin:@{{hostName}}:{{port}}:{{database}}","parameters":{"port":"1521","hostName":"localhost","database":"database"}},{"name":"IBM Db2","categoryType":"supported","categoryName":"Supported"},{"name":"Informix","categoryType":"supported","categoryName":"Supported"},{"categoryName":"No SQL & Big Data","categoryType":"nosql_bigdata","classifier":"global","dataSourceProvider":"noSql","driver":"com.helicalinsight.nosql.mongo","name":"Mongodb","parameters":{"collection":"collection","database":"database","hostName":"localhost","port":"27017","sslPort":"3345"},"type":"global.jdbc","url":"mongodb://{{hostName}}:{{port}}/{{database}}"},{"driver":"com.helicalinsight.parquet","databaseDialect":"","name":"Parquet","categoryName":"Flat Files","categoryType":"flat_files","type":"global.jdbc","dataSourceProvider":"tomcat","fileUpload":true,"classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"name":"Microsoft Sqlserver","categoryType":"supported","categoryName":"Supported"},{"name":"Teradata","categoryType":"supported","categoryName":"Supported"},{"driver":"com.helicalinsight.pcap","databaseDialect":"","name":"Pcap","categoryName":"Flat Files","categoryType":"flat_files","type":"global.jdbc","dataSourceProvider":"tomcat","fileUpload":true,"classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"name":"Hsqldb","categoryType":"supported","categoryName":"Supported"},{"name":"Sybase Jdbc2","categoryType":"supported","categoryName":"Supported"},{"name":"Sybase Jdbc4","categoryType":"supported","categoryName":"Supported"},{"name":"Firebirdsql","categoryType":"supported","categoryName":"Supported"},{"driver":"com.helicalinsight.json","databaseDialect":"","name":"Json","categoryName":"Flat Files","categoryType":"flat_files","type":"global.jdbc","dataSourceProvider":"tomcat","fileUpload":true,"classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"driver":"com.helicalinsight.csv","databaseDialect":"","name":"Csv","categoryName":"Flat Files","categoryType":"flat_files","type":"global.jdbc","dataSourceProvider":"tomcat","fileUpload":true,"classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"driver":"org.sqlite.JDBC","databaseDialect":"sqlite","name":"Sqlite","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","

	classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:sqlite:{{database}}","parameters":{"database":"database"}},{"driver":"com.ingres.jdbc.IngresDriver","databaseDialect":"ingres","name":"Ingres","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:ingres://{{hostName}}:{{port}}/{{database}};","parameters":{"port":"II7","hostName":"localhost","database":"database"}},{"driver":"com.mysql.fabric.jdbc.FabricMySQLDriver","databaseDialect":"","name":"Mysql Fabric","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png"},{"driver":"org.apache.derby.jdbc.AutoloadedDriver","databaseDialect":"derby","name":"Derby","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:derby:{{database}}","parameters":{"database":"database"}},{"name":"Ξ Add Driver Ξ","categoryType":"supported","categoryName":"Supported"},{"driver":"org.mariadb.jdbc.Driver","databaseDialect":"mysql","name":"Mariadb","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:mariadb://{{hostName}}:{{port}}/{{database}}","parameters":{"port":"3306","hostName":"localhost","database":"database"}},{"name":"Presto","categoryType":"supported","categoryName":"Supported"},{"driver":"org.apache.hive.jdbc.HiveDriver","databaseDialect":"spark","name":"Hive","categoryName":"Big Data","categoryType":"big_data","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:hive2://{{hostName}}:{{port}}/{{database}}","parameters":{"port":"10001","hostName":"localhost","database":"database"}},{"name":"Access","categoryType":"supported","categoryName":"Supported"},{"driver":"com.mysql.jdbc.Driver","databaseDialect":"mysql","name":"Mysql","categoryName":"RDBMS","categoryType":"rdbms","type":"global.jdbc","dataSourceProvider":"tomcat","classifier":"global","imgUrl":"../images/data_sources/defaut_datasource.png","url":"jdbc:mysql://{{hostName}}:{{port}}/{{database}}","parameters":{"port":"3306","hostName":"localhost","database":"database"}}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK

Screenshot	

Get Datasources list

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

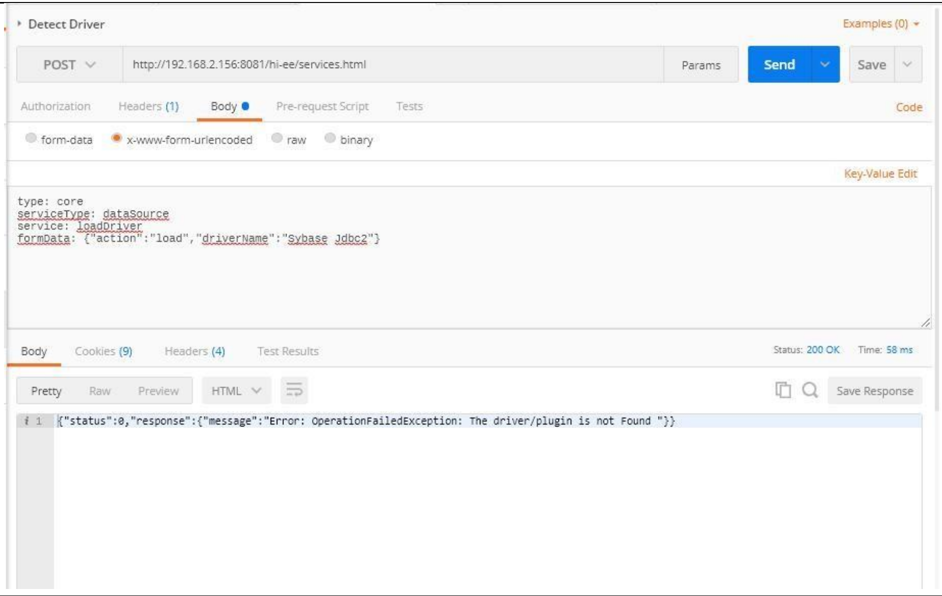
```
type: content
serviceType: static
service: getContents
formData: {"contentId":"Static/DataSourcesList"}
```

Status: 200 OK Time: 396 ms

```
{"status":1,"response":{"driversList":[{"driver":"com.helicalinsight.csv","available":"true","fileUpload":true,"parameter":
{"@delimiter":",","@extensions":".csv","@extractHeader":"true","@type":"text"}},{"driver":"com.helicalinsight.json","available":"true","fileUpload":true,"parameter":
{"@extensions":".json","@type":"json"}},{"driver":"com.helicalinsight.parquet","available":"true","fileUpload":true,"parameter":
{"@type":"parquet","@extensions":".parquet"}},{"driver":"com.helicalinsight.pcap","available":"true","fileUpload":true,"parameter":
{"@type":"pcap","@extensions":".pcap"}},{"driver":"com.helicalinsight.tsv","available":"true","fileUpload":true,"parameter":
{"@delimiter":"\t","@extensions":".tsv","@extractHeader":"true","@type":"text"}},
{"driver":"com.helicalinsight.nosql.mongo","available":"true","url":"mongodb://{{hostName}}:{{port}}/{{database}}","parameters":
{"port":"27017","hostName":"localhost","database":"database","collection":"collection","sslPort":"3345"}},{"url":"jdbc:derby:
{{database}}","driver":"org.apache.derby.jdbc.AutoloadedDriver","available":"true","parameters":{"database":"database"}},{"url":"jdbc:hive2://{{hostName}}:
{{port}}/{{database}}","driver":"org.apache.hive.jdbc.HiveDriver","available":"true","parameters":{"port":"10001","hostName":"localhost","database":"database"}},
{"url":"jdbc:ingres://{{hostName}}:{{port}}/{{database}};","driver":"com.ingres.jdbc.IngresDriver","available":"true","parameters":
```

4.1.6 Detect Driver

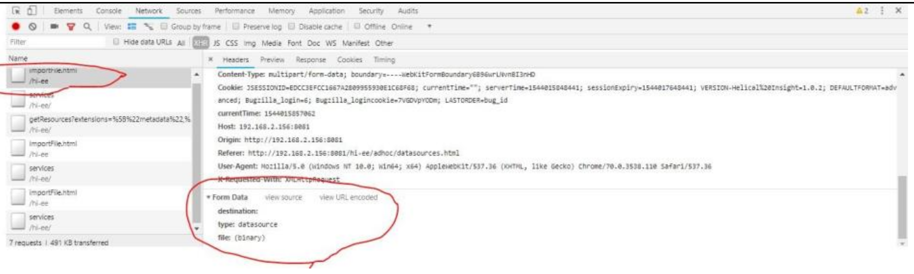
URL	/services.html
Description	User can detect the driver if it is not available in Driver/Plugins folder but if it is available in /lib folder in such case driver will be detected
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=loadDriver&formData={'action':'load','driverName':

	'Sybase Jdbc2'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as static
service:	loadDriver	Service as getContents
formData:	{"action":"load","driverName":"Sybase Jdbc2"}	Action to detect the driver in /lib folder
Response Output(JSON Format):	{"status":0,"response":{"message":"Error: OperationFailedException: The driver/plugin is not Found "}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

```
Detect Driver
POST  http://192.168.2.156:8081/hi-ee/services.html

type: core
serviceType: dataSource
service: loadDriver
formData: {"action":"load","driverName":"Sybase Jdbc2"}

Status: 200 OK   Time: 58 ms

{"status":0,"response":{"message":"Error: OperationFailedException: The driver/plugin is not Found "}}
```

4.1.7 Upload Database Driver/Jar/Zip/Flat(csv, json etc) file

URL	importFile.html
Description	User can upload database driver/Jar/Zip/flat files etc.If there is depedency jar files
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	http://192.168.2.156:8081/hi-ee/services.html

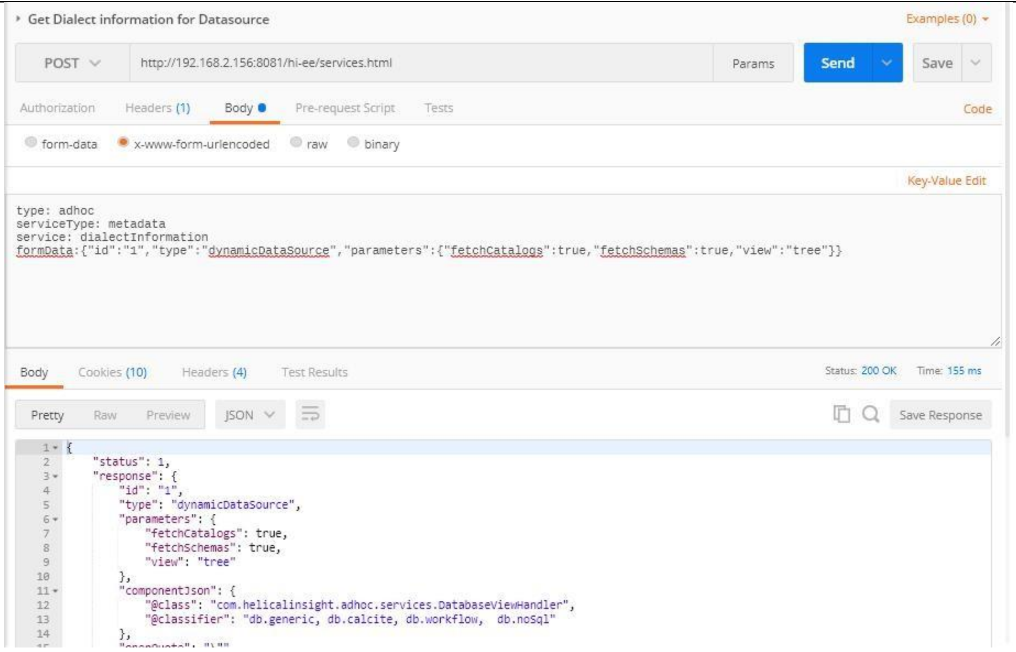
HTTP Request Key	**HTTP Request Value**	**Description**
type:	datasource/csv/csvh/json/tsv/psv/avro	typeas datasource,csv,json,tsv,psv,avro etc.
destination:		destination
file:	(binary)	File will be driver/jar/zip/ file
Response Output(JSON Format):	{"status":1,"response":{"message":"The file has been imported successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

```
Name
importFile.html
/hi-ee
services
/hi-ee/
getResources?extensions=%5B%22metadata%22,%
/hi-ee/
importFile.html
/hi-ee
services
/hi-ee/
importFile.html
/hi-ee
services
/hi-ee/
7 requests | 491 KB transferred

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary6896wrLNvn8I3nHD
Cookie: JSESSIONID=EDCC3EFCC1667A2809955930E1C68F68; currentTime=""; serverTime=1544015848441; sessionExpiry=1544017648441; VERSION-Helical%20Insight=1.0.2; DEFAULTFORMAT=advanced; Bugzilla_login=6; Bugzilla_logincookie=7VGDVpYODm; LASTORDER=bug_id
currentTime: 1544015857062
Host: 192.168.2.156:8081
Origin: http://192.168.2.156:8081
Referer: http://192.168.2.156:8081/hi-ee/adhoc/datasources.html
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
X-Requested-With: XMLHttpRequest
▾ Form Data
destination:
type: datasource
file: (binary)
```

4.1.8 Get Dialect Information for datasource

User will get dialect information as per datasource.

URL	/services.html	
Description	The user will get the dialect information as per provided datasource ID.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=dialectInformation&formData={'id':'1','type':'dynamicDataSource','parameters':{'fetchCatalogs':true,'fetchSchemas':true,'view':'tree'}}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request	**HTTP Request Value**	**Description**

Key		
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	dialectInformation	The service is to get the dialectInformation
formData:	{"id":"1","type":"dynamicDataSource","parameters":{"fetchCatalogs":true,"fetchSchemas":true,"view":"tree"}}	Provide datasource ID to get dialect information
Response Output(JSON Format)	{"status":1,"response":{"id":"1","type":"dynamicDataSource","parameters":{"fetchCatalogs":true,"fetchSchemas":true,"view":"tree"},"componentJson":{"@class":"com.helicalinsight.adhoc.services.DatabaseViewHandler","@classifier":"db.generic, db.calcite, db.workflow, db.noSql"},"openQuote":"\"","closeQuote":"\"","dialectName":"org.hibernate.dialect.DerbyTenSevenDialect"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

Get Dialect information for Datasource

POST http://192.168.2.156:8081/hi-ee/services.html

x-www-form-urlencoded

```
type: adhoc
serviceType: metadata
service: dialectInformation
formData:{"id":"1","type":"dynamicDataSource","parameters":{"fetchCatalogs":true,"fetchSchemas":true,"view":"tree"}}
```

Status: 200 OK Time: 155 ms

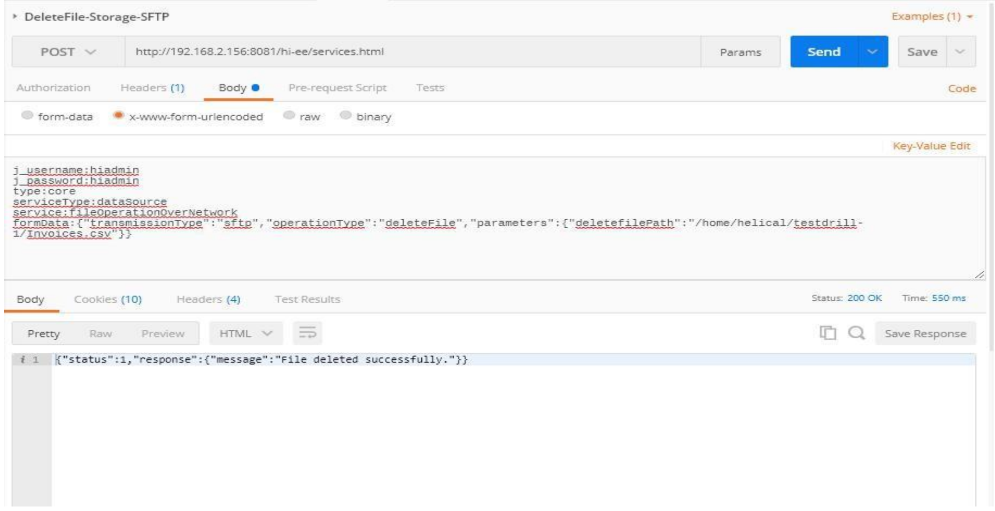
```
{
    "status": 1,
    "response": {
        "id": "1",
        "type": "dynamicDataSource",
        "parameters": {
            "fetchCatalogs": true,
            "fetchSchemas": true,
            "view": "tree"
        },
        "componentJson": {
            "@class": "com.helicalinsight.adhoc.services.DatabaseViewHandler",
            "@classifier": "db.generic, db.calcite, db.workflow,  db.noSql"
        },
```

4.1.9 Middleware:: Delete uploaded file/folder

User can delete uploaded file or folder as per storage Implementation(SFTP,HDFS).

4.1.9.1 Middleware:: Delete uploaded file

URL	/services.html	
Description	The user can delete uploaded file which is uploaded with middleware connection as per storage implementation for ex.sftp,hdfs	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=fileOperationOverNetwork&formData={'transmissionType':'sftp','operationType':'deleteFile','parameters':{'deletefilePath':'/home/helical/testdrill-1/Invoices.csv'}}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	fileOperationOverNetwork	The service is to delete uploaded file for fileOperationOverNetwork
formData:	{"transmissionType":"sftp","operationType":"deleteFile","parameters":{"deletefilePath":"/home/helical/testdrill-1/Invoices.csv"}}	transmissionType: user can provided storage imple type for.Ex.stfp/hdfs deletefilePath: provide file path for deletion of uploaded file.
Response Output(JSON Format)	{"status":1,"response":{"message":"File deleted successfully."}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

DeleteFile-Storage-SFTP

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

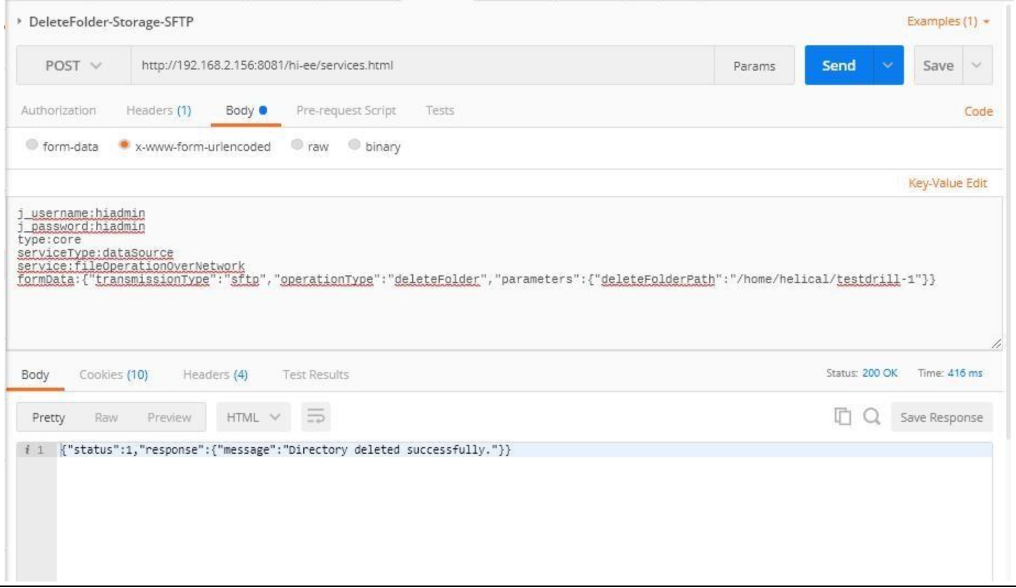
```
j_username:hiadmin
j_password:hiadmin
type:core
serviceType:dataSource
service:fileOperationOverNetwork
formData:{"transmissionType":"sftp","operationType":"deleteFile","parameters":{"deletefilePath":"/home/helical/testdrill-1/Invoices.csv"}}
```

Status: 200 OK Time: 550 ms

```
{"status":1,"response":{"message":"File deleted successfully."}}
```

4.1.9.2 Middleware:: Delete uploaded folder

URL	/services.html	
Description	The user can delete uploaded folder/created which is uploaded with middleware connection as per storage implementation for ex.sftp,hdfs	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=fileOperationOverNetwork&formData={'transmissionType':'sftp','operationType':'deleteFolder','parameters':{'deleteFolderPath':'/home/helical/testdrill-1'}}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	fileOperationOverNetwork	The service is to delete uploaded folder for fileOperationOverNetwork

formData:	{"transmissionType":"sftp","operationType":"deleteFolder","parameters":{"deleteFolderPath":"/home/helical/testdrill-1"}}	transmissionType: user can provided storage imple type for.Ex.stfp/hdfs deleteFolderPath: provide folderpath for deletion of uploaded/created folder.
Response Output(JSON Format)	{"status":1,"response":{"message":"Directory deleted successfully."}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot	 DeleteFolder-Storage-SFTP Examples (1) POST Params Send Save Authorization Headers (1) Body Pre-request Script Tests Code form-data x-www-form-urlencoded raw binary Key-Value Edit Body Cookies (10) Headers (4) Test Results Pretty Raw Preview HTML Save Responsehttp://192.168.2.156:8081/hi-ee/services.html j_username:hiadmin j_password:hiadmin type:core serviceType:dataSource service:fileOperationOverNetwork formData:{"transmissionType":"sftp","operationType":"deleteFolder","parameters":{"deleteFolderPath":"/home/helical/testdrill-1"}} Status: 200 OK Time: 416 ms {"status":1,"response":{"message":"Directory deleted successfully."}}	

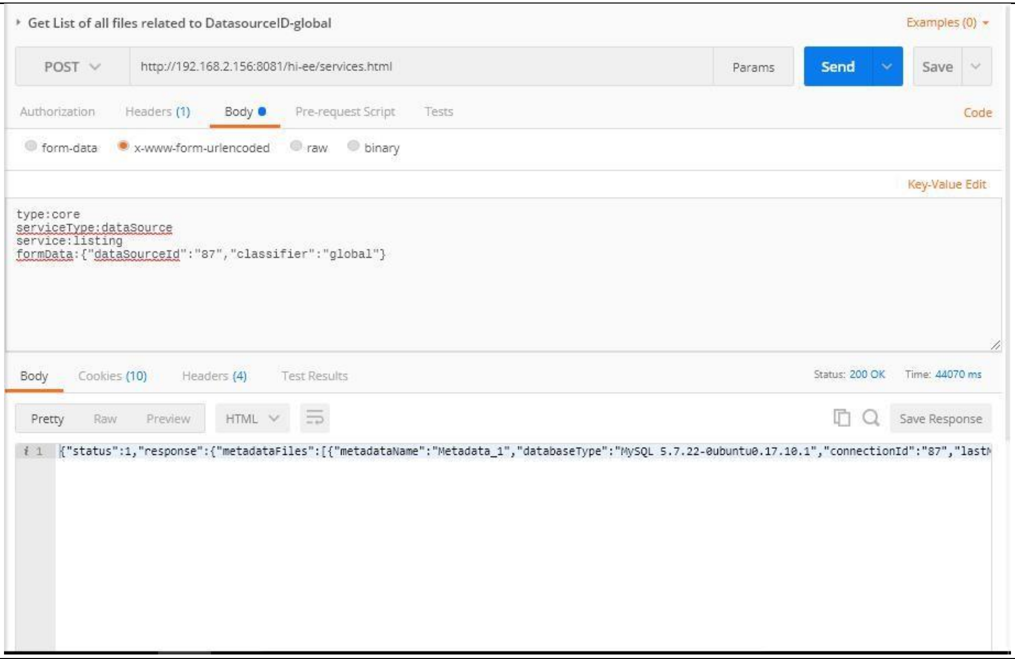
4.1.10 Delete Data source

User can delete data source.There are two types of deletion of data source simple and cascade data source delete.With “*simple*” delete only data source will be deleted and with “*cascade*” delete all dependent resources(reports,metadata etc) along with data source will be deleted.Based on dataSourceId datasource will be deleted which can be global or efwd if the dataSourceId is efwd then need to provide the location also in the fromdata.

4.1.10.1 Get all files related to global dataSourceID(managed datasource)

URL	/services.html
Description	The user will get all files related to global datasourceID(Managed datasource)
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.

Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'dataSourceId':'33','classifier':'global'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	listing	The service is to list the resources related to provided global datasourceID
formData:	{"dataSourceId":"87","classifier":"global"}	dataSourceId: ID of the datasource Classifier : global which is managed datasource
Response Output(JSON Format)	{"status":1,"response":{"metadataFiles":[{"metadataName":"Metadata_1","databaseType":"MySQL 5.7.22-0ubuntu0.17.10.1","connectionId":"87","lastModified":"1543918235000","name":"Metadata_1","path":"1543917308637/4876d715-921d-4539-b06a-b12cb7e252a9.metadata","folderName":"/Testing Input Parameter","reportDetails":[{"reportFileName":"4f98c97d-df9b-4ae8-9724-a649e0468a8f.report","reportName":"report","metadataFileName":"4876d715-921d-4539-b06a-b12cb7e252a9.metadata","location":"/1543917308637/","savedReports":[],"designerReports":[{"designerReportName":"dashboard","reportFileName":"4f98c97d-df9b-4ae8-9724-a649e0468a8f.report","reportDirectory":"1543917308637","efwFileName":"27c714d3-dd48-4f15-a16e-8a16d00a881d.efw"}]}]}]}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Get List of all files related to DatasourceID-global

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

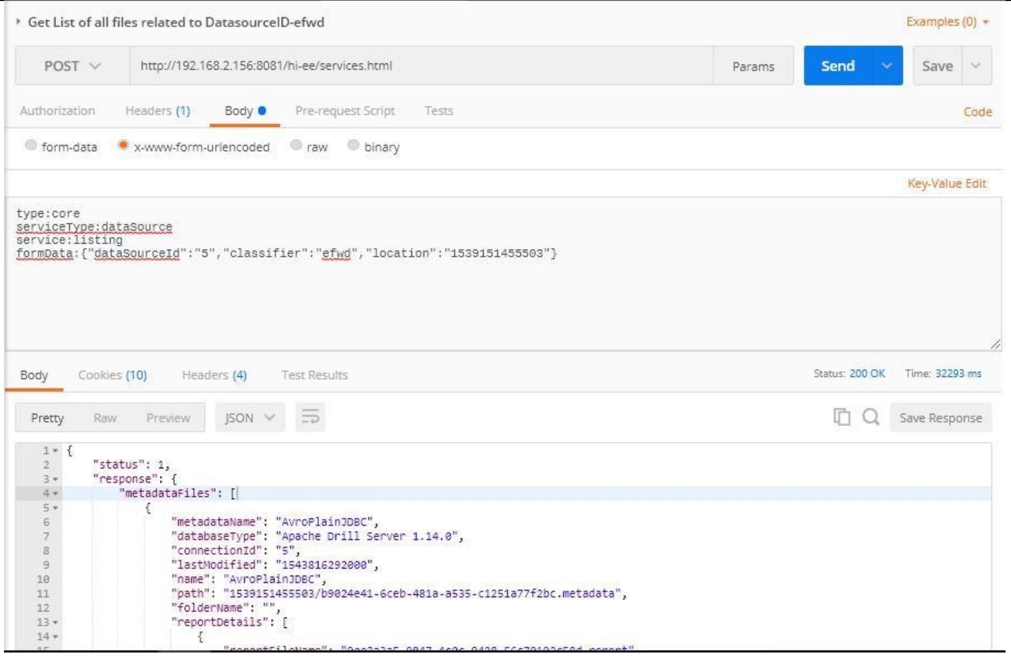
```
type:core
serviceType:dataSource
service:listing
formData:{"dataSourceId":"87","classifier":"global"}
```

Status: 200 OK Time: 44070 ms

```
{"status":1,"response":{"metadataFiles":[{"metadataName":"Metadata_1","databaseType":"MySQL 5.7.22-0ubuntu0.17.10.1","connectionId":"87","lastM
```

4.1.10.2 Get all files related to efwd dataSourceID(Plain JDBC datasource)

URL	/services.html	
Description	The user will get all files related to efwd datasourceID(Plain JDBC datasource)	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'dataSourceId':'5','classifier':'efwd','location':'1539151455503'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	listing	The service is to list the resources

		related to provided global datasourceID
formData:	{"dataSourceId":"5","classifier":"efwd","location":"1539151455503"}	dataSourceId: ID of the datasource Classifier : efwd which is plain JDBC datasource location : location of efwd file
Response Output(JSON Format)	{"status":1,"response":{"metadataFiles":[{"metadataName":"AvroPlainJDBC","databaseType":"Apache Drill Server 1.14.0","connectionId":"5","lastModified":"1543816292000","name":"AvroPlainJDBC","path":"1539151455503/b9024e41-6ceb-481a-a535-c1251a77f2bc.metadata","folderName":"","reportDetails":[{"reportFileName":"9ee2a3a5-9847-4c0c-9428-56c79192c58d.report","reportName":"AvroPlainJDBCReport","metadataFileName":"b9024e41-6ceb-481a-a535-c1251a77f2bc.metadata","location":"/1539151455503/","savedReports":[],"designerReports":[]}]}]}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

Get List of all files related to DatasourceID-efwd

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:core
serviceType:dataSource
service:listing
formData:{"dataSourceId":"5","classifier":"efwd","location":"1539151455503"}
```

Status: 200 OK Time: 32293 ms

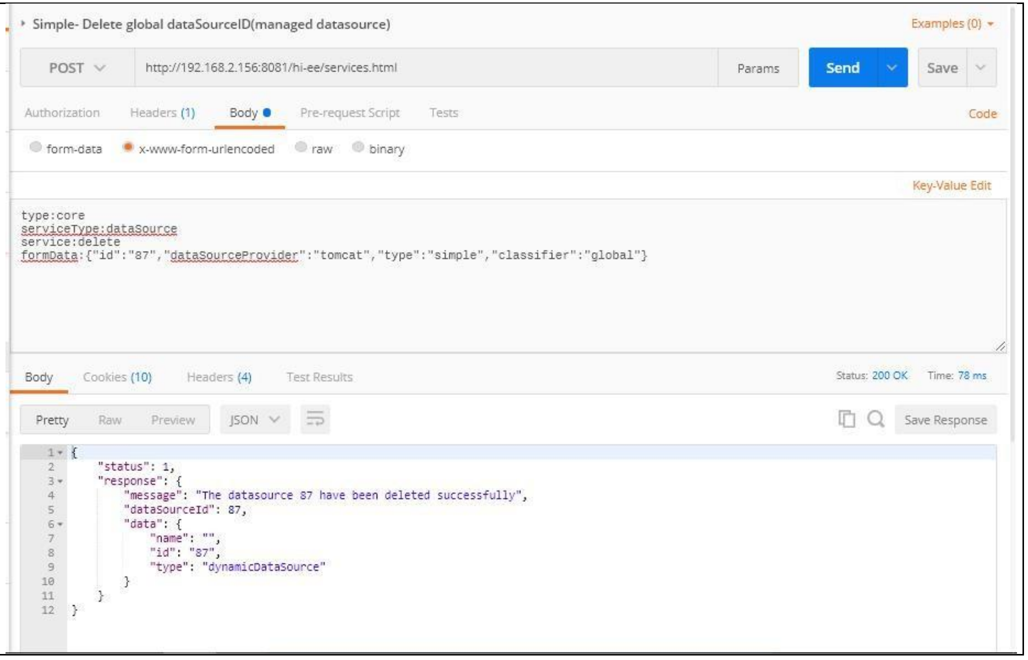
```
{
    "status": 1,
    "response": {
        "metadataFiles": [
            {
                "metadataName": "AvroPlainJDBC",
                "databaseType": "Apache Drill Server 1.14.0",
                "connectionId": "5",
                "lastModified": "1543816292000",
                "name": "AvroPlainJDBC",
                "path": "1539151455503/b9024e41-6ceb-481a-a535-c1251a77f2bc.metadata",
                "folderName": "",
                "reportDetails": [
                    {
```

4.1.10.3 Delete global dataSourceID(managed datasource)

4.1.10.3.1 Simple- Delete global dataSourceID(managed datasource)

URL	/services.html

Description	The user can delete global datasource(managed datasource) with simple type which will delete only datasource.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=delete&formData={'id':'87','dataSourceProvider':'tomcat','type':'simple','classifier':'global'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	delete	The service is to delete datasource with provided global datasourceID
formData:	{"id":"87","dataSourceProvider":"tomcat","type":"simple","classifier":"global"}	dataSourceId: ID of the datasource Classifier : global which is managed datasource
Response Output(JSON Format)	{"status":1,"response":{"message":"The datasource 87 have been deleted successfully","dataSourceId":87,"data":{"name":"","id":"87","type":"dynamicDataSource"}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

Simple- Delete global dataSourceID(managed datasource)

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

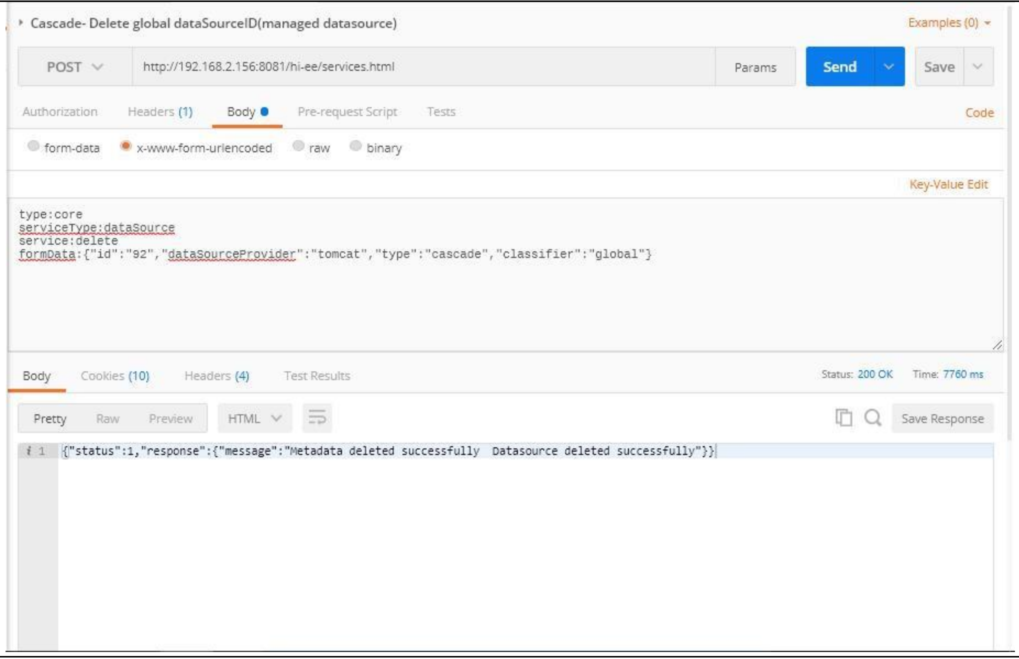
```
type:core
serviceType:dataSource
service:delete
formData:{"id":"87","dataSourceProvider":"tomcat","type":"simple","classifier":"global"}
```

Status: 200 OK Time: 78 ms

```
{
    "status": 1,
    "response": {
        "message": "The datasource 87 have been deleted successfully",
        "dataSourceId": 87,
        "data": {
            "name": "",
            "id": "87",
            "type": "dynamicDataSource"
        }
    }
}
```

4.1.10.3.2 Cascade- Delete global dataSourceID(managed datasource)

URL	/services.html	
Description	The user can delete global datasource(managed datasource) with cascade type which will delete all dependent resources along with datasource.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'dataSourceId':'5','classifier':'efwd','location':'1539151455503'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core

serviceType:	dataSource	serviceType as dataSource
service:	delete	The service is to delete the resources related to provided global datasourceID
formData:	{"id":"92","dataSourceProvider":"tomcat","type":"cascade","classifier":"global"}	dataSourceId: ID of the datasource Classifier : global which is managed datasource
Response Output(JSON Format)	{"id":"92","dataSourceProvider":"tomcat","type":"cascade","classifier":"global"}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8081/hi-ee/services.html

form-data [✓ x-www-form-urlencoded] raw binary

```
type:core
serviceType:dataSource
service:delete
formData:{"id":"92","dataSourceProvider":"tomcat","type":"cascade","classifier":"global"}
```

Status: 200 OK Time: 7760 ms

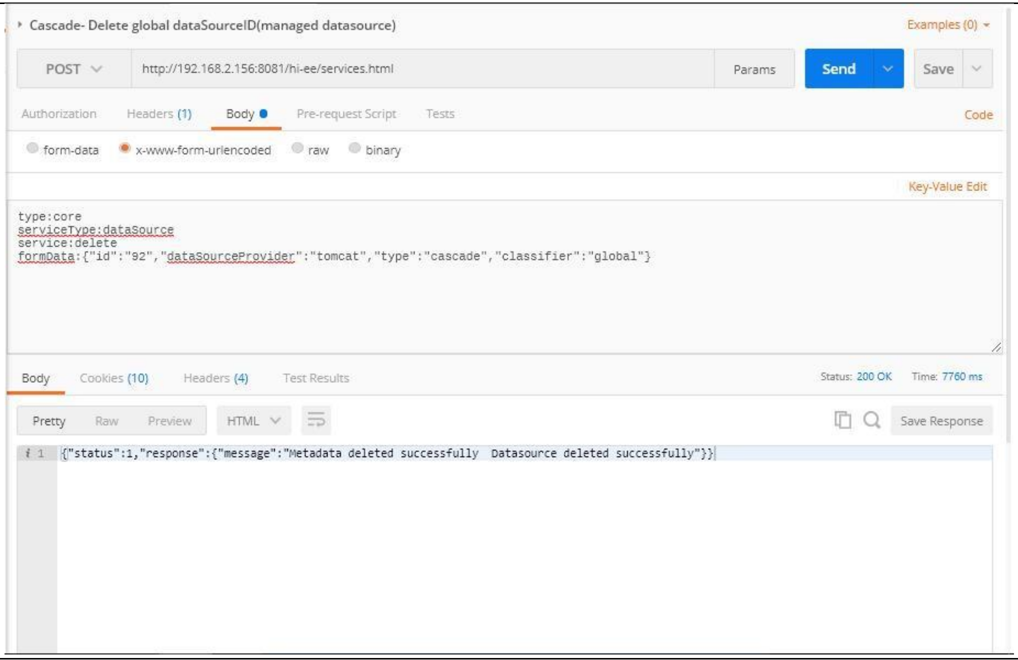
```
{"status":1,"response":{"message":"Metadata deleted successfully  Datasource deleted successfully"}}
```

4.1.10.4 Delete efwd dataSourceID(Plain JDBC datasource)

4.1.10.4.1 Simple- Delete efwd dataSourceID(Plain JDBC datasource)

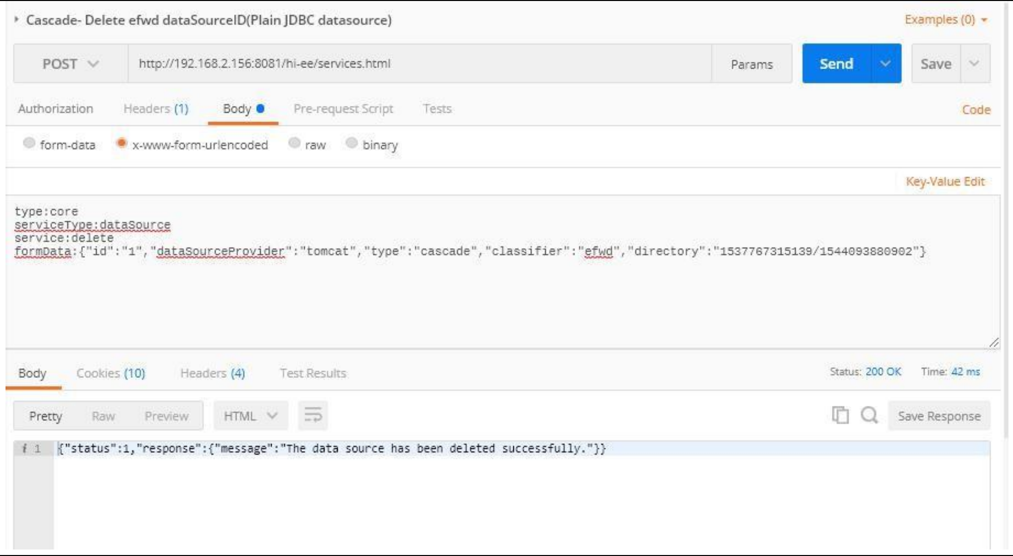
URL	/services.html
Description	The user can delete efwd datasource(plain JDBC datasource) with simple type

	which will delete only datasource.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=delete&formData={'id':'1','dataSourceProvider':'tomcat','type':'simple','classifier':'efwd','directory':'1537766417348/1541492426251/1541492777596'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	delete	The service is to delete the provided efwd datasourceID
formData:	{"id":"1","dataSourceProvider":"tomcat","type":"simple","classifier":"efwd","directory":"1537766417348/1541492426251/1541492777596"}	dataSourceId: ID of the datasource Classifier : efwd which is plain JDBC datasource directory: location of efwd file
Response Output(JSON Format)	{"status":1,"response":{"message":"The data source has been deleted successfully."}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Cascade- Delete global dataSourceID(managed datasource)
	POST http://192.168.2.156:8081/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:core`
	`serviceType:dataSource`
	`service:delete`
	`formData:{"id":"92","dataSourceProvider":"tomcat","type":"cascade","classifier":"global"}`
	Status: 200 OK Time: 7760 ms
	`{"status":1,"response":{"message":"Metadata deleted successfully  Datasource deleted successfully"}}`

4.1.10.4.2 Cascade- Delete efwd dataSourceID(Plain JDBC datasource)

URL	/services.html	
Description	The user can delete efwd datasource(Plain JDBC datasource) with cascade type which will delete all dependent resources along with datasource.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'dataSourceId':'5','classifier':'efwd','location':'1539151455503'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core

serviceType:	dataSource	serviceType as dataSource
service:	delete	The service is to delete the resources related to provided efwd datasourceID
formData:	{"id":"1","dataSourceProvider":"tomcat","type":"cascade","classifier":"efwd","directory":"1537767315139/1544093880902"}	dataSourceId: ID of the datasource Classifier : efwd which is plain JDBC datasource directory: location of efwd file
Response Output(JSON Format)	{"status":1,"response":{"message":"The data source has been deleted successfully."}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

Cascade- Delete efwd dataSourceID(Plain JDBC datasource)

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:core
serviceType:dataSource
service:delete
formData:{"id":"1","dataSourceProvider":"tomcat","type":"cascade","classifier":"efwd","directory":"1537767315139/1544093880902"}
```

Status: 200 OK Time: 42 ms

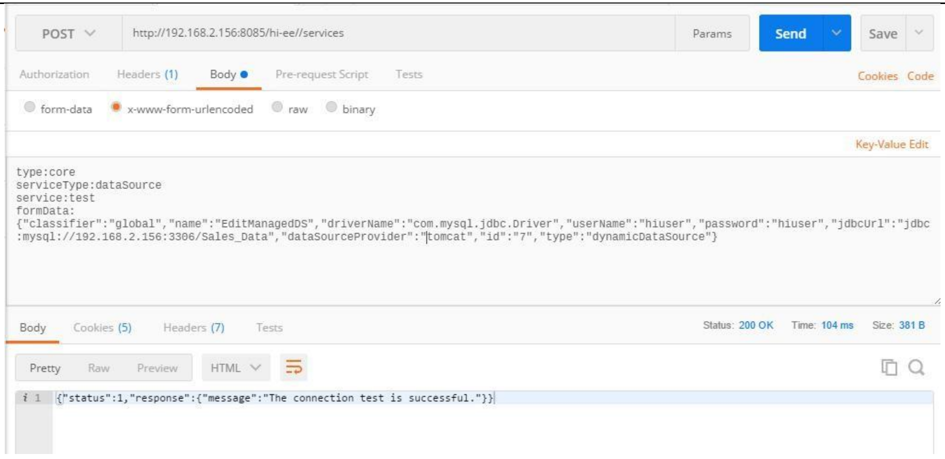
```
{"status":1,"response":{"message":"The data source has been deleted successfully."}}
```

4.2.2 Test Edited Managed Datasource

URL	/services	
Description	It allows user to test the edited connection details while updating the managed datasource connection , so that before saving the datasource we can test the provided connection. Note : If the connection details(host,username,password,dbname) are not correct , then you will get an Exception.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Note : Here we are testing the MySQL db connection using managed datasource , same way you can edit different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=test&formData={'classifier':'global','name':'EditManagedDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/Sales_Data','dataSourceProvider':'tomcat','id':'7','type':'dynamicDataSource'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	test	The service is to test the provided connection details while creating managed datasource connection.
formData:	{"classifier":"global","name":"EditManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:33	name : name of the datasource id: This is the id of the existing datasource which we are going to modify.

	06/Sales_Data","dataSourceProvider":"tomcat","id":"7","type":"dynamicDataSource"}	

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The connection test is successful"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. If all the credential and jdbc url is correct the user will get success message.	
Service Status	200 OK	
Screenshot		

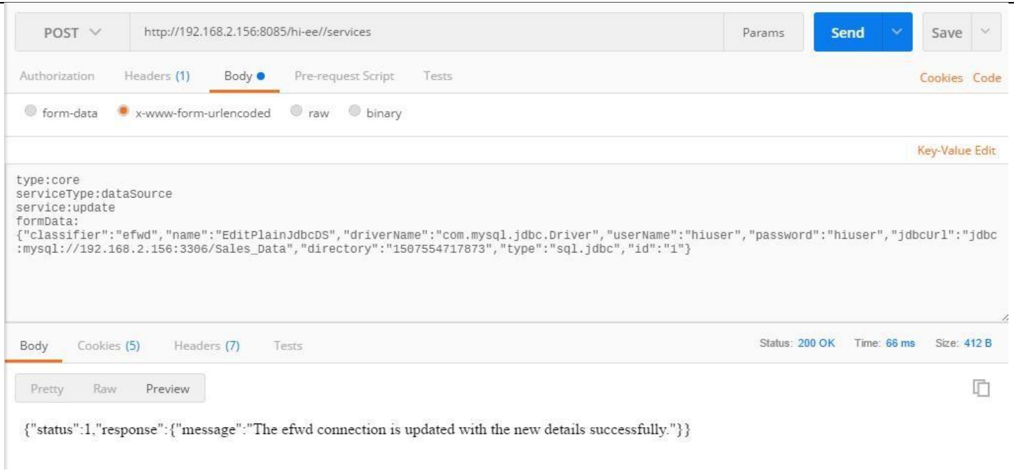
```
POST  http://192.168.2.156:8085/hi-ee//services
Body: x-www-form-urlencoded

type:core
serviceType:dataSource
service:test
formData:
{"classifier":"global","name":"EditManagedDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc
:mysql://192.168.2.156:3306/Sales_Data","dataSourceProvider":"tomcat","id":"7","type":"dynamicDataSource"}

Status: 200 OK   Time: 104 ms   Size: 381 B

{"status":1,"response":{"message":"The connection test is successful."}}
```

4.2.3 Edit Plain JDBC Datasource

URL	/services
Description	The user can update any changes in the plain Jdbc datasources.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Note : Here we are creating the MySQL db connection using plain Jdbc datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :**

	curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=update&formData={'classifier':'efwd','name':'EditPlainJdbcDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/Sales_Data','directory':1507554717873','type':'sql.jdbc','id':'1'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	update	The service is update to modifiy existing datasource connection
formData:	{"classifier":"efwd","name":"EditPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/Sales_Data","directory":"1507554717873","type":"sql.jdbc","id":"1"}	name : name of the datasource id: This is the id of the existing datasource which we are going to modify.

Response Output(JSON Format)	{
	"status":1,"response":{"message":"The efwd connection is updated with the new details successfully."}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. The respective datasource is modified and saved in the backend.The updated details get saved in .efwd file at stored location.	
Service Status	200 OK	
Screenshot		

```
type:core
serviceType:dataSource
service:update
formData:
{"classifier":"efwd","name":"EditPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc
:mysql://192.168.2.156:3306/Sales_Data","directory":"1507554717873","type":"sql.jdbc","id":"1"}
```

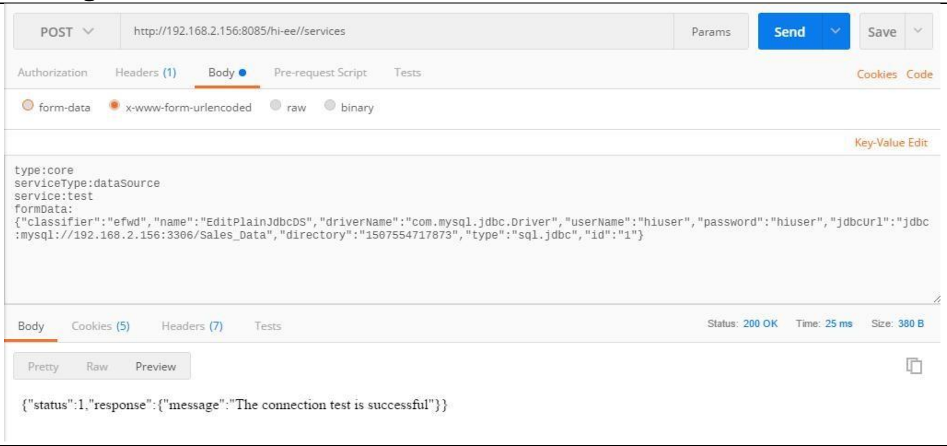
```
{"status":1,"response":{"message":"The efwd connection is updated with the new details successfully."}}
```

4.2.4 Test Edited Plain JDBC Datasource

URL	/services	
Description	It allows user to test the edited connection details while updating the plain Jdbc datasource connection , so that before saving the datasource we can test the provided connection. Note : If the connection details(host,username,password,dbname) are not correct , then you will get an Exception.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Note : Here we are creating the MySQL db connection using plain Jdbc datasource , same way you can create different supported datasource connections.** **Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=test&formData={'classifier':'efwd','name':'EditPlainJdbcDS','driverName':'com.mysql.jdbc.Driver','userName':'hiuser','password':'hiuser','jdbcUrl':'jdbc:mysql://192.168.2.156:3306/Sales_Data','directory':1507554717873','type':'sql.jdbc','id':'1'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	test	The service is to test the provided connection details while creating plain Jdbc datasource connection.
formData:	{"classifier":"efwd","name":"EditPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysq	name : name of the datasource id: This is the id of the existing datasource which we are going to modify.

	l://192.168.2.156:3306/Sales_Data","directory":"1507554717873","type":"sql.jdbc","id":"1"}	

Response Output(JSON Format)	{
	"status":1,
	"response":{"message":"The connection test is successful"}
	}	

Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message. If all the credential and jdbc url is correct the user will get success message.	
Screenshot	 POST http://192.168.2.156:8085/hi-ee//services x-www-form-urlencoded type:core serviceType:dataSource service:test formData: {"classifier":"efwd","name":"EditPlainJdbcDS","driverName":"com.mysql.jdbc.Driver","userName":"hiuser","password":"hiuser","jdbcUrl":"jdbc:mysql://192.168.2.156:3306/Sales_Data","directory":"1507554717873","type":"sql.jdbc","id":"1"} Status: 200 OK Time: 25 ms Size: 380 B {"status":1,"response":{"message":"The connection test is successful"}}	

4.2.5 Datasource Cache Management

Datasource Cache Management is the new implementation which is implemented to cache the datasource connection which helps to increase the metadata performance.

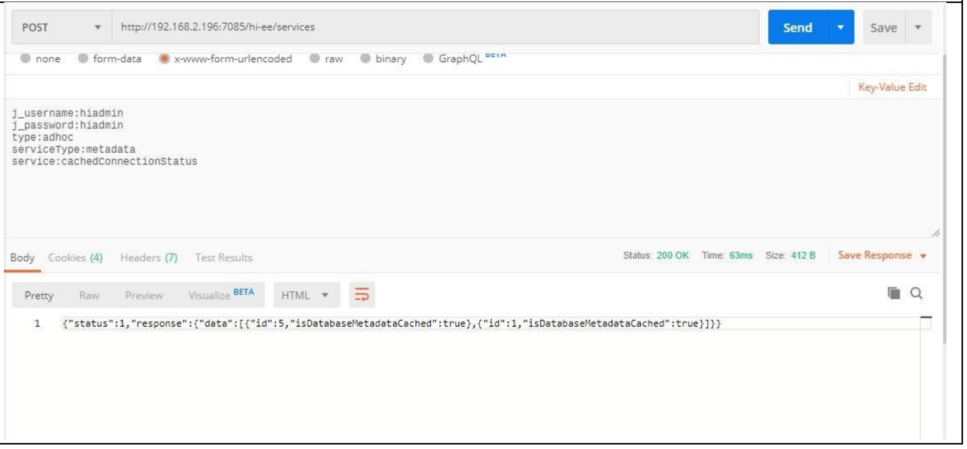
Below are the service API's to get more info about datasource cache .

4.2.5.1 Datasource Cache Status

URL	/services
Description	With this service API user will get datasource connection status.
Pre-requisite	User should have logged in before accessing the service.[Refer login

	module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=cachedConnectionStatus' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as core
serviceType:	metadata	serviceType as dataSource
service:	cachedConnectionStatus	The service is get the cached connection status

Response Output(JSON Format)	{"status":1,"response":{"data":[{"id":5,"isDatabaseMetadataCached":true},{"id":1,"isDatabaseMetadataCached":true}]}
	}	

Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message with datasource ConnectionID.	
Screenshot		

POST http://192.168.2.196:7085/hi-ee/services

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA

```
j_username:hiadmin
j_password:hiadmin
type:adhoc
serviceType:metadata
service:cachedConnectionStatus
```

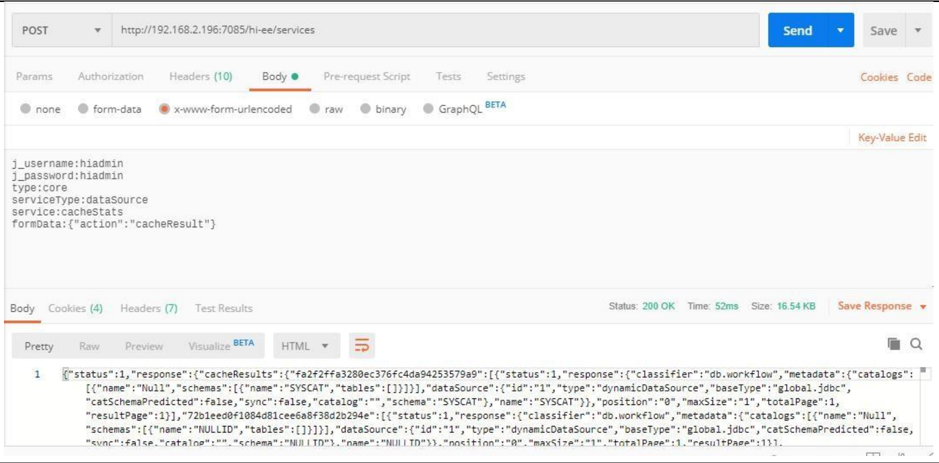
Status: 200 OK Time: 63ms Size: 412 B

```
{"status":1,"response":{"data":[{"id":5,"isDatabaseMetadataCached":true},{"id":1,"isDatabaseMetadataCached":true}]}}
```

4.2.5.2 Datasource Cache Result

URL	/services

Description	It allows user to get the cache result of cached datasource.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=cacheStats&formData={"action":"cacheResult"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	cacheStats	The service is to get the cache stats or info
formData:	{"action":"cacheResult"}	Action is to get the cache result.
Response Output(JSON Format)	{"status":1,"response":{"cacheResults":{"fa2f2ffa3280ec376fc4da94253579a9":[{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null","schemas":[{"name":"SYSCAT","tables":[]}]}],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,"sync":false,"catalog":"","schema":"SYSCAT"},"name":"SYSCAT"}},"position":"0","maxSize":"1","totalPage":1,"resultPage":1}]}}}	
Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message along with cache result and related information	

Screenshot	

POST http://192.168.2.196:7085/hi-ee/services

Body: x-www-form-urlencoded

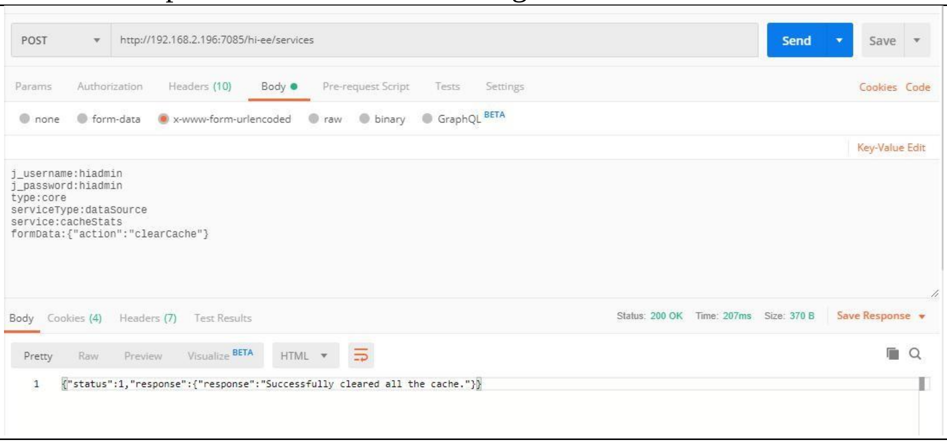
```
j_username:hiadmin
j_password:hiadmin
type:core
serviceType:dataSource
service:cacheStats
formData:{"action":"cacheResult"}
```

Status: 200 OK Time: 52ms Size: 16.54 KB

```
{"status":1,"response":{"cacheResults":{"fa2f2ffa3280ec376fc4da94253579a9":[{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":
    [{"name":"Null","schemas":[{"name":"SYSCAT","tables":[]}]}],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc",
    "catSchemaPredicted":false,"sync":false,"catalog":"","schema":"SYSCAT"},"name":"SYSCAT"}},"position":"0","maxSize":"1","totalPage":1,
    "resultPage":1}],"72b1eed0f1084d81cee6a8f38d2b294e":[{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null",
    "schemas":[{"name":"NULLID","tables":[]}]}],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,
    "sync":false,"catalog":"","schema":"NULLID"},"name":"NULLID"}},"position":"0","maxSize":"1","totalPage":1,"resultPage":1}],
```

4.2.5.3 Datasource Clear Cache

URL	/services	
Description	It allows user to clear the cache datasources from memory	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=cacheStats&formData={"action":"clearCache"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core

serviceType:	dataSource	serviceType as dataSource
service:	cacheStats	The service is to get the cache stats or info
formData:	{"action":"clearCache"}	Action is to clear the cache.
Response Output(JSON Format)	{"status":1,"response":{"response":"Successfully cleared all the cache."}}	
Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message	
Screenshot		

POST http://192.168.2.196:7085/hi-ee/services

Body: x-www-form-urlencoded

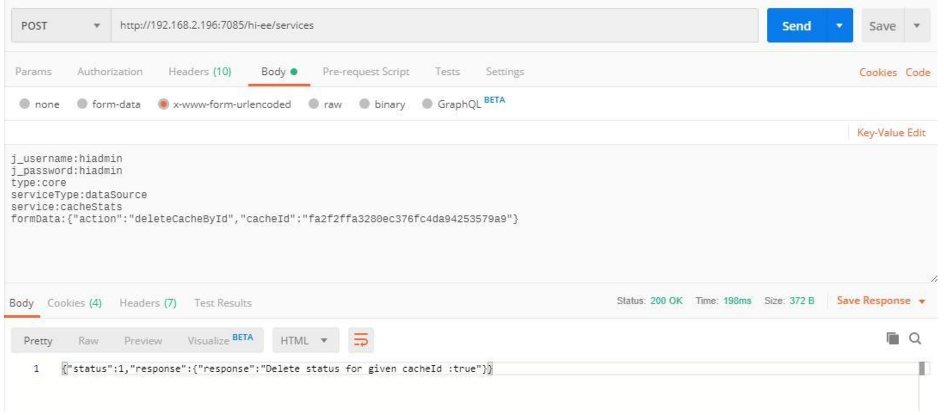
```
j_username:hiadmin
j_password:hiadmin
type:core
serviceType:dataSource
service:cacheStats
formData:{"action":"clearCache"}
```

Status: 200 OK Time: 207ms Size: 370 B

```
{"status":1,"response":{"response":"Successfully cleared all the cache."}}
```

4.2.5.3 Datasource Cache Delete By CacheId

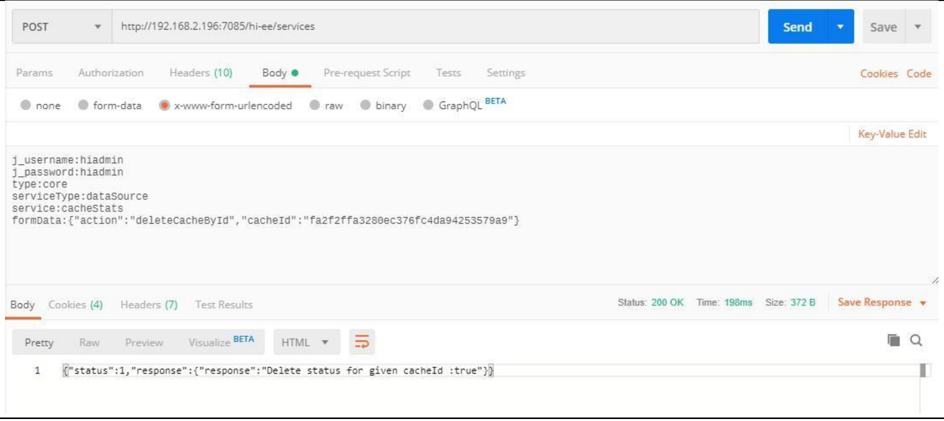
URL	/services
Description	It allows user to delete the cache datasources from memory.For that user need to provide the cacheID (it is a cache key which is stored in (CACHE_DATASOURCE) table of hiee database.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services

	Access through Curl command : curl --data 'j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=cacheStats&formData={"action":"deleteCacheById","cacheId":"fa2f2ffa3280ec376fc4da94253579a9"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	cacheStats	The service is to get the cache stats or info
formData:	{"action":"deleteCacheById","cacheId":"fa2f2ffa3280ec376fc4da94253579a9"}	Action is to delete the cache by id which is the cachekey from cache_datasource table.
Response Output(JSON Format)	{"status":1,"response":{"response":"Delete status for given cacheId :true"}}	
Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message	
Screenshot	 POST http://192.168.2.196:7085/hi-ee/services Body: x-www-form-urlencoded j_username:hiadmin j_password:hiadmin type:core serviceType:dataSource service:cacheStats formData:{"action":"deleteCacheById","cacheId":"fa2f2ffa3280ec376fc4da94253579a9"} Status: 200 OK Time: 198ms Size: 372 B {"status":1,"response":{"response":"Delete status for given cacheId :true"}}	

4.2.5.3 Datasource Cache Read By CacheId

URL	/services
Description	It allows user to read tthe cache datasources from memory.For that user need to provide the cacheID (it is a cache key which is stored in (CACHE_DATASOURCE) table of hiee database.
Pre-requisite	User should have logged in before accessing the service.[Refer login

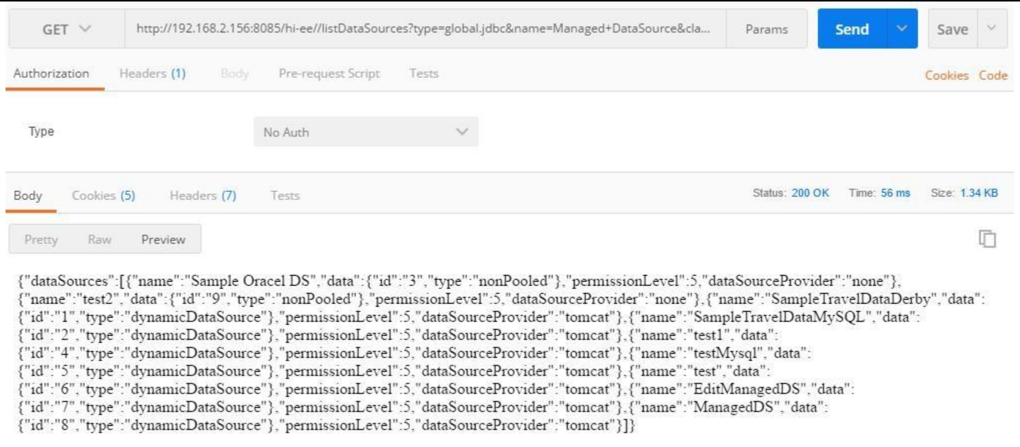
	module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=cacheStats&formData={"action":"readCacheById","cacheId":"ea78540cc974c373834fbf4e1a2b97ac"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	cacheStats	The service is to get the cache stats or info
formData:	{"action":"readCacheById","cacheId":"ea78540cc974c373834fbf4e1a2b97ac"}	Action is to read the cache by id which is the cachekey from cache_datasource table.
Response Output(JSON Format)	{"status":1,"response":{"cacheResults":[{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null","schemas":[{"name":"SQLJ","tables":[]}]}],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,"sync":false,"catalog":"","schema":"SQLJ"},"name":"SQLJ"}},"position":"0","maxSize":"1","totalPage":1,"resultPage":1}]}}	
Service Status	200 OK	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message along with cached datasource details	

Screenshot	 POST http://192.168.2.196:7085/hi-ee/services
	Body: x-www-form-urlencoded
	`j_username:hiadmin`
	`j_password:hiadmin`
	`type:core`
	`serviceType:dataSource`
	`service:cacheStats`
	`formData:{"action":"deleteCacheById","cacheId":"fa2f2ffa3280ec376fc4da94253579a9"}`
	Status: 200 OK Time: 198ms Size: 372 B
	`{"status":1,"response":{"response":"Delete status for given cacheId :true"}}`

4.3 List Datasource

URL	/listDataSources
Description	It shows the list of the existing datasources according to requested type and classifier described in HTTP Request Key-Value section.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST,GET**
Example	**Note : Access through browser is the GET method for listing out the datasource.We can list all type of data which is already created. Under response section we are taking the example of managed Datasource.** **Access through browser :** List of datasources for managed datasource: http://192.168.2.156:8085/hi-ee//listDataSources?type=global.jdbc&name=Managed+DataSource&classifier=global

	List of datasources for Plain JDBC datasource: http://192.168.2.156:8085/hi-ee//listDataSources?type=sql.jdbc&name=Plain+Jdbc+DataSource&classifier=efwd **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=global.jdbc&name=Managed+DataSource&classifier=global" http://192.168.2.156:8085/hi-ee//listDataSources -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	global.jdbc	global.jdbc/ sql.jdbc
name:	Managed+DataSource	Managed+DataSource / Plain+Jdbc+DataSource
classifier:	global	Values can be global/efwd. • If the classifier is global then details are stored in globalConnections.xml • If the classifier is efwd then details are stored in respective efwd file.
Response Output(JSON Format)	{"dataSources":[{"name":"Sample Oracel DS","data":{"id":"3","type":"nonPooled"},"permissionLevel":5,"dataSourceProvider":"none"},{"name":"test2","data":{"id":"9","type":"nonPooled"},"permissionLevel":5,"dataSourceProvider":"none"},{"name":"SampleTravelDataDerby","data":{"id":"1","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"SampleTravelDataMySQL","data":{"id":"2","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"test1","data":{"id":"4","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"testMysql","data":{"id":"5","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"test","data":{"id":"6","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"EditManagedDS","data":{"id":"7","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"ManagedDS","data":{"id":"8","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"}]}	
Description of Response Output:	The response we get as the datsource array with name of the datasource, if of datasource with type of datasource. Depending upon the **type parameter** and **classifier parameter** the different lists are generated , following are the different parameters : **name** : Name of datasource **data** : Array having dataid and type of connecion	

	permissionLevel : permissionLevel of datasource **dataSourceProvider** : Datasource provider name etc
Service Status	200 OK
Screenshot	

```
{"dataSources":[{"name":"Sample Oracel DS","data":{"id":"3","type":"nonPooled"},"permissionLevel":5,"dataSourceProvider":"none"},
{"name":"test2","data":{"id":"9","type":"nonPooled"},"permissionLevel":5,"dataSourceProvider":"none"},{"name":"SampleTravelDataDerby","data":
{"id":"1","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"SampleTravelDataMySQL","data":
{"id":"2","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"test1","data":
{"id":"4","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"testMysql","data":
{"id":"5","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"test","data":
{"id":"6","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"EditManagedDS","data":
{"id":"7","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"},{"name":"ManagedDS","data":
{"id":"8","type":"dynamicDataSource"},"permissionLevel":5,"dataSourceProvider":"tomcat"}]}
```

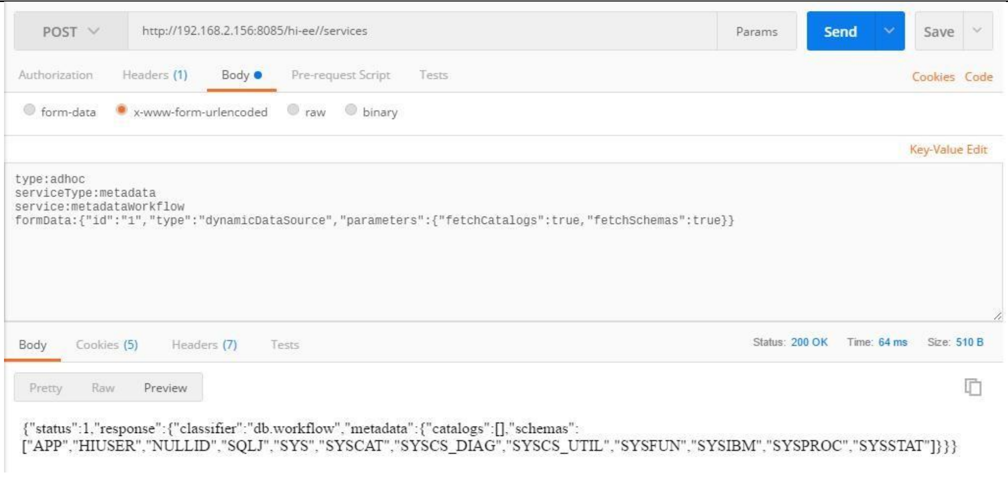
4.4 Metadata Operations

4.4.1 Metadata Create

4.4.1.1 Retrieve catalogs and schemas :

URL	/services
Description	It fetch the list of available catalogs and schemas for selected datasource. When you select datasource for creation of metadata it will gives you the list of catalogs and schemas.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :**

	curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=metadataWorkflow&formData={'id':'1','type':'dynamicDataSource','parameters':{'fetchCatalogs':true,'fetchSchemas':true}}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	metadataWorkflow	Service as metadataWorkflow to create metadata.
formData:	{ "id":"1","type":"dynamicDataSource","parameters":{"fetchCatalogs":true,"fetchSchemas":true} }	formData contains the id which is the datasource connection ID . Type is the type of datasource, parameters include the fetchCatalogs and fetchSchemas true/false values.
Response Output(JSON Format)	{ "status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[],"schemas":["APP","HIUSER","NULLID","SQLJ","SYS","SYSCAT","SYSCS_DIAG","SYSCS_UTIL","SYSFUN","SYSIBM","SYSPROC","SYSSTAT"]}} }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. The metadata with list of catalogs and schemas get return as per selected datasource. **classifier** : name of classifier **metadata** : metadata array **catalogs** : list of catalogs **schemas** : Schema array with schema name nad related information.	
Service Status	200 OK	

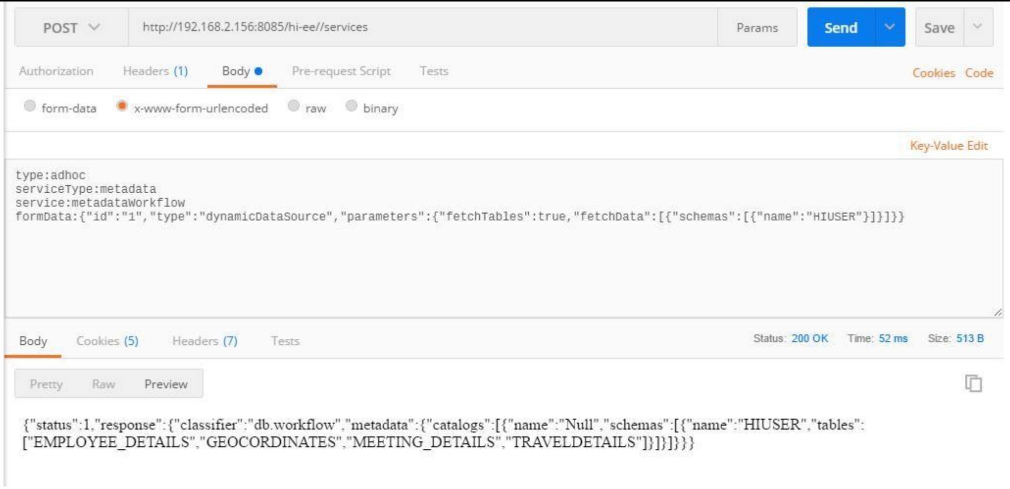
Screenshot	 POST http://192.168.2.156:8085/hi-ee//services Body: x-www-form-urlencoded type:adhoc serviceType:metadata service:metadataWorkflow formData:{"id":"1","type":"dynamicDataSource","parameters":{"fetchCatalogs":true,"fetchSchemas":true}} Status: 200 OK Time: 64 ms Size: 510 B {"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[],"schemas":["APP","HIUSER","NULLID","SQLJ","SYS","SYSCAT","SYSCS_DIAG","SYSCS_UTIL","SYSFUN","SYSIBM","SYSPROC","SYSSTAT"]}}}

4.4.1.2 Retrieve tables of selected schema:

URL	/services
Description	It fetch the list of available tables of selected schema. When you select schema for creation of metadata it will give you the list of tables present in the schema.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=metadataWorkflow&formData={'id':'1','type':'dynamicDataSource','parameters':{'fetchTables':true,'fetchData':[{'schemas':[{'name':'HIUSER'}]}]}}" http://192.168.2.156:8085/hi-ee//services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	metadataWorkflow	Service as metadataWorkflow to create metadata.
formData:	{"id":"1","type":"dynamicDataSource","parameters":{"fetchTables":true,"fetchData":[{"schemas":[{"name":"HIUSER"}]}]}}	formData contains the id which is the datasource connection ID . Type is the type of datasource, parameters include the fetchtables as true and fetchData array contains the name of the schema which we are going to use for metadata create.

Response Output(JSON Format)	{
	"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[],"schemas":["APP","HIUSER","NULLID","SQLJ","SYS","SYSCAT","SYSCS_DIAG","SYSCS_UTIL","SYSFUN","SYSIBM","SYSPROC","SYSSTAT"]}}
	}	

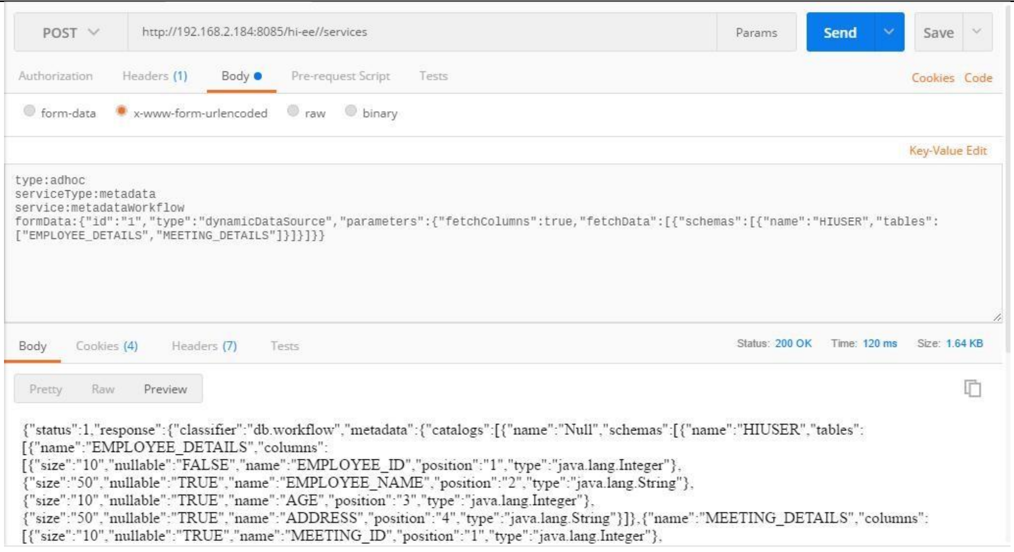
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. **classifier** : name of classifier **metadata** : metadata array **catalogs** : list of catalogs **schemas** : Schema array with schema name nad related information. The metadata array with list of tables get return as per selected schema.	
Service Status	200 OK	
Screenshot		

```
type:adhoc
serviceType:metadata
service:metadataWorkflow
formData:{"id":"1","type":"dynamicDataSource","parameters":{"fetchTables":true,"fetchData":[{"schemas":[{"name":"HIUSER"}]}]}}
```

```
{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null","schemas":[{"name":"HIUSER","tables":
["EMPLOYEE_DETAILS","GEOCORDINATES","MEETING_DETAILS","TRAVELDETAILS"]}]}]}}}
```

4.4.1.3 Retrieve columns of selected tables:

URL	/services	
Description	It fetch the list of associated columns of selected tables. When you select tables for creation of metadata it will give you the list of columns present in the table.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=metadataWorkflow&formData={'id':'1','type':'dynamicDataSource','parameters':{'fetchColumns':true,'fetchData':[{'schemas':[{'name':'HIUSER','tables':['EMPLOYEE_DETAILS','MEETING_DETAILS']}]}]}}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	metadataWorkflow	Service as metadataWorkflow to create metadata.
formData:	{"id":"1","type":"dynamicDataSource","parameters":{"fetchColumns":true,"fetchData":[{"schemas":[{"name":"HIUSER","tables":["EMPLOYEE_DETAILS","MEETING_DETAILS"]}]}]}}	formData contains the id which is the datasource connection ID . Type is the type of datasource, parameters include the fetchColumns as true and fetchData array contains the name of the schema with selected tables which we are going to use for metadata create.
Response Output(JSON Format)	{"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null","schemas":[{"name":"HIUSER","tables":[{"name":"EMPLOYEE_DETAILS","columns":[{"size":"10","nullable":"FALSE","name":"EMPLOYEE_ID","position":"1","type":"java.lang.Integer"},{"size":"50","nullable":"TRUE","name":"EMPLOYEE_NAME","position":"2","type":"java.lang.String"},{"size":"10","nullab	

	le":"TRUE","name":"AGE","position":"3","type":"java.lang.Integer"},{"size":"50","nullable":"TRUE","name":"ADDRESS","position":"4","type":"java.lang.String"}]},{"name":"MEETING_DETAILS","columns":[{"size":"10","nullable":"TRUE","name":"MEETING_ID","position":"1","type":"java.lang.Integer"},{"size":"29","nullable":"TRUE","name":"MEETING_DATE","position":"2","type":"java.sql.Timestamp"},{"size":"10","nullable":"TRUE","name":"MEETING_BY","position":"3","type":"java.lang.Integer"},{"size":"50","nullable":"TRUE","name":"CLIENT_NAME","position":"4","type":"java.lang.String"},{"size":"50","nullable":"TRUE","name":"MEETING_PURPOSE","position":"5","type":"java.lang.String"},{"size":"50","nullable":"TRUE","name":"MEETING_IMPACT","position":"6","type":"java.lang.String"},{"size":"50","nullable":"TRUE","name":"MEET_CANCELLATION_STATUS","position":"7","type":"java.lang.String"},{"size":"50","nullable":"TRUE","name":"CANCELLATION_REASON","position":"8","type":"java.lang.String"}]}]}]}]}}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. The metadata array with list of selected tables with its columns details get return. Schema is the DB name under this name is the table name with all associated columns details like : size of column , name of column , nullable status of column,position of column , type of column(java class)
Service Status	200 OK
Screenshot	 POST http://192.168.2.184:8085/hi-ee//services Body: x-www-form-urlencoded type:adhoc serviceType:metadata service:metadataWorkflow formData:{"id":"1","type":"dynamicDataSource","parameters":{"fetchColumns":true,"fetchData":[{"schemas":[{"name":"HIUSER","tables":["EMPLOYEE_DETAILS","MEETING_DETAILS"]}]}]}} Status: 200 OK Time: 120 ms Size: 1.64 KB {"status":1,"response":{"classifier":"db.workflow","metadata":{"catalogs":[{"name":"Null","schemas":[{"name":"HIUSER","tables":[{"name":"EMPLOYEE_DETAILS","columns":[{"size":"10","nullable":"FALSE","name":"EMPLOYEE_ID","position":"1","type":"java.lang.Integer"},{"size":"50","nullable":"TRUE","name":"EMPLOYEE_NAME","position":"2","type":"java.lang.String"},{"size":"10","nullable":"TRUE","name":"AGE","position":"3","type":"java.lang.Integer"},{"size":"50","nullable":"TRUE","name":"ADDRESS","position":"4","type":"java.lang.String"}]},{"name":"MEETING_DETAILS","columns":[{"size":"10","nullable":"TRUE","name":"MEETING_ID","position":"1","type":"java.lang.Integer"},

4.4.1.4 Save WorkFlow:

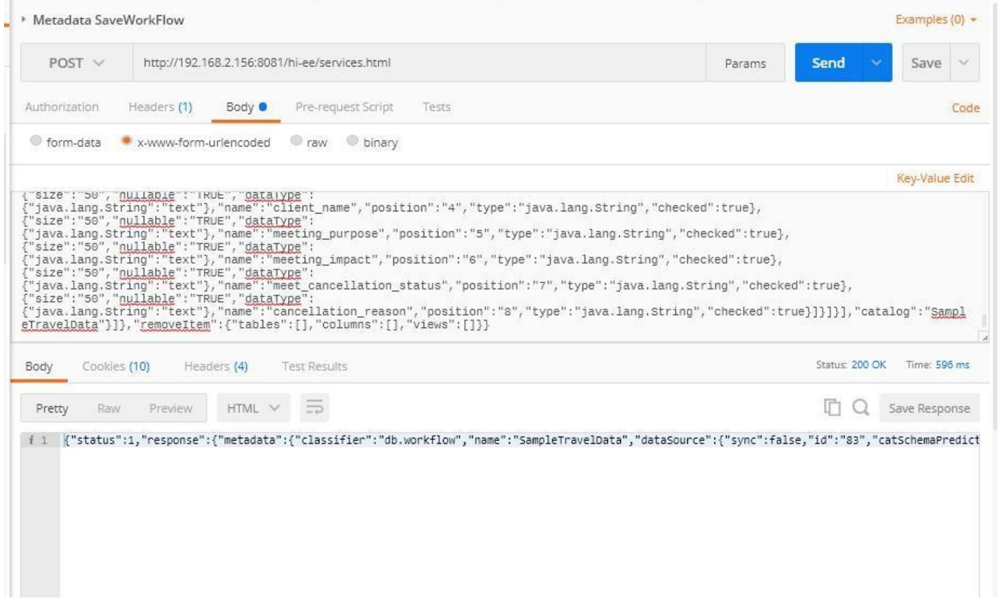
URL	/services
Description	It assigns the unique ID for metadata and all selected schema with its details get return.
Pre-requisite	User should be login before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN

HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=saveWorkflow&formData={'id':'1','type':'dynamicDataSource','metadata':{'catalogs':[{'schemas':[{'name':'HIUSER','tables':[{'name':'EMPLOYEE_DETAILS','columns':[{'size':'10','nullable':'FALSE','name':'EMPLOYEE_ID','position':'1','type':'java.lang.Integer','checked':true},{'size':'50','nullable':'TRUE','name':'EMPLOYEE_NAME','position':'2','type':'java.lang.String','checked':true},{'size':'10','nullable':'TRUE','name':'AGE','position':'3','type':'java.lang.Integer','checked':true},{'size':'50','nullable':'TRUE','name':'ADDRESS','position':'4','type':'java.lang.String','checked':true}]},{'name':'MEETING_DETAILS','columns':[{'size':'10','nullable':'TRUE','name':'MEETING_ID','position':'1','type':'java.lang.Integer','checked':true},{'size':'29','nullable':'TRUE','name':'MEETING_DATE','position':'2','type':'java.sql.Timestamp','checked':true},{'size':'10','nullable':'TRUE','name':'MEETING_BY','position':'3','type':'java.lang.Integer','checked':true},{'size':'50','nullable':'TRUE','name':'CLIENT_NAME','position':'4','type':'java.lang.String','checked':true},{'size':'50','nullable':'TRUE','name':'MEETING_PURPOSE','position':'5','type':'java.lang.String','checked':true},{'size':'50','nullable':'TRUE','name':'MEETING_IMPACT','position':'6','type':'java.lang.String','checked':true},{'size':'50','nullable':'TRUE','name':'MEET_CANCELLATION_STATUS','position':'7','type':'java.lang.String','checked':true},{'size':'50','nullable':'TRUE','name':'CANCELLATION_REASON','position':'8','type':'java.lang.String','checked':true}]}]}]}]}}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	saveWorkflow	Service as saveWorkflow
formData:	{"id":"83","type":"dynamicDataSource","metadata":{"catalogs":[{"schemas":[{"tables":[{"name":"TravelDetails","checked":true,"columns":[{"size":"10","nullable":"FALSE","dataType":{"java.lang.Integer":"numeric"},"name":"travel_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"19","nullable":"TRUE","dataType":{"java.sql.Timestamp":"dateTime"},"name":"travel_date","position":"2","type":"java.sql.Timestamp","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"travel_type","position":"3","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"travel_medium","position":"4","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"source_id","position":"5","type":"java.lang.Integer","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"source_id_error","position":"6","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"source","position":"7","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.I	formData contains the id which is the datasource connection ID . Type is the type of datasource, metdata information includes schema name and tables with associated columns which we are going to use for metadata create.

	nteger":"numeric"},"name":"destination_id","position":"8","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"destination","position":"9","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"travel_cost","position":"10","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"mode_of_payment","position":"11","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"booking_platform","position":"12","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"travelled_by","position":"13","type":"java.lang.Integer","checked":true}]},{"name":"employee_details","checked":true,"columns":[{"size":"10","nullable":"FALSE","dataType":{"java.lang.Integer":"numeric"},"name":"employee_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"employee_name","position":"2","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"age","position":"3","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"address","position":"4","type":"java.lang.String","checked":true}]},{"name":"employee_details_1","checked":true,"columns":[{"size":"10","nullable":"FALSE","dataType":{"java.lang.Integer":"numeric"},"name":"employee_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"employee_name","position":"2","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"age","position":"3","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"address","position":"4","type":"java.lang.String","checked":true}]},{"name":"employee_details_2","checked":true,"columns":[{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"employee_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"employee_name","position":"2","type":"java.lang.String","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"age","position":"3","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"address","position":"4","type":"java.lang.String","checked":true}]},{"name":"geoCordinates","checked":true,"columns":[{"size":"10","nullable":"FALSE","dataType":{"java.lang.Integer":"numeric"},"name":"location_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"location","position":"2","type":"java.lang.String","checked":true},{"size":"22","nullable":"TRUE","dataType":{"java.lang.Double":"numeric"},"name":"latitude","position":"3","type":"java.lang.Double","checked":true},{"size":"22","nullable":"TRUE","dataType":{"java.lang.Double":"numeric"},"name":"longitude","position":"4","type":"java.lang.Double","checked":true}]},{"name":"meeting_details","checked":true,"columns":[{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"meeting_id","position":"1","type":"java.lang.Integer","checked":true},{"size":"19","nullable":"TRUE","dataType":{"java.sql.Timestamp":"dateTime"},"name":"meeting_date","position":"2","type":"java.sql.Timestamp","checked":true},{"size":"10","nullable":"TRUE","dataType":{"java.lang.Integer":"numeric"},"name":"meeting_by","position":"3","type":"java.lang.Integer","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"client_name","position":"4","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"meeting_purpose","position":"5","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"meeting_i	

	mpact","position":"6","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"meet_cancellation_status","position":"7","type":"java.lang.String","checked":true},{"size":"50","nullable":"TRUE","dataType":{"java.lang.String":"text"},"name":"cancellation_reason","position":"8","type":"java.lang.String","checked":true}]}]}],"catalog":"SampleTravelData"}]},"removeItem":{"tables":[],"columns":[],"views":[]}}	
Response Output(JSON Format)	{"status":1,"response":{"metadata":{"classifier":"db.workflow","name":"SampleTravelData","dataSource":{"sync":false,"id":"83","catSchemaPredicted":false,"catalog":"SampleTravelData","schema":"","type":"dynamicDataSource","baseType":"global.jdbc"},"tables":{"TravelDetails":{"id":"19626c57-3bf7-4aac-9839-9dae4886e987","alias":"TravelDetails","columns":{"travel_id":{"alias":"travel_id","fullyQualifiedColumn":"TravelDetails.travel_id","columnId":"a6b1920f-0e96-42d1-bd75-351020dc63ee","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"travel_date":{"alias":"travel_date","fullyQualifiedColumn":"TravelDetails.travel_date","columnId":"9b4121bc-aa0f-4a5d-bfed-e9f8760ec2ec","defaultFunction":"db.generic.groupBy.group","type":{"java.sql.Timestamp":"dateTime"}},"travel_type":{"alias":"travel_type","fullyQualifiedColumn":"TravelDetails.travel_type","columnId":"344a3dba-63b1-4df8-a218-034e5964b258","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"travel_medium":{"alias":"travel_medium","fullyQualifiedColumn":"TravelDetails.travel_medium","columnId":"6887c6bb-6621-43a2-92f7-4749fe507c2e","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"source_id":{"alias":"source_id","fullyQualifiedColumn":"TravelDetails.source_id","columnId":"4c7b72d6-2258-4585-bc7f-69c5c96385bc","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"source_id_error":{"alias":"source_id_error","fullyQualifiedColumn":"TravelDetails.source_id_error","columnId":"c9fd8509-de59-43aa-8f3c-e39807dd1739","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"source":{"alias":"source","fullyQualifiedColumn":"TravelDetails.source","columnId":"0c384085-c32f-4a33-ac8f-9ac95d93b750","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"destination_id":{"alias":"destination_id","fullyQualifiedColumn":"TravelDetails.destination_id","columnId":"39d0f806-b0bc-4a3c-a7e7-75caaf22594f","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"destination":{"alias":"destination","fullyQualifiedColumn":"TravelDetails.destination","columnId":"ee4157c9-22d9-4f3c-8fb2-559afe53a270","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"travel_cost":{"alias":"travel_cost","fullyQualifiedColumn":"TravelDetails.travel_cost","columnId":"c90417a7-99e1-4375-bb0a-e3c5fa19e6a4","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"mode_of_payment":{"alias":"mode_of_payment","fullyQualifiedColumn":"TravelDetails.mode_of_payment","columnId":"239091c4-935e-4b2b-97f6-f35c3762674b","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"booking_platform":{"alias":"booking_platform","fullyQualifiedColumn":"TravelDetails.booking_platform","columnId":"812a6a72-6318-43d4-921f-14923c00255a","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"travelled_by":{"alias":"travelled_by","fullyQualifiedColumn":"TravelDetails.travelled_by","columnId":"6ae646aa-4f75-4c1c-90e2-4d48f4d1aecf","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}}}},"employee_details":{"id":"ac683cac-7cb1-4318-aae5-b86acc8559d4","alias":"employee_details","columns":{"employee_id":{"alias":"employee_id","fullyQualifiedColumn":"employee_details.employee_id","columnId":"7c46558c-754a-4a62-aa41-23235d6c97aa","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"employee_name":{"alias":"employee_name","fullyQualifiedColumn":"employee_details.employee_name","columnId":"d8726b12-1a86-4396-979e-599f0c556a48","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"age":{"alias":"age","fullyQualifiedColumn":"employee_details.age","columnId":"61b9181b-2d05-45e6-adea-4b550c047cd5","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"address":{"alias":"address","fullyQualifiedColumn":"employee_details.address","columnId":"872c85a7-3d5f-48a9-8ec3-	

	fc3df6d29aff","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}}},"employee_details_1":{"id":"6a5f8fdf-3892-4ea6-ab00-734cf78dae46","alias":"employee_details_1","columns":{"employee_id":{"alias":"employee_id","fullyQualifiedColumn":"employee_details_1.employee_id","columnId":"87341398-0739-40f3-b827-2ce9c8a31010","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"employee_name":{"alias":"employee_name","fullyQualifiedColumn":"employee_details_1.employee_name","columnId":"83e739ea-3b89-4502-94bc-85397be08ac7","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"age":{"alias":"age","fullyQualifiedColumn":"employee_details_1.age","columnId":"81f393fe-3c1e-492d-a8f3-a8e36a559b9d","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"address":{"alias":"address","fullyQualifiedColumn":"employee_details_1.address","columnId":"89cdd7bd-ba45-4bb7-867d-c3165f5b3881","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}}},"employee_details_2":{"id":"b5415820-0458-4869-8214-3dbf2cdf58fb","alias":"employee_details_2","columns":{"employee_id":{"alias":"employee_id","fullyQualifiedColumn":"employee_details_2.employee_id","columnId":"2b2a294f-edc6-4534-847c-50b912cabb40","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"employee_name":{"alias":"employee_name","fullyQualifiedColumn":"employee_details_2.employee_name","columnId":"984ddeb8-a3ae-49c8-9c42-4bc1cb77f64a","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"age":{"alias":"age","fullyQualifiedColumn":"employee_details_2.age","columnId":"6cd74369-cc98-4521-9513-c7571067f11d","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"address":{"alias":"address","fullyQualifiedColumn":"employee_details_2.address","columnId":"7e41ec60-05b4-4dca-a657-b00bc07e8e42","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}}},"geoCordinates":{"id":"e8cb6f87-a4c5-412f-96c4-424c42035195","alias":"geoCordinates","columns":{"location_id":{"alias":"location_id","fullyQualifiedColumn":"geoCordinates.location_id","columnId":"f97a09a0-c0da-4eaf-b59d-3586e1f6c28f","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"location":{"alias":"location","fullyQualifiedColumn":"geoCordinates.location","columnId":"637d02ca-8867-4f11-9684-48d3684648b6","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"latitude":{"alias":"latitude","fullyQualifiedColumn":"geoCordinates.latitude","columnId":"68c07c80-f4ed-4fa9-9dce-71152fc3f463","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Double":"numeric"}},"longitude":{"alias":"longitude","fullyQualifiedColumn":"geoCordinates.longitude","columnId":"e04dabd0-d137-43e5-bceb-91df2498ed52","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Double":"numeric"}}}},"meeting_details":{"id":"8802ad50-f981-4147-85ea-7962fbba3733","alias":"meeting_details","columns":{"meeting_id":{"alias":"meeting_id","fullyQualifiedColumn":"meeting_details.meeting_id","columnId":"1f04c09f-aade-4c9c-849b-f97ec7576085","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"meeting_date":{"alias":"meeting_date","fullyQualifiedColumn":"meeting_details.meeting_date","columnId":"41dd7997-54a8-4d1d-9231-6d5df501ae0c","defaultFunction":"db.generic.groupBy.group","type":{"java.sql.Timestamp":"dateTime"}},"meeting_by":{"alias":"meeting_by","fullyQualifiedColumn":"meeting_details.meeting_by","columnId":"4906ded5-d8ef-48f3-8a80-35b7cb5b5073","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"client_name":{"alias":"client_name","fullyQualifiedColumn":"meeting_details.client_name","columnId":"87efa8ad-420f-4399-95e3-609824a3a4fc","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"meeting_purpose":{"alias":"meeting_purpose","fullyQualifiedColumn":"meeting_details.meeting_purpose","columnId":"dfaa00a7-c7f1-4698-be80-ebf4849debd4","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"meeting_impact":{"alias":"meeting_impact","fullyQualifiedColumn":"meeting_details.meeting_impact","columnId":"03a0ce7e-d904-40c7-ac93-6e1114d5aff5","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"meet_cancellation_status":{"alias":"meet_cancellation_status","fullyQualifiedColumn":"meeting_details.meet_cancellation_status","columnId":"cfb581a6-63f1-4076-8c16-1ba64d7838ab","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"cancellation_reason":{"alias":"cancellation_reason","fullyQualifiedColumn":"meeting_details.cancellation_reason","columnId":"42321407-1cd9-43ed-b306-

	993a99d0beca","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}}}},"sets":[["employee_details_1"],["TravelDetails","employee_details","geoCordinates","meeting_details"],["employee_details_2"]],"joins":[{"id":"b2ddcd16-e0a1-48ac-bf03-2d51f2e73a9f","type":"inner","operator":"=","left":{"table":"employee_details","column":"employee_id","alias":"employee_details.employee_id"},"right":{"table":"TravelDetails","column":"travelled_by","alias":"TravelDetails.travelled_by"}},{"id":"74a13656-7f0d-4b4e-a3a3-a647a6b92161","type":"inner","operator":"=","left":{"table":"geoCordinates","column":"location_id","alias":"geoCordinates.location_id"},"right":{"table":"TravelDetails","column":"source_id","alias":"TravelDetails.source_id"}},{"id":"c8fe88d5-42fa-427c-b92c-9f9eb49a704d","type":"inner","operator":"=","left":{"table":"geoCordinates","column":"location_id","alias":"geoCordinates.location_id"},"right":{"table":"TravelDetails","column":"destination_id","alias":"TravelDetails.destination_id"}},{"id":"34116e5d-a424-4e3f-8203-98be1aaba904","type":"inner","operator":"=","left":{"table":"employee_details","column":"employee_id","alias":"employee_details.employee_id"},"right":{"table":"meeting_details","column":"meeting_by","alias":"meeting_details.meeting_by"}}]},"uniqueId":"e04ec0a5-bc83-4842-b7d2-fb163b2eca1f"}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. The metadata array with list of selected tables with its columns details get return with schema name.The unique id for metadata will get assign. **classifier** : name of classifier **metadata** : metadata array **catalogs** : list of catalogs **schemas** : Schema array with schema name nad related information. Schema is the DB name under this name is the table name with all associated columns details like : size of column , name of column , nullable status of column,position of column , type of column(java class)
Service Status	200 OK
Screenshot	

Metadata SaveWorkFlow

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
{"size":"50","nullable":"TRUE","dataType":
{"java.lang.String":"text"},"name":"client_name","position":"4","type":"java.lang.String","checked":true},
{"size":"50","nullable":"TRUE","dataType":
{"java.lang.String":"text"},"name":"meeting_purpose","position":"5","type":"java.lang.String","checked":true},
{"size":"50","nullable":"TRUE","dataType":
{"java.lang.String":"text"},"name":"meeting_impact","position":"6","type":"java.lang.String","checked":true},
{"size":"50","nullable":"TRUE","dataType":
{"java.lang.String":"text"},"name":"meet_cancellation_status","position":"7","type":"java.lang.String","checked":true},
{"size":"50","nullable":"TRUE","dataType":
{"java.lang.String":"text"},"name":"cancellation_reason","position":"8","type":"java.lang.String","checked":true}]}]}],"catalog":"Sampl
eTravelData"}]},"removeItem":{"tables":[],"columns":[],"views":[]}}
```

Status: 200 OK Time: 596 ms

```
{"status":1,"response":{"metadata":{"classifier":"db.workflow","name":"SampleTravelData","dataSource":{"sync":false,"id":"83","catSchemaPredict
```

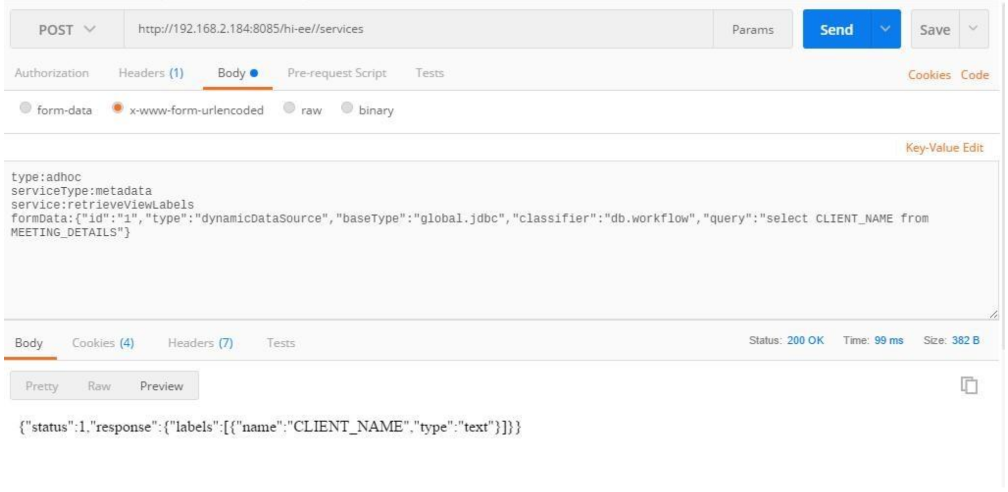
4.4.1.5 Execute View:

URL	/services	
Description	It executes the view query while view creation before saving it.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=retrieveViewLabels&formData={'id':'1','type':'dynamicDataSource','baseType':'global.jdbc','classifier':'db.workflow','query':'select CLIENT_NAME from MEETING_DETAILS'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	retrieveViewLabels	Service as retrieveViewLabels
formData:	{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","classifier":"db.workflow","query":"select CLIENT_NAME from MEETING_DETAILS"}	formData contains the query used for view and the return columns with its details.

Response Output(JSON Format)	{
	"status":1,"response":{"labels":[{"name":"CLIENT_NAME","type":"text"}]}
	}	

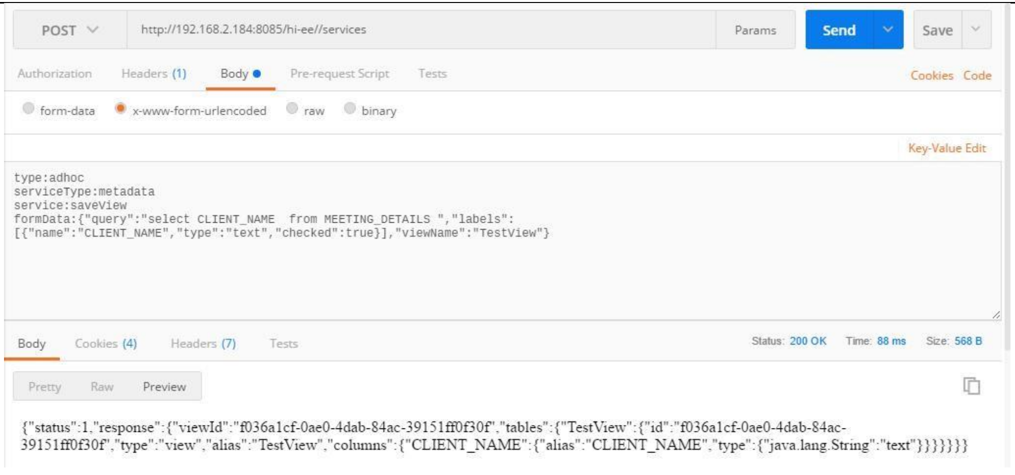
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. The View get executed and returns the fetched column details. **name** : Name of the column **type** : Type of column	

Service Status	200 OK

Screenshot	 POST http://192.168.2.184:8085/hi-ee//services
	x-www-form-urlencoded
	`type:adhoc`
	`serviceType:metadata`
	`service:retrieveViewLabels`
	`formData:{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","classifier":"db.workflow","query":"select CLIENT_NAME from MEETING_DETAILS"}`
	Status: 200 OK Time: 99 ms Size: 382 B
	`{"status":1,"response":{"labels":[{"name":"CLIENT_NAME","type":"text"}]}}`

4.4.1.6 Save View:

URL	/services
Description	It assigns the ID for view and view details will get save.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=saveView&formData={'query':'select CLIENT_NAME from MEETING_DETAILS ','labels':[{'name':'CLIENT_NAME','type':'text','checked':true}],'viewName':'

	TestView'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	saveView	Service as saveView
formData:	{"query":"select CLIENT_NAME from MEETING_DETAILS ","labels":[{"name":"CLIENT_NAME","type":"text","checked":true}], "viewName":"TestView"}	formData contains the query used for view and the return columns with its details.
Response Output(JSON Format)	{"status":1,"response":{"viewId":"f036a1cf-0ae0-4dab-84ac-39151ff0f30f","tables":{"TestView":{"id":"f036a1cf-0ae0-4dab-84ac-39151ff0f30f","type":"view","alias":"TestView","columns":{"CLIENT_NAME":{"alias":"CLIENT_NAME","type":{"java.lang.String":"text"}}}}}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. The View get created and the id for view get assigned with its details.. **viewId** : Id of the created view **tables** : tables array **type** : type as view **columns** : Array with alias name	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.184:8085/hi-ee//services

Body: x-www-form-urlencoded

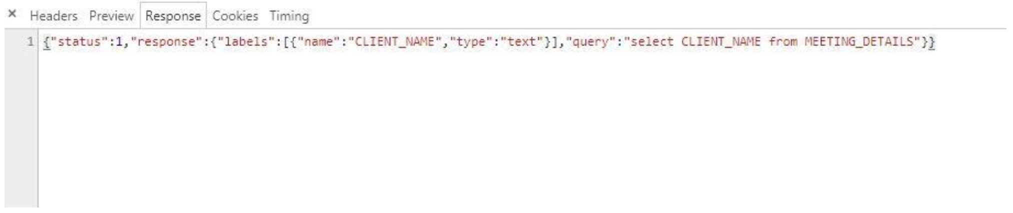
```
type:adhoc
serviceType:metadata
service:saveView
formData:{"query":"select CLIENT_NAME  from MEETING_DETAILS ","labels":
[{"name":"CLIENT_NAME","type":"text","checked":true}],"viewName":"TestView"}
```

Status: 200 OK Time: 88 ms Size: 568 B

```
{"status":1,"response":{"viewId":"f036a1cf-0ae0-4dab-84ac-39151ff0f30f","tables":{"TestView":{"id":"f036a1cf-0ae0-4dab-84ac-
39151ff0f30f","type":"view","alias":"TestView","columns":{"CLIENT_NAME":{"alias":"CLIENT_NAME","type":{"java.lang.String":"text"}}}}}}}
```

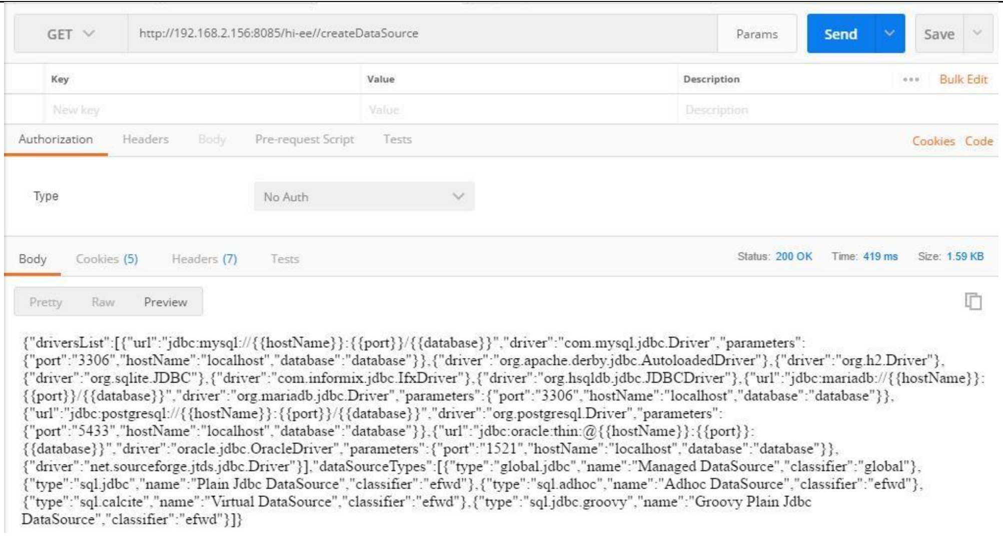
4.4.1.7 Edit View:

URL	/services	
Description	It allows user to edit the existing view while metdata creation.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=retrieveView&formData={'id':'1','type':'dynamicDataSource','baseType':'global.jdbc','classifier':'db.workflow','viewId':'475d53c1-b606-40ca-816b-05d784ea5a50'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	retrieveView	Service as retrieveView
formData:	{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","classifier":"db.workflow","viewId":"475d53c1-b606-40ca-816b-05d784ea5a50"}	formData contains the viewID with datasource details.
Response Output(JSON Format)	{"status":1,"response":{"labels":[{"name":"CLIENT_NAME","type":"text"}],"query":"select CLIENT_NAME from MEETING_DETAILS"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. **viewId** : Id of the created view **tables** : tables array **type** : type as view **columns** : Array with alias name	

	The View get created and the id for view get assigned with its details..
Service Status	200 OK
Screenshot	 `{"status":1,"response":{"labels":[{"name":"CLIENT_NAME","type":"text"}],"query":"select CLIENT_NAME from MEETING_DETAILS"}}`

4.4.1.8 Change DataSource :

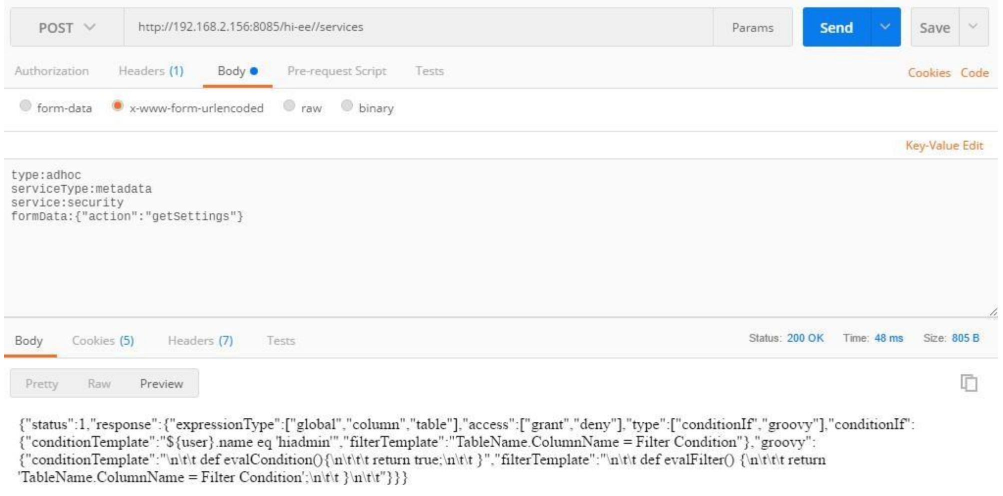
URL	/createDataSource
Description	It allows user to change the datasource while metdata creation.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST,GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//createDataSource **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin" http://192.168.2.156:8085/hi-ee//createDataSource -v
Response Output(JSON Format)	{"driversList":[{"url":"jdbc:mysql://{{hostName}}:{{port}}/{{database}}","driver":"com.mysql.jdbc.Driver","parameters":{"port":"3306","hostName":"localhost","database":"database"}},{"driver":"org.apache.derby.jdbc.AutoloadedDriver"},{"driver":"org.h2.Driver"},{"driver":"org.sqlite.JDBC"},{"driver":"com.informix.jdbc.IfxDriver"},{"driver":"org.hsqldb.jdbc.JDBCDriver"},{"url":"jdbc:mariadb://{{hostName}}:{{port}}/{{database}}","driver":"org.mariadb.jdbc.Driver","parameters":{"port":"3306","hostName":"localhost","database":"database"}},{"url":"jdbc:postgresql://{{hostName}}:{{port}}/{{database}}","driver":"org.postgresql.Driver","parameters":{"port":"5433","hostName":"localhost","database":"database"}},{"url":"jdbc:oracle:thin:@{{hostName}}:{{port}}:{{database}}","driver":"oracle.jdbc.OracleDriver","parameters":{"port":"1521","hostName":"localhost","database":"database"}},{"driver":"net.sourceforge.jtds.jdbc.Driver"}],"dataSourceTypes":[{"type":"global.jdbc","name":"Managed DataSource","classifier":"global"},{"type":"sql.jdbc","name":"Plain Jdbc DataSource","classifier":"efwd"},{"type":"sql.adhoc","name":"Adhoc DataSource","classifier":"efwd"},{"type":"sql.calcite","name":"Virtual DataSource","classifier":"efwd"},{"type":"sql.jdbc.groovy","name":"Groovy Plain Jdbc DataSource","classifier":"efwd"}]}

Description of Response Output:	The response of the API is , it returns the all available driverlist array with each driver details. Parameters include the port number,host,database , driver etc. **driversList** : Array of driverslist **url** : Connection URL **driver** : Name of the driver etc
Service Status	200 OK
Screenshot	

```
{"driversList":[{"url":"jdbc:mysql://{{hostName}}:{{port}}/{{database}}","driver":"com.mysql.jdbc.Driver","parameters":
{"port":"3306","hostName":"localhost","database":"database"}},{"driver":"org.apache.derby.jdbc.AutoloadedDriver"},{"driver":"org.h2.Driver"},
{"driver":"org.sqlite.JDBC"},{"driver":"com.informix.jdbc.IfxDriver"},{"driver":"org.hsqldb.jdbc.JDBCDriver"},{"url":"jdbc:mariadb://{{hostName}}:
{{port}}/{{database}}","driver":"org.mariadb.jdbc.Driver","parameters":{"port":"3306","hostName":"localhost","database":"database"}},
{"url":"jdbc:postgresql://{{hostName}}:{{port}}/{{database}}","driver":"org.postgresql.Driver","parameters":
{"port":"5433","hostName":"localhost","database":"database"}},{"url":"jdbc:oracle:thin:@{{hostName}}:{{port}}:
{{database}}","driver":"oracle.jdbc.OracleDriver","parameters":{"port":"1521","hostName":"localhost","database":"database"}},
{"driver":"net.sourceforge.jtds.jdbc.Driver"}],"dataSourceTypes":[{"type":"global.jdbc","name":"Managed DataSource","classifier":"global"},
{"type":"sql.jdbc","name":"Plain Jdbc DataSource","classifier":"efwd"},{"type":"sql.adhoc","name":"Adhoc DataSource","classifier":"efwd"},
{"type":"sql.calcite","name":"Virtual DataSource","classifier":"efwd"},{"type":"sql.jdbc.groovy","name":"Groovy Plain Jdbc
DataSource","classifier":"efwd"}]}
```

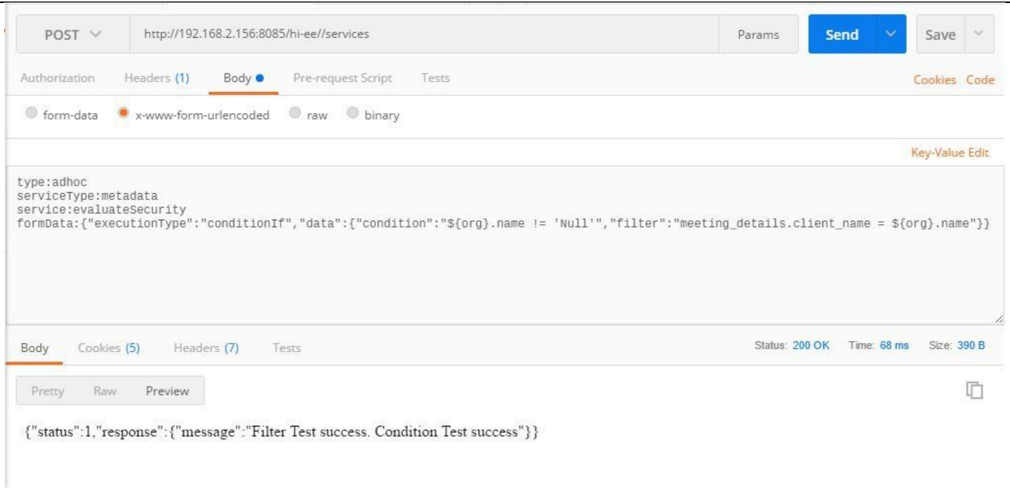
4.4.1.9 Get Security details:

URL	/services
Description	It allows user to get security details while metdata creation.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=security&formData={'action':'gettings'}" http://192.168.2.156:8085/hi-ee//services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	security	Service as security
formData:	{"action":"getSettings"}	formData contains the action as getSettings.
Response Output(JSON Format)	{"status":1,"response":{"expressionType":["global","column","table"],"access":["grant","deny"],"type":["conditionIf","groovy"],"conditionIf":{"conditionTemplate":"${user}.name eq 'hiadmin'","filterTemplate":"TableName.ColumnName = Filter Condition"},"groovy":{"conditionTemplate":"\n\t\t def evalCondition(){\n\t\t\t return true;\n\t\t }","filterTemplate":"\n\t\t def evalFilter() {\n\t\t\t return 'TableName.ColumnName = Filter Condition';\n\t\t }\n\t\t"}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. Metadata security details get return. **expressionType** : expressionType array with expression details **access** : Security access type **type** : type of security **conditionIf** : security conditionIf etc	
Service Status	200 OK	
Screenshot	 POST http://192.168.2.156:8085/hi-ee//services Body: x-www-form-urlencoded type:adhoc serviceType:metadata service:security formData:{"action":"getSettings"} Status: 200 OK Time: 48 ms Size: 805 B {"status":1,"response":{"expressionType":["global","column","table"],"access":["grant","deny"],"type":["conditionIf","groovy"],"conditionIf":{"conditionTemplate":"${user}.name eq 'hiadmin'","filterTemplate":"TableName.ColumnName = Filter Condition"},"groovy":{"conditionTemplate":"\n\t\t def evalCondition(){\n\t\t\t return true;\n\t\t }","filterTemplate":"\n\t\t def evalFilter() {\n\t\t\t return 'TableName.ColumnName = Filter Condition';\n\t\t }\n\t\t"}}}	

4.4.1.10 Validate Security details:

URL	/services	
Description	It allows user to validate requested security details while metdata creation.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=evaluateSecurity&formData={'executionType':'conditionIf','data':{'condition':'${org}.name != 'Null'','filter':'meeting_details.client_name = ${org}.name'}}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	evaluateSecurity	Service as evaluateSecurity
formData:	{"executionType":"conditionIf","data":{"condition":"${org}.name != 'Null'","filter":"meeting_details.client_name = ${org}.name"}}	formData contains the aall metadata security details like executionType , condition and the filter data.
Response Output(JSON Format)	{"status":1,"response":{"message":"Filter Test success. Condition Test success"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with success message.	

Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

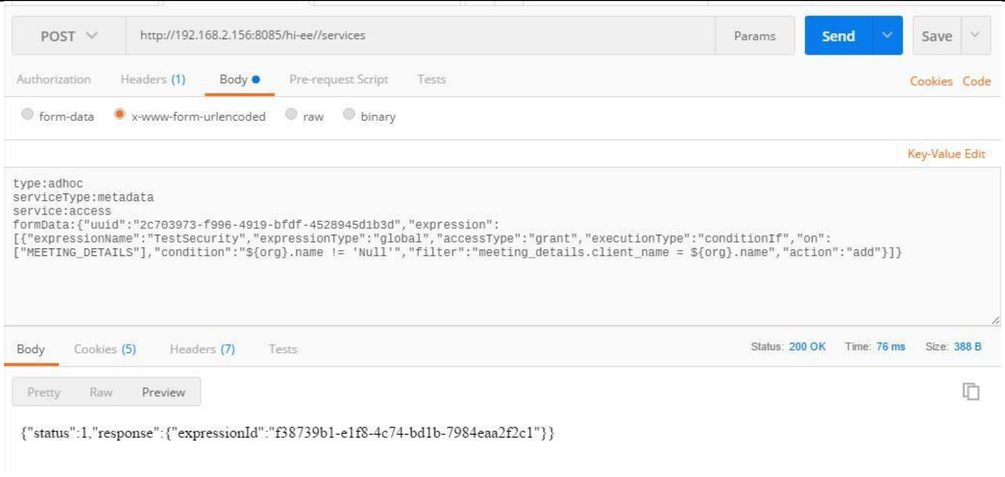
```
type:adhoc
serviceType:metadata
service:evaluateSecurity
formData:{"executionType":"conditionIf","data":{"condition":"${org}.name != 'Null'","filter":"meeting_details.client_name = ${org}.name"}}
```

Status: 200 OK Time: 68 ms Size: 390 B

```
{"status":1,"response":{"message":"Filter Test success. Condition Test success"}}
```

4.4.1.11 Apply Security details:

URL	/services	
Description	It allows user to apply requested security details while metdata creation.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=access&formData={'uuid':'2c703973-f996-4919-bfdf-4528945d1b3d','expression':[{'expressionName':'TestSecurity','expressionType':'global','accessType':'grant','executionType':'conditionIf','on':['MEETING_DETAILS'],'condition':'${org}.name != 'Null'','filter':'meeting_details.client_name = ${org}.name','action':'add'}]}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**

type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	access	Service as access
formData:	{"uuid":"2c703973-f996-4919-bfdf-4528945d1b3d","expression":[{"expressionName":"TestSecurity","expressionType":"global","accessType":"grant","executionType":"conditionIf","on":["MEETING_DETAILS"],"condition":"${org}.name != 'Null'","filter":"meeting_details.client_name = ${org}.name","action":"add"}]}	formData contains the all metadata security details like executionType , condition and the filter data. Alog with uuid assigned for metadata.
Response Output(JSON Format)	{"status":1,"response":{"expressionId":"fb11ee83-8b12-4fb1-94d2-625db4681c9b"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with assigned **expressionId** for security.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8085/hi-ee//services

```
type:adhoc
serviceType:metadata
service:access
formData:{"uuid":"2c703973-f996-4919-bfdf-4528945d1b3d","expression":
[{"expressionName":"TestSecurity","expressionType":"global","accessType":"grant","executionType":"conditionIf","on":
["MEETING_DETAILS"],"condition":"${org}.name != 'Null'","filter":"meeting_details.client_name = ${org}.name","action":"add"}]}
```

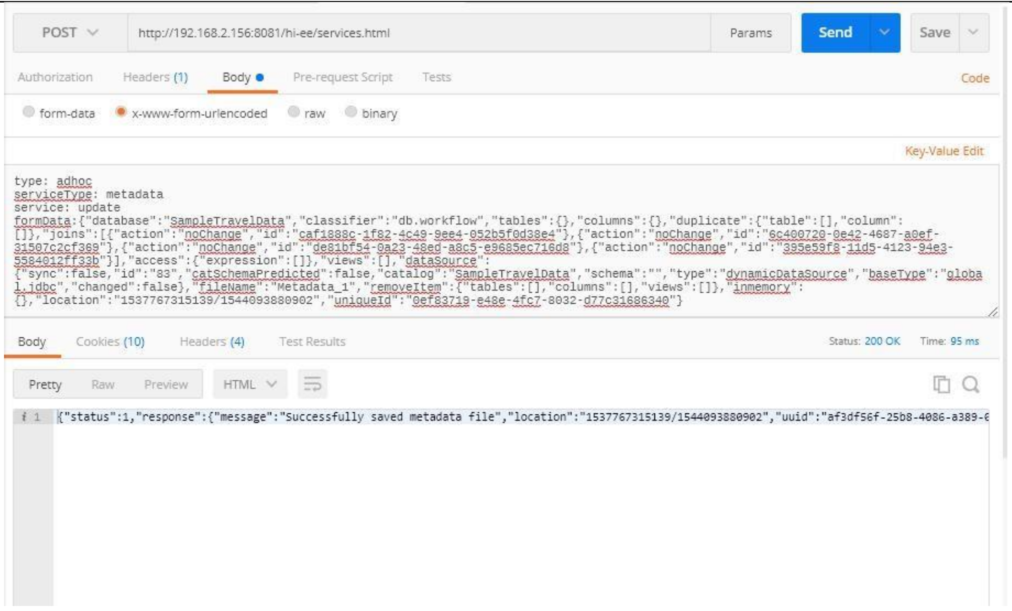
Status: 200 OK Time: 76 ms Size: 388 B

```
{"status":1,"response":{"expressionId":"f38739b1-e1f8-4c74-bd1b-7984eaa2f2c1"}}
```

4.4.1.12 Save Metadata details:

URL	/services
Description	It allows user to save the metdata after applied all the filters , security.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.

Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=update&formData={'database':'HIUSER','classifier':'db.workflow','tables':{'HIUSER.MEETING_DETAILS':'MEETINGDETAILS'},'columns':{},'joins':[{'action':'noChange','id':'c2726f55-f43f-44bc-99f6-122e1b71204a'}],'access':{'expression':[{'expressionId':'fb11ee83-8b12-4fb1-94d2-625db4681c9b','action':'delete'},{'expressionId':'e1520daf-958b-423d-a4e5-ac7a5c5fc1cd','action':'add'}]},'views':['475d53c1-b606-40ca-816b-05d784ea5a50'],'dataSource':{'id':'1','type':'dynamicDataSource','baseType':'global.jdbc'},'location':'1507554717873','uniqueId':'2c703973-f996-4919-bfdf-4528945d1b3d','fileName':'TestMetadata'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	update	Service as update
formData:	{"database":"SampleTravelData","classifier":"db.workflow","tables":{},"columns":{},"duplicate":{"table":[],"column":[]},"joins":[{"action":"noChange","id":"caf1888c-1f82-4c49-9ee4-052b5f0d38e4"},{"action":"noChange","id":"6c400720-0e42-4687-a0ef-31507c2cf369"},{"action":"noChange","id":"de81bf54-0a23-48ed-a8c5-e9685ec716d8"},{"action":"noChange","id":"395e59f8-11d5-4123-94e3-5584012ff33b"}],"access":{"expression":[]},"views":[],"dataSource":{"sync":false,"id":"83","catSchemaPredicted":false,"catalog":"SampleTravelData","schema":"","type":"dynamicDataSource","baseType":"global.jdbc","changed":false},"fileName":"Metadata_1","removeItem":{"tables":[],"columns":[],"views":[]},"inmemory":{},"location":"1537767315139/1544093880902","uniqueId":"0ef83719-e48e-4fc7-8032-d77c31686340"}	formData contains the all metadata details like database details,metadatasecurity details etc.

Response Output(JSON Format)	{"status":1,"response":{"message":"Successfully saved metadata file","location":"1537767315139/1544093880902","uuid":"af3df56f-25b8-4086-a389-0bb4f8561e9d.metadata"}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with success message and the metadata stored **location** with **physical name** of **metadata** file.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
type: adhoc
serviceType: metadata
service: update
formData:{"database":"SampleTravelData","classifier":"db.workflow","tables":{},"columns":{},"duplicate":{"table":[],"column":
[]},"joins":[{"action":"noChange","id":"caf1888c-1f82-4c49-9ee4-052b5f0d38e4"},{"action":"noChange","id":"6c400720-0e42-4687-a0ef-
31507c2cf369"},{"action":"noChange","id":"de81bf54-0a23-48ed-a8c5-e9685ec716d8"},{"action":"noChange","id":"395e59f8-11d5-4123-94e3-
5584012ff33b"}],"access":{"expression":[]},"views":[],"dataSource":
{"sync":false,"id":"83","catSchemaPredicted":false,"catalog":"SampleTravelData","schema":"","type":"dynamicDataSource","baseType":"globa
l.jdbc","changed":false},"fileName":"Metadata_1","removeItem":{"tables":[],"columns":[],"views":[]},"inmemory":
{},"location":"1537767315139/1544093880902","uniqueId":"0ef83719-e48e-4fc7-8032-d77c31686340"}
```

Status: 200 OK Time: 95 ms

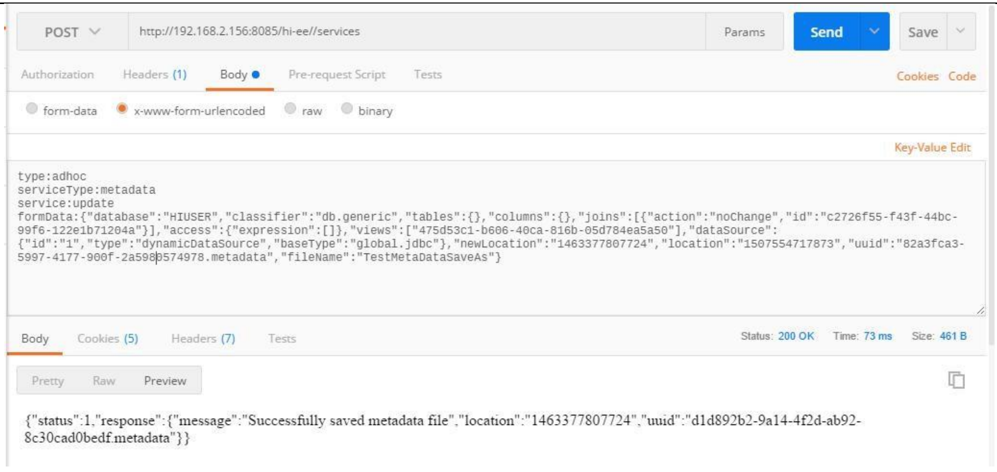
```
{"status":1,"response":{"message":"Successfully saved metadata file","location":"1537767315139/1544093880902","uuid":"af3df56f-25b8-4086-a389-
```

4.4.1.13 Metadata SaveAs:

Note : Metadata saveAs service is same as metadata save service only difference is that it has the extra key values as newLocation,name of file etc.

URL	/services
Description	It allows user to saveAs the metdata at different location with different filename after applied all the filters , security .
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :**

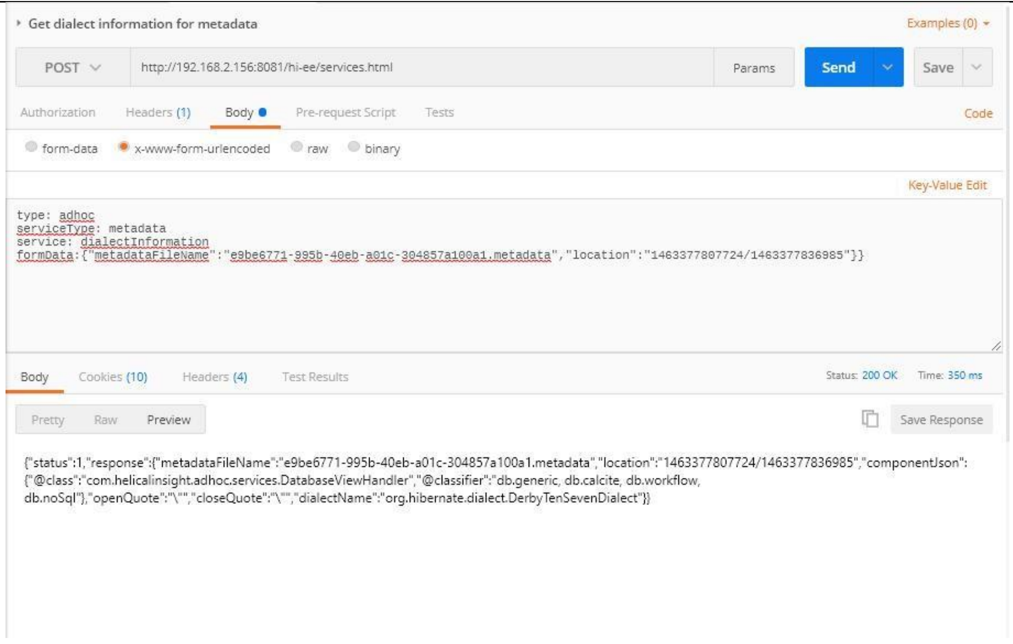
	http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=update&formData={'database':'HIUSER','classifier':'db.generic','tables':{},'columns':{},'joins':[{'action':'noChange','id':'c2726f55-f43f-44bc-99f6-122e1b71204a'}],'access':{'expression':[]},'views':['475d53c1-b606-40ca-816b-05d784ea5a50'],'dataSource':{'id':'1','type':'dynamicDataSource','baseType':'global.jdbc'},'newLocation':'1507554717873','location':'1463377807724','uuid':'c1df0ca2-91ae-4ae0-99fe-07bf158402be.metadata','fileName':'TestMetaDataSaveAs'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	update	Service as update
formData:	{"database":"HIUSER","classifier":"db.generic","tables":{},"columns":{},"joins":[{"action":"noChange","id":"c2726f55-f43f-44bc-99f6-122e1b71204a"}],"access":{"expression":[]},"views":["475d53c1-b606-40ca-816b-05d784ea5a50"],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc"},"newLocation":"1463377807724","location":"1507554717873","uuid":"82a3fca3-5997-4177-900f-2a5980574978.metadata","fileName":"TestMetaDataSaveAs"}	formData contains the all metadata details like database details,metadatasecurity details etc. It contains the metadata file new location.
Response Output(JSON Format)	{"status":1,"response":{"message":"Successfully saved metadata file","location":"1463377807724","uuid":"5e403ba2-8885-457d-8a46-6290908d3cba.metadata"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with success message and the metadata stored new **location** with **physical name** of **metadata file**.	
Service Status	200 OK	

Screenshot	

```
POST http://192.168.2.156:8085/hi-ee//services

type:adhoc
serviceType:metadata
service:update
formData:{"database":"HIUSER","classifier":"db.generic","tables":{},"columns":{},"joins":[{"action":"noChange","id":"c2726f55-f43f-44bc-99f6-122e1b71204a"}],"access":{"expression":[]},"views":["475d53c1-b606-40ca-816b-05d784ea5a50"],"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc"},"newLocation":"1463377807724","location":"1507554717873","uuid":"82a3fca3-5997-4177-900f-2a5980574978.metadata","fileName":"TestMetaDataSaveAs"}

Status: 200 OK  Time: 73 ms  Size: 461 B

{"status":1,"response":{"message":"Successfully saved metadata file","location":"1463377807724","uuid":"d1d892b2-9a14-4f2d-ab92-8c30cad0bedf.metadata"}}
```

4.4.1.14 Get Dialect Information for metadata:

URL	/services.html	
Description	The user will get the dialect information as per provided metadata file.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=dialectInformation&formData={'metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','location':'1463377807724/1463377836985'}}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	dialectInformation	The service is to get the dialectInformation

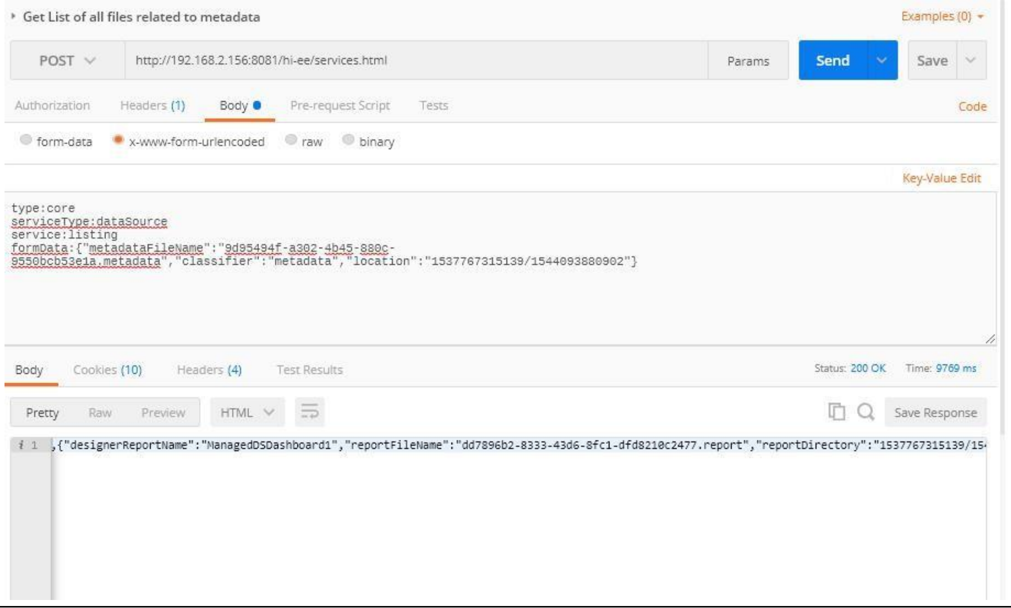
formData:	{"metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","location":"1463377807724/1463377836985"}}	Provide metadata file name and its physical location
Response Output(JSON Format)	{"status":1,"response":{"metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","location":"1463377807724/1463377836985","componentJson":{"@class":"com.helicalinsight.adhoc.services.DatabaseViewHandler","@classifier":"db.generic, db.calcite, db.workflow, db.noSql"},"openQuote":"\"","closeQuote":"\"","dialectName":"org.hibernate.dialect.DerbyTenSevenDialect"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot	 Get dialect information for metadata POST http://192.168.2.156:8081/hi-ee/services.html Body: x-www-form-urlencoded type: adhoc serviceType: metadata service: dialectInformation formData:{"metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","location":"1463377807724/1463377836985"}} Status: 200 OK Time: 350 ms {"status":1,"response":{"metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","location":"1463377807724/1463377836985","componentJson":{"@class":"com.helicalinsight.adhoc.services.DatabaseViewHandler","@classifier":"db.generic, db.calcite, db.workflow, db.noSql"},"openQuote":"\"","closeQuote":"\"","dialectName":"org.hibernate.dialect.DerbyTenSevenDialect"}}	

4.4.1.15 Get all files related to metadata:

URL	/services.html
Description	The user will get all dependent files related to metadata.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :**

	http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'metadataFileName':'9d95494f-a302-4b45-880c-9550bcb53e1a.metadata','classifier':'metadata','location':'1537767315139/1544093880902'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource
service:	listing	The service is to list the resources related to provided global datasourceID
formData:	{"metadataFileName":"9d95494f-a302-4b45-880c-9550bcb53e1a.metadata","classifier":"metadata","location":"1537767315139/1544093880902"}	metadataFileName:name of the metadata file. Classifier :metadata location : location of metadata file
Response Output(JSON Format)	{"status":1,"response":{"adhocReports":[{"reportFileName":"4a4f065d-d06d-4315-8bfe-581740bff5df.report","reportName":"PlainJDBCReport2","metadataFileName":"9d95494f-a302-4b45-880c-9550bcb53e1a.metadata","location":"/1537767315139/1544093880902/","savedReports":[],"designerReports":[{"designerReportName":"PlainJDBCDSReport2","reportFileName":"4a4f065d-d06d-4315-8bfe-581740bff5df.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"6fc19cb2-31ae-4750-96f7-1517c64b5b03.efw"},{"designerReportName":"PlainJDBCDSReport1","reportFileName":"4a4f065d-d06d-4315-8bfe-581740bff5df.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"8ea54e00-2109-4135-a14a-b42cbe95a2c6.efw"}]},{"reportFileName":"a84fcee7-bc94-47af-a5c6-1db225940b6e.report","reportName":"PlainJDBCDSReport1","metadataFileName":"9d95494f-a302-4b45-880c-9550bcb53e1a.metadata","location":"/1537767315139/1544093880902/","savedReports":[],"designerReports":[{"designerReportName":"PlainJDBCDSReport2","reportFileName":"a84fcee7-bc94-47af-a5c6-1db225940b6e.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"6fc19cb2-31ae-4750-96f7-1517c64b5b03.efw"},{"designerReportName":"PlainJDBCDSReport1","reportFileName":"a84fcee7-bc94-47af-a5c6-1db225940b6e.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"8ea54e00-2109-4135-a14a-b42cbe95a2c6.efw"}]},{"reportFileName":"dd7896b2-8333-43d6-8fc1-dfd8210c2477.report","reportName":"ManagedDSReport2","metadataFileName":"9d95494f-a302-4b45-880c-9550bcb53e1a.metadata","location":"/1537767315139/1544093880902/","savedReports":[],"designerReports":[{"designerReportName":"ManagedDSDashboard2","reportFileName":"dd7896b2-8333-43d6-8fc1-	

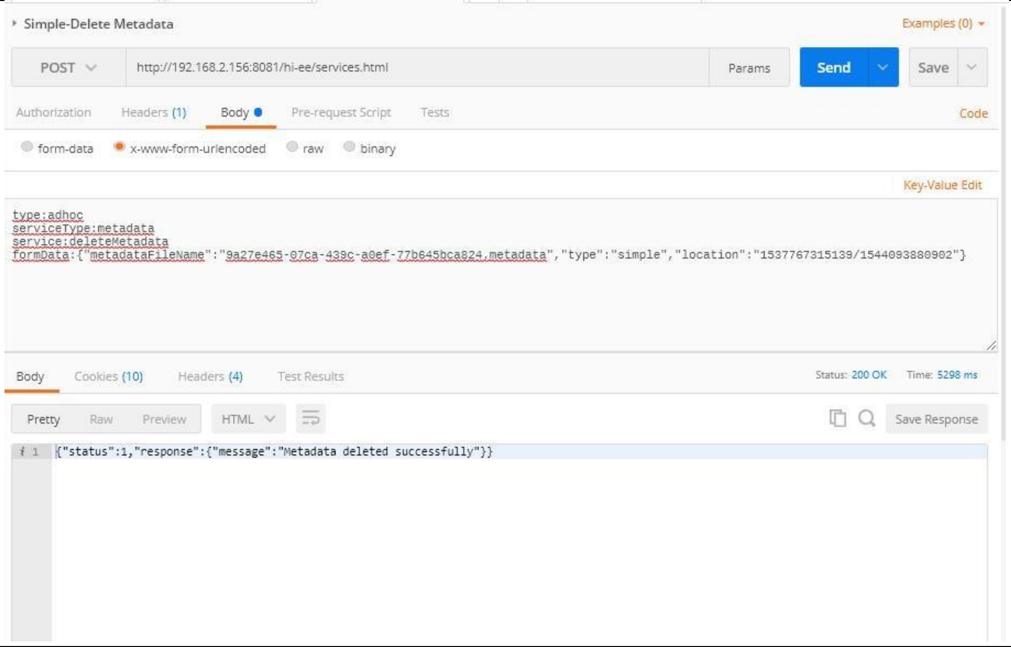
	dfd8210c2477.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"05a64cbd-b949-45f2-ad55-c8f973348465.efw"},{"designerReportName":"ManagedDSDashboard1","reportFileName":"dd7896b2-8333-43d6-8fc1-dfd8210c2477.report","reportDirectory":"1537767315139/1544093880902","efwFileName":"32e31b8e-6e2d-4286-8e15-64e9736a4028.efw"}]}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.
Service Status	200 OK

Screenshot	 Get List of all files related to metadata
	POST http://192.168.2.156:8081/hi-ee/services.html
	Body: x-www-form-urlencoded
	`type:core`
	`serviceType:dataSource`
	`service:listing`
	`formData:{"metadataFileName":"9d95494f-a302-4b45-880c-9550bcb53e1a.metadata","classifier":"metadata","location":"1537767315139/1544093880902"}`
	Status: 200 OK Time: 9769 ms
	`,{"designerReportName":"ManagedDSDashboard1","reportFileName":"dd7896b2-8333-43d6-8fc1-dfd8210c2477.report","reportDirectory":"1537767315139/15`

4.4.1.16 Delete metadata:

4.4.1.16.1 Simple- Delete metadata

URL	/services.html
Description	The user can delete metadata with simple type which will delete only respective metadata file.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :**

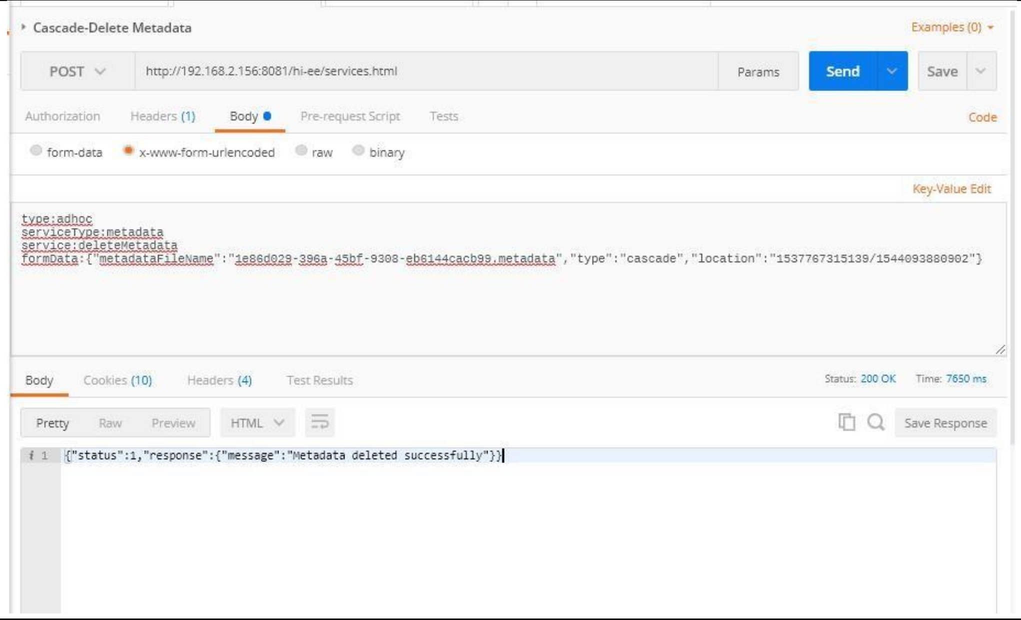
	curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=deleteMetadata&formData={'metadataFileName':'9a27e465-07ca-439c-a0ef-77b645bca824.metadata','type':'simple','location':'1537767315139/1544093880902'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	deleteMetadata	The service is to delete metadata
formData:	{"metadataFileName":"9a27e465-07ca-439c-a0ef-77b645bca824.metadata","type":"simple","location":"1537767315139/1544093880902"}	Metadatafilename with its location is required along with simple as a type
Response Output(JSON Format)	{"status":1,"response":{"message":"Metadata deleted successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

```
type:adhoc
serviceType:metadata
service:deleteMetadata
formData:{"metadataFileName":"9a27e465-07ca-439c-a0ef-77b645bca824.metadata","type":"simple","location":"1537767315139/1544093880902"}
```

```
{"status":1,"response":{"message":"Metadata deleted successfully"}}
```

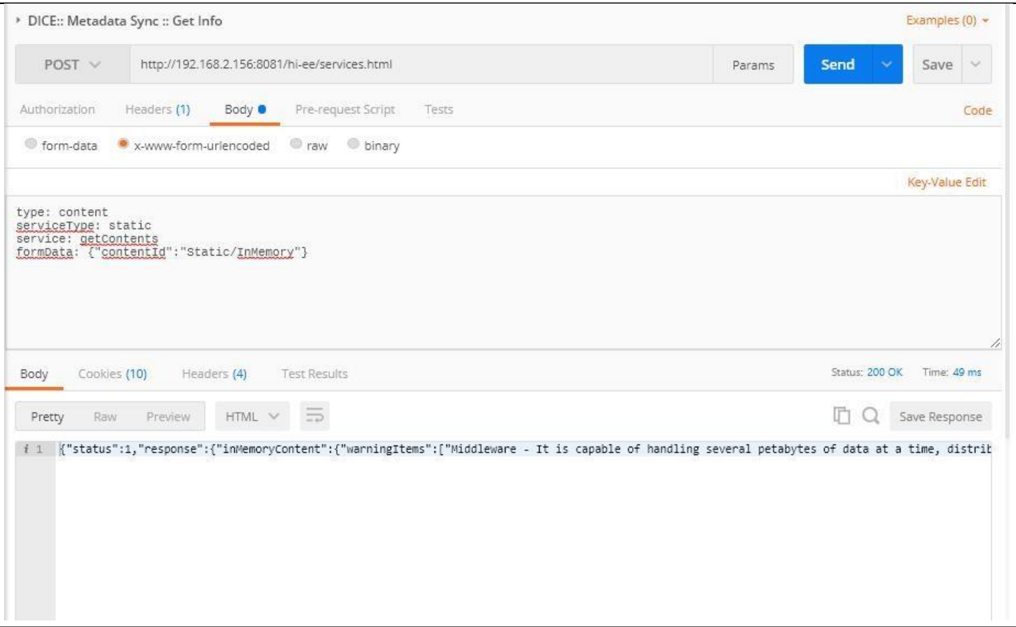
4.4.1.16.2 Cascade- Delete metadata

URL	/services.html	
Description	The user can delete metadata with cascade type which will delete all dependent resources along with metadata file.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=deleteMetadata&formData={'metadataFileName':'9a27e465-07ca-439c-a0ef-77b645bca824.metadata','type':'simple','location':'1537767315139/1544093880902'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	deleteMetadata	The service is to delete metadata
formData:	{"metadataFileName":"1e86d029-396a-45bf-9308-eb6144cacb99.metadata","type":"cascade","location":"1537767315139/1544093880902"}	Metadatafilename with its location is required along with cascade as a type
Response Output(JSON Format)	{"status":1,"response":{"message":"Metadata deleted successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	 Cascade-Delete Metadata POST http://192.168.2.156:8081/hi-ee/services.html Body: x-www-form-urlencoded
	`type:adhoc`
	`serviceType:metadata`
	`service:deleteMetadata`
	`formData:{"metadataFileName":"1e86d029-396a-45bf-9308-eb6144cacb99.metadata","type":"cascade","location":"1537767315139/1544093880902"}`
	Status: 200 OK Time: 7650 ms
	`{"status":1,"response":{"message":"Metadata deleted successfully"}}`

4.4.1.17 DICE :: Metadata Sync :: Get Information

URL	/services.html	
Description	The user will get DICE metadata sync static information.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=content&serviceType=static&service=getContents&formData={'contentId':'Static/InMemory'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	content	type as content
serviceType:	static	serviceType as static

service:	getContents	The service is to getContents
formData:	{"contentId":"Static/InMemory"}	Action for static contents
Response Output(JSON Format)	{"status":1,"response":{"inMemoryContent":{"warningItems":["Middleware - It is capable of handling several petabytes of data at a time, distributed across a cluster of thousands of cooperating physical or virtual servers. It provides distributed task dispatching, scheduling, and basic I/O functionalities, exposed through an application programming interface.","Cache - Caching or persistence are optimisation techniques for (iterative and interactive) Spark computations. They help saving interim partial results so they can be reused in subsequent stages. These interim results as RDDs are thus kept in memory (default) or more solid storages like disk and/or replicated.","Cache Vs Persist - The difference between cache and persist operations is purely syntactic. cache is a synonym of persist or persist(MEMORY_ONLY), i.e. cache is merely persist with the default storage level MEMORY_ONLY.","Warning: Enabling Middleware or Cache/Persist requires a significant Memory. Please Make sure that you have good RAM/Memory"]}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8081/hi-ee/services.html

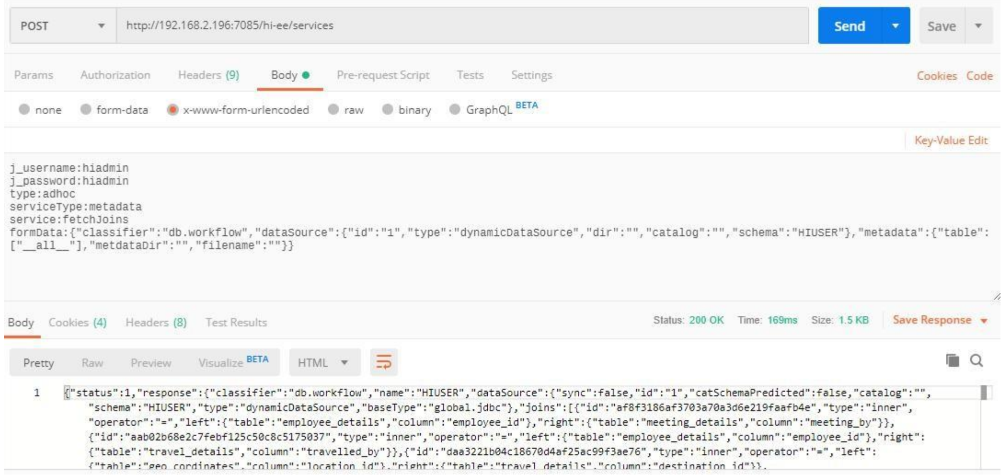
```
type: content
serviceType: static
service: getContents
formData: {"contentId":"Static/InMemory"}
```

Status: 200 OK Time: 49 ms

```
{"status":1,"response":{"inMemoryContent":{"warningItems":["Middleware - It is capable of handling several petabytes of data at a time, distrib
```

4.4.1.18 Fetch metadata Joins

URL	//services	
Description	The user can fetch metadata joins.This service will retrieve all the cachedJoins for all or partial tables which are cached in autoTrigger process that is at the time of Datasource creation.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=fetchJoins&formData={"classifier":"db.workflow","dataSource":{"id":"1","type":"dynamicDataSource","dir":"","catalog":"","schema":"HIUSER"},"metadata":{"table":["_all_"],"metdataDir":"","filename":""}}' http://192.168.2.196:7085/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	fetchJoins	The service is to fetch metadata joins
formData:	{"classifier":"db.workflow","dataSource":{"id":"1","type":"dynamicDataSource","dir":"","catalog":"","schema":"HIUSER"},"metadata":{"table":["__all__"],"metdataDir":"","filename":""}}	Formdata requires the classifier , datasource id,type,directory, catalog,schema and metadata details.Incase to get joins for all tables user need to provide tables value as "_all_"
Response Output(JSON Format)	{"status":1,"response":{"classifier":"db.workflow","name":"HIUSER","dataSource":{"sync":false,"id":"1","catSchemaPredicted":false,"catalog":"","schema":"HIUSER","type":"dynamicDataSource","baseType":"global.jdbc"},"joins":[{"id":"af8f3186af3703a70a3d6e219faafb4e","type":"inner","operator":"=","left":{"table":"employee_details","column":"employee_id"},"right":{"table":"meeting_details","column":"meeting_by"}},{"id":"aab02b68e2c7febf125c50c8c5175037","type":"inner","operator":"=","left":{"table":"employee_details","column":"employee_id"},"right":{"table":"travel_details","column":"travelled_by"}},{"id":"daa3221b04c18670d4af25ac99f3ae76","type":"inner","operator":"=","left":{"table":"geo_cordinates","column":"location_id"},"right":{"table":"travel_details","column":"destination_id"}},{"id":"cdeb5b19799c89335f23ed9b50cc5a22","type":"inner","operator":"=","left":{"table":"geo_cordinates","column":"location_id"},"right":{"table":"travel_details","column":"source_id"}},{"id":"ca21d00c8c87263dedd812f8f74c05b5","type":"inner","operator":"=","left":{"table":"geo_cordinates","column":	

	"location_id"},"right":{"table":"dimdate","column":"dim_id"}}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.Success message along with join details for requested tables or for all tables.
Service Status	200 OK
Screenshot	

POST http://192.168.2.196:7085/hi-ee/services Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data [✓ x-www-form-urlencoded] raw binary GraphQL BETA

Key-Value Edit

```
j_username:hiadmin
j_password:hiadmin
type:adhoc
serviceType:metadata
service:fetchJoins
formData:{"classifier":"db.workflow","dataSource":{"id":"1","type":"dynamicDataSource","dir":"","catalog":"","schema":"HIUSER"},"metadata":{"table":
["__all__"],"metdataDir":"","filename":""}}
```

Body Cookies (4) Headers (8) Test Results Status: 200 OK Time: 169ms Size: 1.5 KB Save Response

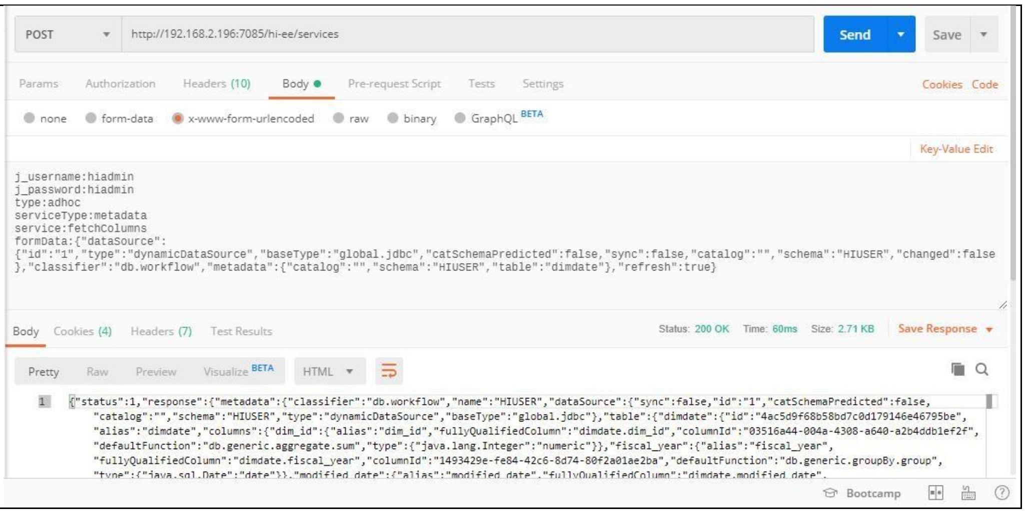
Pretty Raw Preview Visualize BETA HTML

```
{"status":1,"response":{"classifier":"db.workflow","name":"HIUSER","dataSource":{"sync":false,"id":"1","catSchemaPredicted":false,"catalog":"",
   "schema":"HIUSER","type":"dynamicDataSource","baseType":"global.jdbc"},"joins":[{"id":"af8f3186af3703a70a3d6e219faafb4e","type":"inner",
   "operator":"=","left":{"table":"employee_details","column":"employee_id"},"right":{"table":"meeting_details","column":"meeting_by"}},
   {"id":"aab02b68e2c7febf125c50c8c5175037","type":"inner","operator":"=","left":{"table":"employee_details","column":"employee_id"},"right":
   {"table":"travel_details","column":"travelled_by"}},{"id":"daa3221b04c18670d4af25ac99f3ae76","type":"inner","operator":"=","left":
   {"table":"geo_cordinates","column":"location_id"},"right":{"table":"travel_details","column":"destination_id"}},
```

4.4.1.19 Fetch metadata Columns

URL	//services
Description	The user can fetch/get metadata columns for requested tables.This service will retrieve all the cached columns for a particular table which are cached in autoTrigger process that is at the time of datasource creation.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.196:7085/hi-ee/services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=fetchColumns&formData={"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,"sync":false,"catalog":"","schema":"HIUSER","changed":false},"classifier":"db.workflow","metadata":{"catalog":"","schema":"HIUSER","table":"dimdate"},"refresh":true}' http://192.168.2.196:7085/hi-ee/services.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type as adhoc
serviceType:	metadata	serviceType as metadata
service:	fetchColumns fetchColumns	The service is to fetch metadata table columns
formData:	{"dataSource":{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,"sync":false,"catalog":"","schema":"HIUSER","changed":false},"classifier":"db.workflow","metadata":{"catalog":"","schema":"HIUSER","table":"dimdate"},"refresh":true}	For data requires the classifier , datasource id,type,directory, catalog,schema and metadata details.
Response Output(JSON Format)	{"status":1,"response":{"metadata":{"classifier":"db.workflow","name":"HIUSER","dataSource":{"sync":false,"id":"1","catSchemaPredicted":false,"catalog":"","schema":"HIUSER","type":"dynamicDataSource","baseType":"global.jdbc"},"table":{"dimdate":{"id":"4ac5d9f68b58bd7c0d179146e46795be","alias":"dimdate","columns":{"dim_id":{"alias":"dim_id","fullyQualifiedColumn":"dimdate.dim_id","columnId":"03516a44-004a-4308-a640-a2b4ddb1ef2f","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"fiscal_year":{"alias":"fiscal_year","fullyQualifiedColumn":"dimdate.fiscal_year","columnId":"1493429e-fe84-42c6-8d74-80f2a01ae2ba","defaultFunction":"db.generic.groupBy.group","type":{"java.sql.Date":"date"}},"modified_date":{"alias":"modified_date","fullyQualifiedColumn":"dimdate.modified_date","columnId":"5401821a-42a4-4d49-9b4b-d117834b488f","defaultFunction":"db.generic.groupBy.group","type":{"java.sql.Timestamp":"dateTime"}},"date_key":{"alias":"date_key","fullyQualifiedColumn":"dimdate.date_key","columnId":"50959fd2-48ed-4036-be93-003729d449f5","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"day_number":{"alias":"day_number","fullyQualifiedColumn":"dimdate.day_number","columnId":"03dee101-5b65-4a2c-b637-f3acc45450c0","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"fiscal_month_name":{"alias":"fiscal_month_name","fullyQualifiedColumn":"dimdate.fiscal_month_name","columnId":"823a0db2-e4c6-44b6-92c4-582849b723cd","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"fiscal_month_label":{"alias":"fiscal_month_label","fullyQualifiedColumn":"dimdate.fiscal_month_label","columnId":"680d5372-dbe6-48c8-8747-a088b5614384","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"created_date":{"alias":"created_date","fullyQualifiedColumn":"dimdate.created_date","columnId":"0b258dce-983c-4c5f-a380-4da21db23c54","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"created_time":{"alias":"created_time","fullyQualifiedColumn":"dimdate.created_time","columnId":"c5aeee09-8513-469e-9e29-93ed14f63ece","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"rating":{"alias":"rating","fullyQualifiedColumn":"dimdate.rating","columnId":"37a8f0b7-738d-4d1c-8d78-8ab7a476671d","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}}},"name":"dimdate"}}}}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.196:7085/hi-ee/services

Body: x-www-form-urlencoded

```
j_username:hiadmin
j_password:hiadmin
type:adhoc
serviceType:metadata
service:fetchColumns
formData:{"dataSource":
{"id":"1","type":"dynamicDataSource","baseType":"global.jdbc","catSchemaPredicted":false,"sync":false,"catalog":"","schema":"HIUSER","changed":false
},"classifier":"db.workflow","metadata":{"catalog":"","schema":"HIUSER","table":"dimdate"},"refresh":true}
```

Status: 200 OK Time: 60ms Size: 2.71 KB

```
{"status":1,"response":{"metadata":{"classifier":"db.workflow","name":"HIUSER","dataSource":{"sync":false,"id":"1","catSchemaPredicted":false,
    "catalog":"","schema":"HIUSER","type":"dynamicDataSource","baseType":"global.jdbc"},"table":{"dimdate":{"id":"4ac5d9f68b58bd7c0d179146e46795be",
    "alias":"dimdate","columns":{"dim_id":{"alias":"dim_id","fullyQualifiedColumn":"dimdate.dim_id","columnId":"03516a44-004a-4308-a640-a2b4ddb1ef2f",
    "defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"fiscal_year":{"alias":"fiscal_year",
    "fullyQualifiedColumn":"dimdate.fiscal_year","columnId":"1493429e-fe84-42c6-8d74-80f2a01ae2ba","defaultFunction":"db.generic.groupBy.group",
    "type":{"java.sql.Date":"date"}},"modified_date":{"alias":"modified_date","fullyQualifiedColumn":"dimdate.modified_date",
```

4.4.2 Metadata Edit

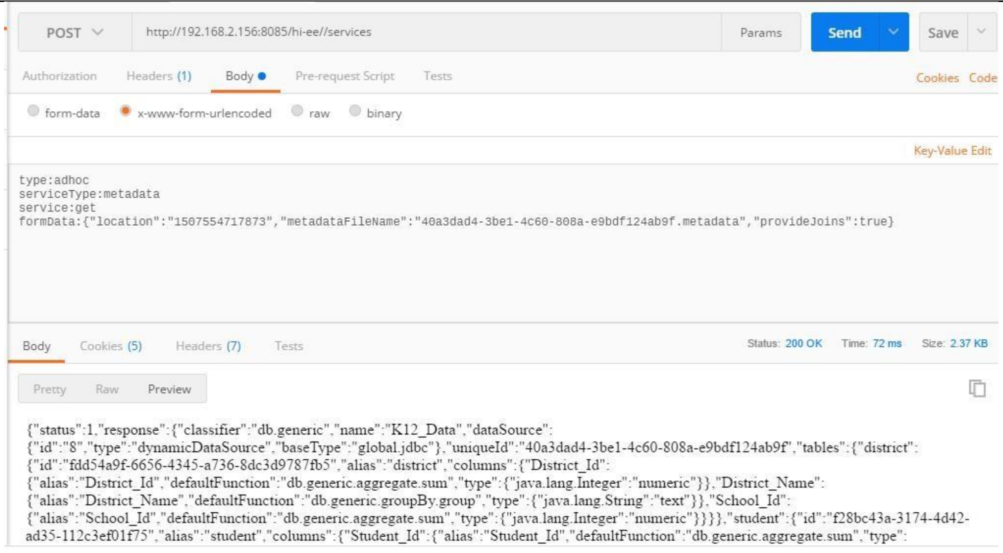
Note : To edit metadata firstly you need to use the below API to edit the particular metadata file , after onwards whatever changes you can do to metata for that refer the above API's of metadata-create section.

For Ex. View , security,Info section of Metadata.

4.4.2.1 Get metadata details:

URL	/services	
Description	It allows user to get the details of existing metdata for edit.It requires the location and the file name to get details.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=get&formData={'location':'1507554717873','metadataFileName':'40a3dad4-3be1-4c60-808a-e9bdf124ab9f.metadata','provideJoins':true}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	get	Service as get

formData:	{"location":"1507554717873","metadataFileName":"40a3dad4-3be1-4c60-808a-e9bdf124ab9f.metadata","provideJoins":true}	formData contains the location and the metadata file name before editing metadata.
Response Output(JSON Format)	{"status":1,"response":{"classifier":"db.generic","name":"K12_Data","dataSource":{"id":"8","type":"dynamicDataSource","baseType":"global.jdbc"},"uniqueId":"40a3dad4-3be1-4c60-808a-e9bdf124ab9f","tables":{"district":{"id":"fdd54a9f-6656-4345-a736-8dc3d9787fb5","alias":"district","columns":{"District_Id":{"alias":"District_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"District_Name":{"alias":"District_Name","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"School_Id":{"alias":"School_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}}}},"student":{"id":"f28bc43a-3174-4d42-ad35-112c3ef01f75","alias":"student","columns":{"Student_Id":{"alias":"Student_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"Student_Name":{"alias":"Student_Name","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"School_Name":{"alias":"School_Name","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Year":{"alias":"Year","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Gender":{"alias":"Gender","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Ethnicity":{"alias":"Ethnicity","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Mother_Tongue":{"alias":"Mother_Tongue","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Grade":{"alias":"Grade","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"Program_Type":{"alias":"Program_Type","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"Teacher_Name":{"alias":"Teacher_Name","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"School_Id":{"alias":"School_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}}}}},"sets":[["student"],["district"]],"joins":[],"metadataName":"TestMetadata","metadataDir":"New Folder"}}	
Description of Response Output:	The response of the API is , it returns the all details of existing metadata like its name , id used security , schema,tables along with column names. **classifier** : name of classifier **name**: name of datasource **dataSource**: datsource array with if , type,baseType of datasource. **metadataName** : Name of metadata file **metadataDir** : Name of metadataDir **tables** : Array with all column details like(id,alias name,default applied function,type of column etc)	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee//services
	Body: x-www-form-urlencoded
	`type:adhoc`
	`serviceType:metadata`
	`service:get`
	`formData:{"location":"1507554717873","metadataFileName":"40a3dad4-3be1-4c60-808a-e9bdf124ab9f.metadata","provideJoins":true}`
	Status: 200 OK Time: 72 ms Size: 2.37 KB
	`{"status":1,"response":{"classifier":"db.generic","name":"K12_Data","dataSource":{"id":"8","type":"dynamicDataSource","baseType":"global.jdbc"},"uniqueId":"40a3dad4-3be1-4c60-808a-e9bdf124ab9f","tables":{"district":{"id":"fdd54a9f-6656-4345-a736-8dc3d9787fb5","alias":"district","columns":{"District_Id":{"alias":"District_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}},"District_Name":{"alias":"District_Name","defaultFunction":"db.generic.groupBy.group","type":{"java.lang.String":"text"}},"School_Id":{"alias":"School_Id","defaultFunction":"db.generic.aggregate.sum","type":{"java.lang.Integer":"numeric"}}}},"student":{"id":"f28bc43a-3174-4d42-ad35-112c3ef01f75","alias":"student","columns":{"Student_Id":{"alias":"Student_Id","defaultFunction":"db.generic.aggregate.sum","type":`

4.4.2.2 Metadata SaveAs

4.5 Report Operations

4.5.1 Create Report

http://192.168.2.184:8085/hi-ee/adhoc/report-create.html

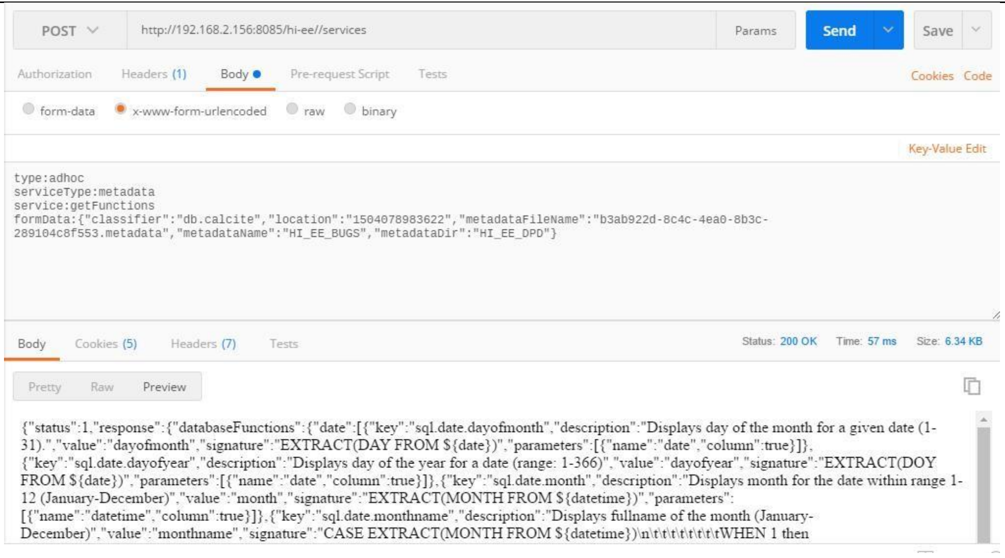
4.5.1.1 Get Metadata :

4.5.1.2 Get Metadata DB Functions:

URL	/services
Description	It allows user to see which database functions are available for selected metadata datasource. Database functions may differ according to database driver used.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST,GET**
Example	**Access through browser :**

	http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=metadata&service=getFunctions&formData={'classifier':'db.generic','location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type as adhoc
serviceType:	metadata	Service type as metadata
service:	getFunctions	Service as getFunctions
formData:	{"classifier":"db.generic","location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder"}	formData contains the all metadata info like location,filename etc.
Response Output(JSON Format)	{"status":1,"response":{"databaseFunctions":{"date":[{"key":"sql.date.year","description":"Displays the year from a given date.","value":"year","signature":"YEAR(${date})","parameters":[{"name":"date","column":true}]},{"key":"sql.date.day","description":"The DAY function returns the day part of a value. The result of the function is an integer between 1 and 31","value":"day","signature":"DAY(${date})","parameters":[{"name":"date","column":true}]},{"key":"sql.date.month","description":"Displays month from the date within range 1-12 (January-December)","value":"month","signature":"month(${datetime})","parameters":[{"name":"datetime","column":true}]},{"key":"sql.date.monthyear","description":"Displays month and year in (month-year) format ","value":"month-year","signature":"CAST(month(${column}) AS CHAR(20))|| '-' ||CAST(YEAR(${column}) AS CHAR(20))","parameters":[{"name":"column","column":true,"defaultValue":"0"}]},{"key":"sql.date.quarter","description":"Displays quarter of the year for a date (1-4)","value":"quarter","signature":"CASE MONTH(${column})\n\t\t\t\t\t\t\t\tWHEN < 4 THEN 1\n\t\t\t\t\t\t\t\tWHEN BETWEEN 4 AND 6 then 2\n\t\t\t\t\t\t\t\tWHEN BETWEEN 7 AND 9 then 3\n\t\t\t\t\t\t\t\tWHEN BETWEEN 10 AND 12 then 4\n\t\t\t\t\t\t\tEND\t\n\t\t\t\t\t\t\t\t","parameters":[{"name":"column","column":true,"defaultValue":"0"}]},{"key":"sql.date.monthname","description":"Displays month name for a date","value":"monthname","signature":"CASE MONTH(${column})\n\t\t\t\t\t\t\t\tWHEN 1 then 'January'\n\t\t\t\t\t\t\t\tWHEN 2 then 'February' \n\t\t\t\t\t\t\t\tWHEN 3 then 'March'\n\t\t\t\t\t\t\t\tWHEN 4 then 'April'\n\t\t\t\t\t\t\t\tWHEN 5 then 'May'\n\t\t\t\t\t\t\t\tWHEN 6 then 'June'\n\t\t\t\t\t\t\t\tWHEN 7 then 'July'\n\t\t\t\t\t\t\t\tWHEN 8 then 'August'\n\t\t\t\t\t\t\t\tWHEN 9 then 'September'\n\t\t\t\t\t\t\t\tWHEN 10 then 'October'\n\t\t\t\t\t\t\t\tWHEN 11 then 'November'\n\t\t\t\t\t\t\t\tWHEN 12 then 'December'\n\n\t\t\t\t\t\t\tEND\t\n\t\t\t\t\t\t\t\t","parameters":[{"name":"column","column":true,"defaultValue":"0"}]},{"key":"sql.text.date","description":"Extracts the date from the date and time value","value":"date","signature":"DATE(${column})","parameters":[{"name":"column","column":true}]}],"dateTime":[{"key":"sql.dateTime.hour","description":"Displays the hour portion of a time(24).","value":"hour","signature":"HOUR(${time})","parameters":[{"name":"time","column":true}]},{"key":"sql.dateTime.minute","description":"Displays a minute from a	

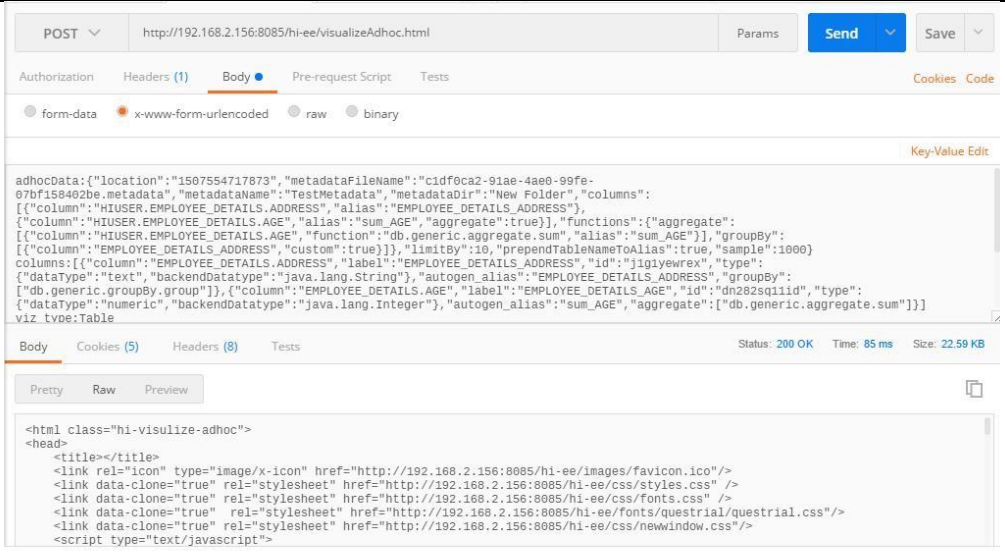
	time.","value":"minute","signature":"MINUTE(${datetime})","parameters":[{"name":"datetime","column":true}]},{"key":"sql.dateTime.currentdate","description":"The CURRENT_DATE function returns the current date.","value":"currentdate","signature":" (VALUES CURRENT_DATE)","parameters":[]},{"key":"sql.dateTime.currenttime","description":"The CURRENT_TIME function returns the current time.","value":"currenttime","signature":" (VALUES CURRENT_TIME) ","parameters":[]},{"key":"sql.dateTime.currenttimestamp","description":"The CURRENT_TIMESTAMP function returns the current timestamp;","value":"current","signature":" (VALUES CURRENT_TIMESTAMP) ","parameters":[]}],"numeric":[{"key":"sql.numeric.bigint","description":"BIGINT function returns a 64-bit integer representation of a number or character string in the form of an integer constant.","value":"bigint","signature":"BIGINT(${decimal}) ","parameters":[{"name":"decimal","column":true,"defaultValue":"0"}]},{"key":"sql.numeric.cast","description":"The CAST function converts a value from one data type to another and provides a data type to a dynamic parameter (?) or a NULL value.","value":"cast","signature":"CAST(${text} AS ${format})","parameters":[{"name":"text","column":true,"defaultValue":"0"},{"name":"format"}]},{"key":"sql.numeric.ceiling","description":"Displays the smallest integer value not less than the number specified.","value":"ceiling","signature":"CEIL(${decimal})","parameters":[{"name":"decimal","column":true,"defaultValue":"0"}]},{"key":"sql.numeric.floor","description":"Displays the largest value not greater than a number specified.","value":"floor","signature":"FLOOR(${decimal})","parameters":[{"name":"decimal","column":true,"defaultValue":"0"}]},{"key":"sql.numeric.mod","description":"Displays the remainder of a number divided by another number.","value":"mod","signature":"MOD(${number},${divisor})","parameters":[{"name":"number","column":true,"defaultValue":"0"},{"name":"divisor","defaultValue":"10"}]},{"key":"sql.numeric.radians","description":"Converts the value of a number from degrees to radians.","value":"radians","signature":"RADIANS(${number})","parameters":[{"name":"number","column":true}]},{"key":"sql.numeric.sqrt","description":"Displays the square root of a non-negative number.","value":"sqrt","signature":"SQRT(${number})","parameters":[{"name":"number","column":true}]},{"key":"sql.number.avg","description":"Displays the average value of a numeric column.","value":"avg","signature":"avg(${distAll} ${column})","parameters":[{"name":"distAll","column":false,"defaultValue":"ALL"},{"name":"column","column":true}]}],"text":[{"key":"sql.text.lower","description":"Converts all characters in the specified string to lowercase.","value":"lower","signature":"LOWER(${text})","parameters":[{"name":"text","column":true}]},{"key":"sql.text.upper","description":"It converts all the characters in a string to uppercase characters.","value":"upper","signature":"UPPER(${text})","parameters":[{"name":"text","column":true}]},{"key":"sql.text.trim","description":"It removes all specified characters either from the beginning or the ending of the string.\n ","value":"trim","signature":"TRIM(${text})","parameters":[{"name":"text","column":true}]},{"key":"sql.text.ltrim","description":"It removes all space characters from the left-side of a string.","value":"ltrim","signature":"LTRIM(${text})","parameters":[{"name":"text","column":true}]}, {"key":"sql.text.locate","description":"The LOCATE function is used to search for a string within another string. If the desired string is found, LOCATE returns the index at which it is found. If the desired string is not found, LOCATE returns 0.","value":"locate","signature":"LOCATE(${text1},${text2})","parameters":[{"name":"text1","column":true},{"name":"text2","column":true}]},{"key":"sql.text.substr","description":"Displays the substring in a string from startPosition to the lengthof substring specified.","value":"substr","signature":"SUBSTR(${string},${startPosition},${lengthOfString})","parameters":[{"name":"string","column":true},{"name":"startPosition","defaultValue":"1"},{"name":"lengthOfString","defaultValue":"3"}]},{"key":"sql.text.concat","description":"Helps in joining two string","value":"concat","signature":" (${text1} || ${text2}) ","parameters":[{"name":"text1","column":true},{"name":"text2","column":true}]}]},"functions":{"db.generic.aggregate.avg":"avg","db.generic.aggregate.count":"count","db.generic.aggregate.distinct":"distinct","db.generic.aggregate.max":"max","db.generic.aggregate.min":"min","db.generic.aggregate.sum":"sum","db.generic.groupBy.group":"group by","db.generic.orderBy.order":"order by"}}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with databaseFunctions array having all database functions related to database driver.
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee//services Params Send Save Authorization Headers (1) Body Pre-request Script Tests Cookies Code form-data x-www-form-urlencoded raw binary Key-Value Edit type:adhoc serviceType:metadata service:getFunctions formData:{"classifier":"db.calcite","location":"1504078983622","metadataFileName":"b3ab922d-8c4c-4ea0-8b3c-289104c8f553.metadata","metadataName":"HI_EE_BUGS","metadataDir":"HI_EE_DPD"} Body Cookies (5) Headers (7) Tests Pretty Raw Preview Status: 200 OK Time: 57 ms Size: 6.34 KB {"status":1,"response":{"databaseFunctions":{"date":[{"key":"sql.date.dayofmonth","description":"Displays day of the month for a given date (1-31).","value":"dayofmonth","signature":"EXTRACT(DAY FROM ${date})","parameters":[{"name":"date","column":true}]},{"key":"sql.date.dayofyear","description":"Displays day of the year for a date (range: 1-366)","value":"dayofyear","signature":"EXTRACT(DOY FROM ${date})","parameters":[{"name":"date","column":true}]},{"key":"sql.date.month","description":"Displays month for the date within range 1-12 (January-December)","value":"month","signature":"EXTRACT(MONTH FROM ${datetime})","parameters":[{"name":"datetime","column":true}]},{"key":"sql.date.monthname","description":"Displays fullname of the month (January-December)","value":"monthname","signature":"CASE EXTRACT(MONTH FROM ${datetime})\n\t\t\t\t\t\t\tWHEN 1 then

4.5.1.3 Generate table:

URL	visualizeAdhoc.html
Description	It allows user to generate table report using adhoc report. For creating table report we need to pass the adhoc data information,columns info, visualize type etc. mentioned in HTTP Request Key-value section.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&adhocData={'location':'1507554717873','metadataFileName':'c1df0ca2-91ae-4ae0-99fe-07bf158402be.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':10,'prependTableNameToAlias':tr

	ue,'sample':1000}&columns=[{'column':'EMPLOYEE_DETAILS.ADDRESS','label':'EMPLOYEE_DETAILS_ADDRESS','id':'j1g1yewrex','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'EMPLOYEE_DETAILS_ADDRESS','groupBy':['db.generic.groupBy.group']},{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'dn282sq11id','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_AGE','aggregate':['db.generic.aggregate.sum']}]&viz_type=Table&settings={'script':null,'vizscriptsEditMultipleMode':false}&database=HIUSER&scripts=[]&customScripts=[]&styles="&customStyles="" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
adhocData:	{"location":"1507554717873","metadataFileName":"c1df0ca2-91ae-4ae0-99fe-07bf158402be.metadata","metadataName":"Test Metadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":10,"prependTableNameToAlias":true,"sample":1000}	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.
columns:	[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"j1g1yewrex","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"dn282sq11id","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).
viz_type:	Table	Type of visualization
settings:	{"script":null,"vizscriptsEditMultipleMode":false}	Settings related to applied scripts if any.
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	[]	Applied Customscripts if any
styles:	“”	Applied styles if any

customStyles:	“”	Applied Customstyles if any
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html

form-data [✓ x-www-form-urlencoded] raw binary

```
adhocData:{"location":"1507554717873","metadataFileName":"c1df0ca2-91ae-4ae0-99fe-
07bf158402be.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":10,"prependTableNameToAlias":true,"sample":1000}
columns:[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"j1g1yewrex","type":
{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":
["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"dn282sq11id","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]
viz_type:Table
```

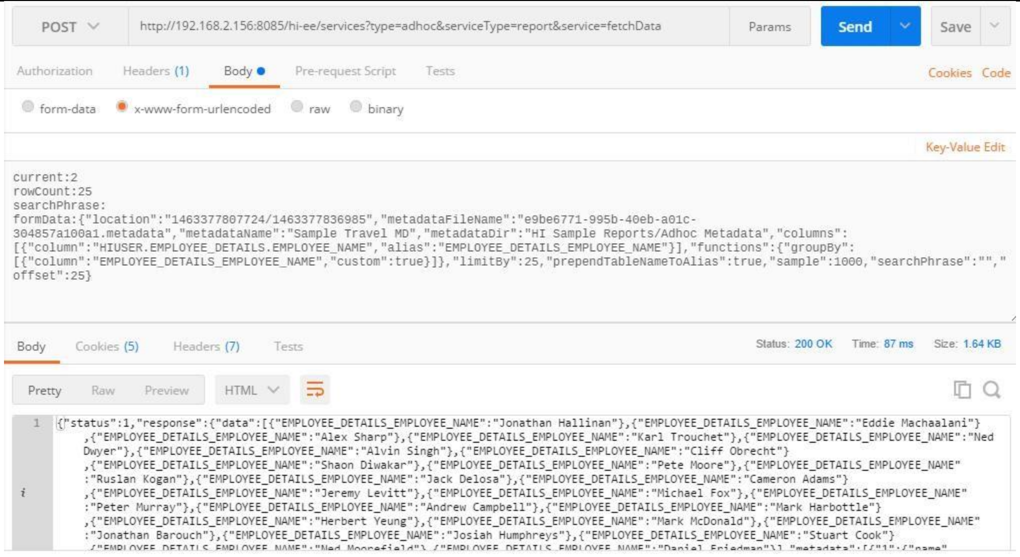
Status: 200 OK Time: 85 ms Size: 22.59 KB

```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
    <script type="text/javascript">
```

4.5.1.4 Refresh table/Change show entries/Change Report filter/Next-Prev page

URL	services?type=adhoc&serviceType=report&service=fetchData
Description	It allows user to refresh / change show entries or change report filter or next prev operation on generated table.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST,GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services?type=adhoc&serviceType=report&service=fetchData

	Access through Curl command : curl --data "j_username=hiadmin&j_password=hiadmin&current=2&rowCount=25&searchPhrase=&type=adhoc&serviceType=report&service=fetchData&formData={'location':'1463377807724/1463377836985','metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','metadataName':'Sample Travel MD','metadataDir':'HI Sample Reports/Adhoc Metadata','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME','alias':'EMPLOYEE_DETAILS_EMPLOYEE_NAME'}],'functions':{'groupBy':[{'column':'EMPLOYEE_DETAILS_EMPLOYEE_NAME','custom':true}]},'limitBy':25,'prependTableNameToAlias':true,'sample':1000,'searchPhrase':",'offset':25}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
current:	2	Current page denotes the number of the current page.
rowCount:	25	rowCount is the total number of rows which you want to generate for.
searchPhrase:		searchPhrase if any
formData:	{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME","alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME"}],"functions":{"groupBy":[{"column":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","custom":true}]},"limitBy":25,"prependTableNameToAlias":true,"sample":1000,"searchPhrase":"","offset":25}	formData contains the metadata info like location,filename and the report columns etc.
Response Output(JSON Format)	{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jonathan Hallinan"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Eddie Machaalani"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alex Sharp"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Karl Trouchet"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ned Dwyer"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alvin Singh"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cliff Obrecht"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Shaon Diwakar"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Pete Moore"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ruslan Kogan"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jack Delosa"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cameron Adams"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jeremy Levitt"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Michael Fox"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Peter Murray"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Andrew Campbell"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Mark Harbottle"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Herbert Yeung"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Mark McDonald"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jonathan	

	Barouch"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Josiah Humphreys"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Stuart Cook"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ned Moorefield"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Daniel Friedman"}],"metadata":[{"1":{"name":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","type":"text"}},{"rows":24}]},"lastModified":1509426687000}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with data array having all retrieved rows of the table and the metadata array with type of column name with rows count . Lastmodified date.
Service Status	200 OK
Screenshot	

POST http://192.168.2.156:8085/hi-ee/services?type=adhoc&serviceType=report&service=fetchData

Body: x-www-form-urlencoded

```
current:2
rowCount:25
searchPhrase:
formData:{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-
304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME","alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME"}],"functions":{"groupBy":
[{"column":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","custom":true}]},"limitBy":25,"prependTableNameToAlias":true,"sample":1000,"searchPhrase":"","
offset":25}
```

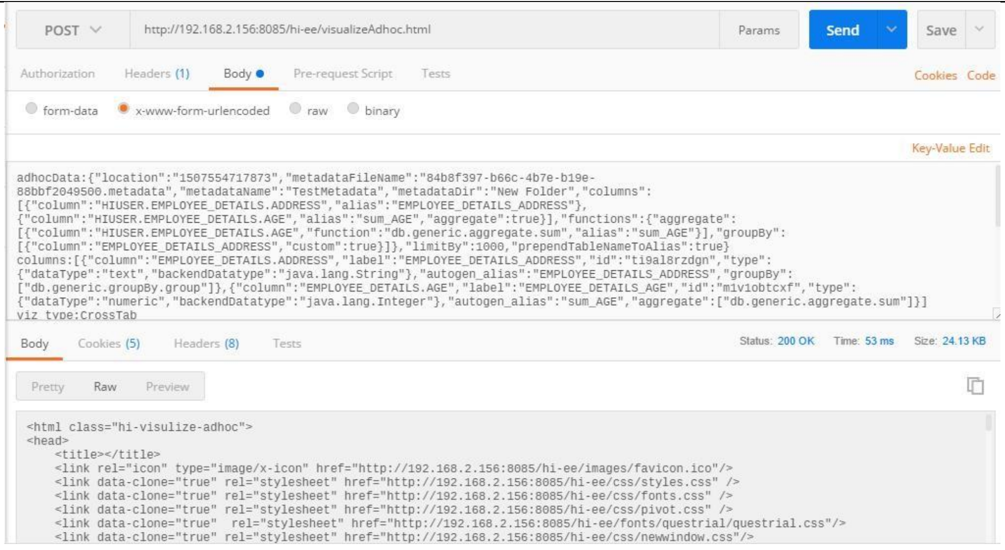
Status: 200 OK Time: 87 ms Size: 1.64 KB

```
{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jonathan Hallinan"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Eddie Machaalani"}
    ,{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alex Sharp"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Karl Trouchet"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ned
    Dwyer"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alvin Singh"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cliff Obrecht"}
    ,{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Shaon Diwakar"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Pete Moore"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME"
    :"Ruslan Kogan"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jack Delosa"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cameron Adams"}
    ,{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Jeremy Levitt"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Michael Fox"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME"
    :"Peter Murray"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Andrew Campbell"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Mark Harbottle"}
    ,{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Herbert Yeung"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Mark McDonald"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME"
    :"Jonathan Barouch"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Josiah Humphreys"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Stuart Cook"}
    ,{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ned Moorefield"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Daniel Friedman"}],"metadata":[{"1":{"name"
```

4.5.1.5 Generate CrossTab:

URL	visualizeAdhoc.html
Description	It allows user to generate cross tab report using adhoc report. For creating cross tab report we need to pass the adhoc data information,columns info, visualize type etc. mentioned in HTTP Request Key-value section.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :**

	http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&adhocData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':1000,'prependTableNameToAlias':true}&columns=[{'column':'EMPLOYEE_DETAILS.ADDRESS','label':'EMPLOYEE_DETAILS_ADDRESS','id':'ti9al8rzdgn','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'EMPLOYEE_DETAILS_ADDRESS','groupBy':['db.generic.groupBy.group']},{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'m1v1obtcxf','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_AGE','aggregate':['db.generic.aggregate.sum']}]&viz_type=CrossTab&settings={"script":null,"vizscriptsEditMultipleMode":false,"crossTabRows":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"m1v1obtcxf","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}],"crossTabCols":[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ti9al8rzdgn","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]}],"crossTabVals":[]}&scripts=[]&customScripts=[]&styles="&customStyles="" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
adhocData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.
columns:	[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ti9al8rzdgn","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"m1v1obtcxf","type":{"dataType":"numeric","backendDatatyp	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).

	e":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]	
viz_type:	CrossTab	Type of visualization
settings:	{"script":null,"vizscriptsEditMultipleMode":false,"crossTabRows":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"m1v1obtcxf","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}],"crossTabCols":[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ti9al8rzdgn","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]}],"crossTabVals":[]}	Settings related to applied scripts if any.
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	[]	Applied Customscripts if any
styles:	“”	Applied styles if any
customStyles:	“”	Applied Customstyles if any
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html

Body: x-www-form-urlencoded

```
adhocData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-
88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}
columns:[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ti9al8rzdgn","type":
{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":
["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"m1v1obtcxf","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]
viz_type:CrossTab
```

Status: 200 OK Time: 53 ms Size: 24.13 KB

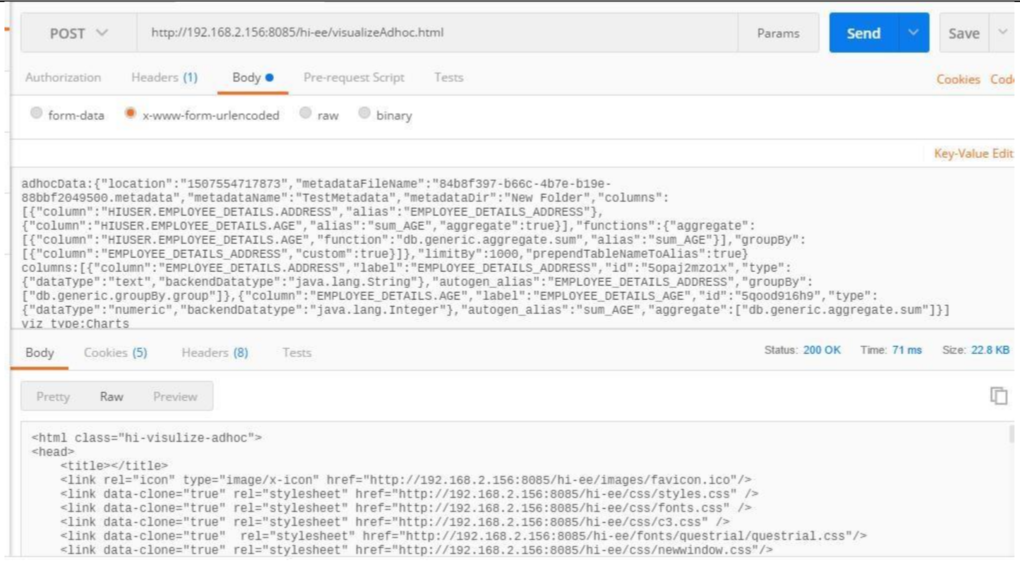
```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/pivot.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
```

4.5.1.6 Generate Charts:

URL	visualizeAdhoc.html
Description	It allows user to generate chart report using adhoc report. For creating chart report we need to pass the adhoc data information,columns info, visualize type etc. mentioned in HTTP Request Key-value section. Note : While generating charts there are different types of charts having different chart groups to know the group of chart refer chartTypes.txt
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST,GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&adhocData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':1000,'prependTableNameToAlias':true}&columns=[{'column':'EMPLOYEE_DETAILS.ADDRESS','label':'EMPLOYEE_DETAILS_ADDRESS','id':'5opaj2mzo1x','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'EMPLOYEE_DETAILS_ADDRESS','groupBy':['db.generic.groupBy.group']},{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'5qood916h9','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_AGE','aggregate':['db.generic.aggregate.sum']}]&viz_type=Charts&settings={'script':null,'vizscriptsEditMultipleMode':false,'type':'AreaChart','vizType':'Charts','chartGroup':'c3Axis'}&scripts=[]&customScripts=[]&styles="&customStyles="" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v

HTTP Request Key	**HTTP Request Value**	**Description**

adhocData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"Test Metadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.
columns:	[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"5opaj2mzo1x","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"5qood916h9","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).
viz_type:	Charts	Type of visualization
settings:	{"script":null,"vizscriptsEditMultipleMode":false,"type":"AreaChart","vizType":"Charts","chartGroup":"c3Axis"}	Settings related to applied scripts and the chart type with group of chart. vizType is the type of visualization. *Note: As we used AreaChart which I shaving chartGroup as c3Axis , same like that if you select another type of chart accordingly chartGroup will get change.to know the chartgroup of chart Refer document chartType*
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	[]	Applied Customscripts if any
styles:	“”	Applied styles if any
customStyles:	“”	Applied Customstyles if any
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	

Screenshot	

```
adhocData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-
88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}
columns:[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"5opaj2mzo1x","type":
{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":
["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"5qood916h9","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]
viz type:Charts
```

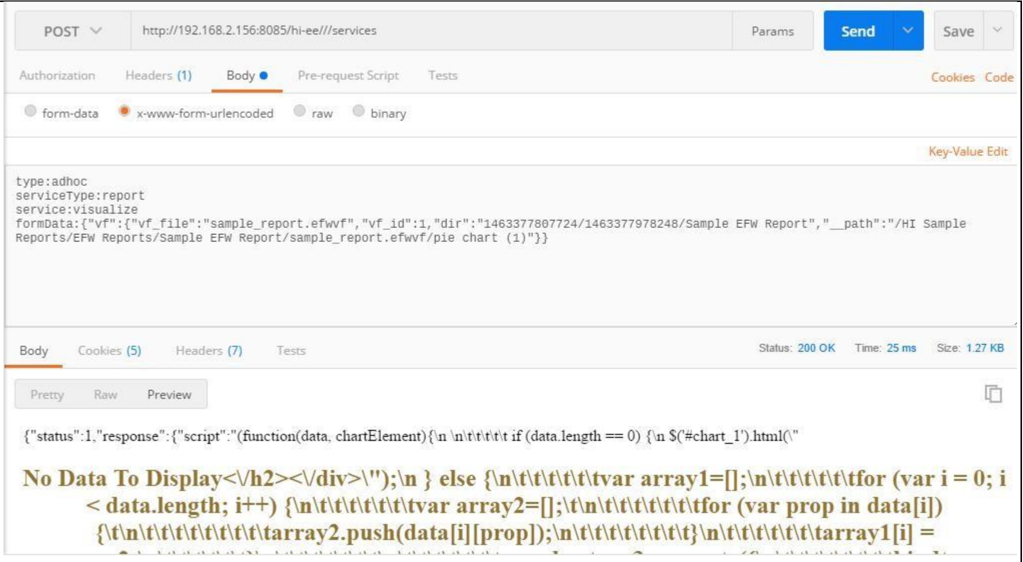
```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/c3.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
```

4.5.1.7 Generate VF Report:

4.5.1.7.1 SELECT VF REPORT:

URL	/services
Description	It allows user to select the VF file for VF report creation. Requires the .vf file name and directory location.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=visualize&formData={'vf':{'vf_file':'sample_report.efwvf','vf_id':1,'dir':'1463377807724/1463377978248/Sample EFW Report','_path':'/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)'}}" http://192.168.2.156:8085/hi-ee/services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	visualize	service as visualize
formData:	{"vf":{"vf_file":"sample_report.efwvf","vf_id":1,"dir":"1463377807724/1463377978248/Sample EFW Report","_path":"/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)"}}	formData contains vf details like vf file name , vf id, directoy of vf file and the path of file.
Response Output(JSON Format)	{ "status":1,"response":{"script":"(function(data, chartElement){\n \n\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\"<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display<\/h2><\/div>\");\n } else {\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor (var prop in data[i]) {\t\n\t\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\tarray1[i] = array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\tbindto: '#chart_1',\n\t\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\ttype: 'pie'\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\ttooltip: {\n\t\t\t\t\t\t\t\t show: true\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t});\t\n\t\t\t\t}\n\t\t\t\n })([\"${data}\"])"}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with script function.	
Service Status	200 OK	

Screenshot	

```
type:adhoc
serviceType:report
service:visualize
formData:{"vf":{"vf_file":"sample_report.efwvf","vf_id":1,"dir":"1463377807724/1463377978248/Sample EFW Report","__path":"/HI Sample
Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)"}}
```

```
{"status":1,"response":{"script":"(function(data, chartElement){\n \n\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\"

No Data To Display<\/h2><\/div>\");\n } else {\n\t\t\t\t\tvar array1=[];\n\t\t\t\t\tfor (var i = 0; i
< data.length; i++) {\n\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\tfor (var prop in data[i])
{\t\n\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\tarray1[i] =
```

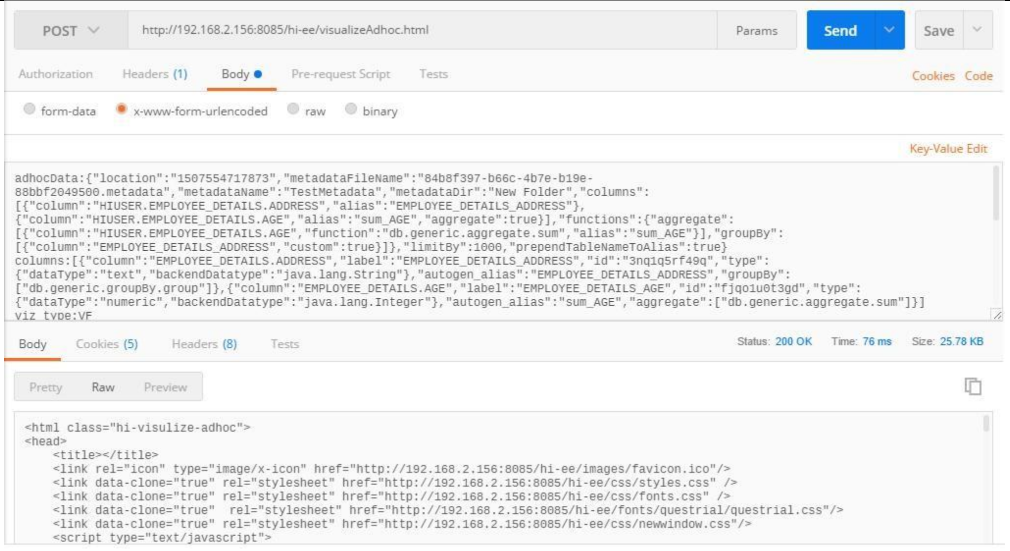
4.5.1.7.2 GENERATE VF REPORT:

URL	visualizeAdhoc.html
Description	It allows user to generate adhoc report using VF report file. For creating chart report we need to pass the adhoc data information,columns info, visualize type etc. mentioned in HTTP Request Key-value section.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST,GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&adhocData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'li

	mitBy':1000,'prependTableNameToAlias':true}&columns=[{'column':'EMPLOYEE_DETAILS.ADDRESS','label':'EMPLOYEE_DETAILS_ADDRESS','id':'3nq1q5rf49q','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'EMPLOYEE_DETAILS_ADDRESS','groupBy':['db.generic.groupBy.group']},{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'fjqo1u0t3gd','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_AGE','aggregate':['db.generic.aggregate.sum']}]&viz_type=VF&settings={'vf_file':'sample_report.efwvf','vf_id':1,'dir':'1463377807724/1463377978248/Sample EFW Report','_path':'/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)','script':'(function(data, chartElement){\n \n\t\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\'<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display</h2></div>\');\n } else {\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor (var prop in data[i]) {\t\n\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\tarray1[i] = array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\tbindto: '#chart_1',\n\t\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\ttype: 'pie'\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\ttooltip: {\n\t\t\t\t\t\t\t\t show: true\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t});\t\n\t\t\t\t\t}\n\t\t\t\t\n })(window.data)','type':'VF'}&scripts=[]&customScripts=[]&styles=''&customStyles=''" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
adhocData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"Test Metadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.
columns:	[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"3nq1q5rf49q","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"fjqo1u0t3gd","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).
viz_type:	VF	Type of visualization

settings:	{"vf_file":"sample_report.efwvf","vf_id":1,"dir": "1463377807724/1463377978248/Sample EFW Report","_path":"/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)","script":"(function(data, chartElement){\n \n\t\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\"<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display</h2></div>\");\n } else {\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor (var prop in data[i]) {\t\n\t\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\tarray1[i] = array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\t\tbindto: '#chart_1',\n\t\t\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\t\ttype: 'pie'\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\ttooltip: {\n\t\t\t\t\t\t\t\t show: true\n\t\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\t});\t\n\t\t\t\t\t}\n\t\t\t\n })(window.data)","type":"VF"}	Settings related to vf file and related details .vizType is the type of visualization.
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	[]	Applied Customscripts if any
styles:	“”	Applied styles if any
customStyles:	“”	Applied Customstyles if any
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html

```
adhocData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-
88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}
columns:[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"3nq1q5rf49q","type":
{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":
["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"fjqo1u0t3gd","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]
viz_type:VF
```

Status: 200 OK Time: 76 ms Size: 25.78 KB

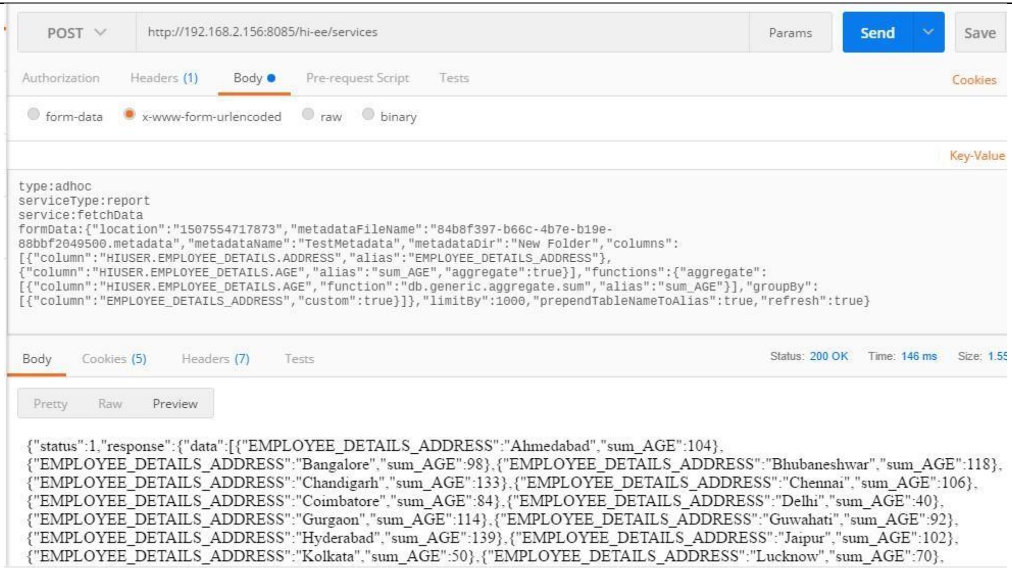
```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
    <script type="text/javascript">
```

4.5.1.8 Fetch Adhoc Report Data:

Note : To get any type of report(table,chart,crossTab,VF etc) data we need to use this API. According to requirement formData will get change which is nothing but the filename,location,columns,aggregate functions,groupBy,limit etc.

URL	services
Description	It allows user to get the any type of adhoc report data. Note : Here we are taking example to get VF report data.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=fetchData&formData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'E

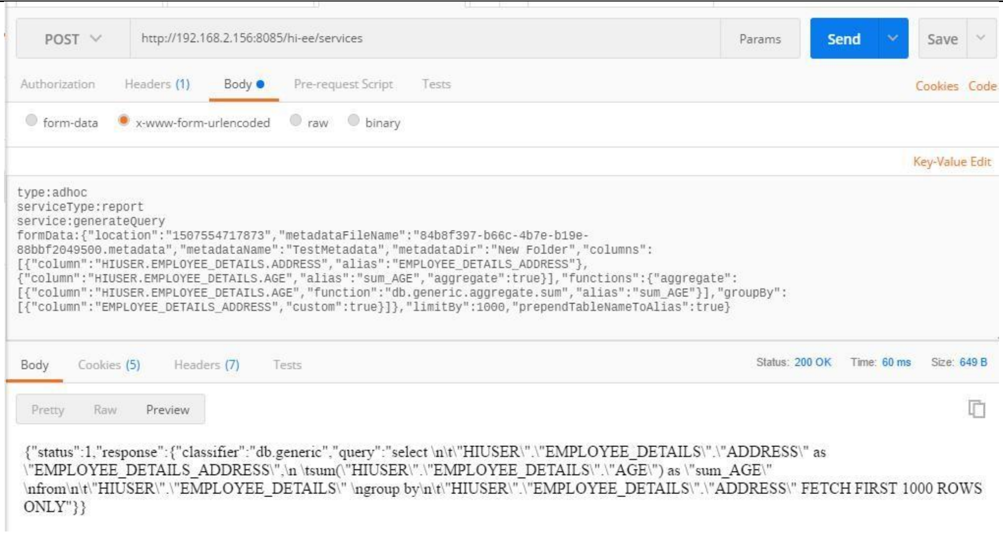
	MPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':1000,'prependTableNameToAlias':true,'refresh':true}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	fetchData	service as fetchData
formData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true,"refresh":true}	formData contains report name, location,columns,aggregate functions,limit,groupBy etc. Details.
Response Output(JSON Format)	{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_ADDRESS":"Ahmedabad","sum_AGE":104},{"EMPLOYEE_DETAILS_ADDRESS":"Bangalore","sum_AGE":98},{"EMPLOYEE_DETAILS_ADDRESS":"Bhubaneshwar","sum_AGE":118},{"EMPLOYEE_DETAILS_ADDRESS":"Chandigarh","sum_AGE":133},{"EMPLOYEE_DETAILS_ADDRESS":"Chennai","sum_AGE":106},{"EMPLOYEE_DETAILS_ADDRESS":"Coimbatore","sum_AGE":84},{"EMPLOYEE_DETAILS_ADDRESS":"Delhi","sum_AGE":40},{"EMPLOYEE_DETAILS_ADDRESS":"Gurgaon","sum_AGE":114},{"EMPLOYEE_DETAILS_ADDRESS":"Guwahati","sum_AGE":92},{"EMPLOYEE_DETAILS_ADDRESS":"Hyderabad","sum_AGE":139},{"EMPLOYEE_DETAILS_ADDRESS":"Jaipur","sum_AGE":102},{"EMPLOYEE_DETAILS_ADDRESS":"Kolkata","sum_AGE":50},{"EMPLOYEE_DETAILS_ADDRESS":"Lucknow","sum_AGE":70},{"EMPLOYEE_DETAILS_ADDRESS":"Mumbai","sum_AGE":97},{"EMPLOYEE_DETAILS_ADDRESS":"Mysore","sum_AGE":65},{"EMPLOYEE_DETAILS_ADDRESS":"Nagpur","sum_AGE":81},{"EMPLOYEE_DETAILS_ADDRESS":"New Delhi","sum_AGE":75},{"EMPLOYEE_DETAILS_ADDRESS":"Noida","sum_AGE":76},{"EMPLOYEE_DETAILS_ADDRESS":"Pune","sum_AGE":75},{"EMPLOYEE_DETAILS_ADDRESS":"Ranchi","sum_AGE":64},{"EMPLOYEE_DETAILS_ADDRESS":"Thiruvananthapuram","sum_AGE":71}],"metadata":[{"1":{"name":"EMPLOYEE_DETAILS_ADDRESS","type":"text"},"2":{"name":"sum_AGE","type":"numeric"}},{"rows":21}]}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with VF report data with metdata array and the total number of rows.	
Service Status	200 OK	

Screenshot	 POST http://192.168.2.156:8085/hi-ee/services Params Send Save
	Authorization Headers (1) Body Pre-request Script Tests Cookies
	form-data [✓ x-www-form-urlencoded] raw binary Key-Value
	`type:adhoc`
	`serviceType:report`
	`service:fetchData`
	`formData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true,"refresh":true}`
	Body Cookies (5) Headers (7) Tests Status: 200 OK Time: 146 ms Size: 1.55
	Pretty Raw Preview
	`{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_ADDRESS":"Ahmedabad","sum_AGE":104},{"EMPLOYEE_DETAILS_ADDRESS":"Bangalore","sum_AGE":98},{"EMPLOYEE_DETAILS_ADDRESS":"Bhubaneshwar","sum_AGE":118},{"EMPLOYEE_DETAILS_ADDRESS":"Chandigarh","sum_AGE":133},{"EMPLOYEE_DETAILS_ADDRESS":"Chennai","sum_AGE":106},{"EMPLOYEE_DETAILS_ADDRESS":"Coimbatore","sum_AGE":84},{"EMPLOYEE_DETAILS_ADDRESS":"Delhi","sum_AGE":40},{"EMPLOYEE_DETAILS_ADDRESS":"Gurgaon","sum_AGE":114},{"EMPLOYEE_DETAILS_ADDRESS":"Guwahati","sum_AGE":92},{"EMPLOYEE_DETAILS_ADDRESS":"Hyderabad","sum_AGE":139},{"EMPLOYEE_DETAILS_ADDRESS":"Jaipur","sum_AGE":102},{"EMPLOYEE_DETAILS_ADDRESS":"Kolkata","sum_AGE":50},{"EMPLOYEE_DETAILS_ADDRESS":"Lucknow","sum_AGE":70},`

4.5.1.9 Check SQL Editor:

URL	services
Description	It allows user to see the auto generated SQL query and returns the query as response.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=generateQuery&formData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSE

	R.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':1000,'prependTableNameToAlias':true}" http://192.168.2.156:8085/hi-ee/services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	generateQuery	service as generateQuery
formData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}	formData contains report name, location,columns,aggregate functions,limit,groupBy etc. Details.
Response Output(JSON Format)	{"status":1,"response":{"classifier":"db.generic","query":"select \n\t\"HIUSER\".\"EMPLOYEE_DETAILS\".\"ADDRESS\" as \"EMPLOYEE_DETAILS_ADDRESS\",\n \tsum(\"HIUSER\".\"EMPLOYEE_DETAILS\".\"AGE\") as \"sum_AGE\" \nfrom\n\n\t\"HIUSER\".\"EMPLOYEE_DETAILS\" \ngroup by\n\n\t\"HIUSER\".\"EMPLOYEE_DETAILS\".\"ADDRESS\" FETCH FIRST 1000 ROWS ONLY"}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with the db classifier and the query.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/services

Authorization | Headers (1) | Body | Pre-request Script | Tests

form-data | x-www-form-urlencoded | raw | binary

```
type:adhoc
serviceType:report
service:generateQuery
formData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-
88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":1000,"prependTableNameToAlias":true}
```

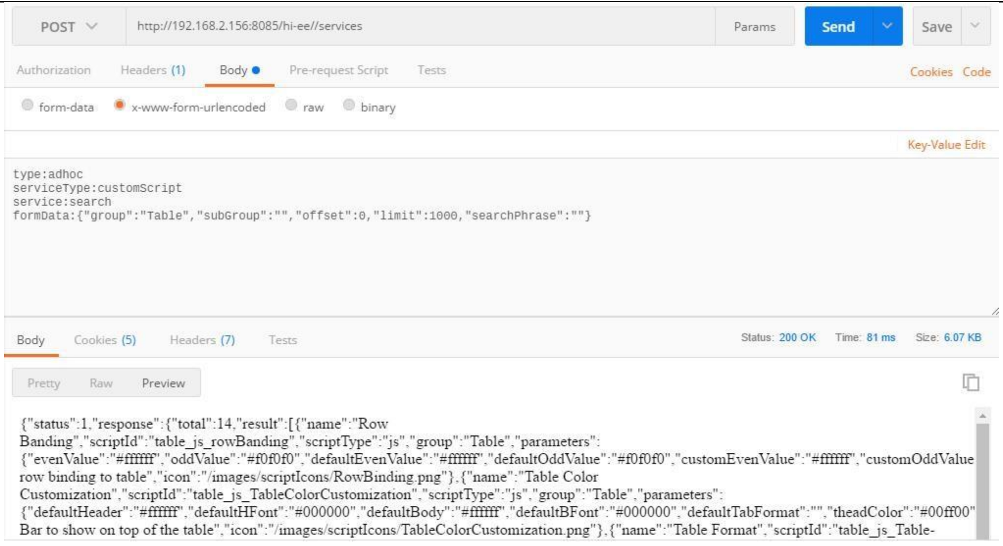
Status: 200 OK Time: 60 ms Size: 649 B

```
{"status":1,"response":{"classifier":"db.generic","query":"select \n\t\"HIUSER\".\"EMPLOYEE_DETAILS\".\"ADDRESS\" as
\"EMPLOYEE_DETAILS_ADDRESS\",\n \tsum(\"HIUSER\".\"EMPLOYEE_DETAILS\".\"AGE\") as \"sum_AGE\"
\nfrom\n\t\"HIUSER\".\"EMPLOYEE_DETAILS\" \ngroup by\n\t\"HIUSER\".\"EMPLOYEE_DETAILS\".\"ADDRESS\" FETCH FIRST 1000 ROWS
ONLY"}}
```

4.5.1.10 Search Report properties:

Note : Accroding to visualization types the custom scripts get changed.

URL	/services	
Description	It allows user to search custom script for report type. Using custom scripts you can apply different visualization changes for report components.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN , ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=customScript&service=search&formData={'group':'Table','subGroup':'','offset':0,'limit':1000,'searchPhrase':''}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**

type:	adhoc	Type of module
serviceType:	customScript	serviceType as customScript
service:	search	service as search
formData:	{"group":"Table","subGroup":"","offset":0,"limit":1000,"searchPhrase":""}	formData contains type of group subgroup etc information.
Response Output(JSON Format)	{"status":1,"response":{"total":14,"result":[{"name":"Row Banding","scriptId":"table_js_rowBanding","scriptType":"js","group":"Table","parameters":{"evenValue":"#ffffff","oddValue":"#f0f0f0","defaultEvenValue":"#ffffff","defaultOddValue":"#f0f0f0","customEvenValue":"#ffffff","customOddValue":"#f0f0f0","switch":"false"},"renderOn":"simpleModel","isEditable":true,"description":"Adds row binding to table","icon":"/images/scriptIcons/RowBinding.png"}]}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with the total number of scripts available for visualization type with script details(name,id,type,group etc).	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

```
type:adhoc
serviceType:customScript
service:search
formData:{"group":"Table","subGroup":"","offset":0,"limit":1000,"searchPhrase":""}
```

Status: 200 OK Time: 81 ms Size: 6.07 KB

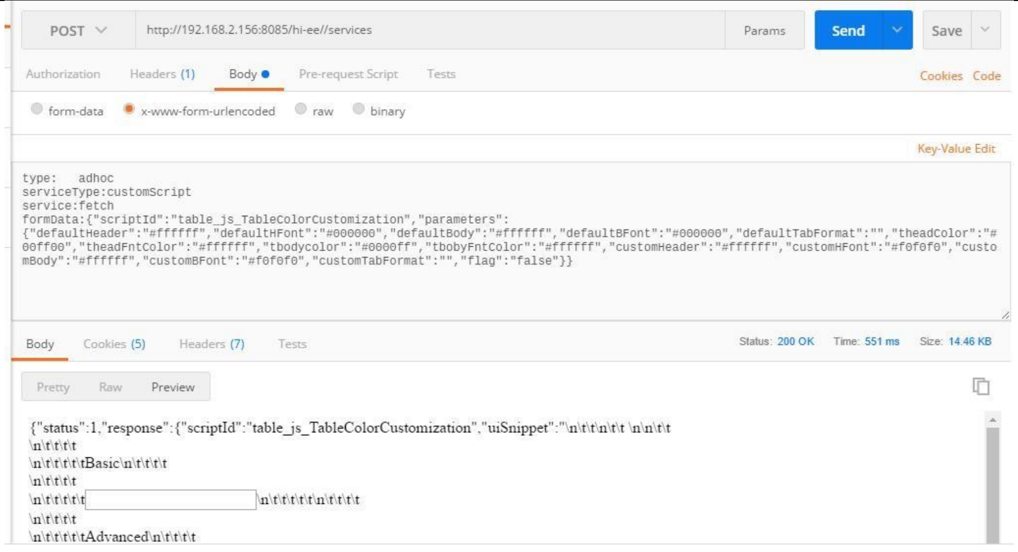
```
{"status":1,"response":{"total":14,"result":[{"name":"Row
Banding","scriptId":"table_js_rowBanding","scriptType":"js","group":"Table","parameters":
{"evenValue":"#ffffff","oddValue":"#f0f0f0","defaultEvenValue":"#ffffff","defaultOddValue":"#f0f0f0","customEvenValue":"#ffffff","customOddValue
row binding to table","icon":"/images/scriptIcons/RowBinding.png"},{"name":"Table Color
Customization","scriptId":"table_js_TableColorCustomization","scriptType":"js","group":"Table","parameters":
{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#00ff00"
Bar to show on top of the table","icon":"/images/scriptIcons/TableColorCustomization.png"},{"name":"Table Format","scriptId":"table_js_Table-
```

4.5.1.11 Fetch selected Custom Script settings:

Note : According to visualization types the custom scripts get changed.

URL	/services
Description	It allows user to search custom script for report type. Using custom scripts you can apply different visualization changes for report components.
Pre-requisite	User should have logged in before accessing the service.[Refer login module]

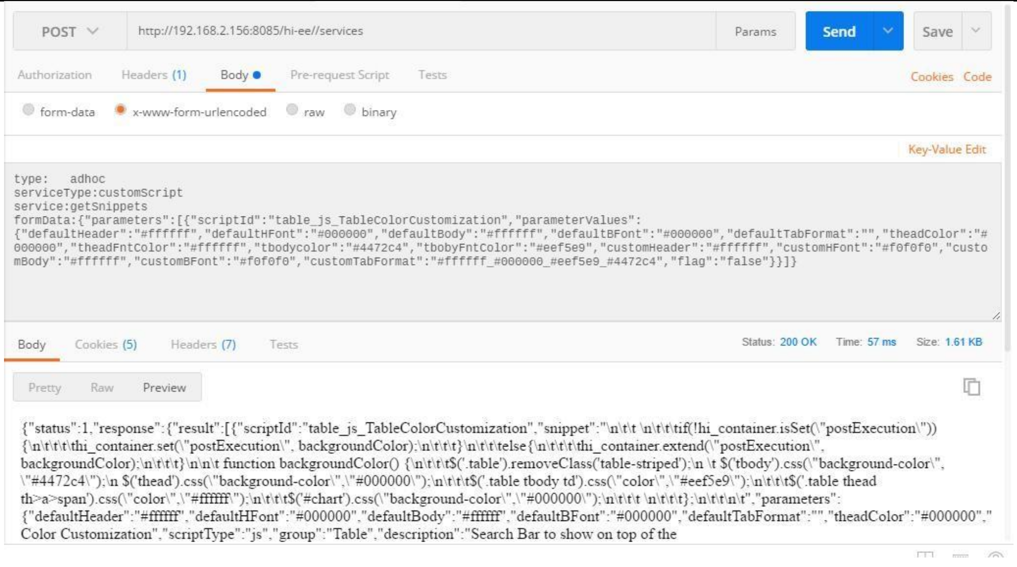
	If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN , ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=customScript&service=fetch&formData={'scriptId':'table_js_TableColorCustomization','parameters':{'defaultHeader':'#ffffff','defaultHFont':'#000000','defaultBody':'#ffffff','defaultBFont':'#000000','defaultTabFormat':'','theadColor':'#00ff00','theadFntColor':'#ffffff','tbodycolor':'#0000ff','tbobyFntColor':'#ffffff','customHeader':'#ffffff','customHFont':'#f0f0f0','customBody':'#ffffff','customBFont':'#f0f0f0','customTabFormat':'','flag':'false'}}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	customScript	serviceType as customScript
service:	fetch	service as fetch
formData:	{"scriptId":"table_js_TableColorCustomization","parameters":{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#00ff00","theadFntColor":"#ffffff","tbodycolor":"#0000ff","tbobyFntColor":"#ffffff","customHeader":"#ffffff","customHFont":"#f0f0f0","customBody":"#ffffff","customBFont":"#f0f0f0","customTabFormat":"","flag":"false"}}	formData contains script information like scriptID and the selected parameters details.
Response Output(JSON Format)	{"status":1,"response":{Script information}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with selected script information. Note : Script information is the selected script information like scriptID , style details,name of the script etc.	
Service Status	200 OK	

Screenshot	

```
POST  http://192.168.2.156:8085/hi-ee//services

type:   adhoc
serviceType:customScript
service:fetch
formData:{"scriptId":"table_js_TableColorCustomization","parameters":
{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#
00ff00","theadFntColor":"#ffffff","tbodycolor":"#0000ff","tbobyFntColor":"#ffffff","customHeader":"#ffffff","customHFont":"#f0f0f0","custo
mBody":"#ffffff","customBFont":"#f0f0f0","customTabFormat":"","flag":"false"}}

Status: 200 OK   Time: 551 ms   Size: 14.46 KB

{"status":1,"response":{"scriptId":"table_js_TableColorCustomization","uiSnippet":"\n\t\t\n\t\t \n\n\t\t
\n\t\t\t\t
\n\t\t\t\t\tBasic\n\t\t\t\t
\n\t\t\t\t
\n\t\t\t\t\t                    \n\t\t\t\t\t\n\t\t\t\t
\n\t\t\t\t
\n\t\t\t\t\tAdvanced\n\t\t\t\t
```

4.5.1.12 Get edited Report Properties :

URL	/services
Description	It allows user to search custom script for report type. Using custom scripts you can apply different visualization changes for report components.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=customScript&service=getSnippets&formData={'parameters':[{'scriptId':'table_js_TableColorCustomization','parameterValues':{'defaultHeader':'#ffffff','defaultHFont':'#000000','defaultBody':'#ffffff','defaultBFont':'#000000','defaultTabFormat':'','theadColor':'#000000','theadFntColor':'#ffffff','tbodycolor':'#4472c4','tbobyFntColor':'#eef5e9','customHeader':'#ffffff','customHFont':'#f0f0f0','customBody':'#ffffff','customBFont':'#f0f0f0','customTabFormat':'#ffffff_#000000_#eef5e9_#4472c4','flag':'false'}}]}"

	http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	customScript	serviceType as customScript
service:	getsnippets	service as getsnippets
formData:	{"parameters":[{"scriptId":"table_js_TableColorCustomization","parameterValues":{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#000000","theadFntColor":"#ffffff","tbodycolor":"#4472c4","tbobyFntColor":"#eef5e9","customHeader":"#ffffff","customHFont":"#f0f0f0","customBody":"#ffffff","customBFont":"#f0f0f0","customTabFormat":"#ffffff_#000000_#eef5e9_#4472c4","flag":"false"}}]}	formData contains script information like scriptID and the selected parameters details.
Response Output(JSON Format)	{"status":1,"response":{Script information}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with edited report property information.	
Service Status	200 OK	
Screenshot		

```
type:   adhoc
serviceType:customScript
service:getSnippets
formData:{"parameters":[{"scriptId":"table_js_TableColorCustomization","parameterValues":
{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#
000000","theadFntColor":"#ffffff","tbodycolor":"#4472c4","tbobyFntColor":"#eef5e9","customHeader":"#ffffff","customHFont":"#f0f0f0","custo
mBody":"#ffffff","customBFont":"#f0f0f0","customTabFormat":"#ffffff_#000000_#eef5e9_#4472c4","flag":"false"}}]}
```

```
{"status":1,"response":{"result":[{"scriptId":"table_js_TableColorCustomization","snippet":"\n\t\t \n\t\t\tif(!hi_container.isSet(\"postExecution\"))
{\n\t\t\thi_container.set(\"postExecution\", backgroundColor);\n\t\t\t}\n\t\t\telse{\n\t\t\thi_container.extend(\"postExecution\",
backgroundColor);\n\t\t\t}\n\n\t function backgroundColor() {\n\t\t\t$('.table').removeClass('table-striped');\n \t $('tbody').css(\"background-color\",
\"#4472c4\");\n $('thead').css(\"background-color\",\"#000000\");\n\t\t\t$('.table tbody td').css(\"color\",\"#eef5e9\");\n\t\t\t$('.table thead
th>a>span').css(\"color\",\"#ffffff\");\n\t\t\t$('#chart').css(\"background-color\",\"#000000\");\n\t\t\t \n\t\t\t};\n\t\t\n\t","parameters":
{"defaultHeader":"#ffffff","defaultHFont":"#000000","defaultBody":"#ffffff","defaultBFont":"#000000","defaultTabFormat":"","theadColor":"#000000","
Color Customization","scriptType":"js","group":"Table","description":"Search Bar to show on top of the
```

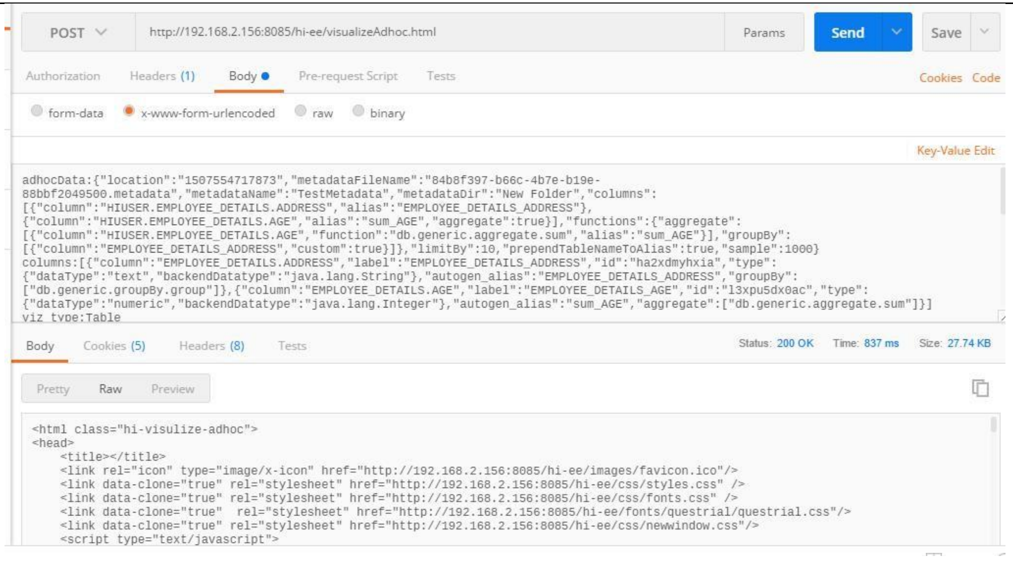
4.5.1.13 Apply Report Properties :

Note : Accroding to visualization types the custom scripts get changed.After selection of report properties we need to apply that report property.

URL	visualizeAdhoc.html
Description	It allows user to apply the report properties on different visualization types(table,charts,crosstab,VF) etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST,GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&adhocData={'location':'1507554717873','metadataFileName':'84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata','metadataName':'TestMetadata','metadataDir':'New Folder','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.ADDRESS','alias':'EMPLOYEE_DETAILS_ADDRESS'},{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','alias':'sum_AGE','aggregate':true}],'functions':{'aggregate':[{'column':'HIUSER.EMPLOYEE_DETAILS.AGE','function':'db.generic.aggregate.sum','alias':'sum_AGE'}],'groupBy':[{'column':'EMPLOYEE_DETAILS_ADDRESS','custom':true}]},'limitBy':10,'prependTableNameToAlias':true,'sample':1000}&columns=[{'column':'EMPLOYEE_DETAILS.ADDRESS','label':'EMPLOYEE_DETAILS_ADDRESS','id':'ha2xdmyhxia','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'EMPLOYEE_DETAILS_ADDRESS','groupBy':['db.generic.groupBy.group']},{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'l3xpu5dx0ac','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_AGE','aggregate':['db.generic.aggregate.sum']}]&viz_type=Table&settings={'vf_file':'sample_report.efwvf','vf_id':1,'dir':'1463377807724/1463377978248/Sample EFW Report','_path':'/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)','script':'(function(data, chartElement){\n \n\t\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\'<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display</h2></div>\');\n } else {\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor (var prop in data[i]) {\t\n\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\tarray1[i] = array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\tbindto: '#chart_1',\n\t\t\t\t\t\t\tdata:

	{\n\t\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\t\ttype: 'pie'\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\ttooltip: {\n\t\t\t\t\t\t\t\t show: true\n\t\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\t});\t\n\t\t\t\t\t}\n\t\t\t\n })(window.data)','type':'VF'}&scripts=[]&customScripts=['\n\t\t \n\t\t\tif(!hi_container.isSet(\'postExecution\')){\n\t\t\t\thi_container.set(\'postExecution\', backgroundColor);\n\t\t\t}\n\t\t\telse{\n\t\t\t\thi_container.extend(\'postExecution\', backgroundColor);\n\t\t\t}\n\n\t function backgroundColor(){\n\t\t\t$('.table').removeClass('table-striped');\n \t $('tbody').css(\'background-color\', \'#4472c4\');\n $('thead').css(\'background-color\',\'#000000\');\n\t\t\t$('.table tbody td').css(\'color\',\'#eef5e9\');\n\t\t\t$('.table thead th>a>span').css(\'color\',\'#ffffff\');\n\t\t\t$('#chart').css(\'background-color\',\'#000000\');\n\t\t\t \n\t\t\t};\n\t\t\n\n\t']&styles=''&customStyles=''" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
adhocData:	{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-88bbf2049500.metadata","metadataName":"Test Metadata","metadataDir":"New Folder","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":10,"prependTableNameToAlias":true,"sample":1000}	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.
columns:	[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ha2xdmyhxia","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"l3xpu5dx0ac","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).
viz_type:	Table	Type of visualization
settings:	{"vf_file":"sample_report.efwvf","vf_id":1,"dir":"1463377807724/1463377978248/Sample EFW Report","_path":"/HI Sample Reports/EFW Reports/Sample EFW Report/sample_report.efwvf/pie chart (1)","script":"(function(data, chartElement){\n \n\t\t\t\t\t if (data.length == 0) {\n $('#chart_1').html(\"<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display</h2></div>\");\n } else {\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor (var prop in data[i]) {\t\n\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\	Settings related to vf file and related details .vizType is the type of visualization.

	t\t\t\t\t\t}\n\t\t\t\t\t\tarray1[i] = array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\tbindto: '#chart_1',\n\t\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\ttype: 'pie'\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\ttooltip: {\n\t\t\t\t\t\t\t show: true\n\t\t\t\t\t\t\t}\n\t\t\t\t\t\t});\t\n\t\t\t\t\t}\n\t\t\t\t\t\t\n })(window.data)","type":"VF"}	
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	["\n\t\t \n\t\t\tif(!hi_container.isSet(\"postExecution\")){\n\t\t\t\thi_container.set(\"postExecution\", backgroundColor);\n\t\t\t}\n\t\t\telse{\n\t\t\t\thi_container.extend(\"postExecution\", backgroundColor);\n\t\t\t}\n\n\t function backgroundColor() {\n\t\t\t$('.table').removeClass('table-striped');\n\t $('tbody').css(\"background-color\", \"#4472c4\");\n $('thead').css(\"background-color\",\"#000000\");\n\t\t\t$('.table tbody td').css(\"color\",\"#eef5e9\");\n\t\t\t$('.table thead th>a>span').css(\"color\",\"#ffffff\");\n\t\t\t$('#chart').css(\"background-color\",\"#000000\");\n\t\t\t \n\t\t\t};\n\t\t\n\t"]	Applied Customscripts /report property
styles:	“”	Applied styles if any
customStyles:	“”	Applied Customstyles if any
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html

Body: x-www-form-urlencoded

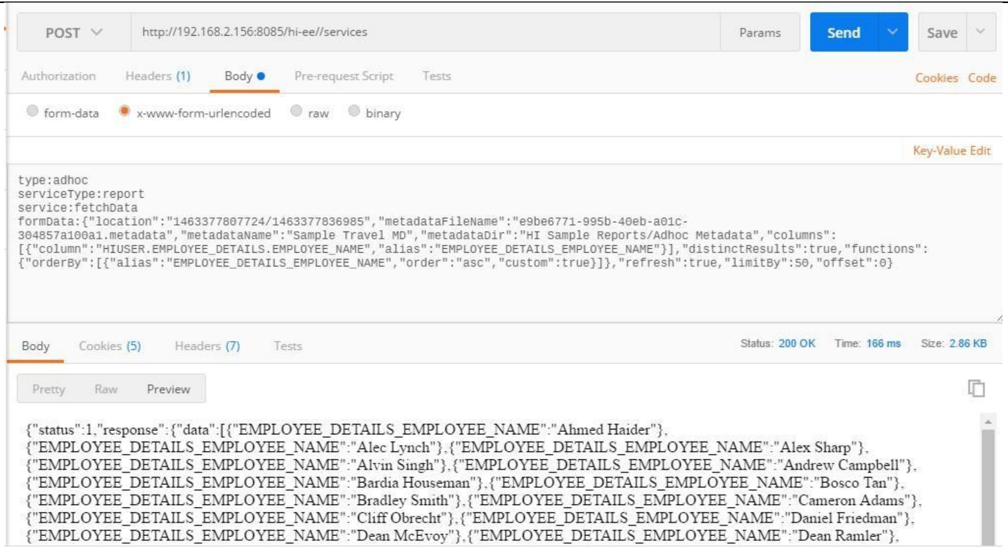
```
adhocData:{"location":"1507554717873","metadataFileName":"84b8f397-b66c-4b7e-b19e-
88bbf2049500.metadata","metadataName":"TestMetadata","metadataDir":"New Folder","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.ADDRESS","alias":"EMPLOYEE_DETAILS_ADDRESS"},
{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","alias":"sum_AGE","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.EMPLOYEE_DETAILS.AGE","function":"db.generic.aggregate.sum","alias":"sum_AGE"}],"groupBy":
[{"column":"EMPLOYEE_DETAILS_ADDRESS","custom":true}]},"limitBy":10,"prependTableNameToAlias":true,"sample":1000}
columns:[{"column":"EMPLOYEE_DETAILS.ADDRESS","label":"EMPLOYEE_DETAILS_ADDRESS","id":"ha2xdmyhxia","type":
{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"EMPLOYEE_DETAILS_ADDRESS","groupBy":
["db.generic.groupBy.group"]},{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"l3xpu5dx0ac","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_AGE","aggregate":["db.generic.aggregate.sum"]}]
viz_type:Table
```

Status: 200 OK Time: 837 ms Size: 27.74 KB

```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
    <script type="text/javascript">
```

4.5.1.14 Add auto search filter in report:

URL	/services
Description	It allows user to add auto search filter in the adhoc report .
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=fetchData&formData={'location':'1463377807724/1463377836985','metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','metadataName':'Sample Travel MD','metadataDir':'HI Sample Reports/Adhoc Metadata','columns':[{'column':'HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME','alias':'EMPLOYEE_DETAILS_EMPLOYEE_NAME'}],'distinctResults':true,'functions':{'orderBy':[{'alias':'EMPLOYEE_DETAILS_EMPLOYEE_NAME','order':'asc','custom':true}]},'refresh':true,'limitBy':50,'offset':0}"

	http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	fetchData	service as fetchData
formData:	{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":[{"column":"HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME","alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME"}],"distinctResults":true,"functions":{"orderBy":[{"alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","order":"asc","custom":true}]},"refresh":true,"limitBy":50,"offset":0}	formData contains the metadata information like metadata name , directory,schema name , applied search filter column etc.
Response Output(JSON Format)	{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ahmed Haider"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alec Lynch"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alex Sharp"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alvin Singh"}],"metadata":[{"1":{"name":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","type":"text"}},{"rows":4}]}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with data array having all rows on which search filter is applied.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee//services

Body: x-www-form-urlencoded

```
type:adhoc
serviceType:report
service:fetchData
formData:{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-
304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":
[{"column":"HIUSER.EMPLOYEE_DETAILS.EMPLOYEE_NAME","alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME"}],"distinctResults":true,"functions":
{"orderBy":[{"alias":"EMPLOYEE_DETAILS_EMPLOYEE_NAME","order":"asc","custom":true}]},"refresh":true,"limitBy":50,"offset":0}
```

Status: 200 OK Time: 166 ms Size: 2.86 KB

```
{"status":1,"response":{"data":[{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Ahmed Haider"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alec Lynch"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alex Sharp"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Alvin Singh"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Andrew Campbell"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Bardia Houseman"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Bosco Tan"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Bradley Smith"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cameron Adams"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Cliff Obrecht"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Daniel Friedman"},
{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Dean McEvoy"},{"EMPLOYEE_DETAILS_EMPLOYEE_NAME":"Dean Ramler"},
```

4.5.1.15 Apply filter on report :

Note : To apply any filter on adhoc report , according to filter different parameter values of filter will get change .Below we just taken one example.

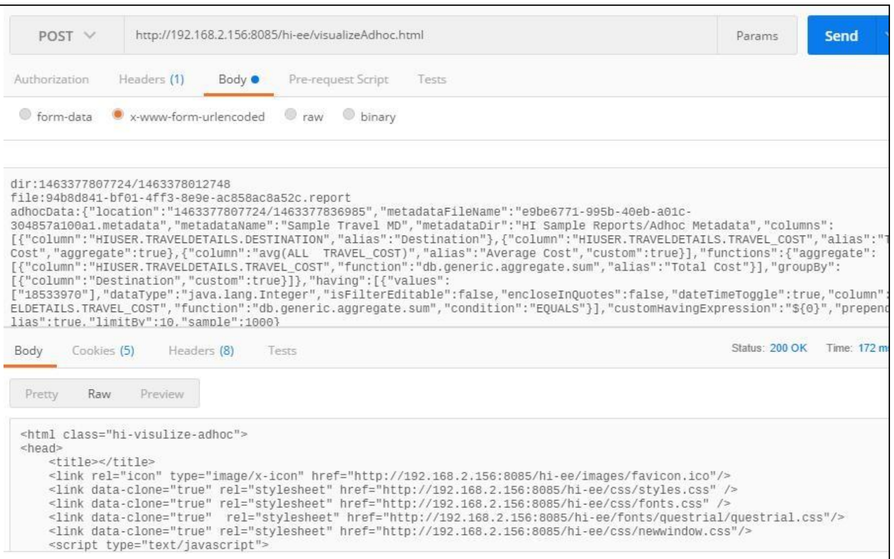
URL	visualizeAdhoc.html
Description	It allows user to apply the filters on different visualization types(table,charts,crosstab,VF) etc.According to applied filter you will get report data.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report&adhocData={'location':'1463377807724/1463377836985','metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','metadataName':'Sample Travel MD','metadataDir':'HI Sample Reports/Adhoc Metadata','columns':[{'column':'HIUSER.TRAVELDETAILS.DESTINATION','alias':'Destination'},{'column':'HIUSER.TRAVELDETAILS.TRAVEL_COST','alias':'Total Cost','aggregate':true},{'column':'avg(ALL TRAVEL_COST)','alias':'Average Cost','custom':true}],'functions':{'aggregate':[{'column':'HIUSER.TRAVELDETAILS.TRAVEL_COST','function':'db.generic.aggregate.sum','alias':'Total Cost'}],'groupBy':[{'column':'Destination','custom':true}]},'having':[{'values':['18533970'],'dataType':'java.lang.Integer','isFilterEditable':false,'encloseInQuotes':false,'dateTimeToggle':true,'column':'HIUSER.TRAVELDETAILS.TRAVEL_COST','function':'db.generic.aggregate.sum','condition':'EQUALS'}],'customHavingExpression':'${0}','prependTableNameToAlias':true,'limitBy':10,'sample':1000}&columns=[{'column':'TRAVELDETAILS.DESTINATION','label':'TRAVELDETAILS_DESTINATION','id':'k53y6kqskbo','type':{'dataType':'text','backendDatatype':'java.lang.String'},'autogen_alias':'TRAVELDETAILS_DESTINATION','groupBy':['db.generic.groupBy.group'],'alias':'Destination'},{'column':'TRAVELDETAILS.TRAVEL_COST','label':'TRAVELDETAILS_TRAVEL_COST','id':'eyo

	hqmqiq2a','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'sum_TRAVEL_COST','aggregate':['db.generic.aggregate.sum'],'alias':'Total Cost'},{'label':'Custom Column','custom':true,'column':'avg(ALL TRAVEL_COST)','alias':'Average Cost','id':'q3hqgcj7u8'}]&viz_type=Table&settings=""&scripts=[]&customScripts=[]&styles="&customStyles="&getFilters[0][values][]=18533970&getFilters[0][mode]=auto&getFilters[0][dataType]=numeric&getFilters[0][dataSource][location]=1463377807724/1463377836985&getFilters[0][dataSource][metadataFileName]=e9be6771-995b-40eb-a01c-304857a100a1.metadata&getFilters[0][dataSource][metadataName]=Sample Travel MD&getFilters[0][dataSource][metadataDir]=HI Sample Reports/Adhoc Metadata&getFilters[0][valuesMode]=auto&getFilters[0][isFilterEditable]=false&getFilters[0][parameters][]=Total Cost&getFilters[0][name]=Total Cost&getFilters[0][encloseInQuotes]=false&getFilters[0][dateTimeToggle]=true&getFilters[0][label]=Total Cost&getFilters[0][column]=TRAVELDETAILS.TRAVEL_COST&getFilters[0][backendDataType]=java.lang.Integer&getFilters[0][adhoc]=true&getFilters[0][condition]=EQUALS&getFilters[0][database]=HIUSER&getFilters[0][aggregate][]=db.generic.aggregate.sum" http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html -v

HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	Physical name of directory
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	Physical name of report file
adhocData:	{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":[{"column":"HIUSER.TRAVELDETAILS.DESTINATION","alias":"Destination"},{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","alias":"Total Cost","aggregate":true},{"column":"avg(ALL TRAVEL_COST)","alias":"Average Cost","custom":true}],"functions":{"aggregate":[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.sum","alias":"Total Cost"}],"groupBy":[{"column":"Destination","custom":true}]},"having":[{"values":["18533970"],"dataType":"java.lang.Integer","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"column":	adhocData is the passed data information related to metadata its name , location with used columns along with functions applied.

	"HIUSER.TRAVELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.sum","condition":"EQUALS"}],"customHavingExpression":"${0}","prependTableNameToAlias":true,"limitBy":10,"sample":1000}	
columns:	[{"column":"TRAVELDETAILS.DESTINATION","label":"TRAVELDETAILS_DESTINATION","id":"k53y6kqskbo","type":{"dataType":"text","backendDatatype":"java.lang.String"},"autogen_alias":"TRAVELDETAILS_DESTINATION", "groupBy":["db.generic.groupBy.group"],"alias":"Destination"},{"column":"TRAVELDETAILS.TRAVEL_COST","label":"TRAVELDETAILS_TRAVEL_COST","id":"eyohqmqiq2a","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"sum_TRAVEL_COST","aggregate":["db.generic.aggregate.sum"],"alias":"Total Cost"},{"label":"Custom Column","custom":true,"column":"avg(ALL TRAVEL_COST)","alias":"Average Cost","id":"q3hqgcj7u8"}]	Columns contains all selected column information(name,dataType,alias,aggregate functions applied etc.).
viz_type:	Table	Type of visualization
settings:	“”	Settings related to type of visualization.
database:	HIUSER	Name of database used
scripts:	[]	Applied scripts if any
customScripts:	["\n\t\t \n\t\t\tif(!hi_container.isSet(\"postExecution\")){\n\t\t\t\thi_container.set(\"postExecution\", backgroundColor);\n\t\t\t}\n\t\t\telse{\n\t\t\t\thi_container.extend(\"postExecution\", backgroundColor);\n\t\t\t}\n\n\t function backgroundColor() {\n\t\t\t$('.table').removeClass('table-striped');\n \t $('tbody').css(\"background-color\", \"#4472c4\");\n $('thead').css(\"background-color\",\"#000000\");\n\t\t\t$('.table tbody td').css(\"color\",\"#eef5e9\");\n\t\t\t$('.table thead th>a>span').css(\"color\",\"#ffffff\");\n\t\t\t$('#chart').css(\"background-color\",\"#000000\");\n\t\t\t \n\t\t\t};\n\t\t\n\t"]	Applied Customscripts /report property

styles:	“”	Applied styles if any
customStyles:	“”	Applied Customstyles if any
getFilters[0][values][]:	18533970	Filter value selected
getFilters[0][mode]:	auto	Type of search mode
getFilters[0][dataType]:	numeric	Type of data
getFilters[0][dataSource][location]:	1463377807724/1463377836985	Location of datasource
getFilters[0][dataSource][metadataFileName]:	e9be6771-995b-40eb-a01c-304857a100a1.metadata	Physical name of metadata
getFilters[0][dataSource][metadataName]:	Sample Travel MD	Name of metadata
getFilters[0][dataSource][metadataDir]:	HI Sample Reports/Adhoc Metadata	Directory name of metadata
getFilters[0][valuesMode]:	auto	Mode of values
getFilters[0][isFilterEditable]:	false	Filter is editable or not : true/false
getFilters[0][parameters][]:	Total Cost	Parameters of filter
getFilters[0][name]:	Total Cost	Name if filter
getFilters[0][encloseInQuotes]:	false	Enclose in quotes or not : true/false
getFilters[0][dateTimeToggle]:	true	Dtaetime toggle value true/false
getFilters[0][label]:	Total Cost	Label of filter
getFilters[0][column]:	TRAVELDETAILS.TRAVEL_COST	Name of column name
getFilters[0][backendDataType]:	java.lang.Integer	Type of java class for numeric value
getFilters[0][adhoc]:	true	Value of adhoc filter : true/false
getFilters[0][condition]:	EQUALS	Condition of filter
getFilters[0][database]:	HIUSER	Name of database
getFilters[0][aggregate][]:	db.generic.aggregate.sum	Used aggregate function
Response Output(JSON Format)	The response we get from API is the report html contents.	
Service Status	200 OK	

Screenshot	

POST http://192.168.2.156:8085/hi-ee/visualizeAdhoc.html Params Send

Authorization Headers (1) Body ● Pre-request Script Tests

form-data [✓ x-www-form-urlencoded] raw binary

```
dir:1463377807724/1463378012748
file:94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report
adhocData:{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-
304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata","columns":
[{"column":"HIUSER.TRAVELDETAILS.DESTINATION","alias":"Destination"},{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","alias":"
Cost","aggregate":true},{"column":"avg(ALL  TRAVEL_COST)","alias":"Average Cost","custom":true}],"functions":{"aggregate":
[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.sum","alias":"Total Cost"}],"groupBy":
[{"column":"Destination","custom":true}]},"having":[{"values":
["18533970"],"dataType":"java.lang.Integer","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"column":
ELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.sum","condition":"EQUALS"}],"customHavingExpression":"${0}","prepend
lias":true,"limitBy":10,"sample":1000}
```

Body Cookies (5) Headers (8) Tests Status: 200 OK Time: 172 m

Pretty Raw Preview

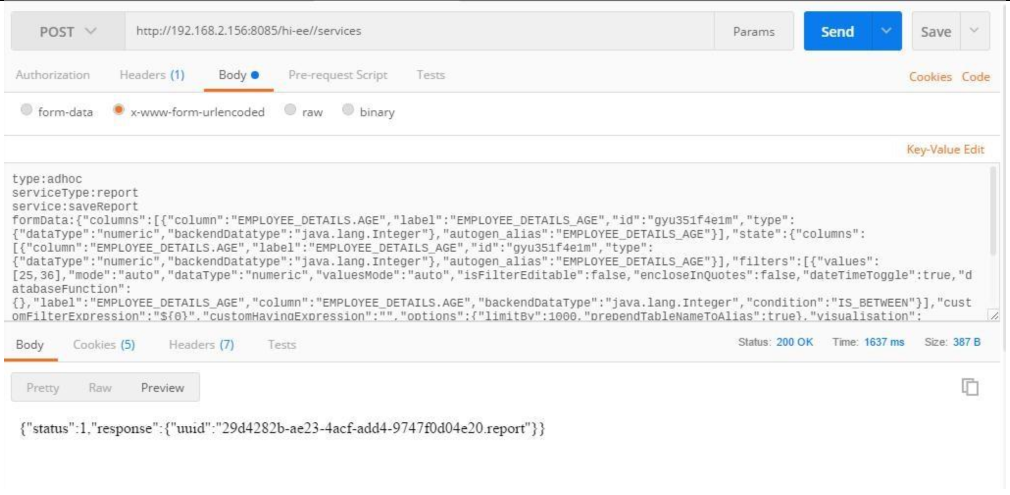
```
<html class="hi-visulize-adhoc">
<head>
    <title></title>
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css" />
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css" />
    <link data-clone="true"  rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
    <link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/newwindow.css"/>
    <script type="text/javascript">
```

4.5.1.16 Save adhoc report :

URL	/services
Description	It allows user to save the adhoc report .
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=saveReport&formData={'columns':[{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'gyu351f4e1m','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'EMPLOYEE_DETAILS_AGE'}],'state':{'columns':[{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'gyu351f4e1m','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'EMPLOYEE_DETAILS_AGE'}],'filters':[{'values':[25,36],'mode':'auto','dataType':'numeric','valuesMode':'auto','isFilterEditable':false,'encloseInQuotes':false,'dateTimeToggle':true,'databaseFunction':{},'label':'EMPLO

	YEE_DETAILS_AGE','column':'EMPLOYEE_DETAILS.AGE','backendDataType':'java.lang.Integer','condition':'IS_BETWEEN'}],'customFilterExpression':'${0}','customHavingExpression':'','options':{'limitBy':1000,'prependTableNameToAlias':true},'visualisation':{'type':'Table','chartGroup':'','selectedType':'Table','settings':{'script':null,'vizscriptsEditMultipleMode':false},'vizSelectedScripts':[]},'scripts':[],'styles':'','customStyles':'','customScripts':[]},'classifier':'db.generic','metadata':{'location':'1463377807724/1463377836985','metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','metadataName':'Sample Travel MD','metadataDir':'HI Sample Reports/Adhoc Metadata'},'location':'1507554717873','uuid':'29d4282b-ae23-4acf-add4-9747f0d04e20.report'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	saveReport	service as saveReport
formData:	{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"state":{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"filters":[{"values":[25,36],"mode":"auto","dataType":"numeric","valuesMode":"auto","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"databaseFunction":{},"label":"EMPLOYEE_DETAILS_AGE","column":"EMPLOYEE_DETAILS.AGE","backendDataType":"java.lang.Integer","condition":"IS_BETWEEN"}],"customFilterExpression":"${0}","customHavingExpression":"","options":{"limitBy":1000,"prependTableNameToAlias":true},"visualisation":{"type":"Table","chartGroup":"","selectedType":"Table","settings":{"script":null,"vizscriptsEditMultipleMode":false},"vizSelectedScripts":[]},"scripts":[],"styles":"","customStyles":"","customScripts":[]},"classifier":"db.generic","metadata":{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata"},"location":"1507554717873","uuid":"29d4282b-ae23-4acf-add4-9747f0d04e20.report"}	formData contains the report related information like columns ,filters,functions used ,metadata information, report uuid stored etc.
Response Output(JSON Format)	{"status":1,"response":{"uuid":"29d4282b-ae23-4acf-add4-9747f0d04e20.report"}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with uuid of the report.	

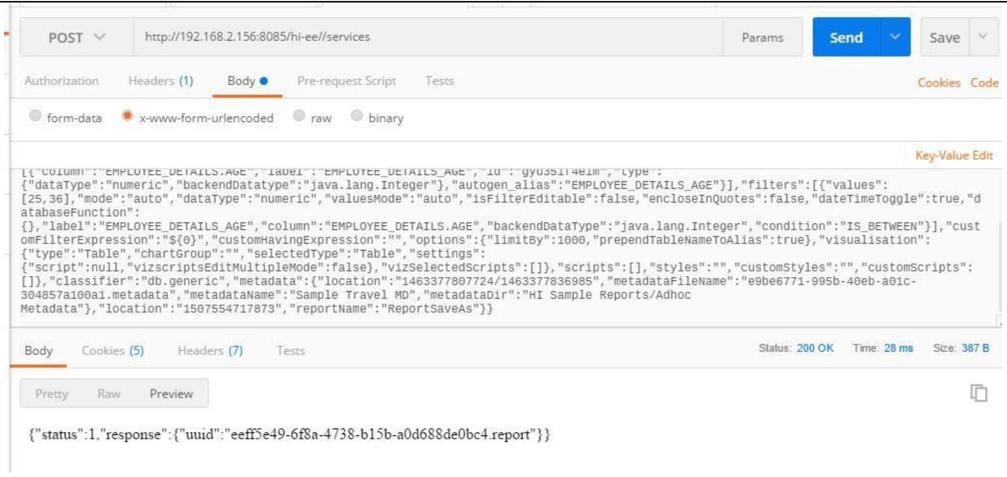
Service Status	200 OK

Screenshot	 POST http://192.168.2.156:8085/hi-ee//services · Body: x-www-form-urlencoded
	`type:adhoc`
	`serviceType:report`
	`service:saveReport`
	`formData:{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"state":{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"filters":[{"values":[25,36],"mode":"auto","dataType":"numeric","valuesMode":"auto","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"databaseFunction":{},"label":"EMPLOYEE_DETAILS_AGE","column":"EMPLOYEE_DETAILS.AGE","backendDataType":"java.lang.Integer","condition":"IS_BETWEEN"}],"customFilterExpression":"${0}","customHavingExpression":"","options":{"limitBy":1000,"prependTableNameToAlias":true},"visualisation":`
	Status: 200 OK Time: 1637 ms Size: 387 B
	`{"status":1,"response":{"uuid":"29d4282b-ae23-4acf-add4-9747f0d04e20.report"}}`

4.5.1.17 Adhoc report SaveAs:

URL	/services
Description	It allows user to re-save the adhoc report .
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN , ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=saveReport&formData={'columns':[{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'gyu351f4e1m','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'EMPLOYEE_DETAILS_AGE'}],'state':{'columns':[{'column':'EMPLOYEE_DETAILS.AGE','label':'EMPLOYEE_DETAILS_AGE','id':'gyu351f4e1m','type':{'dataType':'numeric','backendDatatype':'java.lang.Integer'},'autogen_alias':'EMPLOYEE_DETAILS_AGE'}],'filters':[{'values':[25,36],'mode':'auto','dataType':'numeric','valuesMode':'auto','isFilterEditable':false,'encloseInQuotes':false,'dateTimeToggle':true,'databaseFunction':{},'label':'EMPLOYEE_DETAILS_AGE','column':'EMPLOYEE_DETAILS.AGE','backendDataType':'java.lang.Integer','condition':'IS_BETWEEN'}],'customFilterExpression':'${0}','cust

	omHavingExpression':'','options':{'limitBy':1000,'prependTableNameToAlias':true},'visualisation':{'type':'Table','chartGroup':'','selectedType':'Table','settings':{'script':null,'vizscriptsEditMultipleMode':false},'vizSelectedScripts':[]},'scripts':[],'styles':'','customStyles':'','customScripts':[]},'classifier':'db.generic','metadata':{'location':'1463377807724/1463377836985','metadataFileName':'e9be6771-995b-40eb-a01c-304857a100a1.metadata','metadataName':'Sample Travel MD','metadataDir':'HI Sample Reports/Adhoc Metadata'},'location':'1507554717873','reportName':'ReportSaveAs'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	saveReport	service as saveReport
formData:	{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"state":{"columns":[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351f4e1m","type":{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"filters":[{"values":[25,36],"mode":"auto","dataType":"numeric","valuesMode":"auto","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"databaseFunction":{},"label":"EMPLOYEE_DETAILS_AGE","column":"EMPLOYEE_DETAILS.AGE","backendDataType":"java.lang.Integer","condition":"IS_BETWEEN"}],"customFilterExpression":"${0}","customHavingExpression":"","options":{"limitBy":1000,"prependTableNameToAlias":true},"visualisation":{"type":"Table","chartGroup":"","selectedType":"Table","settings":{"script":null,"vizscriptsEditMultipleMode":false},"vizSelectedScripts":[]},"scripts":[],"styles":"","customStyles":"","customScripts":[]},"classifier":"db.generic","metadata":{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc Metadata"},"location":"1507554717873","reportName":"ReportSaveAs"}}	formData contains the report related information like columns ,filters,functions used ,metadata information, report name at different location instead of uuid stored etc.
Response Output(JSON	{"status":1,"response":{"uuid":"eeff5e49-6f8a-4738-b15b-a0d688de0bc4.report"}}	

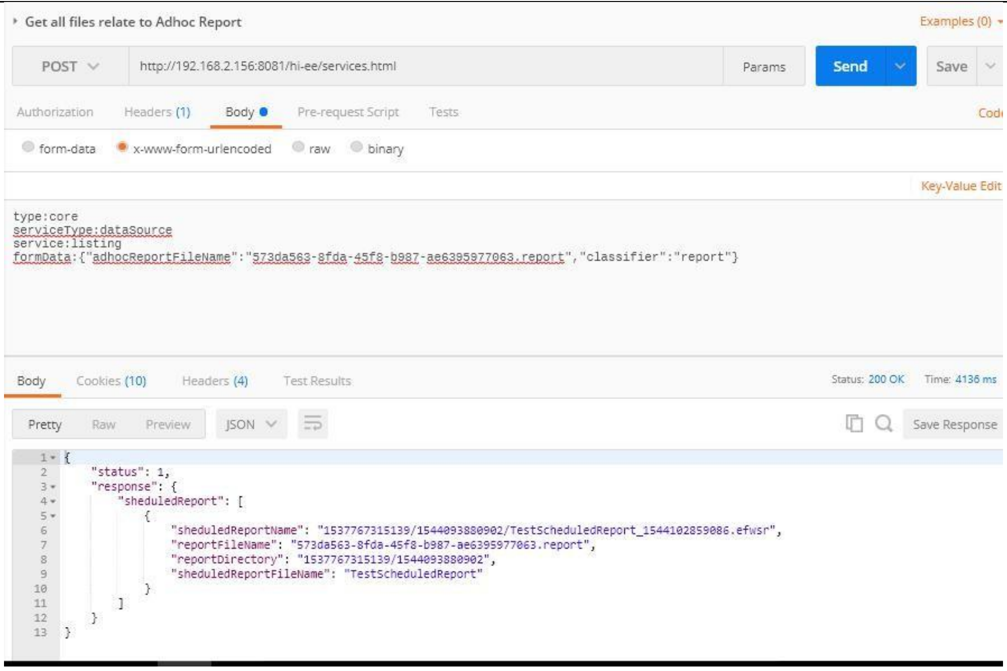
Format)	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with uuid of the report.
Service Status	200 OK
Screenshot	

```
[{"column":"EMPLOYEE_DETAILS.AGE","label":"EMPLOYEE_DETAILS_AGE","id":"gyu351l4elm","type":
{"dataType":"numeric","backendDatatype":"java.lang.Integer"},"autogen_alias":"EMPLOYEE_DETAILS_AGE"}],"filters":[{"values":
[25,36],"mode":"auto","dataType":"numeric","valuesMode":"auto","isFilterEditable":false,"encloseInQuotes":false,"dateTimeToggle":true,"d
atabaseFunction":
{},"label":"EMPLOYEE_DETAILS_AGE","column":"EMPLOYEE_DETAILS.AGE","backendDataType":"java.lang.Integer","condition":"IS_BETWEEN"}],"cust
omFilterExpression":"${0}","customHavingExpression":"","options":{"limitBy":1000,"prependTableNameToAlias":true},"visualisation":
{"type":"Table","chartGroup":"","selectedType":"Table","settings":
{"script":null,"vizscriptsEditMultipleMode":false},"vizSelectedScripts":[]},"scripts":[],"styles":"","customStyles":"","customScripts":
[]},"classifier":"db.generic","metadata":{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-
304857a100a1.metadata","metadataName":"Sample Travel MD","metadataDir":"HI Sample Reports/Adhoc
Metadata"},"location":"1507554717873","reportName":"ReportSaveAs"}}
```

```
{"status":1,"response":{"uuid":"eeff5e49-6f8a-4738-b15b-a0d688de0bc4.report"}}
```

4.5.1.18 Get all files related to adhoc Report

URL	/services.html	
Description	The user will get scheduled report files related to adhoc report.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8081/hi-ee/services.html **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=core&serviceType=dataSource&service=listing&formData={'metadataFileName':'9d95494f-a302-4b45-880c-9550bcb53e1a.metadata','classifier':'metadata','location':'1537767315139/1544093880902'}" http://192.168.2.156:8081/hi-ee/services.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	core	type as core
serviceType:	dataSource	serviceType as dataSource

service:	listing	The service is to list the resources related to provided global datasourceID
formData:	{"adhocReportFileName":"573da563-8fda-45f8-b987-ae6395977063.report","classifier":"report"}	adhocreportFile:name of the metadata file. Classifier :report
Response Output(JSON Format)	{"status":1,"response":{"sheduledReport":[{"sheduledReportName":"1537767315139/1544093880902/TestScheduledReport_1544102859086.efwsr","reportFileName":"573da563-8fda-45f8-b987-ae6395977063.report","reportDirectory":"1537767315139/1544093880902","sheduledReportFileName":"TestScheduledReport"}]}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status.	
Service Status	200 OK	
Screenshot		

Get all files relate to Adhoc Report

POST http://192.168.2.156:8081/hi-ee/services.html

Body: x-www-form-urlencoded

```
type:core
serviceType:dataSource
service:listing
formData:{"adhocReportFileName":"573da563-8fda-45f8-b987-ae6395977063.report","classifier":"report"}
```

Status: 200 OK Time: 4136 ms

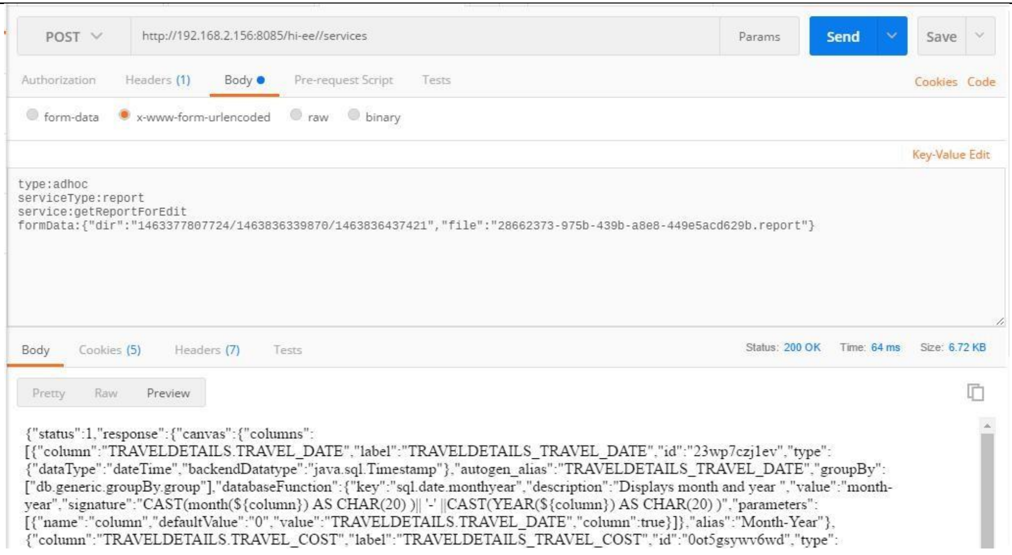
```
{
    "status": 1,
    "response": {
        "sheduledReport": [
            {
                "sheduledReportName": "1537767315139/1544093880902/TestScheduledReport_1544102859086.efwsr",
                "reportFileName": "573da563-8fda-45f8-b987-ae6395977063.report",
                "reportDirectory": "1537767315139/1544093880902",
                "sheduledReportFileName": "TestScheduledReport"
            }
        ]
    }
}
```

4.5.2 Edit Report

Note : To edit any adhoc report you need to use below API , once the report get opened in edit mode to do any changes in report the API's are same as API's used in create adhoc report .You can refer API's from Edit Report to till report save/saveAS to do changes in report .

API to open adhoc report in Edit Mode :

URL	/services	
Description	It allows user toedit the adhoc report	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN , ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&type=adhoc&serviceType=report&service=getReportForEdit&formData={'dir':'1463377807724/1463836339870/1463836437421','file':'28662373-975b-439b-a8e8-449e5acd629b.report'}" http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	Type of module
serviceType:	report	serviceType as report
service:	getReportForEdit	service as getReportForEdit
formData:	{"dir":"1463377807724/1463836339870/1463836437421","file":"28662373-975b-439b-a8e8-449e5acd629b.report"}	formData contains directory and the filename for edit
Response Output(JSON Format)	{"status":1,"response":{Script information}}	
Description of Response Output :	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status with edited reportdetails like column details.	
Service Status	200 OK	

Screenshot	

```
type:adhoc
serviceType:report
service:getReportForEdit
formData:{"dir":"1463377807724/1463836339870/1463836437421","file":"28662373-975b-439b-a8e8-449e5acd629b.report"}
```

{"status":1,"response":{"canvas":{"columns":
[{"column":"TRAVELDETAILS.TRAVEL_DATE","label":"TRAVELDETAILS_TRAVEL_DATE","id":"23wp7czj1ev","type":
{"dataType":"dateTime","backendDatatype":"java.sql.Timestamp"},"autogen_alias":"TRAVELDETAILS_TRAVEL_DATE","groupBy":
["db.generic.groupBy.group"],"databaseFunction":{"key":"sql.date.monthyear","description":"Displays month and year ","value":"month-
year","signature":"CAST(month(${column}) AS CHAR(20))|| '-' ||CAST(YEAR(${column}) AS CHAR(20))","parameters":
[{"name":"column","defaultValue":"0","value":"TRAVELDETAILS.TRAVEL_DATE","column":true}]},"alias":"Month-Year"},
{"column":"TRAVELDETAILS.TRAVEL_COST","label":"TRAVELDETAILS_TRAVEL_COST","id":"0ot5gsywv6wd","type":

5. Dashboard Designer Module

Dashboard designer provides interpanel communication.With dashboard designer user can view collection of different reports in a single canvas , these reports can be in different forms like Table, Charts, and Map etc.It provides customization and some additional components are available in dashboard designer.

Dashboard designer page : http://192.168.2.156:8085/hi-ee/designer.html

5.1 Create Dashboard

To create dashboard using designer there are only following API's:

1. Refresh repository

2. Save Dashboard

3. SaveAs Dasboard

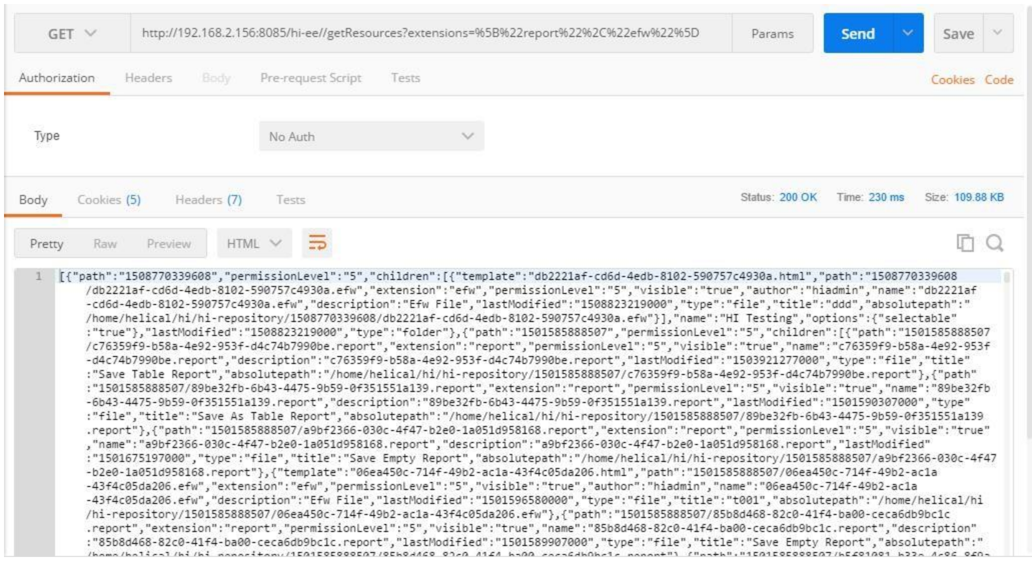
5.1.1 Refresh Repository

This API refresh the repository by which we can add updated/new reports(with extension as efw and report) to dashboard.

URL	/getResources?extensions=%5B%22report%22%2C%22efw%22%5D
Description	Loads all the efw reports and the adhoc reports resources i.e file system whichever is accessible to the logged in user. It will show you the all efw report and the adhoc reports resources related details like its subfolder , name ,fileType,permission etc.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_USER, ROLE_ADMIN
HTTP Request Method	**GET**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//getResources?extensions=%5B%22report%22%2C%22efw%22%5D **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&extensions=%5B%22report%22%2C%22efw%22%5D" http://192.168.2.156:8085/hi-ee//getResources -v

Response Output(JSON Format)	
	```
	[
	  {
	    "path": "1508770339608",
	    "permissionLevel": "5",
	    "children": [
	      {
	        "template": "db2221af-cd6d-4edb-8102-590757c4930a.html",
	        "path": "1508770339608/db2221af-cd6d-4edb-8102-590757c4930a.efw",
	        "extension": "efw",
	        "permissionLevel": "5",
	        "visible": "true",
	        "author": "hiadmin",
	        "name": "db2221af-cd6d-4edb-8102-590757c4930a.efw",
	        "description": "Efw File",
	        "lastModified": "1508823219000",
	        "type": "file",
	        "title": "ddd",
	        "absolutepath": "/home/helical/hi/hi-repository/1508770339608/db2221af-cd6d-4edb-8102-
	590757c4930a.efw"
	      }
	    ],
	    "name": "HI Testing",
	```

	"options": {
	"selectable": "true"
	},
	"lastModified": "1508823219000",
	"type": "folder"
	}
	]

Description of Response Output:	The response returned is the JSON array of all efw report and the adhoc report resources having the different paths of the repository , its permission(Click for more details) , name of the folder , lastmodified timestamp , type etc. It returns the children array which is the sub-folder/file of the path having all details(name,type,title,path) related to children file/folder. • PermissionLevel: This key have the permission of the resource for the respective user. • lastModified holds the timestamp information for the file/folder when it was last modified/access. Path holds the physical name of the file/folder.
Service Status	200 OK
Screenshot	

```
[{"path":"1508770339608","permissionLevel":"5","children":[{"template":"db2221af-cd6d-4edb-8102-590757c4930a.html","path":"1508770339608
    /db2221af-cd6d-4edb-8102-590757c4930a.efw","extension":"efw","permissionLevel":"5","visible":"true","author":"hiadmin","name":"db2221af
    -cd6d-4edb-8102-590757c4930a.efw","description":"Efw File","lastModified":"1508823219000","type":"file","title":"ddd","absolutepath":"
    /home/helical/hi/hi-repository/1508770339608/db2221af-cd6d-4edb-8102-590757c4930a.efw"}],"name":"HI Testing","options":{"selectable"
    :"true"},"lastModified":"1508823219000","type":"folder"},{"path":"1501585888507","permissionLevel":"5","children":[{"path":"1501585888507
    /c76359f9-b58a-4e92-953f-d4c74b7990be.report","extension":"report","permissionLevel":"5","visible":"true","name":"c76359f9-b58a-4e92-953f
    -d4c74b7990be.report","description":"c76359f9-b58a-4e92-953f-d4c74b7990be.report","lastModified":"1503921277000","type":"file","title"
    :"Save Table Report","absolutepath":"/home/helical/hi/hi-repository/1501585888507/c76359f9-b58a-4e92-953f-d4c74b7990be.report"},{"path"
    :"1501585888507/89be32fb-6b43-4475-9b59-0f351551a139.report","extension":"report","permissionLevel":"5","visible":"true","name":"89be32fb
    -6b43-4475-9b59-0f351551a139.report","description":"89be32fb-6b43-4475-9b59-0f351551a139.report","lastModified":"1501590307000","type"
    :"file","title":"Save As Table Report","absolutepath":"/home/helical/hi/hi-repository/1501585888507/89be32fb-6b43-4475-9b59-0f351551a139
    .report"},{"path":"1501585888507/a9bf2366-030c-4f47-b2e0-1a051d958168.report","extension":"report","permissionLevel":"5","visible":"true"
    ,"name":"a9bf2366-030c-4f47-b2e0-1a051d958168.report","description":"a9bf2366-030c-4f47-b2e0-1a051d958168.report","lastModified"
    :"1501675197000","type":"file","title":"Save Empty Report","absolutepath":"/home/helical/hi/hi-repository/1501585888507/a9bf2366-030c-4f47
    -b2e0-1a051d958168.report"},{"template":"06ea450c-714f-49b2-ac1a-43f4c05da206.html","path":"1501585888507/06ea450c-714f-49b2-ac1a
    -43f4c05da206.efw","extension":"efw","permissionLevel":"5","visible":"true","author":"hiadmin","name":"06ea450c-714f-49b2-ac1a
    -43f4c05da206.efw","description":"Efw File","lastModified":"1501596580000","type":"file","title":"t001","absolutepath":"/home/helical/hi
    /hi-repository/1501585888507/06ea450c-714f-49b2-ac1a-43f4c05da206.efw"},{"path":"1501585888507/85b8d468-82c0-41f4-ba00-ceca6db9bc1c
    .report","extension":"report","permissionLevel":"5","visible":"true","name":"85b8d468-82c0-41f4-ba00-ceca6db9bc1c.report","description"
    :"85b8d468-82c0-41f4-ba00-ceca6db9bc1c.report","lastModified":"1501589907000","type":"file","title":"Save Empty Report","absolutepath":"
```

5.1.2 Save Dashboard

Note : While dashboard creation we can add different available components to it according to components the requested form data will get change, so after creation of dashboard we can save it , to save dashboard use below API.

URL	/services
Description	It allows user to save the created dashboard.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwdd&service=designer&formData={"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-title\">Destination wise monthly travel expense</div><div class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div

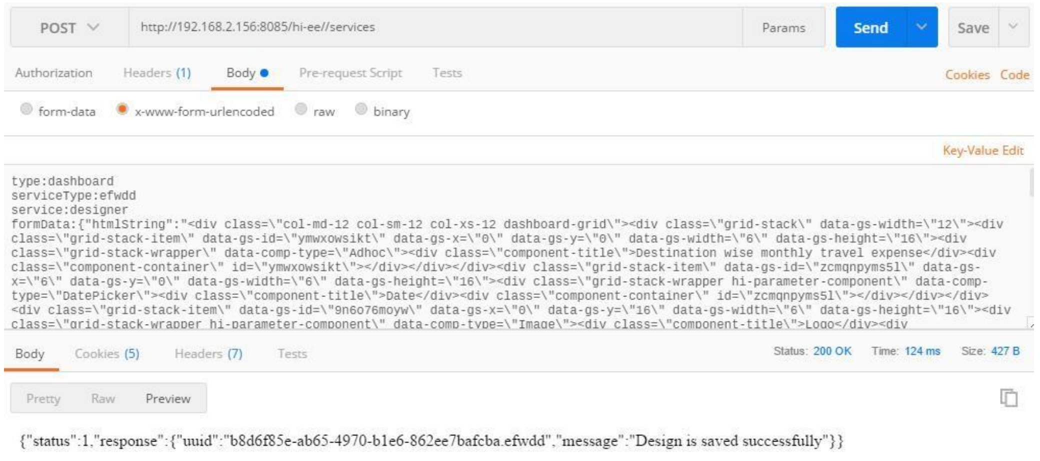
	class=\"component-container\" id=\"9n6o76moyw\"></div></div></div></div></div>\n<script>$(document).ready(function(){\n$(\".grid-stack\").gridstack({width: 12, cellHeight: \"10px\", verticalMargin: \"10px\", staticGrid: true});\n $.gridStackCSSGenerator(12, \"hi-gridstack\");\n\n var dashboard = Dashboard|| {},\n gs = $(\".grid-stack\").data(\"gridstack\");\n dashboard.setGridOptions = function(options) {\n if (!_.isObject(options)) return;\n\n if (typeof options.width === \"number\") {\n gs.setGridWidth(options.width);\n $.gridStackCSSGenerator(options.width, \"hi-gridstack\");\n }\n\n if (typeof options.verticalMargin !== \"undefined\") {\n gs.verticalMargin(options.verticalMargin);\n }\n\n if (typeof options.horizontalMargin !== \"undefined\") {\n var marginStyles = document.getElementById(\"hi-gridstack-margins\");\n\n if (!marginStyles) {\n marginStyles = document.createElement(\"style\");\n marginStyles.type = \"text/css\";\n marginStyles.rel = \"stylesheet\";\n marginStyles.id = \"hi-gridstack-margins\";\n document.getElementsByTagName(\"head\")[0].appendChild(marginStyles);\n }\n\n var css = \".grid-stack .grid-stack-item .grid-stack-wrapper {\" +\n \"margin:\" + options.horizontalMargin + \";\" +\n \"width: calc(100% - \" + options.horizontalMargin + \" -\" + options.horizontalMargin+ \");\" +\n \"}\";\n\n marginStyles.textContent = css;\n }\n\n if (typeof options.cellHeight !== \"undefined\") {\n gs.cellHeight(options.cellHeight);\n }\n };\n\n$(\"<style id='hi-designer-styles'></style>\").appendTo(\"head\")\nDashboard.setVariable(\"Year\", [\"2015\"]);\nDashboard.setVariable(\"Month Name\", [\"'February'\"]);\n\nDashboard.init([{\"type\":\"adhoc\",\"options\":{\"dir\":\"1463377807724\\u002F1463378012748\",\"file\":\"e211af96-d633-4dc7-9460-3612970317fd.report\",\"ext\":\"report\"},\"uid\":\"ymwxowsikt\",\"name\":\"ymwxowsikt\",\"label\":\"Destination wise monthly travel expense\",\"executeAtStart\":true,\"htmlElementId\":\"#ymwxowsikt\"},{\"parameters\":null,\"name\":\"zcmqnpyms5l\",\"uid\":\"zcmqnpyms5l\",\"label\":\"Date\",\"requestParameters\":{},\"executeAtStart\":\"\",\"type\":\"datepicker\",\"options\":{\"displayFormat\":\"DD\\u002FMM\\u002FYYYY\",\"outputFormat\":\"\"},\"htmlElementId\":\"#zcmqnpyms5l\"},{\"type\":\"image\",\"options\":{\"src\":\"https:\\u002F\\u002Fwww.webstix.com\\u002Fwp-content\\u002Fuploads\\u002F2017\\u002F01\\u002FGoogle_Chrome_for_Android-_Android_5.0_Logo.png\",\"id\":\"img1\",\"alt\":\"\",\"boxComp\":\"Image\",\"boxPx\":\"\",\"boxColor\":\"#FFFFFF\",\"bgOpacity\":\"0.1\"},\"uid\":\"9n6o76moyw\",\"name\":\"9n6o76moyw\",\"executeAtStart\":\"\",\"requestParameters\":{},\"label\":\"Logo\",\"htmlElementId\":\"#9n6o76moyw\"}]);\n\n});</script>","state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadata":{"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options":{"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460-3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise monthly travel expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms5l":{"gs_attr":{"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms5l","uid":"zcmqnpyms5l","label":"Date","requestParameters":{},"executeAtStart":"","type":"dashboard-component","options":{"compType":"DatePicker","displayFormat":"DD/MM/YYYY","outputFormat":""}},"9n6o76moyw":{"type":"dashboard-component","options":{"compType":"Image","src":"https://www.webstix.com/wp-content/uploads/2017/01/Google_Chrome_for_Android-_Android_5.0_Logo.png","id":"img1","alt":"","boxComp":"Image","boxPx":"","boxColor":"#FFFFFF","bgOpacity":"0.1"},"uid":"9n6o76moyw","name":"9n6o76moyw","executeAtStart":"","requestParameters":{},"label":"Logo","gs_attr":{"x":0,"y":16,"height":16,"width":6}}},"css":"","script":""},"dir":"1508770339608","fileName":"DashboardSave"}' http://192.168.2.156:8085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as adhoc dashboard type.
serviceType:	efwdd	Servicetype as efwdd
service:	designer	Service to dashboard designer.

formData:	{"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-title\">Destination wise monthly travel expense</div><div class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div class=\"component-container\" id=\"9n6o76moyw\"></div></div></div></div></div>\n<script>$(document).ready(function(){\n$(\".grid-stack\").gridstack({width: 12, cellHeight: \"10px\", verticalMargin: \"10px\", staticGrid: true});\n $.gridStackCSSGenerator(12, \"hi-gridstack\");\n\n var dashboard = Dashboard || {},\n gs = $(\".grid-stack\").data(\"gridstack\");\n dashboard.setGridOptions = function(options) {\n if (!_.isObject(options)) return;\n\n if (typeof options.width === \"number\") {\n gs.setGridWidth(options.width);\n $.gridStackCSSGenerator(options.width, \"hi-gridstack\");\n }\n\n if (typeof options.verticalMargin !== \"undefined\") {\n gs.verticalMargin(options.verticalMargin);\n }\n\n if (typeof options.horizontalMargin !== \"undefined\") {\n var marginStyles = document.getElementById(\"hi-gridstack-margins\");\n\n if (!marginStyles) {\n marginStyles = document.createElement(\"style\");\n marginStyles.type = \"text/css\";\n marginStyles.rel = \"stylesheet\";\n marginStyles.id = \"hi-gridstack-margins\";\n document.getElementsByTagName(\"head\")[0].appendChild(marginStyles);\n }\n\n var css = \".grid-stack .grid-stack-item .grid-stack-wrapper {\" +\n \"margin:\" + options.horizontalMargin + \";\" +\n \"width: calc(100% - \" + options.horizontalMargin + \" - \" + options.horizontalMargin+ \");\" +\n	Formdata contains the dashboard html contents and the component array with all the added components to it with location of dashboard file and the file name.

	\"}\";\n\n marginStyles.textContent = css;\n }\n\n if (typeof options.cellHeight !== \"undefined\") {\n gs.cellHeight(options.cellHeight);\n }\n };\n\n$(\"<style id='hi-designer-styles'></style>\").appendTo(\"head\")\nDashboard.setVariable(\"Year\", [\"2015\"]);\nDashboard.setVariable(\"Month Name\", [\"'February'\"]);\n\nDashboard.init([{\"type\":\"adhoc\",\"options\":{\"dir\":\"1463377807724\\u002F1463378012748\",\"file\":\"e211af96-d633-4dc7-9460-3612970317fd.report\",\"ext\":\"report\"},\"uid\":\"ymwxowsikt\",\"name\":\"ymwxowsikt\",\"label\":\"Destination wise monthly travel expense\",\"executeAtStart\":true,\"htmlElementId\":\"#ymwxowsikt\"},{\"parameters\":null,\"name\":\"zcmqnpyms5l\",\"uid\":\"zcmqnpyms5l\",\"label\":\"Date\",\"requestParameters\":{},\"executeAtStart\":\"\",\"type\":\"datepicker\",\"options\":{\"displayFormat\":\"DD\\u002FMM\\u002FYYYY\",\"outputFormat\":\"\"},\"htmlElementId\":\"#zcmqnpyms5l\"},{\"type\":\"image\",\"options\":{\"src\":\"https:\\u002F\\u002Fwww.webstix.com\\u002Fwp-content\\u002Fuploads\\u002F2017\\u002F01\\u002FGoogle_Chrome_for_Android-_Android_5.0_Logo.png\",\"id\":\"img1\",\"alt\":\"\",\"boxComp\":\"Image\",\"boxPx\":\"\",\"boxColor\":\"#FFFFFF\",\"bgOpacity\":\"0.1\"},\"uid\":\"9n6o76moyw\",\"name\":\"9n6o76moyw\",\"executeAtStart\":\"\",\"requestParameters\":{},\"label\":\"Logo\",\"htmlElementId\":\"#9n6o76moyw\"}]);\n\n});</script>","state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadata":{"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options":{"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460-3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise monthly travel expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms5l":{"gs_attr":{"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms5l","uid":"zcmqnpyms5l","label":"Date","requestParameters":{},"executeAtStart":"","type":"dashboard-component","options":{"compType":"DatePicker","displayFormat":"DD/MM/YYYY","outputFormat":""}},"9n6o76moyw":{"type":"dashboard-component","options":{"compType":"Image","src":"https://www.webstix.com/wp-content/uploads/2017/01/Google_Chrome_for_Android-_Android_5.0_Logo.png","id":"img1","alt":"","boxComp":"Image","boxPx":"","boxColor":	

	"#FFFFFF","bgOpacity":"0.1"},"uid":"9n6o76moyw","name":"9n6o76moyw","executeAtStart":"","requestParameters":{},"label":"Logo","gs_attr":{"x":0,"y":16,"height":16,"width":6}}},"css":"","script":""},"dir":"1508770339608","fileName":"DashboardSave"}	

Response Output(JSON Format)	{
	"status":1,"response":{"uuid":"b8d6f85e-ab65-4970-b1e6-862ee7bafcba.efwdd","message":"Design is saved successfully"}
	}	

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message with generated uuid for dashboard.	
Service Status	200 OK	
Screenshot		

http://192.168.2.156:8085/hi-ee//services

```
type:dashboard
serviceType:efwdd
service:designer
formData:{"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div
class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div
class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-title\">Destination wise monthly travel expense</div><div
class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-
x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-
type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div>
<div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div
class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div
```

Status: 200 OK Time: 124 ms Size: 427 B

```
{"status":1,"response":{"uuid":"b8d6f85e-ab65-4970-b1e6-862ee7bafcba.efwdd","message":"Design is saved successfully"}}
```

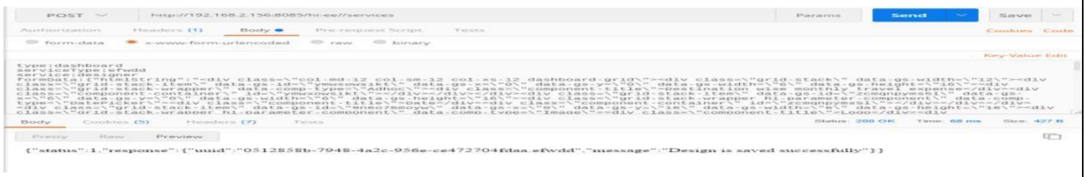
5.1.3 SaveAs Dashboard

URL	/services
Description	It allows user to re-save the existing dashboard
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwdd&service=designer&formData={"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-title\">Destination wise monthly travel expense</div><div class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div class=\"component-container\" id=\"9n6o76moyw\"></div></div></div></div></div>\n<script>$(document).ready(function(){\n$(\".grid-stack\").gridstack({width: 12, cellHeight: \"10px\", verticalMargin: \"10px\", staticGrid: true});\n $.gridStackCSSGenerator(12, \"hi-gridstack\");\n\n var dashboard = Dashboard|| {},\n gs = $(\".grid-

	stack\").data(\"gridstack\");\n dashboard.setGridOptions = function(options) {\n if (!_.isObject(options)) return;\n\n if (typeof options.width === \"number\") {\n gs.setGridWidth(options.width);\n $.gridStackCSSGenerator(options.width, \"hi-gridstack\");\n }\n\n if (typeof options.verticalMargin !== \"undefined\") {\n gs.verticalMargin(options.verticalMargin);\n }\n\n if (typeof options.horizontalMargin !== \"undefined\") {\n var marginStyles = document.getElementById(\"hi-gridstack-margins\");\n\n if (!marginStyles) {\n marginStyles = document.createElement(\"style\");\n marginStyles.type = \"text/css\";\n marginStyles.rel = \"stylesheet\";\n marginStyles.id = \"hi-gridstack-margins\";\n document.getElementsByTagName(\"head\")[0].appendChild(marginStyles);\n }\n\n var css = \".grid-stack .grid-stack-item .grid-stack-wrapper {\" +\n \"margin:\" + options.horizontalMargin + \";\" +\n \"width: calc(100% - \" + options.horizontalMargin + \" - \" + options.horizontalMargin+ \");\" +\n \"}\";\n\n marginStyles.textContent = css;\n }\n\n if (typeof options.cellHeight !== \"undefined\") {\n gs.cellHeight(options.cellHeight);\n }\n };\n\n$(\"<style id='hi-designer-styles'></style>\").appendTo(\"head\")\nDashboard.setVariable(\"Year\", [\"2015\"]);\nDashboard.setVariable(\"Month Name\", [\"'February'\"]);\n\nDashboard.init([{\"type\":\"adhoc\",\"options\":{\"dir\":\"1463377807724\\u002F1463378012748\",\"file\":\"e211af96-d633-4dc7-9460-3612970317fd.report\",\"ext\":\"report\"},\"uid\":\"ymwxowsikt\",\"name\":\"ymwxowsikt\",\"label\":\"Destination wise monthly travel expense\",\"executeAtStart\":true,\"htmlElementId\":\"#ymwxowsikt\"},{\"parameters\":null,\"name\":\"zcmqnpyms5l\",\"uid\":\"zcmqnpyms5l\",\"label\":\"Date\",\"requestParameters\":{},\"executeAtStart\":\"\",\"type\":\"datepicker\",\"options\":{\"displayFormat\":\"DD\\u002FMM\\u002FYYYY\",\"outputFormat\":\"\"},\"htmlElementId\":\"#zcmqnpyms5l\"},{\"type\":\"image\",\"options\":{\"src\":\"https:\\u002F\\u002Fwww.webstix.com\\u002Fwp-content\\u002Fuploads\\u002F2017\\u002F01\\u002FGoogle_Chrome_for_Android-_Android_5.0_Logo.png\",\"id\":\"img1\",\"alt\":\"\",\"boxComp\":\"Image\",\"boxPx\":\"\",\"boxColor\":\"#FFFFFF\",\"bgOpacity\":\"0.1\"},\"uid\":\"9n6o76moyw\",\"name\":\"9n6o76moyw\",\"executeAtStart\":\"\",\"requestParameters\":{},\"label\":\"Logo\",\"htmlElementId\":\"#9n6o76moyw\"}]);\n\n});</script>","state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadata":{"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options":{"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460-3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise monthly travel expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms5l":{"gs_attr":{"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms5l","uid":"zcmqnpyms5l","label":"Date","requestParameters":{},"executeAtStart":"","type":"dashboard-component","options":{"compType":"DatePicker","displayFormat":"DD/MM/YYYY","outputFormat":""}},"9n6o76moyw":{"type":"dashboard-component","options":{"compType":"Image","src":"https://www.webstix.com/wp-content/uploads/2017/01/Google_Chrome_for_Android-_Android_5.0_Logo.png","id":"img1","alt":"","boxComp":"Image","boxPx":"","boxColor":"#FFFFFF","bgOpacity":"0.1"},"uid":"9n6o76moyw","name":"9n6o76moyw","executeAtStart":"","requestParameters":{},"label":"Logo","gs_attr":{"x":0,"y":16,"height":16,"width":6}}},"css":"","script":""},"dir":"1508846783518","fileName":"DashboardSaveAs"}' http://192.168.2.156:8085/hi-ee//services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as adhoc dashboard type.
serviceType:	efwdd	Servicetype as efwdd
service:	designer	Service to dashboard designer.
formData:	{"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-	Formdata contains the dashboard html contents and the component array with all the added components to it with location of dashboard file and the file name.

	title\">Destination wise monthly travel expense</div><div class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div class=\"component-container\" id=\"9n6o76moyw\"></div></div></div></div></div>\n<script>$(document).ready(function(){\n$(\".grid-stack\").gridstack({width: 12, cellHeight: \"10px\", verticalMargin: \"10px\", staticGrid: true});\n $.gridStackCSSGenerator(12, \"hi-gridstack\");\n\n var dashboard = Dashboard || {},\n gs = $(\".grid-stack\").data(\"gridstack\");\n dashboard.setGridOptions = function(options) {\n if (!_.isObject(options)) return;\n\n if (typeof options.width === \"number\") {\n gs.setGridWidth(options.width);\n $.gridStackCSSGenerator(options.width, \"hi-gridstack\");\n }\n\n if (typeof options.verticalMargin !== \"undefined\") {\n gs.verticalMargin(options.verticalMargin);\n }\n\n if (typeof options.horizontalMargin !== \"undefined\") {\n var marginStyles = document.getElementById(\"hi-gridstack-margins\");\n\n if (!marginStyles) {\n marginStyles = document.createElement(\"style\");\n marginStyles.type = \"text/css\";\n marginStyles.rel = \"stylesheet\";\n marginStyles.id = \"hi-gridstack-margins\";\n document.getElementsByTagName(\"head\")[0].appendChild(marginStyles);\n }\n\n var css = \".grid-stack .grid-stack-item .grid-stack-wrapper {\" +\n \"margin:\" + options.horizontalMargin + \";\" +\n \"width: calc(100% - \" + options.horizontalMargin + \" - \" + options.horizontalMargin+ \");\" +\n \"}\";\n\n marginStyles.textContent = css;\n }\n\n if (typeof options.cellHeight !== \"undefined\") {\n gs.cellHeight(options.cellHeight);\n }\n };\n\n$(\"<style id='hi-designer-styles'></style>\").appendTo(\"head\")\nDashboard.setVariable(\"Year\", [\"2015\"]);\nDashboard.setVariable(\"Month Name\", [\"'February'\"]);\n\nDashboard.init([{\"type\":\"adhoc\",\"options\":{\"dir\":\"1463377807724\\u002F1463378012748\",\"file\":\"e211af96-d633-4dc7-9460-3612970317fd.report\",\"ext\":\"report\"},\"uid\":\"ymwxowsikt\",\"name\":\"ymwxowsikt\",\"label\":\"Destination wise monthly travel expense\",\"executeAtStart\":true,\"htmlElementId\":\"#ymwxowsikt\"},{\"parameters\":null,\"name\":\"zcmqnpyms5l\",\"uid\":\"zcmqnpyms5l\",\"label\":\"Date\",\"requestParameters\":{},\"executeAtStart\":\"\",\"type\":\"datepicker\",\"options\":{\"displayFormat\":\"DD\\u002FMM\\u002FYYYY\",\"outputFormat\":\"\"},\"htmlElementId\":\"#zcmqnpyms5l\"},{\"type\":\"image\",\"options\":{\"src\":\"https:\\u002F\\u002Fwww.webstix.com\\u002Fwp-content\\u002Fuploads\\u002F2017\\u002F01\\u002FGoogle_Chrome_for_Android-_Android_5.0_Logo.png\",\"id\":\"img1\",\"alt\":\"\",\"boxComp	

	\":\"Image\",\"boxPx\":\"\",\"boxColor\":\"#FFFFFF\",\"bgOpacity\":\"0.1\"},\"uid\":\"9n6o76moyw\",\"name\":\"9n6o76moyw\",\"executeAtStart\":\"\",\"requestParameters\":{},\"label\":\"Logo\",\"htmlElementId\":\"#9n6o76moyw\"}]);\n\n});</script>","state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadata":{"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options":{"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460-3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise monthly travel expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms5l":{"gs_attr":{"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms5l","uid":"zcmqnpyms5l","label":"Date","requestParameters":{},"executeAtStart":"","type":"dashboard-component","options":{"compType":"DatePicker","displayFormat":"DD/MM/YYYY","outputFormat":""}},"9n6o76moyw":{"type":"dashboard-component","options":{"compType":"Image","src":"https://www.webstix.com/wp-content/uploads/2017/01/Google_Chrome_for_Android-_Android_5.0_Logo.png","id":"img1","alt":"","boxComp":"Image","boxPx":"","boxColor":"#FFFFFF","bgOpacity":"0.1"},"uid":"9n6o76moyw","name":"9n6o76moyw","executeAtStart":"","requestParameters":{},"label":"Logo","gs_attr":{"x":0,"y":16,"height":16,"width":6}}},"css":"","script":""},"dir":"1508846783518","fileName":"DashboardSaveAs"}	
Response Output(JSON Format)	{ "status":1,"response":{"uuid":"0512858b-7948-4a2c-956e-ce472704fdaa.efwdd","message":"Design is saved successfully"} }	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message with generated uuid for dashboard.	
Service Status	200 OK	
Screenshot		

POST http://192.168.2.156:8085/hi-ee//services

```
type:dashboard
serviceType:efwdd
service:designer
formData:{"htmlString":"<div class=\"col-md-12 col-sm-12 col-xs-12 dashboard-grid\"><div class=\"grid-stack\" data-gs-width=\"12\"><div
class=\"grid-stack-item\" data-gs-id=\"ymwxowsikt\" data-gs-x=\"0\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div
class=\"grid-stack-wrapper\" data-comp-type=\"Adhoc\"><div class=\"component-title\">Destination wise monthly travel expense</div><div
class=\"component-container\" id=\"ymwxowsikt\"></div></div></div><div class=\"grid-stack-item\" data-gs-id=\"zcmqnpyms5l\" data-gs-
x=\"6\" data-gs-y=\"0\" data-gs-width=\"6\" data-gs-height=\"16\"><div class=\"grid-stack-wrapper hi-parameter-component\" data-comp-
type=\"DatePicker\"><div class=\"component-title\">Date</div><div class=\"component-container\" id=\"zcmqnpyms5l\"></div></div></div>
<div class=\"grid-stack-item\" data-gs-id=\"9n6o76moyw\" data-gs-x=\"0\" data-gs-y=\"16\" data-gs-width=\"6\" data-gs-height=\"16\"><div
class=\"grid-stack-wrapper hi-parameter-component\" data-comp-type=\"Image\"><div class=\"component-title\">Logo</div><div
```

Status: 200 OK Time: 68 ms Size: 427 B

```
{"status":1,"response":{"uuid":"0512858b-7948-4a2c-956e-ce472704fdaa.efwdd","message":"Design is saved successfully"}}
```

5.2 Edit Dashboard

Page : http://192.168.2.156:8085/hi-ee/designer-edit.html

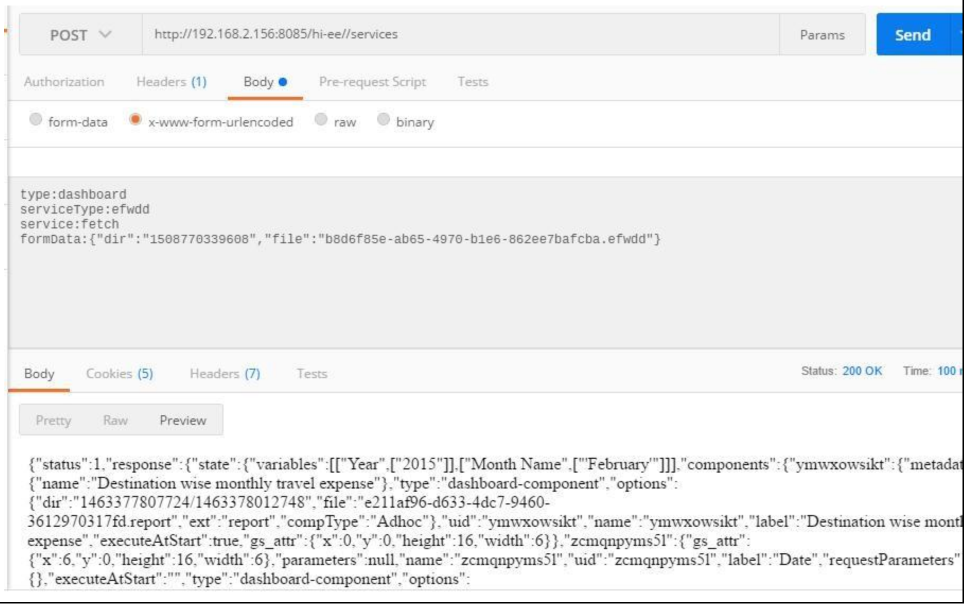
Note : Below API is to select dashboard file which you want to edit .So after that you can do changes in dashboard file and save file /saveAs file

Fetch

URL	/services
Description	It allows user to edit the existing dashboard using dashboard designer.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwdd&service=fetch&formData={"dir":"1508770339608","file":"b8d6f85e-ab65-4970-b1e6-862ee7bafcba.efwdd"}' http://192.168.2.156:8085/hi-ee//services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as adhoc dashboard type.
serviceType:	efwdd	Servicetype as efwdd
service:	fetch	Service to fetch dashboard designer for edit.
formData:	{"dir":"1508770339608","file":"cbc8a246-18f2-4c42-8355-1b28345c8797.efwdd"}	Formdata contains the dashboard file name and the directory of file to edit the dashboard .

Response Output(JSON Format)	{"status":1,"response":{"state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadata":{"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options":{"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460-3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise monthly travel expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms5l":{"gs_attr":{"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms5l","uid":"zcmqnpyms5l","label":"Date","requestParameters":{},"executeAtStart":"","type":"dashboard-component","options":{"compType":"DatePicker","displayFormat":"DD/MM/YYYY","outputFormat":""}},"9n6o76moyw":{"type":"dashboard-component","options":{"compType":"Image","src":"https://www.webstix.com/wp-content/uploads/2017/01/Google_Chrome_for_Android-_Android_5.0_Logo.png","id":"img1","alt":"","boxComp":"Image","boxPx":"","boxColor":"#FFFFFF","bgOpacity":"0.1"},"uid":"9n6o76moyw","name":"9n6o76moyw","executeAtStart":"","requestParameters":{},"label":"Logo","gs_attr":{"x":0,"y":16,"height":16,"width":6}}},"css":"","script

	":""},"reportName":"DashboardSave"}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the details of variables, components used in dashboard alog with name of dashboard file.
Service Status	200 OK
Screenshot	 POST http://192.168.2.156:8085/hi-ee//services Params Send Authorization Headers (1) Body Pre-request Script Tests form-data x-www-form-urlencoded raw binary Body Cookies (5) Headers (7) Tests Status: 200 OK Time: 100 Pretty Raw Preview type:dashboard serviceType:efwdd service:fetch formData:{"dir":"1508770339608","file":"b8d6f85e-ab65-4970-b1e6-862ee7bafcba.efwdd"} {"status":1,"response":{"state":{"variables":[["Year",["2015"]],["Month Name",["'February'"]]],"components":{"ymwxowsikt":{"metadat {"name":"Destination wise monthly travel expense"},"type":"dashboard-component","options": {"dir":"1463377807724/1463378012748","file":"e211af96-d633-4dc7-9460- 3612970317fd.report","ext":"report","compType":"Adhoc"},"uid":"ymwxowsikt","name":"ymwxowsikt","label":"Destination wise montl expense","executeAtStart":true,"gs_attr":{"x":0,"y":0,"height":16,"width":6}},"zcmqnpyms51":{"gs_attr": {"x":6,"y":0,"height":16,"width":6},"parameters":null,"name":"zcmqnpyms51","uid":"zcmqnpyms51","label":"Date","requestParameters" {},"executeAtStart":"","type":"dashboard-component","options":

6. DASHBOARD

Dashboard APIs are accessible in HI module, Designer module and Instant BI.

Needed JS files

- jquery.js
- bootstrap.js
- backbone.js
- moment.js
- jquery-ui.js
- d3.js
- daterangepicker.js
- select.js
- gridstack.js
- c3.js
- H3.js
- validator.js
- dashboard.js

- hdiui.js
- utilities.js
- user_utils.js
- tooltip.js
- file-browser.js

6.1 Dashboard Components

The components that are available to use are:

1. Button
2. Charts
3. Custom
4. Date Picker
5. Date-Range Picker
6. Select
7. Select2
8. Adhoc
9. Text
10. Slider
11. Input

All the components have some basic and common configuration. The configuration options are listed below.

Option	**Format**	**Default**	**Description**
name	string	(required)	This will be used as **id** for the component and also refereeing it in other options.
label	string	Not mandatory	This is the label of component you can give any label for component.
type	string	(required)	Type of the component to be used. Each type is explained below in detail.
options	object	Depends on **type**	Each component has options and the options are explained in their respective component.
listeners	array	[ ]	A list of variables on whose change, this component needs to be updated.
parameters	array	[ ]	List of variables that are to be set by this component. This also depends on **type**.
htmlElementId	string	(required)	This is jQuery selector of the HTML element. Any selector can be used and for best practice, use **id.**
requestParameters	object	{ }	List of parameters that are to be sent with update request of the component. These parameters consist of **key** and **value** pairs.
executeAtStart	Boolean	false	This specifies if the component is set to be updated or initialised after Dashboard.init()
map	int	Depends on **type**	This is set along with **requestParameters** to the Data-Layer and must watch for **map-id** there.

6.1.1 Button

Button component can be accessed by using **type: “button”**. If user intends to trigger component update manually instead of using listeners, then this component is helpful. This component has one additional configuration and also has different behaviour for **executeAtStart**.

6.1.1.1 Additional Configuration

Option	**Format**	**Default**	**Description**
triggers	array	[ ]	Array should contain name properties which are used while defining the components. It will trigger update of all the components that are present in the array (on click).
executeAtStart	boolean	false	For button, this specifies if the components which are specified in the array should trigger on update or initialization after Dashboard.init()

6.1.1.2 Options

Option	**Format**	**Default**	**Description**
display	String	“Submit”	The text to be displayed on button.
classes	String	“btn-primary”	To modify / add custom action and add extra CSS properties, this can be done with space separated string of classes to the button.
Examples	1. Button will be filled with blue background colour with its text as “Go”. `{ display: “Go” }` 2. Button will be filled with green background colour with its text as “Success”. `{ display: “Success”, classes: “btn-success btn-block” }` 3. Button will be filled with red background colour with its text as “Submit”. `{ classes: “btn-danger” }` **Example:** Dashboard.init([{"name":"k3gw7izmwho","uid":"k3gw7izmwho","label":"Submit","requestParameters":{},"executeAtStart":"","type":"button","triggers":null,"options":{"uid":"k3gw7izmwho","display":"","classes":"btn-primary"},"htmlElementId":"#k3gw7izmwho"}]);		

Note: Go through bootstrap classes in bootstrap frame-work to get more insight.

6.1.2 Charts

Charts component can be accessed by using **type: “chart”**. Apart from common configurations, chart component has one additional configuration.

6.1.2.1 Additional Configuration

Option	Format	Default	Description
vf	object	(required)	It has keys namely **file,id** and **dir**. **file**: The name of the file that contains the information about the chart visualization. **id**: The id of the chart form the visualization file, that is to be displayed. **dir:** Directory of the vf file

6.1.2.2 Options

Option	Format	Default	Description
actions	array	[]	Creates a custom menu and it must be an array of objects. Each object must contain a key and value pair where key will be used as id of the element and value. The objects allow for grouping of controls. It has to be noted that the actions are to be specified manually by the user.
menuLabel	String	“Actions”	The label to be used for custom menu.
Example	Dashboard.init([{"name":"a2fyghpympt","uid":"a2fyghpympt","label":"Chart","requestParameters":{"start_date":"start_date","end_date":"end_date"},"executeAtStart":true,"type":"chart","listeners":["start_date","end_date"],"vf":{"id":1,"file":"sample_report.efwvf","dir":"1463377807724\u002F1463377978248\u002FSample EFW Report"},"htmlElementId":"#a2fyghpympt"}]);		

6.1.2.3 Custom

Select component can be accessed by using **type: “custom”**. This component can be used to write user specified custom script and the configuration includes:

6.1.2.4 Configuration

Option	Format	Default	Description
name	string	(required)	This will be used as **id** for the component and also refereeing it in other options.
type	string	(required)	Type is custom.
listeners	array	[ ]	A list of variables on whose change, **customScript** will be called.
htmlElementId	string	(required)	This is jQuery selector of the HTML element. Any selector can be used and for best practice, use **id.**
executeAtStart	Boolean	false	This specifies if the component is set to be updated or initialised after Dashboard.init()
customScript	function	(required)	This function has two arguments. First is the reference to the element of jQuery object given

			in htmlElementId and the second is an object containing the variables specified in listeners with names a key and its value as value Custom script is nothing but the required javascript.

6.1.2.5 Options

Option	**Format**	**Default**	**Description**
Example	Dashboard.init([{"name":"wedb26kldm","uid":"wedb26kldm","label":"Custom","requestParameters":{},"executeAtStart":true,"type":"custom","listeners":["start_date","end_date"],"customScript":function anonymous(elem,params /*``*/) { var startDate=Dashboard.getVariable('start_date'); var newstartDate=startDate.concat('Helical'); Dashboard.setVariable('ModifiedstartDate',newstartDate); },"htmlElementId":"#wedb26kldm"}]);		

6.1.3 Date Picker and Date-Range Picker

Date Picker component and Date-Range Picker component can be accessed by using **type: “datepicker”** and **type: “daterangepicker”**. Both components almost share identical options and the options are:

6.1.3.1 Common Options

Option	**Format**	**Default**	**Description**
displayFormat	string	“YYYY-MM-DD”	Determines date format that has to be displayed to the user. The format used is similar to moment.js library.
outputFormat	string	“YYYY-MM-DD”	Determines the format of the date that is stored in the variable and used while querying databases. The format used is similar to moment.js library.

6.1.3.2 Extra options for daterangepicker

Option	**Format**	**Default**	**Description**
separator	string	“ to ”	The separator to be used while displaying the date.

6.1.3.3 Date formats

	Format	**Output**
Month	M	1 2 ... 11 12
	Mo	1st 2nd ... 11th 12th
	MM	01 02 ... 11 12
	MMM	Jan Feb ... Nov Dec
	MMMM	January February ... November December
Quarter	Q	1 2 3 4
Day of Month	D	1 2 ... 30 31

	Do	1st 2nd ... 30th 31st
	DD	01 02 ... 30 31
Day of Year	DDD	1 2 ... 364 365
	DDDo	1st 2nd ... 364th 365th
	DDDD	001 002 ... 364 365
Day of Week	d	0 1 ... 5 6
	do	0th 1st ... 5th 6th
	dd	Su Mo ... Fr Sa
	ddd	Sun Mon ... Fri Sat
	dddd	Sunday Monday ... Friday Saturday
Day of Week (Locale)	e	0 1 ... 5 6
Day of Week (ISO)	E	1 2 ... 6 7
Week of Year	w	1 2 ... 52 53
	wo	1st 2nd ... 52nd 53rd
	ww	01 02 ... 52 53
Week of Year (ISO)	W	1 2 ... 52 53
	Wo	1st 2nd ... 52nd 53rd
	WW	01 02 ... 52 53
Year	YY	70 71 ... 29 30
	YYYY	1970 1971 ... 2029 2030
Week Year	gg	70 71 ... 29 30
	gggg	1970 1971 ... 2029 2030
Week Year (ISO)	GG	70 71 ... 29 30
	GGGG	1970 1971 ... 2029 2030
AM / PM	A	AM PM
	a	am pm
Hour	H	0 1 ... 22 23
	HH	00 01 ... 22 23
	h	1 2 ... 11 12
	hh	01 02 ... 11 12
Minute	m	0 1 ... 58 59
	mm	00 01 ... 58 59
Second	s	0 1 ... 58 59
	ss	00 01 ... 58 59
Fractional Second	S	0 1 ... 8 9
	SS	0 1 ... 98 99
	SSS	0 1 ... 998 999

Timezone	Z	-07:00 -06:00 ... +6:00 +7:00
	ZZ	-0700 -0600 ... +600 +700
Unix Timestamp	X	1360013296
Examples	**For Date Pickers**	
	19/10/16	{ displayFormat: “DD/MM/YY” }
	2016/10/16	{ displayFormat: “YYYY/MM/DD” }
	10-19-2016	{ displayFormat: “DD-MM-YYYY” }
	Dashboard.init([{"parameters":null,"name":"oksxkn6s3q8","uid":"oksxkn6s3q8","label":"Date Picker","requestParameters":{},"executeAtStart":"","type":"datepicker","options":{"uid":"oksxkn6s3q8","iframe":true,"displayFormat":"DD-MM-YYYY","outputFormat":"DD-MM-YYYY"},"htmlElementId":"#oksxkn6s3q8"}]);	
	For Date Range Pickers	
	19/10/16 29/10/16	{ displayFormat: “DD/MM/YY” }

	2016/10/19
	2016/10/29
		{
		displayFormat: “YYYY/MM/DD”,
		separator: “ upto ”
		}

	19/10/16
	29/10/16
		{
		displayFormat: “DD/MM/YY”,
		}

	10-19-2016
	10-29-2016
		{
		displayFormat: “MM/DD/YYYY”,
		}

	Dashboard.init([{"parameters":["start_date","end_date"],"name":"wp1bpw2tjde","uid":"wp1bpw2tjde","label":"Date-Range Picker","requestParameters":{},"executeAtStart":"","type":"daterangepicker","options":{"uid":"wp1bpw2tjde","iframe":true,"displayFormat":"YYYY-MM-DD HH:mm:ss","outputFormat":"YYYY-MM-DD HH:mm:ss"},"htmlElementId":"#wp1bpw2tjde"}]);	

6.1.4 Select

Select component can be accessed by using **type: “select”**. Select has two modes which are single and multiple which can be set via **options**.

Option	**Format**	**Default**	**Description**
multiple	Boolean	false	This option will determine whether the component must be single (false) or multi-select (true).
display	String	(required)	The key / column of the data should be used as display value.
value	String	(required)	The key / column of the data should be used as actual value.

Examples	1. When single select is enabled.
	{
	multiple: false,
	display: “TRAVELDETAILS_TRAVEL_TYPE”,
	value: “TRAVELDETAILS_TRAVEL_TYPE”
	}Dashboard.init([{"dataSource":{"location":"1463377807724\u002F1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata"},"parameters":null,"name":"atrzmesz06","uid":"atrzmesz06","alias":"TRAVELDETAILS_TRAVEL_TYPE","label":"Select2","column":"TRAVELDETAILS.TRAVEL_TYPE","adhoc":true,"requestParameters":{},"executeAtStart":true,"type":"select2","database":"HIUSER","options":{"display":"TRAVELDETAILS_TRAVEL_TYPE","value":"TRAVELDETAILS_TRAVEL_TYPE","multiple":false,"uid":"atrzmesz06","iframe":true,"placeholder":"Please select a value"},"listeners":null,"htmlElementId":"#atrzmesz06"}]);
	
	2. When multi-select is enabled:
	{
	multiple: true,
	display: “TRAVELDETAILS_TRAVEL_TYPE”,
	value: “TRAVELDETAILS_TRAVEL_TYPE”
	}

	Dashboard.init([{"dataSource":{"location":"1463377807724\u002F1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata"},"parameters":null,"name":"atrzmesz06","uid":"atrzmesz06","alias":"TRAVELDETAILS_TRAVEL_TYPE","label":"Select","column":"TRAVELDETAILS.TRAVEL_TYPE","adhoc":true,"requestParameters":{},"executeAtStart":true,"type":"select","database":"HIUSER","options":{"display":"TRAVELDETAILS_TRAVEL_TYPE","value":"TRAVELDETAILS_TRAVEL_TYPE","multiple":false,"uid":"atrzmesz06","iframe":true},"listeners":null,"htmlElementId":"#atrzmesz06"}]);

Note: It has to be noted that **multiple: false** will set the variable as string where as **multiple: true** will set the variable as an array.

6.1.5 Select with search (Select2)

Select2 component can be used by using **type: “select2”**. Select2 has two modes which are single and multiple which can be set via **options**.Select2 is the select single or multiple options with search enabled.

Option	**Format**	**Default**	**Description**
multiple	Boolean	False	This option will determine whether the component must be single (false) or multi-select (true).
display	String	(required)	The key / column of the data should be used as display value.
value	String	(required)	The key / column of the data should be used as actual value.

placeholder	String	“Please select a value”	This text will be shown when no option is selected.

Examples	1. With single select:
	{
	multiple: false,
	placeholder: “Please select a value”,
	display: “TRAVELDETAILS_TRAVEL_TYPE”,
	value: “TRAVELDETAILS_TRAVEL_TYPE”
	}
	
	2. With multiple select:
	{
	multiple: true,
	placeholder: “Please select a value”,
	display: “TRAVELDETAILS_TRAVEL_TYPE”,
	value: “TRAVELDETAILS_TRAVEL_TYPE”
	}		

	Dashboard.init([{"dataSource":{"location":"1463377807724\u002F1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata"},"parameters":null,"name":"atrzmesz06","uid":"atrzmesz06","alias":"TRAVELDETAILS_TRAVEL_TYPE","label":"Select 2","column":"TRAVELDETAILS.TRAVEL_TYPE","adhoc":true,"requestParameters":{},"executeAtStart":true,"type":"select2","database":"HIUSER","options":{"display":"TRAVELDETAILS_TRAVEL_TYPE","value":"TRAVELDETAILS_TRAVEL_TYPE","multiple":false,"uid":"atrzmesz06","iframe":true,"placeholder":"Please select a value"},"listeners":null,"htmlElementId":"#atrzmesz06"}]);		

Note: Select2 component will also provide search facility on variables in options. To explore more, dive into Select2 plugin to get insight.

6.1.6 Adhoc

Adhoc component can be used by using **type: “adhoc”**. Adhoc has **options** for report object with directory and ereport file. The main advantage of adhoc component, it is having iframe modes as true/false whether to render adhoc component with/without iframe.

Option	**Format**	**Default**	**Description**
iframe	Boolean	true	This option will determine whether the component must without iframe(false) or with iframe (true).
styles	object	Default styles	Adhoc component default styles with box color , border etc.
report	object	(required)	Report object having the dir name and file name of report.
Examples	Dashboard.init([{"name":"gj2cfumzu18","uid":"gj2cfumzu18","label":"Adhoc","requestParameters":{},"executeAtStart":true,"type":"adhoc","options":{"uid":"gj2cfumzu18","iframe":true,"styles":{"boxComp":"Adhoc","borderComp":"Adhoc","value":true,"newValue":true,"boxPx":"","borderPx":"1px","boxColor":"#ffffff","borderColor":"#9d9fa1"},"report":{"dir":"1463377807724\u002F1463378012748","file":"94b8d841-bf01-4ff3-8e9e-		

	ac858ac8a52c.report"}},"listeners":null,"htmlElementId":"#gj2cfumzu18"}]);

6.1.7 Text

Text component can be used by using **type: "text"**. Text has **options** for writing text with text visualizations like textarea , fontsize, fontstyle,alignment , font family etc options.We can set **id** for the text component .

Option	**Format**	**Default**	**Description**
id	Number	Not mandatory	id for the text component
color	Text	Default font color	Font color for text
align	text	Default color	Alignment of the text
bgOpacity	Number	Default opacity	Background color opacity
fontsStyle	text	Default fontstyle	Text Fontstyle
fontSize	Number	Default fontsize	Text Fontsize
fontFamily	text	Default font	Text font
fontWeight	text	Default font weight	Text font weight
boxComp	text	Text as boxComp	Box component is text
bgcolor	text	Default bgcolor	Background color
Examples	Dashboard.init([{"type":"text","options":{"color":"#000000","align":"start","bgOpacity":"1","fontsStyle":"normal","uid":"bdte21z1g9d","fontSize":"12","value":true,"fontFamily":"Comic Sans MS","boxPx":"","boxColor":"#FFFFFF","textArea":"Sample Text Component","boxComp":"Text","id":"","fontsWeight":"normal","bgcolor":"#FFFFFF"},"uid":"bdte21z1g9d","name":"bdte21z1g9d","executeAtStart":"","requestParameters":{},"label":"","htmlElementId":"#bdte21z1g9d"}]);		

6.1.8 Slider

Slider component can be used by using **type: "slider"**. Slider has **options** for report object with directory and ereport file. The main advantage of adhoc component, it is having iframe modes as true/false whether to render adhoc component with/without iframe.

Option	**Format**	**Default**	**Description**
dataSource	object	(required)	This option will determine the dir name and metadata file name.
min	object	(required)	Min value dahsboard variable used to set min value for slider
max	object	(required)	Max value dahsboard variable used to set min value for slider
display	text	Not mandatory	Name of column on which slider is getting

			applied
value	text	Not mandatory	Name of column on which slider is getting applied
dataType	Numeric	default	Type of data for min-max values
aggregate	array	(required)	Array contains min-max column details with aggregate function.
Examples	Dashboard.init([{"orderBy":"","dataSource":{"location":"1513330650237", "metadataFileName":"a34dd8c2-ac7a-4eae-a357-6cce68af2cc9.metadata"},"parameters":["360_surveys_start_date"],"name":"mvf0anpcv3s","uid":"mvf0anpcv3s","alias":"360_surveys_start_date","databaseFunction":{"functionName":"sql.date.month","dataType":"numeric", "parameters":{"datetime":"360_surveys.start_date"}},"label":"Slider","column":"360_surveys.start_date","adhoc":true,"requestParameters":{},"executeAtStart":true,"type":"slider","database":"360_envision","refresh":true,"columns":[{"column":"360_surveys.start_date","alias":"max","aggregate":true},{"column":"360_surveys.start_date","alias":"min","aggregate":true}],"options":{"min":"min","max":"max","display":"360_surveys_start_date","value":"360_surveys_start_date","dataType":"numeric","uid":"mvf0anpcv3s","iframe":true},"aggregate":[{"column":"360_surveys.start_date","function":"db.generic.aggregate.max","alias":"max"},{"column":"360_surveys.start_date","function":"db.generic.aggregate.min","alias":"min"}],"listeners":null, "htmlElementId":"#mvf0anpcv3s"}]);		

6.1.9 Input

Input component can be used by using **type: “input”**. Input has **options** for report object with directory and ereport file. The main advantage of adhoc component, it is having iframe modes as true/false whether to render adhoc component with/without iframe.

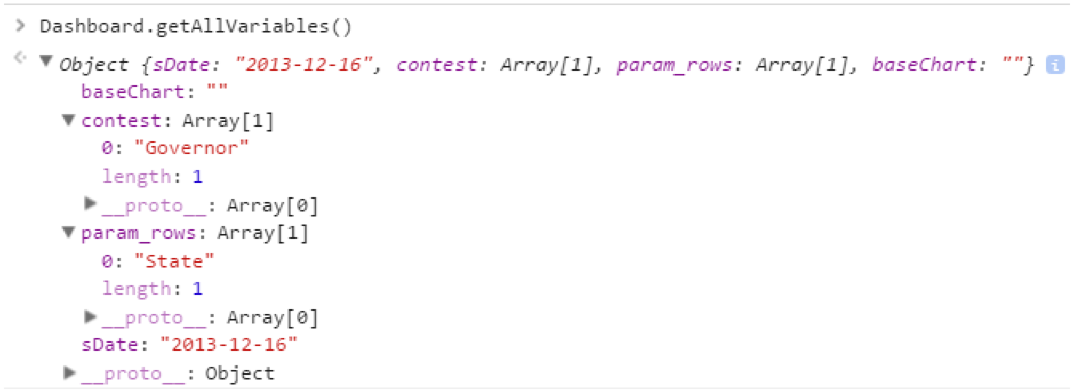
Option	**Format**	**Default**	**Description**
parameters	object	None selected	This option will contains the report parameters which are assigned to input
requestParameters	object	None selected	This option will contains the request parameters which are assigned to input
Examples	Dashboard.init([{"parameters":["start_date"],"name":"c8ps5oqj896","uid":"c8ps5oqj896","label":"Input","requestParameters":{"start_date":"start_date"},"executeAtStart":true,"type":"input","options":{"uid":"c8ps5oqj896","iframe":true},"listeners":null,"htmlElementId":"#c8ps5oqj896"}]);		

6.2 Dashboard.getAllVariables()

This method is used to get all the variables that are set in the report.

Description	This method is used to get all the variables that are set in the report. Variables can be retrieved before (default variables) and after triggering variables in the report.
Parameters	**Description**

dir	Directory of the report
file	Actual name of the report with its extension.

Example	**Step – 1:**
	Open a report. Below are the **dir** and **file** of the opened report.
	
	Parameter / **Values**
	Dir: 03 DemTech/DemTech
	File: Demo.efw
	
	Step – 2:
	Open browser's console and type the below JavaScript code to get all the variables of the opened report.
	
	Dashboard.getAllVariables();
	
	
	`> Dashboard.getAllVariables()`
	`<· ▼Object {sDate: "2013-12-16", contest: Array[1], param_rows: Array[1], baseChart: ""}`
	`baseChart: ""`
	`▼contest: Array[1]`
	`0: "Governor"`
	`length: 1`
	`▶__proto__: Array[0]`
	`▼param_rows: Array[1]`
	`0: "State"`
	`length: 1`
	`▶__proto__: Array[0]`
	`sDate: "2013-12-16"`
	`▶__proto__: Object`
	Figure 1: Get all variables of the opened report

6.3 Dashboard.setVariable(key, value)

Prerequisites:

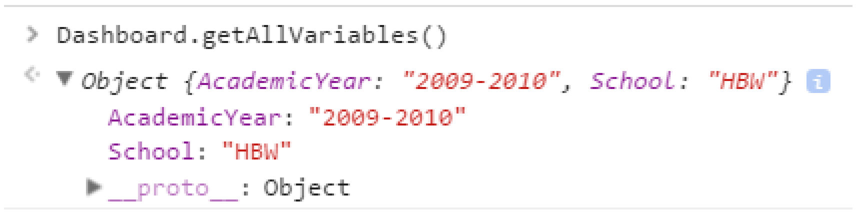
1. This method takes one or two arguments. If no argument is passed, "false" will be written as result.
2. If two arguments are given, the first argument must be a string which will be the name of the variable and second argument will be its value.
3. If one argument is given, it must be an object with key and value pairs in which key will be the name of the variable and value will be its value.
4. If a variable already exists with the same name, it will be updated set variable and set variable can be used anywhere and any number of times.

Description	Sets the variable in the opened report.
Parameters	**Description**
dir	Directory of the report
file	Actual name of the report with its extension.
Example	**Step – 1:** Open a report. Below are the **dir** and **file** of the opened report.

Parameter	Values
Dir	APSVA Dashboard
File	Demo.efw

Step – 2:
Open browser’s console and type the below JavaScript code to get all the variables of the opened report.

Dashboard.getAllVariables();



```
> Dashboard.getAllVariables()
<· ▼Object {AcademicYear: "2009-2010", School: "HBW"}
      AcademicYear: "2009-2010"
      School: "HBW"
    ▶__proto__: Object
```

Figure 2: Get all variables

Step – 3:
After getting all the variables, suppose if we want to set a new variable or modify the current variable, open browser’s console and type the following JavaScript code:

Foramt:

Dashboard.setVariable(“key”, “value”)

Usage example:

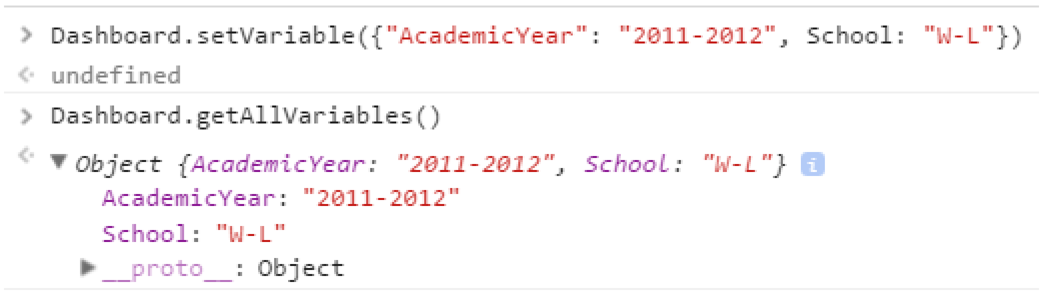
1. If we want to modify existing variables, get the “key” and assign the value to it (modify existing value).

   Dashboard.setVariable(“AcademicYear”, “2011-2012”);

2. We can also set multiple variables at once. For example:

   ```
   Dashboard.setVariable({
       AcademicYear: “2011-2012”,
       School: “W-L”
   })
   ```

   And if Dashboard.getAllVariables() is called, we get updated variables rather than default set variables.



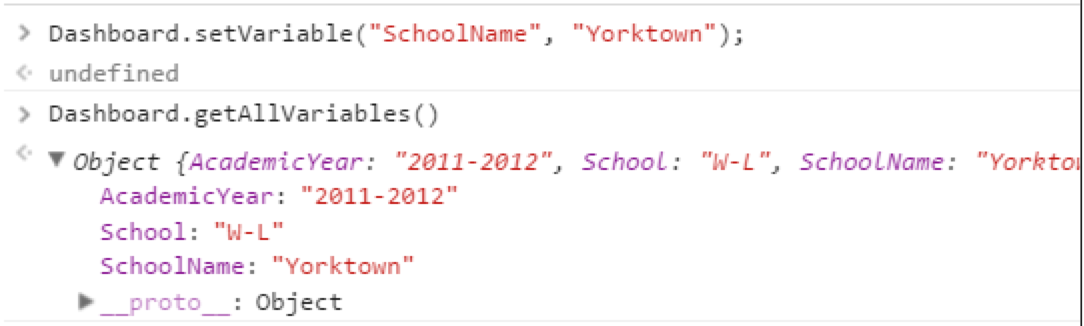
```
> Dashboard.setVariable({"AcademicYear": "2011-2012", School: "W-L"})
<· undefined
> Dashboard.getAllVariables()
<· ▼Object {AcademicYear: "2011-2012", School: "W-L"}
      AcademicYear: "2011-2012"
      School: "W-L"
    ▶__proto__: Object
```

Figure 3: Set multiple variables

3. If we want to set new variable then, we can achieve this by:

   Dashboard.setVariable("SchoolName", "Yorktown")

   Now, to check the variable which is set above, we can achieve this by calling Dashboard.getAllVariables().



```
> Dashboard.setVariable("SchoolName", "Yorktown");
< undefined
> Dashboard.getAllVariables()
< ▼Object {AcademicYear: "2011-2012", School: "W-L", SchoolName: "Yorkto
     AcademicYear: "2011-2012"
     School: "W-L"
     SchoolName: "Yorktown"
   ▶__proto__: Object
```

Figure 4: Set new variable

6.4 Dashboard.getVariable(varName)

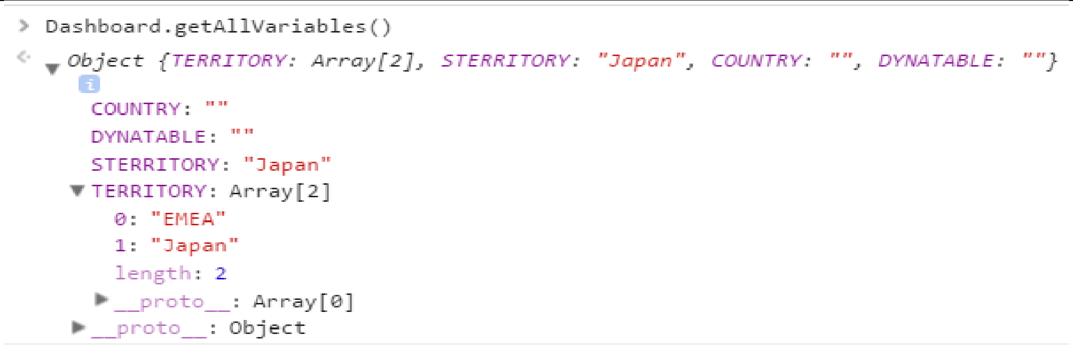
This method takes one argument which will be name of the variable.

Description	This method will retrieve the variable name and returns the value of the variable.
Parameters	**Description**
dir	Directory of the report
file	Actual name of the report with its extension.
Example	**Step – 1:** Open a report. Below are the **dir** and **file** of the opened report.

Parameter	**Values**
Dir	HelicalDemo
File	HelicalDemoFile.efw

Step – 2:
Open browser's console and type the below JavaScript code to get all the variables of the opened report.

Dashboard.getAllVariables();

	 ```
> Dashboard.getAllVariables()
< ▼Object {TERRITORY: Array[2], STERRITORY: "Japan", COUNTRY: "", DYNATABLE: ""}
COUNTRY: ""
DYNATABLE: ""
STERRITORY: "Japan"
▼TERRITORY: Array[2]
0: "EMEA"
1: "Japan"
length: 2
▶__proto__: Array[0]
▶__proto__: Object
`` Figure 5: Get all variables **Step – 3:** After getting all the variables, suppose if we want to get a variable, open browser's console and type the following JavaScript code: **Foramt:** Dashboard.getVariable("varName") **Usage example:** To get the value of the provided variable name, type the following JavaScript code in browser's console: Dashboard.getVariable("STERRITORY") Above JavaScript code will return **"Japan"** in browser's console.

6.5 Dashboard.addComponent(component)

This API needs component as a mandatory argument and the component is an object with certain parameters in it. Please refer components section for the parameters.

Description	This API adds a new component in the components that are already present.
Parameters	**Description**
name	Name of the component
type	Type of the component which can be button/select etc.
options	Options that are set in the defined component.
htmlElementId	Every component requires one id that refers the html element.
Example	**Step – 1:** Open a report. Below are the **dir** and **file** of the opened report. **Parameter** | **Values** Dir | APSVA Dashboard File | Demo.efw **Step – 2:** In this example, we have considered component as button and if no options are provided to the component, default display message and classes that are default will be considered and the component will be added. While creating the component, name is mandatory, type is mandatory and htmlElementId.
```

Step – 3:
Example:

```
var buttonComponent = {
                name: "Btn",
                type: "button",
                options:{
                       display: "Success",
                       classes: "btn btn-success"
                },
                htmlElementId: "#year1"
}

Dashboard.addComponent(buttonComponent)
```

Step – 4:
To check if the component is added or not, call this API in console:

Dashboard.componentViews

6.6 Dashboard.removeComponent(component)

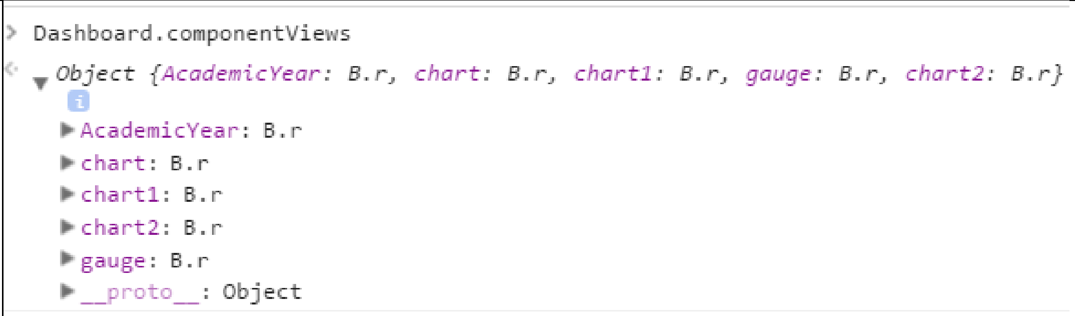
This method takes one argument as mandatory else it will return undefined.

Description	This method will remove the entire component from the opened report.
Parameters	**Description**
dir	Directory of the report
file	Actual name of the report with its extension.
component	Name of the component to reset.
Example	**Step – 1:** Open a report. Below are the **dir** and **file** of the opened report.

Parameter	**Values**
Dir	APSVA Dashboard
File	Demo.efw

Step – 2:
Open browser’s console and type the below JavaScript code to get all the component views of the opened report.

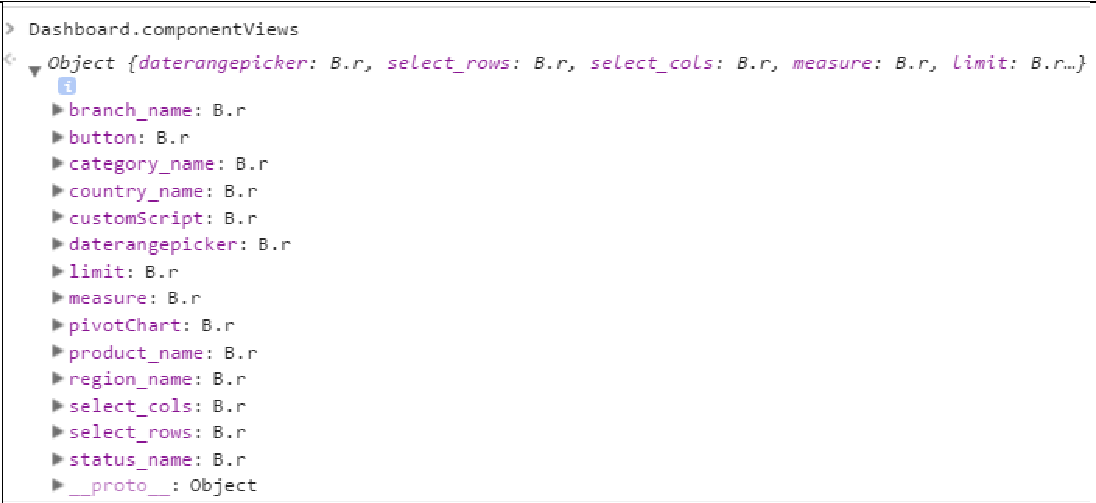
Dashboard.componentViews

	 ``` > Dashboard.componentViews < ▼Object {AcademicYear: B.r, chart: B.r, chart1: B.r, gauge: B.r, chart2: B.r} ▶AcademicYear: B.r ▶chart: B.r ▶chart1: B.r ▶chart2: B.r ▶gauge: B.r ▶__proto__: Object ``` Figure 6: Component views of the opened report **Step – 3:** To remove specific component, type below JavaScript code to achieve this: **Dashboard.removeComponent("AcademicYear")** The above method will remove the component "AcademicYear" (selector from report) with title "*School Year".

6.7 Dashboard.updateComponent(component)

This method takes one argument as mandatory else it will return false.

Description	This method will update all the values or parameters that are set in that component.
Parameters	**Description**
dir	Directory of the report
file	Actual name of the report with its extension.
component	Name of the component to reset.
Example	**Step – 1:** Open a report. Below are the **dir** and **file** of the opened report. **Parameter** / **Values** Dir / 04 Intec Capital File / Intec.efw **Step – 2:** Open browser's console and type the below JavaScript code to get all the component views of the opened report. **Dashboard.componentViews**



```
> Dashboard.componentViews
< ▼Object {daterangepicker: B.r, select_rows: B.r, select_cols: B.r, measure: B.r, limit: B.r…}
    ▶branch_name: B.r
    ▶button: B.r
    ▶category_name: B.r
    ▶country_name: B.r
    ▶customScript: B.r
    ▶daterangepicker: B.r
    ▶limit: B.r
    ▶measure: B.r
    ▶pivotChart: B.r
    ▶product_name: B.r
    ▶region_name: B.r
    ▶select_cols: B.r
    ▶select_rows: B.r
    ▶status_name: B.r
    ▶__proto__: Object
```

Figure 7: Component views of the opened report

Step – 3:
To reset or clear specific component, type below JavaScript code to achieve this:

Dashboard.resetComponent("select_rows")

The above method will return "true" in the console.

Step – 4:
Click on Parameters in the report, and check "Select Rows" section. Section should be blank with no values or parameters.

Step – 5:
Now to update the components in "Select Rows", type below JavaScript code to achieve this:

Format:

Dashboard.updateComponent(component)

Usage:

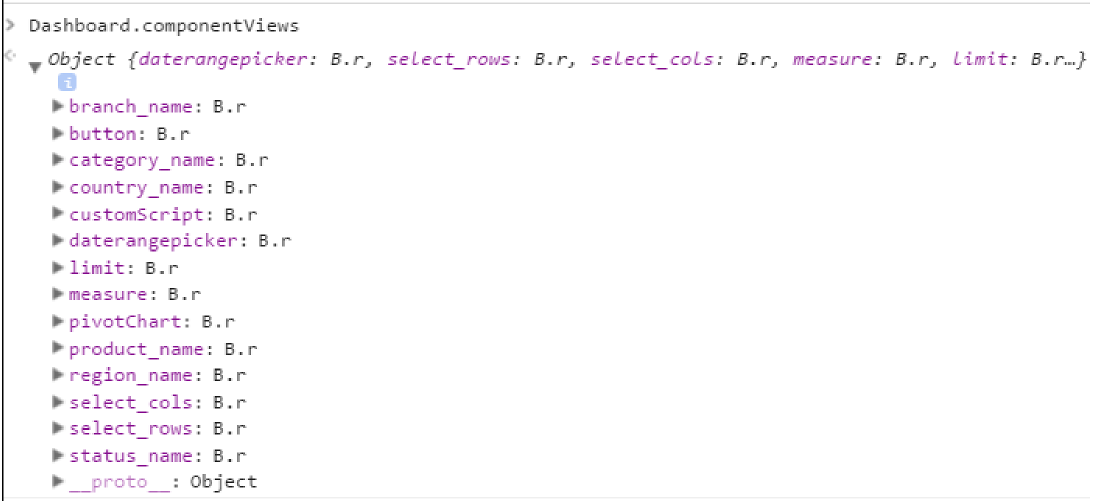
Dashboard.updateComponent("select_rows")

This method will update "select_rows" component and returns "true" in the console.

6.8 Dashboard.resetComponent(component)

This method takes one argument as mandatory else it will return false.

Description	This method will reset all the values or parameters that are set in that component without triggering update.
Parameters	**Description**
dir	Directory of the report
file	Actual name of the report with its extension.
component	Name of the component to reset.

Example	**Step – 1:**
	Open a report. Below are the **dir** and **file** of the opened report.
	
	Parameter | **Values**
	Dir | 04 Intec Capital
	File | Intec.efw
	
	Step – 2:
	Open browser's console and type the below JavaScript code to get all the component views of the opened report.
	
	Dashboard.componentViews
	
	
	`Dashboard.componentViews`
	`Object {daterangepicker: B.r, select_rows: B.r, select_cols: B.r, measure: B.r, limit: B.r…}`
	`branch_name: B.r`
	`button: B.r`
	`category_name: B.r`
	`country_name: B.r`
	`customScript: B.r`
	`daterangepicker: B.r`
	`limit: B.r`
	`measure: B.r`
	`pivotChart: B.r`
	`product_name: B.r`
	`region_name: B.r`
	`select_cols: B.r`
	`select_rows: B.r`
	`status_name: B.r`
	`__proto__: Object`
	Figure 7: Component views of the opened report
	Step – 3:
	To reset or clear specific component, type below JavaScript code to achieve this:
	
	Dashboard.resetComponent("select_rows")
	
	The above method will return "true" in the console.
	
	Step – 4:
	Click on Parameters in the report, and check "Select Rows" section. Section should be blank with no values or parameters.

6.9 Dashboard.init()

Description	This API will initiate all the components that are passed in the API as an array. **Note:** The components should be passed in array.
Example	**Step – 1:** In this example, we have considered created one component of type button and if no options are provided to the component, default display message and classes that are default will be considered and the component will be added. While creating the

	component, name is mandatory, type is mandatory and htmlElementId. This component will show a button.
	Step – 2:
	Example:
	var **buttonComponent** = {
	name: "Btn",
	type: "button",
	options:{
	display: "Success",
	classes: "btn btn-success"
	},
	htmlElementId: "#year1"
	}
	Dashboard.addComponent(buttonComponent)
	Step – 3:
	To initiate the created component, call the API as:
	Dashboard.init([buttonComponent])

6.10 Dashboard.updateComponentOptions(component, options)

Description	This API updates the options parameter that is present in the component. Options vary depending on the type of component. Please refer Dashboard.Components section to know more.
Example	**Step – 1:**
	In this example, we have considered created one component of type button and if no options are provided to the component, default display message and classes that are default will be considered and the component will be added. While creating the component, name is mandatory, type is mandatory and htmlElementId. This component will show a button.
	Step – 2:
	To update the options of the component that is created, we use this API.
	Example:
	var **buttonComponent** = {
	name: "Btn",
	type: "button",
	options:{
	display: "Success",
	classes: "btn btn-success"
	},
	htmlElementId: "#year1"
	}

	buttonComponent.options.display = “Warning”;
	
	Dashboard.updateComponentOptions(buttonComponent, buttonComponent .options)

6.11 DashboardGlobals

DashboardGlobals will get an object with all the url’s that are used in the application.

Object	**Property**	**URL**
DashboarGlobals	baseUrl	It is the url of the running HI application. **Example:** http://localhost:8080/hi/
	solutionLoader	baseUrl + "getSolutionResources.html" This ajax call will get all the solution directories in an array.
	resourceLoader	baseUrl + "getEFWSolution.html" This ajax call will call **dir** and **file name** of the selected file and in response, loads the HTML of that particular file.
	updateService	baseUrl + "executeDatasource.html"
	chartingService	baseUrl + "visualizeData.html"
	exportData	baseUrl + "downloadReport.html" This ajax call is made when report is saved or exported.
	file	Displays the file name of the opened report
	extension	Gets the extension of the opened report
	fileTitle	Will get the title of the opened report
	folderPath	Will get the folder path of the opened report
	productInfo	baseUrl + "getProductInformation.html" This ajax call is made to get the information of the HI profuct (when clicked on About in navigation bar)
	sendMail	baseUrl + "sendMail.html" This ajax is call is made at the time of Email or Scheduling the opened report.
	updateEFWTemplate	baseUrl + "updateEFWTemplate.html"
	sessionUserName	Name of the user who is accessing the application
	sessionUserEmail	Email of the user who is accessing the application
	sessionUserOrganizatio n	If the user has any organization, then organization name will be considered else organization name is empty. For super admin and super user, that is hdiadmin or hdiuser,

		organization name is empty (“”).
	rootDirectoryPermission	Displays the permission level of the user
	provideHTMLExport	Boolean (true / false that is set in the xml page) Boolean value is set whether to export the report through HTML or not.
	enableReportSave	Boolean (true or false that is set in the xml page) This value is set whether to show the filebrowser to save or export the report at particular path or location. If the value is **false**, filebrowser will not be shown and the report will be downloaded directly to the download directory.
	defaultEmailResourceType	url
	saveReport	baseUrl + "saveReport.html" This ajax call is made when a user wants to save the report.
	fsop	baseUrl + "fileSystemOperations.html" This ajax call is made whenever user clicks on **Open, Rename, Edit, Open in new window, Delete, Cut, Paste** options in context menu.
	importFile	baseUrl + "importFile.html" This ajax call is made when user imports a file of extenstion **crt.**
	downloadEnableSavedReport	baseUrl + "downloadEnableSavedReport.html"
	services	baseUrl + "services"
	designerEdit	baseUrl + "designer-edit.html" This ajax call will open the designer report in designer-edit page
	adhocEdit	baseUrl + "adhoc/report-edit.html" This ajax call will open the adhoc report in adhoc-edit page.
	metadataEdit	baseUrl + "adhoc/metadata-edit.html" This ajax call will open the metadata in metadata-edit page.
	adhocReportCreate	baseUrl + "adhoc/report-create.html"
	openAdhoc	baseUrl + "hdi.html" This ajax call will open adhoc report in HI module when dir and file name has been as

		parameters to the url.
	openEfw	baseUrl + "hdi.html" This ajax call will open efw report in HI module when dir and file name has been as parameters to the url.
	visualizeAdhoc	baseUrl + "visualizeAdhoc.html"
DashboardGlobals.controllers	efw	baseUrl + "getEFWSolution.html" This ajax call will call **dir** and **file name** of the selected file and in response, loads the HTML of that particular file.
	efwsr	baseUrl + "executeSavedReport.html" This ajax call will open the saved report file with triggered or selected parameters. These parameters are set in **reportParameters** variable.
	efwfav	baseUrl + "executeFavourite.html"
	report	baseUrl + "hdi.html" This ajax call will render the html page of the HI module.
DashboardGlobals.scheduling	get	baseUrl + "getScheduleData.html" This ajax call will get the scheduled data of the report when the report is scheduled.
	update	baseUrl + "updateScheduleData.html" This ajax call will update the schedule data of the report while scheduling the report.

DashboardGlobals.adminPaths	users	baseUrl+"admin/users"
		
		This ajax call will fetch user’s data.
		
		Example:
		`{`
		`   "users":[`
		`      {`
		`         "slno":"1",`
		`         "id":1,`
		`         "name":"hiadmin",`
		`         "email":"hi@helicaltech.com",`
		`         "enabled":true,`
		`         "organisation":"",`
		`         "orgName":"Null",`
		`         "roles":[`
		`            {`
		`               "id":1,`
		`               "role":"ROLE_ADMIN"`
		`            },`
		`            {`

		"**id**":2, "**role**":"ROLE_USER" }], "**profiles**":[ ] }, …], "**total**": 1 }

	organisations	baseUrl+"admin/organisations"
		
		This ajax will fetch organizations data.
		
		Example:
		{
		"**organisations**":[
		{
		"**slno**":"1",
		"**id**":36,
		"**name**":"Organization Name",
		"**description**":"Description"
		},
		…
		]
		}

	profiles	baseUrl+"admin/profiles"

	roles	baseUrl+"admin/roles"
		
		This ajax call will get the profiles of user or admin.
		
		Example:
		{
		"**total**":2,
		"**roles**":[
		{
		"**slno**":"1",
		"**id**":1,
		"**name**":"ROLE_ADMIN",
		"**organisation**":"",
		"**orgName**":"Null"
		},
		{
		"**slno**":"2",
		"**id**":2,
		"**name**":"ROLE_USER",
		"**organisation**":"",
		"**orgName**":"Null"
		}
		]
		}

	services	baseUrl+"services.html"

DashboardGlobals.optionalReportParams	location	Undefined
	reportname	Undefined

7. Report Community Editor

Report Community Editor is an UI driven method of creating very powerful EFW community reports. Writing SQL queries, call stored procedures, integrate custom visualizations etc and create very very complex reports and dashboards also.User can create EFW report using report community editor. Below are backend API's for Report Community report creation ,deletion,editing etc.

7.1 Create Report CE

Page : http://192.168.2.156:8085/hi-ee/ce-report-edit.html

Create Report CE API is used to create Report CE with available functionality.This is API is used while save and Save as of Report CE after creation.

This service is used to create efwce reports this service handles both save and update operations based up on parameter uuid in form data.This service will create 5 separate files:

*.efw

*.efwce

*.efwvf.

*.efwd

*.html

Update operation will update all above five files.

.efwce file will contain state info which is used to fetch the existing state.

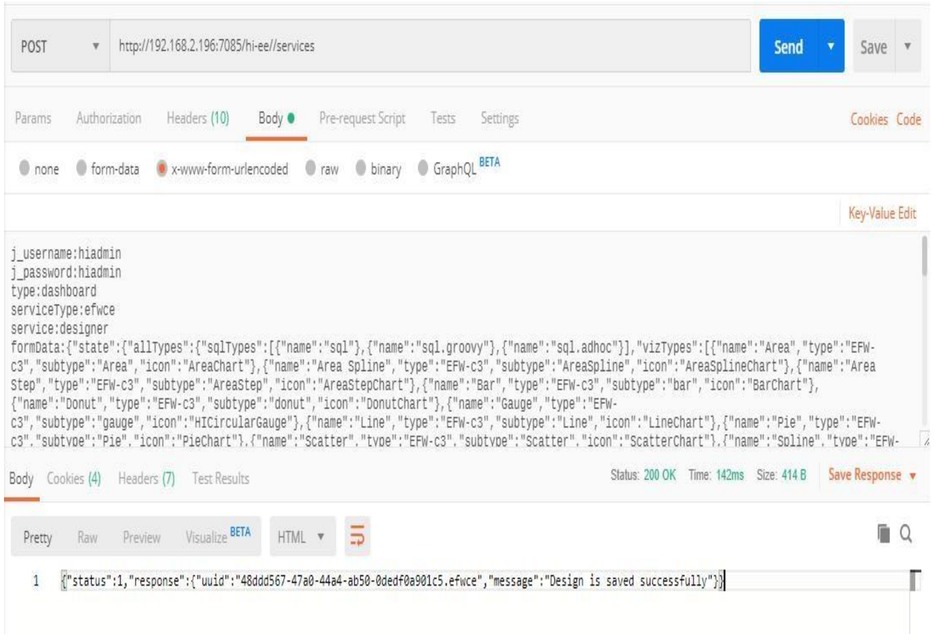
URL	/services
Description	It allows user to create and update(save/save as) the EFWCE report.
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.
Accessible for	ROLE_ADMIN, ROLE_USER
HTTP Request Method	**POST**

Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwce&service=designer&formData={"state":{"allTypes":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut","type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line","type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter","type":"EFW-c3","subtype":"Scatter","icon":"ScatterChart"},{"name":"Spline","type":"EFW-c3","subtype":"Spline","icon":"SplineChart"},{"name":"Step","type":"EFW-c3","subtype":"Step","icon":"StepChart"},{"name":"Cross Tab","type":"EFW-CrossTab","icon":"HICrossTable"},{"name":"Table","type":"EFW-Table","icon":"HITable"},{"name":"Custom","type":"Custom","icon":"VF"}],"connTypes":[{"clazz":"com.helicalinsight.datasource.GlobalJdbcDataSource","classifier":"global","name":"Managed DataSource","type":"global.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.datasource.JDBCDriver","classifier":"efwd","name":"Plain Jdbc DataSource","type":"sql.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.adhoc.SqlAdhocDriver","classifier":"efwd","name":"Adhoc DataSource","type":"sql.adhoc","hidden":"true"},{"clazz":"com.helicalinsight.datasource.ExtJDBCDriver","classifier":"efwd","name":"Groovy Plain Jdbc DataSource","type":"sql.jdbc.groovy","hidden":"false"}],"parameterTypes":[{"name":"Collection"},{"name":"Numeric"},{"name":"String"}]},"datasource":[{"id":1,"configure":"<globalId>1</globalId>","name":"connection1","type":{"name":"Managed DataSource","type":"global.jdbc"}}],"dashboard":{"html":"","css":""},"parameters":[],"reports":[{"id":1,"configure":"var report1 = {\n name: \"report1\",\n type:\"chart\",\n listeners:[\"column_name\"],\n requestParameters :{\n column_name: \"column_name\"\n },\n vf: {\n id: \"1\",\n file: '__ efwvf_name__ '\n },\n efwd : {\n file: ' efwd_file_name '\n },\n htmlElementId : \"#htmlelement_id\", // provide report id here\n executeAtStart: true // it can be true or false\n};","visualisation":"<Dimensions>column_name</Dimensions>\n<Measures>column_name</Measures> \n","name":"report1","vizName":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart","sql":{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}}],"currentEditor":{"type":"reports","name":"report1","component":"sql"},"sqlList":[{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}],"aceText":"","uuid":"","showDatasource":true,"showDashboard":false,"showParameter":true,"showReport":true,"Editor":{"type":"reports","name":"report1","component":"sql","aceText":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\"

	\nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY"},"editing":false,"isEditor":""},"htmlString":"","cssString":"","configurations":{"reports":[{"report":{"id":"1","name":"report1","value":"var dashboard = Dashboard \n dashboard.resetAll() \n \n var report1 = {\n name: \"report1\",\n type:\"chart\",\n listeners:[\"column_name\"],\n requestParameters :{\n column_name: \"column_name\"\n },\n vf : {\n id: \"1\",\n file: '_efwvf_name_'\n },\n efwd : {\n file: '_efwd_file_name_'\n },\n htmlElementId : \"#htmlelement_id\", // provide report id here\n executeAtStart: true // it can be true or false\n};"}}]},"efwd":{"dataSources":{"connections":[{"connection":{"id":"1","type":"global.jdbc","connDetails":{"globalId":"1"}}}]},"dataMaps":[{"dataMap":{"name":"report1","id":"1","type":"sql","connection":"1","query":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n\tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY"}}]},"visualisation":[{"id":"1","name":"report1","type":"EFW-c3","subtype":"Area","DataSource":"1","Dimensions":"column_name","Measures":"column_name"}],"dir":"1570702719854","file":"SampleCEReport1"}' http://192.168.2.196:7085/hi-ee//services -v

HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as dashboard type.
serviceType:	efwce	Service type as efwce
service:	designer	Service to create the efwce report
formData:	{"state":{"allTypes":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut","type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line","type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter","type":"EFW-c3","subtype":"Scatter","icon":"ScatterChart"},{"name":"Spline","type":"EFW-c3","subtype":"Spline","icon":"SplineChart"},{"name":"Step","type":"EFW-c3","subtype":"Step","icon":"StepChart"},{"name":"Cross Tab","type":"EFW-CrossTab","icon":"HICrossTable"},{"name":"Table","type":"EFW-Table","icon":"HITable"},{"name":"Custom","type":"Custom","icon":"VF"}],"connTypes":[{"clazz":"com.helicalinsight.datasource.GlobalJdbcDataSource","classifier":"global","name":"Managed DataSource","type":"global.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.datasource.JDBCDriver","classifier":"efwd","name":"Plain Jdbc DataSource","type":"sql.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.adhoc.SqlAdhocDriver","classifier":"efwd","name":"Adhoc DataSource","type":"sql.adhoc","hidden":"true"},{"clazz":"com.helicalinsight.datasource.ExtJDBCDriver","classifier":"efwd","name":"Groovy Plain Jdbc DataSource","type":"sql.jdbc.groovy","hidden":"false"}],"parameterTypes":[{"name":"Collection"},{"name":"Nume	Formdata contains the efwcefile name and the directory of file to save the created efwce report.Along with filename and directory formdata contains the different available sqlTypes, visualization types, connection types, datasource info, parameters,parameter datatypes,reports, different configurations, SQL's etc.

	ric"},{"name":"String"}]},"datasource":[{"id":1,"configure":"<globalId>1</globalId>","name":"connection1","type":{"name":"Managed DataSource","type":"global.jdbc"}}],"dashboard":{"html":"","css":""},"parameters":[],"reports":[{"id":1,"configure":"var report1 = {\n name: \"report1\",\n type:\"chart\",\n listeners:[\"column_name\"],\n requestParameters :{\n column_name: \"column_name\"\n },\n vf : {\n id: \"1\",\n file: '_efwvf_name_'\n },\n efwd : {\n file: '_efwd_file_name_'\n },\n htmlElementId : \"#htmlelement_id\", // provide report id here\n executeAtStart: true // it can be true or false\n};","visualisation":"<Dimensions>column_name</Dimensions>\n<Measures>column_name</Measures> \n","name":"report1","vizName":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart","sql":{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}}],"currentEditor":{"type":"reports","name":"report1","component":"sql"},"sqlList":[{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}],"aceText":"","uuid":"","showDatasource":true,"showDashboard":false,"showParameter":true,"showReport":true,"Editor":{"type":"reports","name":"report1","component":"sql","aceText":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY"},"editing":false,"isEditor":""},"htmlString":"","cssString":"","configurations":{"reports":[{"report":{"id":"1","name":"report1","value":"var dashboard = Dashboard \n dashboard.resetAll() \n \n var report1 = {\n name: \"report1\",\n type:\"chart\",\n listeners:[\"column_name\"],\n requestParameters :{\n column_name: \"column_name\"\n },\n vf : {\n id: \"1\",\n file: '_efwvf_name_'\n },\n efwd : {\n file: '_efwd_file_name_'\n },\n htmlElementId : \"#htmlelement_id\", // provide report id here\n executeAtStart: true // it can be true or false\n}; "}}]},"efwd":{"dataSources":{"connections":[{"connection":{"id":"1","type":"global.jdbc","connDetails":{"globalId":"1"}}}]},"dataMaps":[{"dataMap":{"name":"report1","id":"1","type":"sql","connection":"1","query":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n \tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY"}}]},"visualisation":[{"id":"1","name":"report1","type":"EFW-c3","subtype":"Area","DataSource":"1","Dimensions":"col	

	umn_name","Measures":"column_name"}],"dir":"1570702719854","file":"SampleCEReport"}	
Response Output(JSON Format)	{"status":1,"response":{"uuid":"48ddd567-47a0-44a4-ab50-0dedf0a901c5.efwce","message":"Design is saved successfully"}}	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the uuid with .efwce file name and the success message	
Service Status	200 OK	
Screenshot		

```
POST http://192.168.2.196:7085/hi-ee//services
Body: x-www-form-urlencoded

j_username:hiadmin
j_password:hiadmin
type:dashboard
serviceType:efwce
service:designer
formData:{"state":{"allTypes":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},
{"name":"Donut","type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line","type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter","type":"EFW-c3","subtype":"Scatter","icon":"ScatterChart"},{"name":"Spline","type":"EFW-

Status: 200 OK  Time: 142ms  Size: 414 B

{"status":1,"response":{"uuid":"48ddd567-47a0-44a4-ab50-0dedf0a901c5.efwce","message":"Design is saved successfully"}}
```

7.2 Edit Report CE

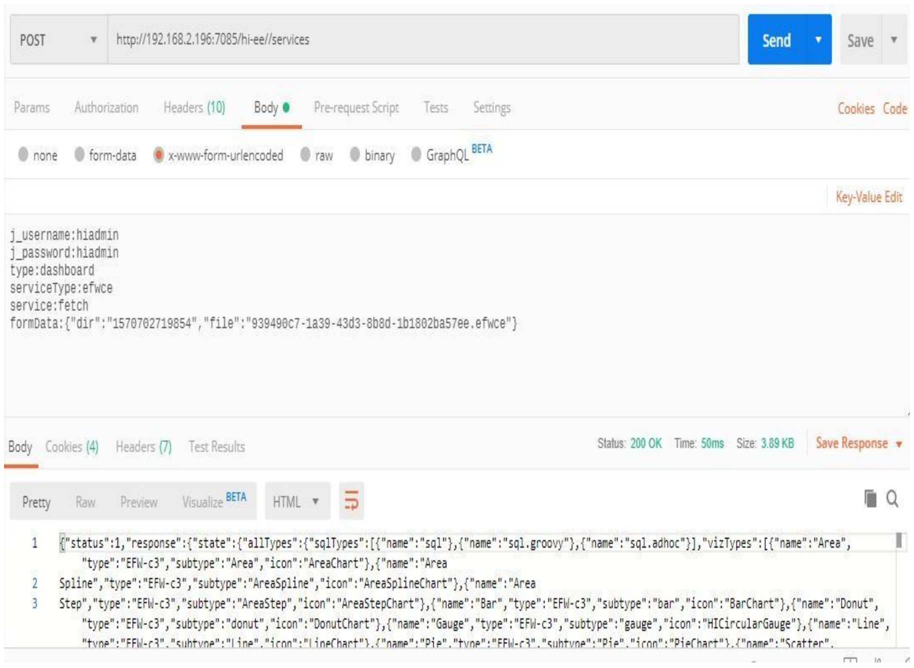
Page : http://192.168.2.156:8085/hi-ee/ce-report-edit.html

Note : Below API is to select efwce file which user want to edit .So after that you can do changes in efwce file and save file /saveAs file is nothing but the Create Report CE API.

This service is used when user wants to edit the existing efwce report this service simply provides the state information which is already saved in efwce file for this service user need to provide efwce fileName and its dir.

URL	/services	
Description	It allows user to fetch/get the EFWCE report for Edit EFWCE.It returns the state of efwce report as response.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwce&service=fetch&formData={"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as dashboard type.
serviceType:	efwce	Service type as efwce
service:	fetch	Service to edit the efwce report
formData:	{"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}	Formdata contains the efwcefile physical name and the directory of file(physical name) where efwce report is saved
Response Output(JSON Format)	{"status":1,"response":{"state":{"allTypes":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut","type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-	

	c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line","type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter","type":"EFW-c3","subtype":"Scatter","icon":"ScatterChart"},{"name":"Spline","type":"EFW-c3","subtype":"Spline","icon":"SplineChart"},{"name":"Step","type":"EFW-c3","subtype":"Step","icon":"StepChart"},{"name":"Cross Tab","type":"EFW-CrossTab","icon":"HICrossTable"},{"name":"Table","type":"EFW-Table","icon":"HITable"},{"name":"Custom","type":"Custom","icon":"VF"}],"connTypes":[{"clazz":"com.helicalinsight.datasource.GlobalJdbcDataSource","classifier":"global","name":"Managed DataSource","type":"global.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.datasource.JDBCDriver","classifier":"efwd","name":"Plain Jdbc DataSource","type":"sql.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.adhoc.SqlAdhocDriver","classifier":"efwd","name":"Adhoc DataSource","type":"sql.adhoc","hidden":"true"},{"clazz":"com.helicalinsight.datasource.ExtJDBCDriver","classifier":"efwd","name":"Groovy Plain Jdbc DataSource","type":"sql.jdbc.groovy","hidden":"false"}],"parameterTypes":[{"name":"Collection"},{"name":"Numeric"},{"name":"String"}]},"datasource":[{"id":1,"configure":" <globalId>1</globalId>","name":"connection1","type":{"name":"Managed DataSource","type":"global.jdbc"}}],"dashboard":{"html":"","css":""},"parameters":[],"reports":[{"id":1,"configure":"var report1 = {\n name: \"report1\",\n type:\"chart\",\n //listeners:[\"column_name\"],\n // requestParameters :{\n // column_name: \"column_name\"\n //},\n vf : {\n id: \"1\",\n file: '_efwvf_name_'\n },\n efwd : {\n file: '_ efwd_file_name_ '\n },\n htmlElementId : \"#htmlelement_id\", // provide report id here\n executeAtStart: true // it can be true or false\n};","visualisation":"<Dimensions>column_name</Dimensions>\n<Measures>column_name</Measures> \n","name":"report1","vizName":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart","sql":{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n\tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}}],"currentEditor":{"type":"dashboard","name":"","component":"css"},"sqlList":[{"id":1,"type":"sql","text":"select \n\t\"HIUSER\".\"employee_details\".\"address\" as \"employee_details_address\",\n\tsum(\"HIUSER\".\"employee_details\".\"age\") as \"sum_age\" \nfrom\n\t\"HIUSER\".\"employee_details\" \ngroup by\n\t\"HIUSER\".\"employee_details\".\"address\" FETCH FIRST 1000 ROWS ONLY","dataSource":{"id":1,"name":"connection1"},"parameters":[]}],"aceText":"","uuid":"939490c7-1a39-43d3-8b8d-1b1802ba57ee","showDatasource":true,"showDashboard":true,"showParameter":true,"showReport":true,"Editor":{"type":"","name":"","component":"","aceText":""},"editing":false,"isEditor":""},"reportName":"SampleCEReport"}}

Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the uuid with .efwce file name and the success message along with state of the efwce report
Service Status	200 OK
Screenshot	

```
POST  http://192.168.2.196:7085/hi-ee//services
```

```
j_username:hiadmin
j_password:hiadmin
type:dashboard
serviceType:efwce
service:fetch
formData:{"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}
```

Status: 200 OK Time: 50ms Size: 3.89 KB

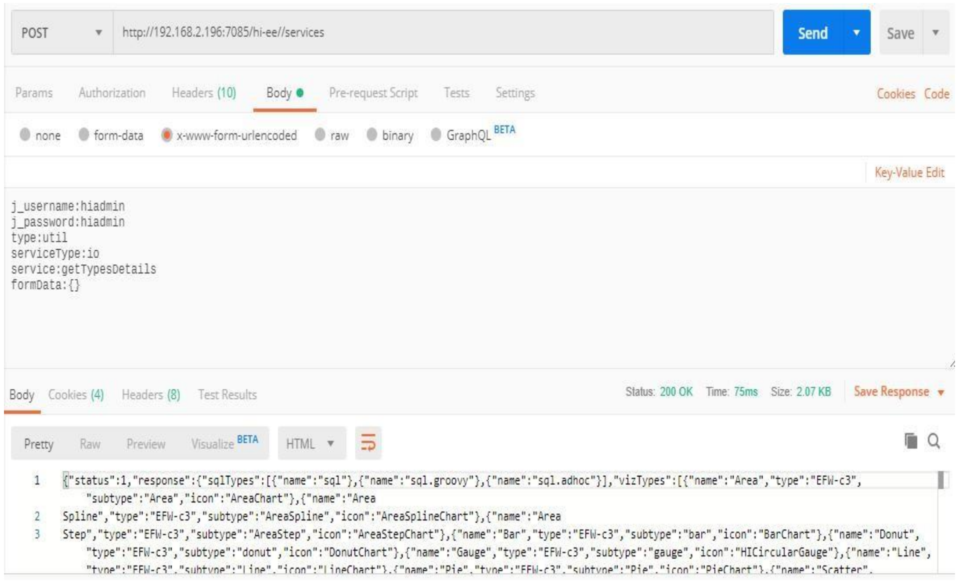
```
{"status":1,"response":{"state":{"allTypes":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area",
    "type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area
Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area
Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut",
    "type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line",
    "type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter",
```

7.3 Get Report CE Types

This service API is used to get the different types which are available for Report CE.

URL	/services	
Description	It allows user to get the EFWCE report types as response.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=util&serviceType=io&service=getTypesDetails&formData={}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	util	Type as util type.
serviceType:	io	Service type as io
service:	getTypesDetails	Service to getTypesDetails of the efwce report
formData:	{}	Formdata as empty

Response Output(JSON Format)	{"status":1,"response":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3","subtype":"Area","icon":"AreaChart"},{"name":"Area Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut","type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line","type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter","type":"EFW-c3","subtype":"Scatter","icon":"ScatterChart"},{"name":"Spline","type":"EFW-c3","subtype":"Spline","icon":"SplineChart"},{"name":"Step","type":"EFW-c3","subtype":"Step","icon":"StepChart"},{"name":"Cross Tab","type":"EFW-CrossTab","icon":"HICrossTable"},{"name":"Table","type":"EFW-Table","icon":"HITable"},{"name":"Custom","type":"Custom","icon":"VF"}],"connTypes":[{"clazz":"com.helicalinsight.datasource.GlobalJdbcDataSource","classifier":"global","name":"Managed

	DataSource","type":"global.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.datasource.JDBCDriver","classifier":"efwd","name":"Plain Jdbc DataSource","type":"sql.jdbc","hidden":"false"},{"clazz":"com.helicalinsight.adhoc.SqlAdhocDriver","classifier":"efwd","name":"Adhoc DataSource","type":"sql.adhoc","hidden":"true"},{"clazz":"com.helicalinsight.datasource.ExtJDBCDriver","classifier":"efwd","name":"Groovy Plain Jdbc DataSource","type":"sql.jdbc.groovy","hidden":"false"}],"parameterTypes":[{"name":"Collection"},{"name":"Numeric"},{"name":"String"}]}}
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as type details of the efwce report
Service Status	200 OK
Screenshot	

POST http://192.168.2.196:7085/hi-ee//services

Body: x-www-form-urlencoded

```
j_username:hiadmin
j_password:hiadmin
type:util
serviceType:io
service:getTypesDetails
formData:{}
```

Status: 200 OK Time: 75ms Size: 2.07 KB

```
{"status":1,"response":{"sqlTypes":[{"name":"sql"},{"name":"sql.groovy"},{"name":"sql.adhoc"}],"vizTypes":[{"name":"Area","type":"EFW-c3",
    "subtype":"Area","icon":"AreaChart"},{"name":"Area
Spline","type":"EFW-c3","subtype":"AreaSpline","icon":"AreaSplineChart"},{"name":"Area
Step","type":"EFW-c3","subtype":"AreaStep","icon":"AreaStepChart"},{"name":"Bar","type":"EFW-c3","subtype":"bar","icon":"BarChart"},{"name":"Donut",
    "type":"EFW-c3","subtype":"donut","icon":"DonutChart"},{"name":"Gauge","type":"EFW-c3","subtype":"gauge","icon":"HICircularGauge"},{"name":"Line",
    "type":"EFW-c3","subtype":"Line","icon":"LineChart"},{"name":"Pie","type":"EFW-c3","subtype":"Pie","icon":"PieChart"},{"name":"Scatter",
```

7.4 Delete Report CE

This service will perform delete operation on efwce report this service will delete all 5 generated files

*.efwce

*.efw

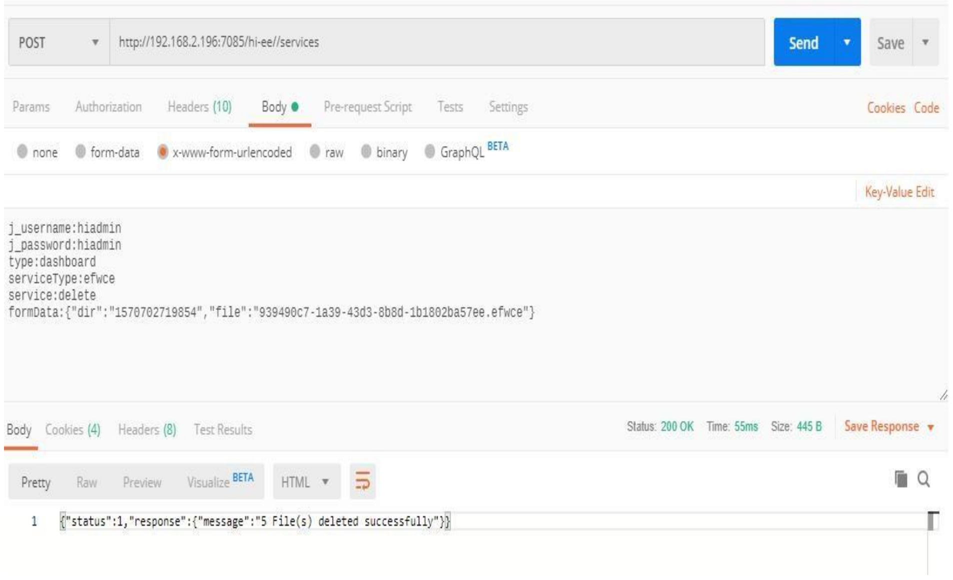
*.efwd

*.efwvf

*.html

This service requires efwce file name with or with out extension and its physical directory.

URL	/services	
Description	It allows user to delete EFWCE report	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN, ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee//services **Access through Curl command :** curl --data 'j_username=hiadmin&j_password=hiadmin&type=dashboard&serviceType=efwce&service=delete&formData={"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}' http://192.168.2.196:7085/hi-ee//services -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	dashboard	Type as dashboard type.
serviceType:	efwce	Service type as efwce
service:	delete	Service to delete the efwce report
formData:	{"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}	Formdata contains the efwcefile physical name and the directory of file(physical name) to delete efwce report.
Response Output(JSON	{"status":1,"response":{"message":"5 File(s) deleted successfully"}}	

Format)	
Description of Response Output:	The response of the API is , it returns the success status value as 1 if it fails returns 0 as the status. It returns response as the success message for delete operation.
Service Status	200 OK
Screenshot	

```
j_username:hiadmin
j_password:hiadmin
type:dashboard
serviceType:efwce
service:delete
formData:{"dir":"1570702719854","file":"939490c7-1a39-43d3-8b8d-1b1802ba57ee.efwce"}
```

```
{"status":1,"response":{"message":"5 File(s) deleted successfully"}}
```

Error and its interpretation

Generally an error occurs when the request is malformed or invalid. The error can also occur on the server side.The following pattern is followed throughout the application. The http status code is 200, and content type is application/json.

The client/browser receives an JSON object of the following structure when an error occurs as given below:

JSON object structure	Description
{ "status": 0, "response": { "message": "Validation Error", "data": { "message": "'columns' No columns selected" } } }	**status 0:**It indicates an error occurred. **response:** This is a JSON object that holds the errordetails. **message:**The message contains the Error message **data:** This is a JSON object (may or may not bepresent in case of error) **message:** This contains the extra/detailed information for the error type.

Example 1: When no columns are selected in the adhoc report generation the following json is obtained.

```
{
   "status": 0,
   "response": {
      "message": "Validation Error",
      "data": {
         "message": "'columns' No columns selected"
      }
   }
}
```

<u>Example 2:</u> SQL Error obtained when the user tries to add multiple group function in a single column

```
{
   "status": 0,
   "response": {
      "message": "Error: SQLException: Invalid use of group function"
   }
}
```

<u>Example 3:</u> SQL Error obtained when the user tries to join different table that are not joined

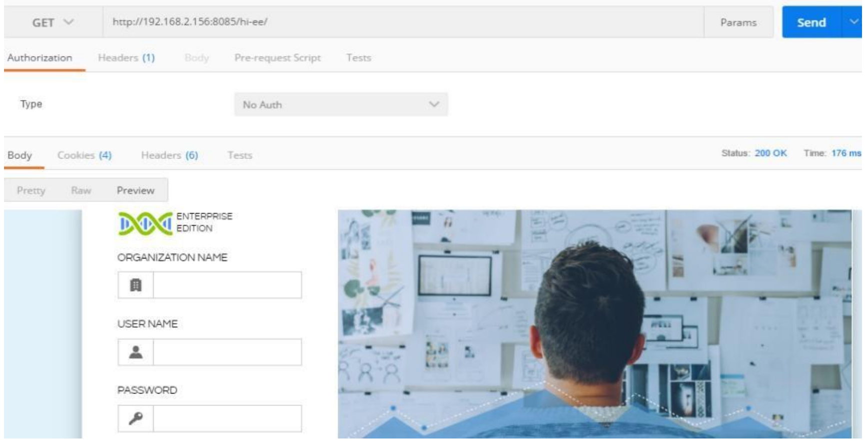
```
{
   "status": 0,
   "response": {
      "message": "Error: QueryBuilderException: Can't prepare sql query. Can't join tables cache and organization."
   }
}
```

- The file extensions can be customized by changing their value in setting.xml extentions element.

Miscellaneous API's

1. Login URL :

URL	/, index.html, login.html/
Description	User Login Page If user already logged in then user will see welcome page for respective user.
Pre-requisite	The Helical Insight Application should be up.
Accessible for	Everyone
HTTP Request Method	**GET ,POST**
Example	http://192.168.2.156:8085/hi-ee/ http://192.168.2.156:8085/index.html http://192.168.2.156:8085/login.html/ Curl command : curl GET/POST http://192.168.2.156:8085/hi-ee/index.html curl GET/POST http://192.168.2.156:8085/hi-ee/login.html/ curl GET/POST http://192.168.2.156:8085/hi-ee/
Response Output	Will get login Page html contents.
Description of Response Output:	The Response output is the login page html contents.
Service Status	200 OK

ScreenShot	 GET http://192.168.2.156:8085/hi-ee/ Params Send Authorization Headers (1) Body Pre-request Script Tests Type No Auth Body Cookies (4) Headers (6) Tests Status: 200 OK Time: 176 ms Pretty Raw Preview ENTERPRISE EDITION ORGANIZATION NAME USER NAME PASSWORD

2 Super admin Login:

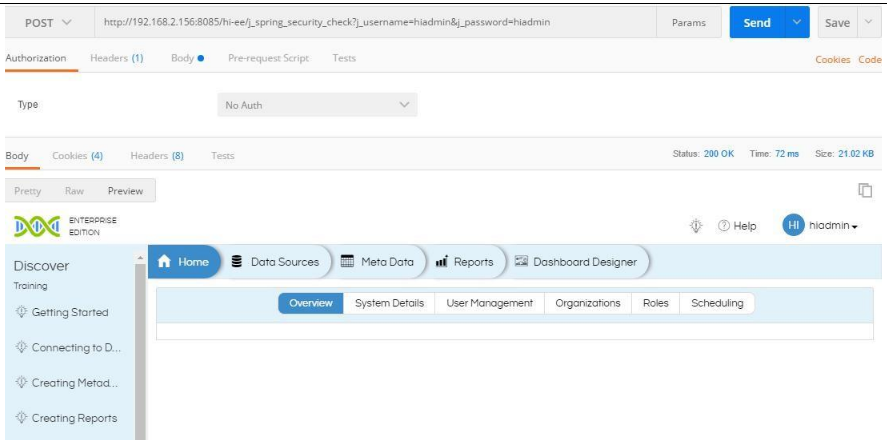
Slash (default)

Pre-authentication : HI resource URL

loginURL j_username=hiadmin&j_password=hiadmin

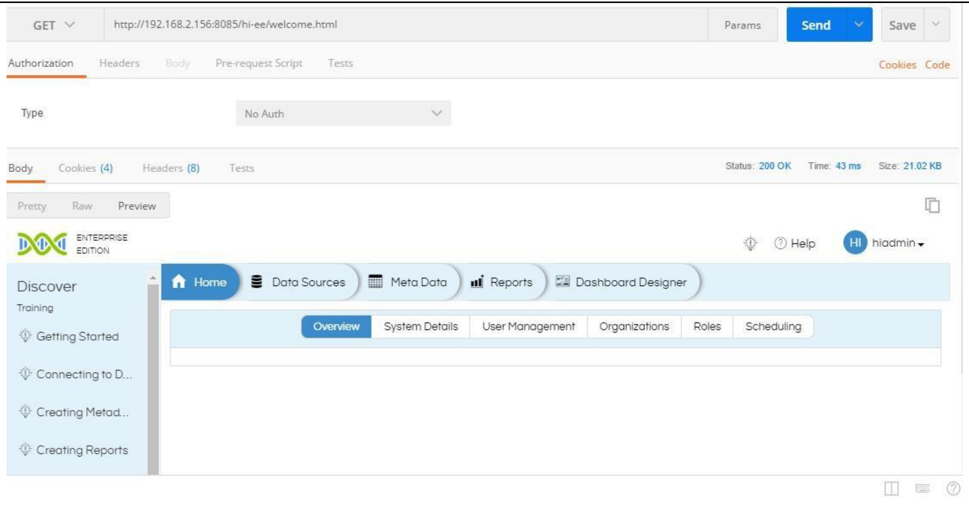
Custom auth :

URL	j_spring_security_check?j_username=hiadmin&j_password=hiadmin	
Description	It performs submit action for login page for login to helical insight application.	
Pre-requisite	The Helical Insight Application should be up.	
Accessible for	Super admin	
HTTP Request Method	**GET , POST**	
HTTP Request Key	**HTTP Request Value**	**Description**
j_organization **(optional)**		Leave the organization name as blank.
j_username	hiadmin	Super admin user name
j_password	hiadmin	Super admin password
Example	http://192.168.2.156:8085/hi-ee/j_spring_security_check?j_username=hiadmin&j_password=hiadmin Curl command :	

Response output	We can see the home page of the HI Application with super admin access.
Description of Response Output:	Will see the home page of HI application
Service Status	200 OK
Screenshot	 POST http://192.168.2.156:8085/hi-ee/j_spring_security_check?j_username=hiadmin&j_password=hiadmin Type: No Auth Status: 200 OK Time: 72 ms Size: 21.02 KB Params Send Save Authorization Headers (1) Body Pre-request Script Tests Cookies Code Body Cookies (4) Headers (8) Tests Pretty Raw Preview ENTERPRISE EDITION Help HI hiadmin Discover Training Getting Started Connecting to D... Creating Metad... Creating Reports Home Data Sources Meta Data Reports Dashboard Designer Overview System Details User Management Organizations Roles Scheduling
Post-action	It will allow super admin related activities. For ex. Create user,organisation,role,datasource,metadata,reports etc

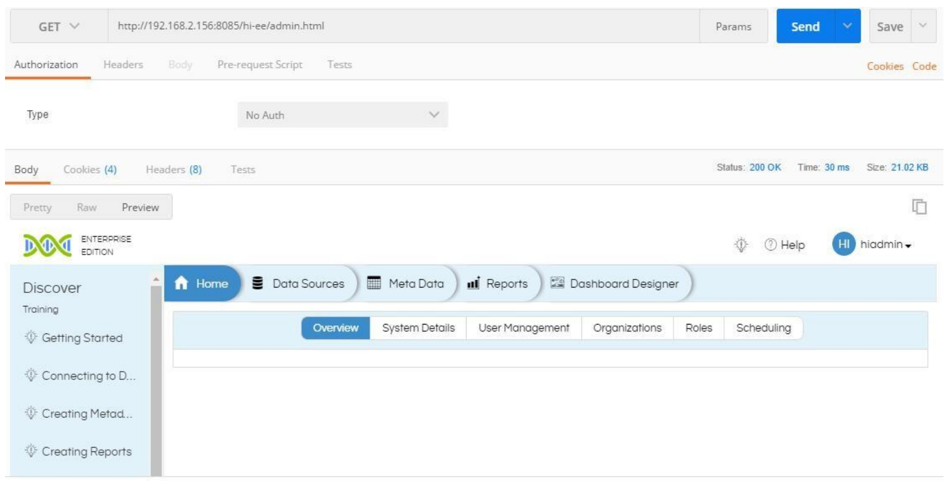
3 Welcome Page :

URL	welcome.html
Description	User will able to see the the welcome page .
Pre-requisite	User should have logged in before accessing the service.**[Refer login module : 1.2 Super admin Login**] If the user is not logged in then you will get login page.
Accessible for	Any authenticated user.
HTTP Request Method	**GET,POST**
Example	http://192.168.2.156:8085/hi-ee/welcome.html Curl commands :
Response Output	Welcome page will come for super admin.

Service Status	200 OK
Screenshot	 GET http://192.168.2.156:8085/hi-ee/welcome.html Type No Auth Status: 200 OK Time: 43 ms Size: 21.02 KB Params Send Save Authorization Headers Body Pre-request Script Tests Cookies Code Body Cookies (4) Headers (8) Tests Pretty Raw Preview ENTERPRISE EDITION Help HI hiadmin Discover Training Getting Started Connecting to D... Creating Metad... Creating Reports Home Data Sources Meta Data Reports Dashboard Designer Overview System Details User Management Organizations Roles Scheduling

4 Admin Page

URL	admin.html
Description	Admin page is the main page of the super admin where you will get system related details.It is used to manage and monitor the organizations ,roles and its users. If user type : Super Admin : Can access / manipulate all users / organization details Organization Admin: Can manipulate its user details.
Pre-requisite	User should have logged in before accessing the service.**[Refer login module : 1.2 Super admin Login**] If the user is not logged in then you will get login page.
Accessible for	Super admin / organisation admin
HTTP Request Method	**GET, POST**
Example	http://192.168.2.156:8085/hi-ee/admin.html
Response Output	Admin main page will come for super admin/organisation admin.
Description of Response Output:	On admin main page we can see the application details , system details etc.
Service Status	200 OK

Screenshot	 GET http://192.168.2.156:8085/hi-ee/admin.html Type: No Auth Status: 200 OK Time: 30 ms Size: 21.02 KB Params Send Save Authorization Headers Body Pre-request Script Tests Cookies Code Body Cookies (4) Headers (8) Tests Pretty Raw Preview ENTERPRISE EDITION Help HI hiadmin Discover Training Getting Started Connecting to D... Creating Metad... Creating Reports Home Data Sources Meta Data Reports Dashboard Designer Overview System Details User Management Organizations Roles Scheduling
Post-action	We can check system details and we can create organisations,roles,users,datasources,metadata,reports etc.

2. URL Embedding

2.1 Access EFW Report

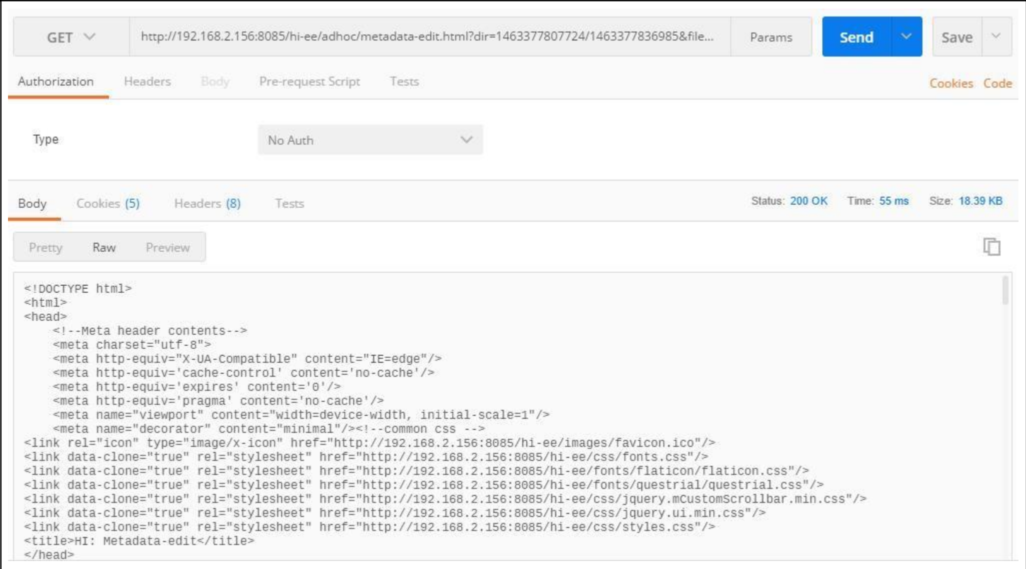
2.2 Access EFWS Report

2.3 Access Adhoc Report

2.4 Load metadata into metadata-edit

URL	adhoc/metadata-edit.html?dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata	
Description	It allows user to load the metadata for metadata editing purpose.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/adhoc/metadata-edit.html?dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata" http://192.168.2.156:8085/hi-ee/adhoc/metadata-edit.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377836985	The directory where the metdata file is present.
file:	e9be6771-995b-40eb-a01c-304857a100a1.metadata	The metadata file which we want to open for editing.
Response Output(JSON Format)	Here response is the requested edit metadata html contents.	

Service Status	200 OK

Screenshot	

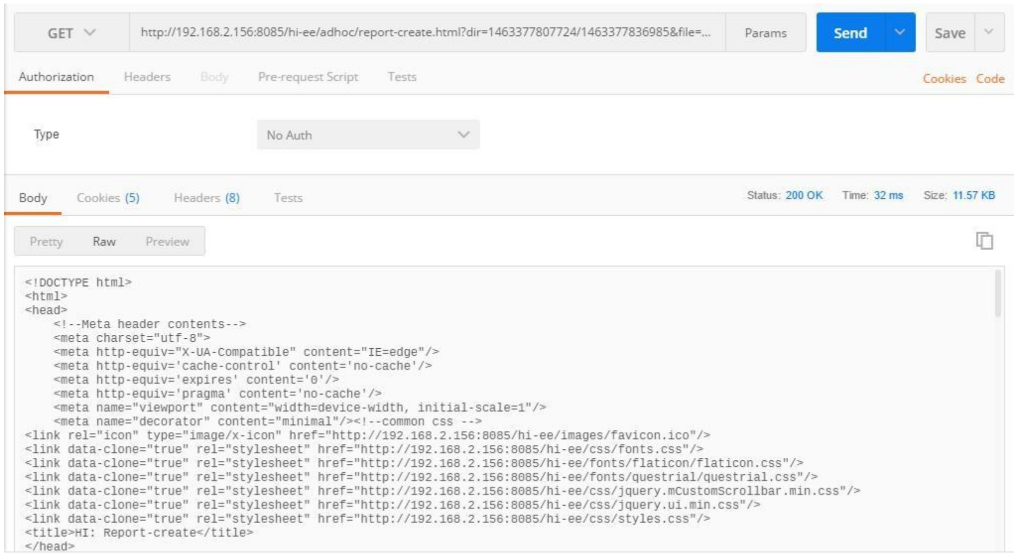
Type: No Auth

Status: 200 OK Time: 55 ms Size: 18.39 KB

```
<!DOCTYPE html>
<html>
<head>
    <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><!--common css -->
<link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.mCustomScrollbar.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
<title>HI: Metadata-edit</title>
</head>
```

2.5 Load metadata into report-create

URL	adhoc/report-create.html?dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata	
Description	It allows user to load the metadata for report creating purpose.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/adhoc/report-create.html?dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377836985&file=e9be6771-995b-40eb-a01c-304857a100a1.metadata" http://192.168.2.156:8085/hi-ee/adhoc/report-create.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463377836985	The directory where the metdata file is present.
file:	e9be6771-995b-40eb-a01c-304857a100a1.metadata	The metadata file which we want to for report creation.
Response Output(JSON Format)	Here response is the html contents of report creation for requested metadata.	
Service Status	200 OK	

Screenshot	

GET http://192.168.2.156:8085/hi-ee/adhoc/report-create.html?dir=1463377807724/1463377836985&file=...

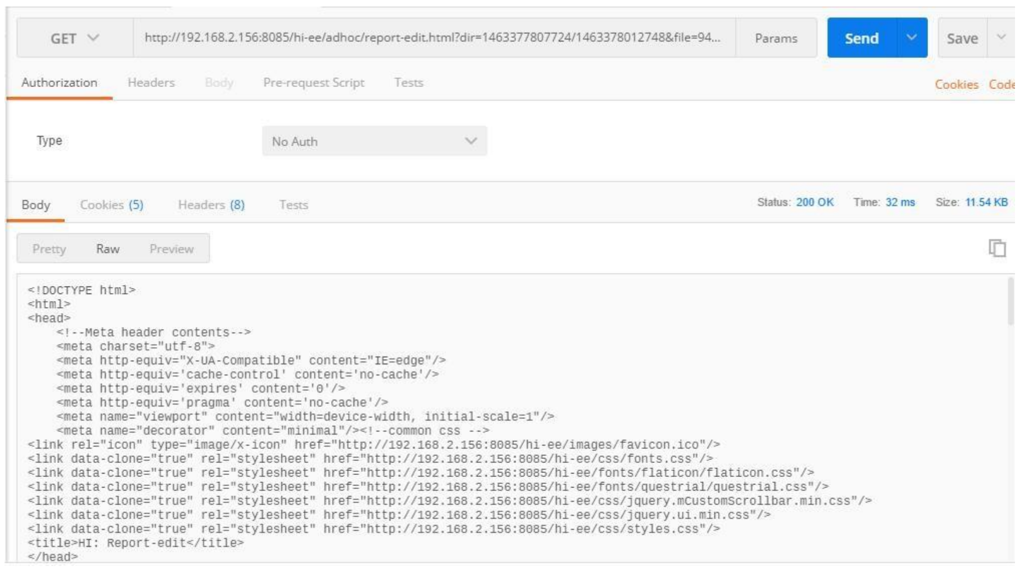
Type: No Auth

Status: 200 OK Time: 32 ms Size: 11.57 KB

```
<!DOCTYPE html>
<html>
<head>
    <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><!--common css -->
<link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.mCustomScrollbar.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
<title>HI: Report-create</title>
</head>
```

2.6 Load adhoc report into report-edit

URL	adhoc/report-edit.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	
Description	It allows user to load the report for report editing purpose.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/adhoc/report-edit.html?dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463378012748&file=94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report" http://192.168.2.156:8085/hi-ee/adhoc/report-edit.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1463378012748	The directory where the report file is present.
file:	94b8d841-bf01-4ff3-8e9e-ac858ac8a52c.report	The report file which we want to edit.
Response Output(JSON Format)	Here response is the requested report edit html contents.	
Service Status	200 OK	

Screenshot	

http://192.168.2.156:8085/hi-ee/adhoc/report-edit.html?dir=1463377807724/1463378012748&file=94...

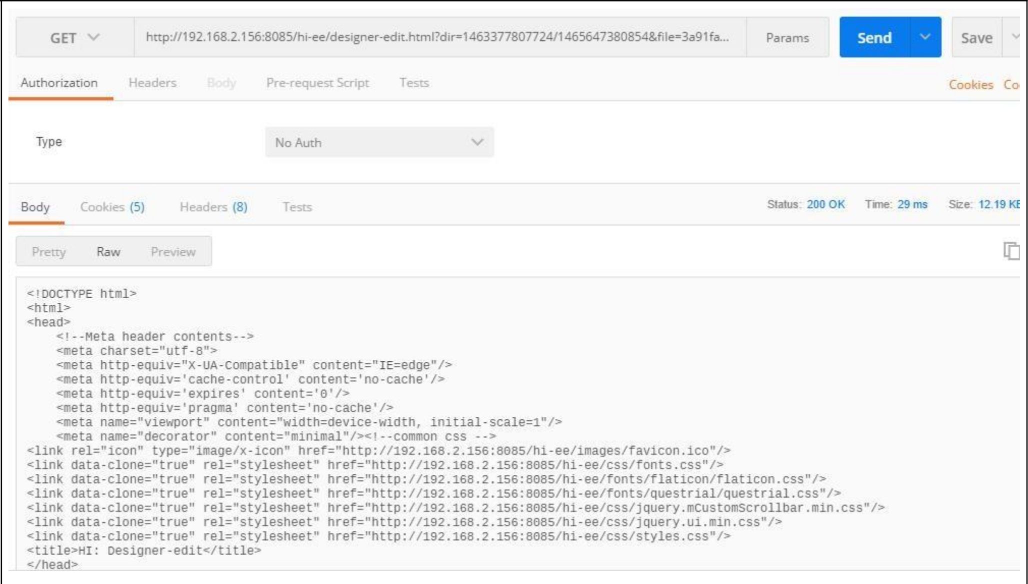
Type: No Auth

Status: 200 OK Time: 32 ms Size: 11.54 KB

```
<!DOCTYPE html>
<html>
<head>
    <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><!--common css -->
<link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.mCustomScrollbar.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
<title>HI: Report-edit</title>
</head>
```

2.7 Load designer into designer-edit

URL	designer-edit.html?dir=1463377807724/1465647380854&file=3a91fae9-6d4d-48fc-a718-83f38613198f.efwdd	
Description	It allows user to load the dashboard designer file for dasboard editing purpose.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**GET,POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/designer-edit.html?dir=1463377807724/1465647380854&file=3a91fae9-6d4d-48fc-a718-83f38613198f.efwdd **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1465647380854&file=3a91fae9-6d4d-48fc-a718-83f38613198f.efwdd" http://192.168.2.156:8085/hi-ee/designer-edit.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
dir:	1463377807724/1465647380854	The directory where the dashboard designer file is present.
file:	3a91fae9-6d4d-48fc-a718-83f38613198f.efwdd	The dashboard designer file which we want to edit.
Response Output(JSON Format)	Here response is the requested dashboard designer edit html contents.	
Service Status	200 OK	

Screenshot	

GET http://192.168.2.156:8085/hi-ee/designer-edit.html?dir=1463377807724/1465647380854&file=3a91fa...

Authorization: Type No Auth

Status: 200 OK Time: 29 ms Size: 12.19 KB

```
<!DOCTYPE html>
<html>
<head>
    <!--Meta header contents-->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta http-equiv='cache-control' content='no-cache'/>
    <meta http-equiv='expires' content='0'/>
    <meta http-equiv='pragma' content='no-cache'/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta name="decorator" content="minimal"/><!--common css -->
<link rel="icon" type="image/x-icon" href="http://192.168.2.156:8085/hi-ee/images/favicon.ico"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/fonts.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/flaticon/flaticon.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/fonts/questrial/questrial.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.mCustomScrollbar.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/jquery.ui.min.css"/>
<link data-clone="true" rel="stylesheet" href="http://192.168.2.156:8085/hi-ee/css/styles.css"/>
<title>HI: Designer-edit</title>
</head>
```

2.8 URL Printing

2.8.1 URL Printing for EFW report

2.8.2 URL Printing for EFWSR report

2.8.3 URL Printing for Adhoc report

2.9 Change Report parameters through URL

2.9.1 Change Report parameters through URL for EFW report

2.9.2 Change Report parameters through URL for EFWSR report

2.9.3 Change Report parameters through URL for Adhoc report

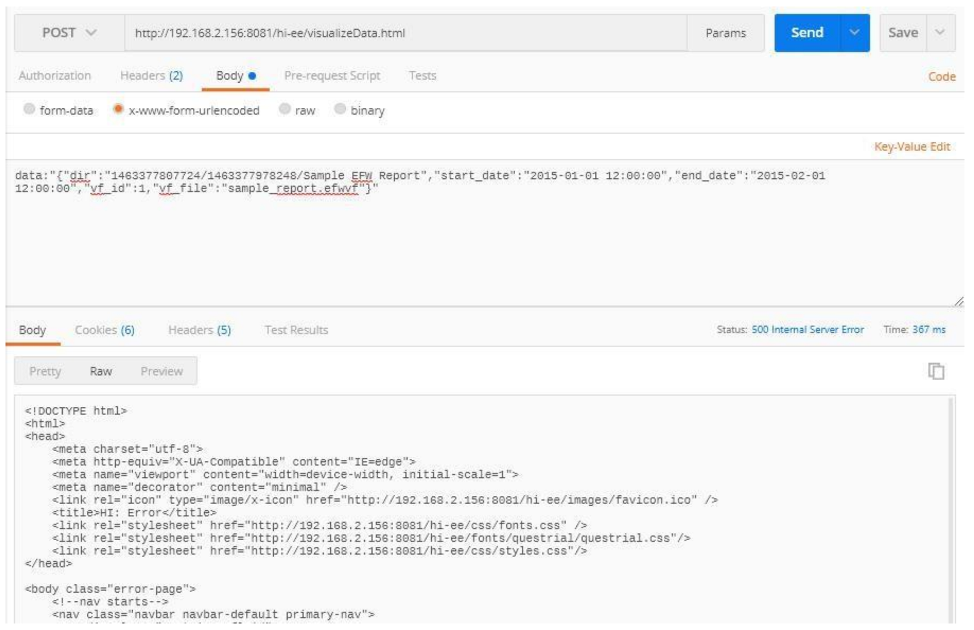
2.10 URL : Cache Refresh

Cache refresh through URL works for dashboard EFW reports , normal efw reports,adhoc reports and for efwsr reports.

Before cache refresh user will get last modified datetime and after cache refresh report will be refresh with recent data.

2.10.1 URL : Cache Refresh for EFW Report

URL	http://192.168.2.156:8081/hi-ee/visualizeData.html	
Description	It allows user to refresh the report cache using URL for EFW report.	
URL	hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&mode=open&refresh=true	
Description	It allows user to refresh the report cache using URL for EFW report. With cache refresh for efw report refresh=true will not be in formdata.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&mode=open&refresh=true **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample%20EFW%20Report&file=sample_report.efw&mode=open&refresh=true" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
data:	"{"dir":"1463377807724/1463377978248/Sample EFW Report","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00","vf_id":1,"vf_file":"sample_report.efwvf"}"	data key contains the report details like its directory, parameters,vf id and name of the vf file
Response Output(JSON Format)	Here response html contents of the requested report.	
Service Status	200 OK	

Screenshot	

```
data:"{"dir":"1463377807724/1463377978248/Sample EFW Report","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01
12:00:00","vf_id":1,"vf_file":"sample_report.efwvf"}"
```

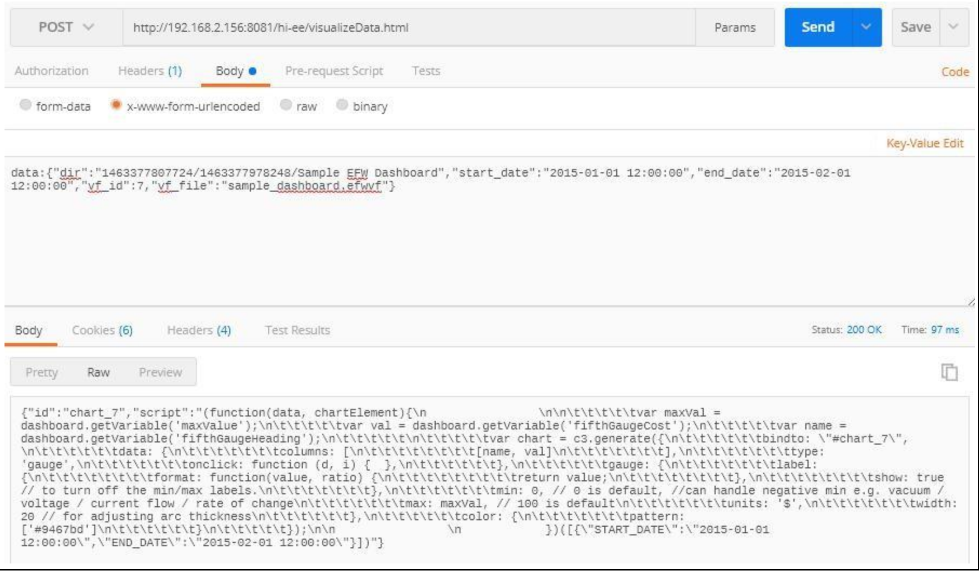
Status: 500 Internal Server Error Time: 367 ms

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="decorator" content="minimal" />
    <link rel="icon" type="image/x-icon" href="http://192.168.2.156:8081/hi-ee/images/favicon.ico" />
    <title>HI: Error</title>
    <link rel="stylesheet" href="http://192.168.2.156:8081/hi-ee/css/fonts.css" />
    <link rel="stylesheet" href="http://192.168.2.156:8081/hi-ee/fonts/questrial/questrial.css"/>
    <link rel="stylesheet" href="http://192.168.2.156:8081/hi-ee/css/styles.css"/>
</head>

<body class="error-page">
    <!--nav starts-->
    <nav class="navbar navbar-default primary-nav">
```

2.10.2 URL : Cache Refresh for Dashboard EFW Report

URL	http://192.168.2.156:8081/hi-ee/visualizeData.html	
Description	It allows user to refresh the dashboard EFW report cache using URL for dashboard EFW report.	
URL	hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Dashboard&file=sample_dashboard.efw&mode=open&refresh=true	
Description	It allows user to refresh the dashboard EFW report cache using URL for dashboard EFW reports.With cache refresh for dashboard EFW report refresh=true key will not be present in formdata.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Dashboard&file=sample_dashboard.efw&mode=open&refresh=true **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463377978248/Sample%20EFW%20Dashboard&file=sample_dashboard.efw&mode=open&refresh=true" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
data:	data:{"dir":"1463377807724/1463377978248/Sample EFW Dashboard","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00","vf_id":7,"vf_file":"sample_dashboard.efwvf"}	data key contains the report details like its directory, parameters,vf id and name of the vf file
Response Output(JSON Format)	Here response will be the dashboard efw reports data.	

Service Status	200 OK

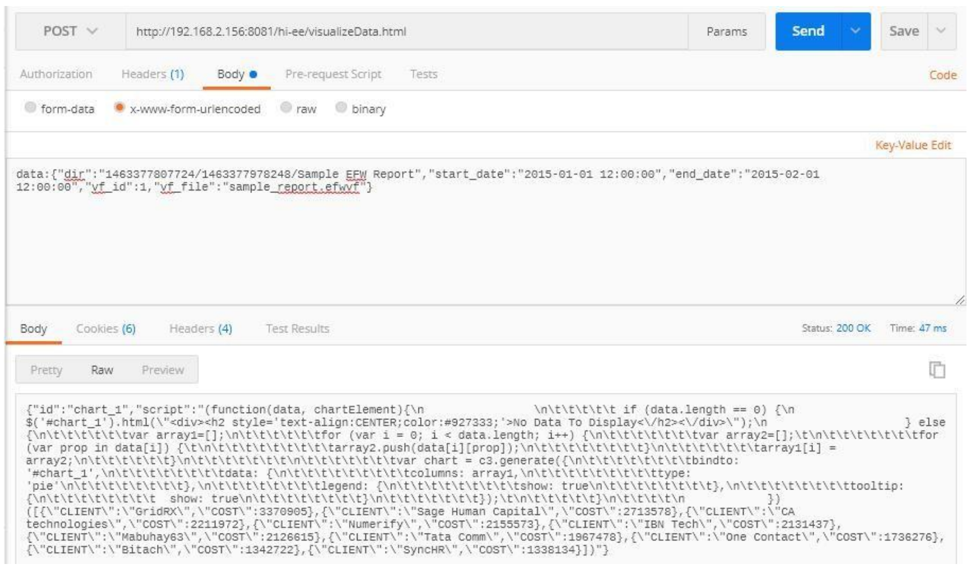
Screenshot	

```
data:{"dir":"1463377807724/1463377978248/Sample EFW Dashboard","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01
12:00:00","vf_id":7,"vf_file":"sample_dashboard.efwvf"}
```

```
{"id":"chart_7","script":"(function(data, chartElement){\n                \n\n\t\t\t\t\tvar maxVal =
dashboard.getVariable('maxValue');\n\t\t\t\t\tvar val = dashboard.getVariable('fifthGaugeCost');\n\t\t\t\t\tvar name =
dashboard.getVariable('fifthGaugeHeading');\n\t\t\t\t\t\n\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\tbindto: \"#chart_7\",
\n\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\tcolumns: [\n\t\t\t\t\t\t\t\t[name, val]\n\t\t\t\t\t\t\t],\n\t\t\t\t\t\t\ttype:
'gauge',\n\t\t\t\t\t\t\tonclick: function (d, i) {  },\n\t\t\t\t\t\t},\n\t\t\t\t\t\tgauge: {\n\t\t\t\t\t\t\tlabel:
{\n\t\t\t\t\t\t\t\tformat: function(value, ratio) {\n\t\t\t\t\t\t\t\t\treturn value;\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\tshow: true
// to turn off the min/max labels.\n\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\tmin: 0, // 0 is default, //can handle negative min e.g. vacuum /
voltage / current flow / rate of change\n\t\t\t\t\t\t\tmax: maxVal, // 100 is default\n\t\t\t\t\t\t\tunits: '$',\n\t\t\t\t\t\t\twidth:
20 // for adjusting arc thickness\n\t\t\t\t\t\t},\n\t\t\t\t\t\tcolor: {\n\t\t\t\t\t\t\tpattern:
['#9467bd']\n\t\t\t\t\t\t}\n\t\t\t\t\t});\n\n              \n            })([{\"START_DATE\":\"2015-01-01
12:00:00\",\"END_DATE\":\"2015-02-01 12:00:00\"}])"}
```

2.10.3 URL : Cache Refresh for EFWSR Report

URL	http://192.168.2.156:8081/hi-ee/visualizeData.html	
Description	It allows user to refresh the EFWSR report cache using URL for EFWSR report.	
URL	hi-ee/hi.html?dir=1463377807724/1463377978248/Sample%20EFW%20Dashboard&file=sample_dashboard.efw&mode=open&refresh=true	
Description	It allows user to refresh the EFWSR report cache using URL for EFWSR reports.With cache refresh for EFWSR report refresh=true key will not be present in formdata.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi-ee/hi.html?dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&mode=open&refresh=true **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1472554245045&file=SavedReport_1472554274862.efwsr&mode=open&refresh=true" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
data:	"{"dir":"1463377807724/1463377978248/Sample EFW Report","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01 12:00:00","vf_id":1,"vf_file":"sample_report.efwvf"}"	data key contains the report details like its directory, parameters,vf id and name of the vf file

Response Output(JSON Format)	Here response will be the efwsr reports data.
Service Status	200 OK
Screenshot	

http://192.168.2.156:8081/hi-ee/visualizeData.html

```
data:{"dir":"1463377807724/1463377978248/Sample EFW Report","start_date":"2015-01-01 12:00:00","end_date":"2015-02-01
12:00:00","vf_id":1,"vf_file":"sample_report.efwvf"}
```

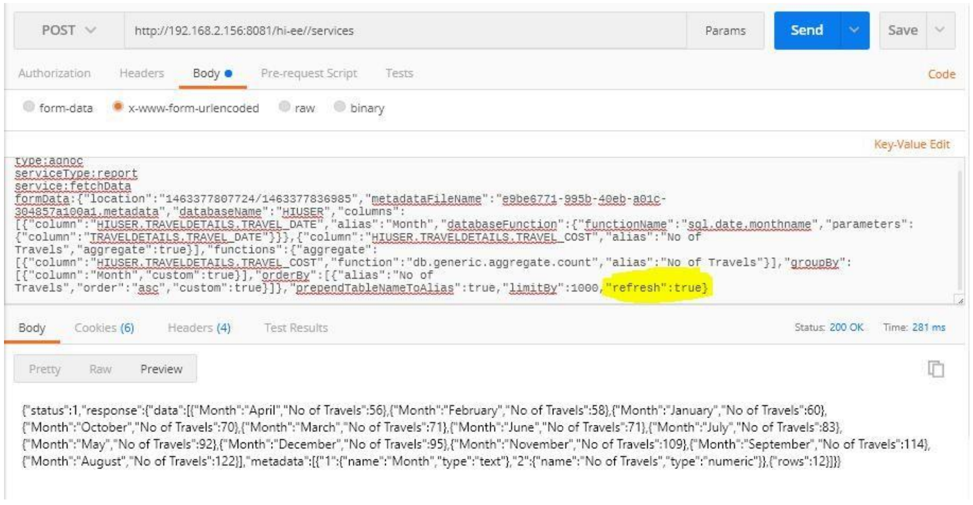
Status: 200 OK Time: 47 ms

```
{"id":"chart_1","script":"(function(data, chartElement){\n                \n\t\t\t\t\t if (data.length == 0) {\n
$('#chart_1').html(\"<div><h2 style='text-align:CENTER;color:#927333;'>No Data To Display<\/h2><\/div>\");\n                } else
{\n\t\t\t\t\t\tvar array1=[];\n\t\t\t\t\t\tfor (var i = 0; i < data.length; i++) {\n\t\t\t\t\t\t\tvar array2=[];\t\n\t\t\t\t\t\t\tfor
(var prop in data[i]) {\t\n\t\t\t\t\t\t\t\t\tarray2.push(data[i][prop]);\n\t\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\tarray1[i] =
array2;\n\t\t\t\t\t\t}\n\t\t\t\t\t\t\n\t\t\t\t\t\tvar chart = c3.generate({\n\t\t\t\t\t\t\t\tbindto:
'#chart_1',\n\t\t\t\t\t\t\t\tdata: {\n\t\t\t\t\t\t\t\t\tcolumns: array1,\n\t\t\t\t\t\t\t\t\ttype:
'pie'\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\tlegend: {\n\t\t\t\t\t\t\t\t\tshow: true\n\t\t\t\t\t\t\t\t},\n\t\t\t\t\t\t\t\ttooltip:
{\n\t\t\t\t\t\t\t\t  show: true\n\t\t\t\t\t\t\t\t}\n\t\t\t\t\t\t\t});\t\n\t\t\t\t\t}\n\t\t\t\t\n            })
([{\"CLIENT\":\"GridRX\",\"COST\":3370905},{\"CLIENT\":\"Sage Human Capital\",\"COST\":2713578},{\"CLIENT\":\"CA
technologies\",\"COST\":2211972},{\"CLIENT\":\"Numerify\",\"COST\":2155573},{\"CLIENT\":\"IBN Tech\",\"COST\":2131437},
{\"CLIENT\":\"Mabuhay63\",\"COST\":2126615},{\"CLIENT\":\"Tata Comm\",\"COST\":1967478},{\"CLIENT\":\"One Contact\",\"COST\":1736276},
{\"CLIENT\":\"Bitach\",\"COST\":1342722},{\"CLIENT\":\"SyncHR\",\"COST\":1338134}])"}
```

2.10.4 URL : Cache Refresh for Adhoc Report

URL	http://192.168.2.156:8081/hi-ee//services	
Description	It allows user to refresh the report cache using URL for adhoc report.	
URL	hi.html?dir=1463377807724/1463983915686/1463838054907&file=d1560c88-be0d-4380-8225-8a8df4eb53bf.report&mode=open&refresh=true	
Description	It allows user to refresh the report cache using URL for adhoc reports. With cache refresh for adhoc report refresh=true key will be present in formdata.	
Pre-requisite	User should have logged in before accessing the service.[Refer login module] If the user is not logged in then you will get login page.	
Accessible for	ROLE_ADMIN,ROLE_USER	
HTTP Request Method	**POST**	
Example	**Access through browser :** http://192.168.2.156:8085/hi-ee/hi.html?dir=1463377807724/1463983915686/1463838054907&file=d1560c88-be0d-4380-8225-8a8df4eb53bf.report&mode=open&refresh=true **Access through Curl command :** curl --data "j_username=hiadmin&j_password=hiadmin&dir=1463377807724/1463983915686/1463838054907&file=d1560c88-be0d-4380-8225-8a8df4eb53bf.report&mode=open&refresh=true" http://192.168.2.156:8085/hi-ee/hi.html -v	
HTTP Request Key	**HTTP Request Value**	**Description**
type:	adhoc	type of service as adhoc
serviceType:	report	serviceType as report
service:	fetchData	service as fetchData

formData:	{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-304857a100a1.metadata","databaseName":"HIUSER","columns":[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_DATE","alias":"Month","databaseFunction":{"functionName":"sql.date.monthname","parameters":{"column":"TRAVELDETAILS.TRAVEL_DATE"}}},{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","alias":"No of Travels","aggregate":true}],"functions":{"aggregate":[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.count","alias":"No of Travels"}],"groupBy":[{"column":"Month","cust	formdata contains the location of the adhoc report file and other report details like columns,applied db function,agg functions , alias name etc. The main parameter in formdata is refresh value as true through which cache will be refreshed.

	om":true}],"orderBy":[{"alias":"No of Travels","order":"asc","custom":true}]},"prependTableNameToAlias":true,"limitBy":1000,"refresh":true}	
Response Output(JSON Format)	Response will be the status 1 with response data with refreshed report data.	
Service Status	200 OK	
Screenshot		

```
type:adhoc
serviceType:report
service:fetchData
formData:{"location":"1463377807724/1463377836985","metadataFileName":"e9be6771-995b-40eb-a01c-
304857a100a1.metadata","databaseName":"HIUSER","columns":
[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_DATE","alias":"Month","databaseFunction":{"functionName":"sql.date.monthname","parameters":
{"column":"TRAVELDETAILS.TRAVEL_DATE"}}},{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","alias":"No of
Travels","aggregate":true}],"functions":{"aggregate":
[{"column":"HIUSER.TRAVELDETAILS.TRAVEL_COST","function":"db.generic.aggregate.count","alias":"No of Travels"}],"groupBy":
[{"column":"Month","custom":true}],"orderBy":[{"alias":"No of
Travels","order":"asc","custom":true}]},"prependTableNameToAlias":true,"limitBy":1000,"refresh":true}
```

```
{"status":1,"response":{"data":[{"Month":"April","No of Travels":56},{"Month":"February","No of Travels":58},{"Month":"January","No of Travels":60},
{"Month":"October","No of Travels":70},{"Month":"March","No of Travels":71},{"Month":"June","No of Travels":71},{"Month":"July","No of Travels":83},
{"Month":"May","No of Travels":92},{"Month":"December","No of Travels":95},{"Month":"November","No of Travels":109},{"Month":"September","No of Travels":114},
{"Month":"August","No of Travels":122}],"metadata":[{"1":{"name":"Month","type":"text"},"2":{"name":"No of Travels","type":"numeric"}},{"rows":12}]}}
```

Future Scope :

1. Helical Workflow (HWF)
2. Template Edit
3. Instant BI