

Rest API for CICS IA

Table of Contents

1. Abstract:.....	3
2. Description:.....	3
3. Architecture:	3
4. Benifits	4
5. Skills Required	4
6. PRE-REQUISITES	4
7. Installationa & Configuration.....	5
a. INSTALLATION (War Deployment).....	5
b. CONFIGURATION.....	5
c. Log file congiguration (Optional)	6
8. Step by Step Usage Guide	7
A. Accessing Swagger Documentation:	7
B. Accessing the First API	8
C. Firing a Request for Swagger UI.....	10
D. API Response.....	10
E. Sample Comparison of CICS IA Plugin with CICS IA API	11
9. Important notes	13

1. ABSTRACT:

Support Pack CICS IA API can be used to demonstrate how runtime data collected by the CICS IA host collectors can be exposed as a Rest Endpoints which can be accessed by browser or can be integrated into any of the web application.

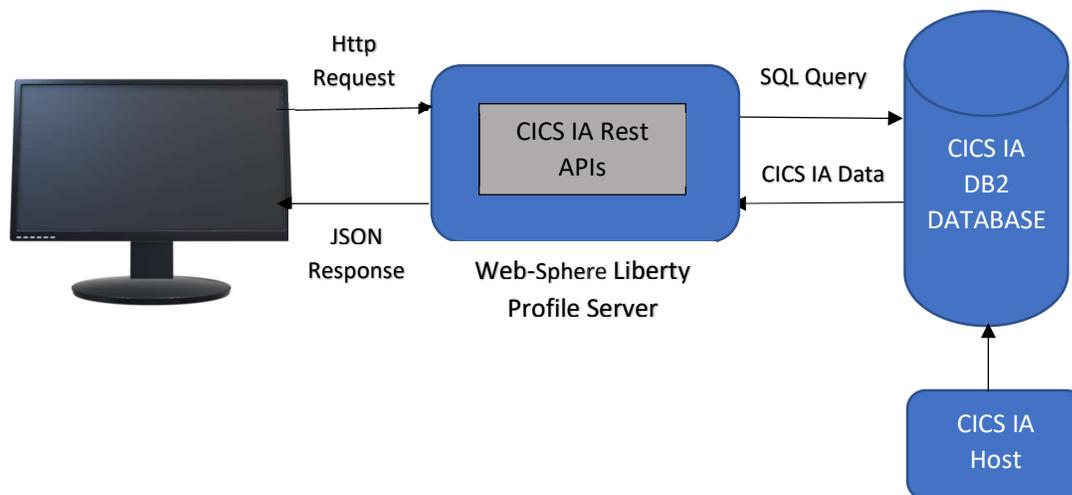
2. DESCRIPTION:

CICS IA has runtime data collected for a CICS Application. The CICS IA Plugin visualizes the data in the form meaningful to the user. However, the integration of this data with other tools is little difficult to achieve, Also the Users needs to live with whatever views and reports that CICS IA provides of the shelf.

Rest Endpoints to the data collected by CICS IA would make sense in such integrations or in any application which like to read the runtime data collected by CICS IA. JSON data is easy to parse and can be integrated into any tool which support HTTP protocol.

Also, Users can bring data together and build their own views and reports as per their needs.

3. ARCHITECTURE:



Component Diagram: Figure 1

As shown in the Figure 1, The CICS IA Rest API is a Web API project to expose the CICS IA Runtime Data as rest End points

- REST APIs is an architectural style
- REST APIs are bundled as a Web Application and hosted on a Web Server
- Rest APIs are hosted on a Web Sphere Liberty profile running on Zos machine
- Clients can fire HTTP requests which invokes the REST APIs

- The Server is configured to connect DB2 on Zos using DB2 Connect Jars.
- The Host collects the runtime data in the CICS IA DB2 tables
- REST API queries the DB for data, processes it, and sends HTTP response containing JSON data.
- The Users can use the browser to find for the Dependency, Affinity or Command Flow data by using the specific API endpoints

4. BENEFITS

- Enables use of modern UI libraries for better visualization of data.
- REST API architecture enables loose coupling between UI and business logic.
- Enables easy integration with other tools.
- Generation of complex reports is possible by using modern reporting frameworks by using JSON response.
- Can create test cases for user applications based on data collected by CICS IA

5. SKILLS REQUIRED

- Basic understanding on the CICS IA and the runtime data that it collects
- Understanding on the JSON and how to interpret the JSON data
- Understanding on how to consume the API response from the existing application, if users want to integrate CICS IA API in the existing project
- Basic understanding on WLP, important configuration files and how to configure wlp for setting up of the CICS IA Rest API

6. PRE-REQUISITES

- Websphere Liberty Profile should be running on Zos Machine where the CICS IA database is installed and listening on a specific port.
- DB2 database schema contains the data collected by CICS IA collector.

- DB2 Connect jars (db2jcc4.jar and db2jcc_license_cisuz.jar) are present in the DB2 directory and the same is referenced in the Server.xml of the WLP server
- Know how on generating a certificate from any certificate issuing authority if users want to configure HTTPS

7. INSTALLATION & CONFIGURATION

A. INSTALLATION (WAR DEPLOYMENT)

- Install the WLP server on the LPAR of your choice. Please refer the following IBM Knowledge center guide:
https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_ins_installz.html
- Create the Server by appropriate name to deploy the CICS IA Rest API War file
Refer the following IBM Knowledge center link
https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/twlp_setup_new_server.html
- Deploy the war file on to the server created by copying and pasting the war file in below location

```
cd path_to_liberty/wlp/usr/servers/"Your_Server_Name"\dropins
```

B. CONFIGURATION

- Go to server.xml present at the below location and Edit the Server.xml to configure the data source for the API's

```
cd path_to_liberty/wlp/usr/servers/"Your_Server_Name"
```

```
<library id="DB2T2LibRef">
  <fileset dir="Path_To_DB2_Jars" includes="*.jar"/>
</library>

<dataSource id="jdbc/CICSIADB" type="javax.sql.DataSource"
jndiName="jdbc/CICSIADB">

<jdbcDriver libraryRef="DB2T2LibRef"/>
<properties.db2.jcc password="Password" user="User_Name"
portNumber="Db_Port" serverName="Domain_Name of the Server"
databaseName="Location for the Database">
```

```
currentSchema="Default_Schema"  
currentFunctionPath="Default_Schema"/>  
</dataSource>
```

Path_To_DB2_Jars: Location of the DB2 connect jar file (db2jcc4.jar and db2jcc_license_cisuz.jar)

Password : DB2 user password

User_Name: DB2 username.

Db_Port : DB2 Port Number.

Domain_Name of the Server : Host name or IP address where the DB2 is installed

Location for the Database : DB2 Location parameter

CurrentSchma : is the schema name of the database that you want to connect to.

CurrentFunctionPath : Schema for executing the stored procedures and functions.

- You Can also configure the Http or Https port using the following configurations

```
<httpEndpoint id="defaultHttpEndpoint" host="*"   
httpsPort="Https_Port_Number" httpPort="Http_Port_Number">  
<sslOptions sslRef="defaultSSLConfig"/>  
</httpEndpoint>
```

- Now you can start and stop the server you have created by following command
Open a command line, then change directory to the wlp/bin directory.

Where path_to_liberty is the location you installed Liberty on your operating system.

```
cd path_to_liberty/wlp/bin
```

to Start

```
server start server_name
```

to Stop

```
server stop server_name
```

C. LOG FILE CONGIGURATION (OPTIONAL)

WLP provide default logging with its installation under the directory

"path_to_liberty/wlp/usr/servers/{server_name}/logs

CICS IA API project gives its own logging as part of the deployment. After the deployment and starting up the server. Users can edit the values present in log4j.properties file present in the following location

"path_to_liberty/wlp/usr/servers/{server_name}/apps/expanded/cics
ia.war/WEB-INF/classes/

Some important Parameters in the file are as below

log4j.appender.file.File: to Specify the log file location on your machine

log4j.rootLogger: to Specify level of logging Like Debug, INFO, ERROR and WARN

8. STEP BY STEP USAGE GUIDE

Once the War file is successfully deployed on a Websphere Liberty Profile server and configured to connect to a DB2 database. The Users can access the application by following URL

URL: [{Protocol}://{Hostname}:{Port}/cicsia](#)

Protocol: Http or Https, based on how the application is configured

Hostname: Host Name or IP Address of the Machine where the CICS IA Rest Api is hosted

Port: Port number on which the WLP server is listening for request

Cicsia: Context root of the CICS IA API Application.

A. ACCESSING SWAGGER DOCUMENTATION:

Users can access the swagger documentation for API by accessing the above URL. Sample Swagger homepage is as shown below

The swagger documentation displays a list of all the APIs that are exposed for the users.

The API' are categorised as below

Program: API's for program (Programs, Program Dependencies and more)

Transaction: API's for Transaction (Transaction List, Transaction Dependencies and more)

Webservice: API's for program (Webservice, Webservice Dependencies and more)

Regions: API's for Region (Region List, Region Dependencies and more)

Platforms: API's for Platform (Platform, Platform Dependencies and more)

Command Flow: API's for Command flow (Command Flows and Command Flow Details)

Thread-Safe Report: API's for Thread-Safe Report (Thread Safe report for Programs and Transaction, Detailed and Summary Reports)

Affinity Report: API get an Affinity Report for given Region

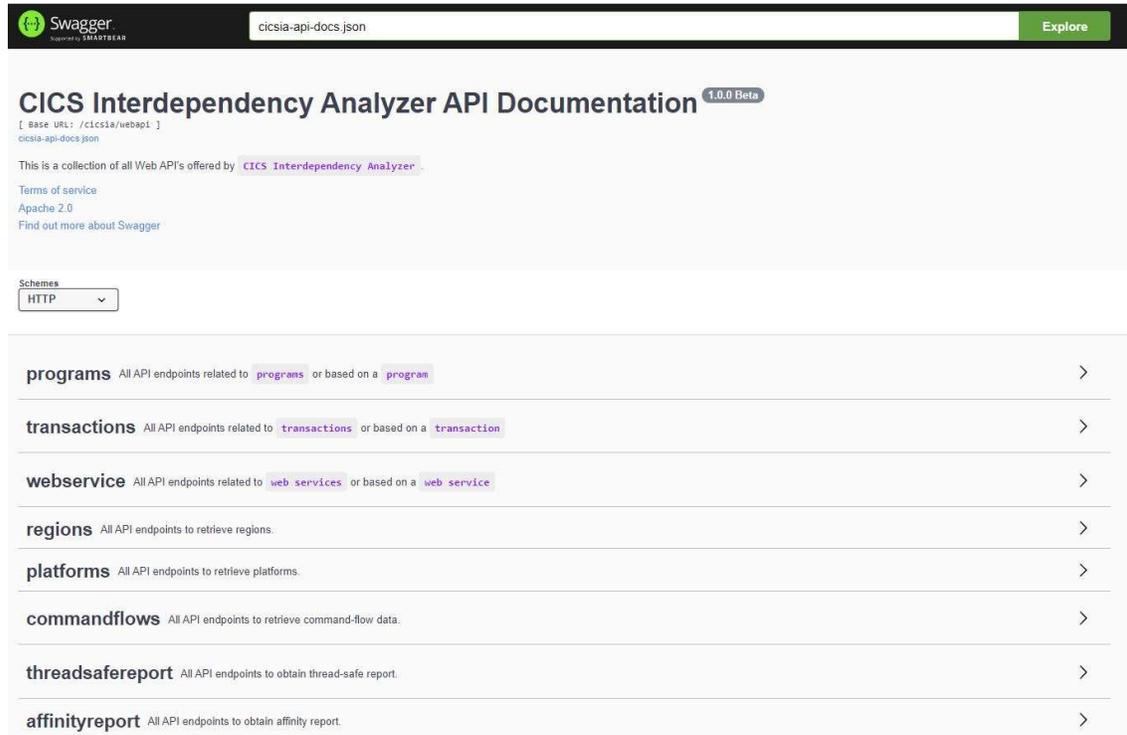


Figure 2

B. ACCESSING THE FIRST API

Users can access a specific API by going to the respective category, expanding it, and selecting an API from the list. This will give additional details about what all information one needs to pass to make a successful API call. Below Picture “Figure 3” shows the sample web page which gets displayed when accessing the specific API endpoint.

As highlighted in the picture below,

Try it out is a clickable link to make an API call

regionId is an optional parameter, user can pass in order to filter the results. The mandatory parameters are displayed with “*required” link next to the parameter in the Swagger page as shown in Figure 4.

programs All API endpoints related to `programs` or based on a `program`

GET `/programs` returns a lists of all programs

This returns a list of all program

Parameters Try it out

Name	Description
<code>regionId</code> string (query)	Region id to filter and list programs of a specific CICS region

Responses Response content type: application/json

Code	Description
200	<p>ok</p> <p>Example Value Model</p> <pre>[{ "programName": "string", "applID": "string", "collectionId": "string", "function": "string" }]</pre>

Figure 3

In the Figure 4, ProgramId is a mandatory parameter, which needs to be provided in order for API to execute successfully. Whereas parameters regionId and resourceType are optional and can be used to further refine the searching of an API.

GET `/programs/usesresources` API to get all the resources used by the given program

API to get all the resources used by the given program. The list will be filtered by type and region if the values are not null.

Parameters Try it out

Name	Description
<code>programId</code> required string (query)	Id of the program
<code>regionId</code> string (query)	Region id to filter and list resources of a specific CICS region
<code>resourceType</code> string (query)	Resource type to filter and list resources of a specific type

Responses Response content type: application/json

Code	Description
200	<p>ok</p> <p>Example Value Model</p> <pre>[{ "applId": "string", "function": "string", "type": "string", "object": "string", "parent": "string" }]</pre>

Figure 4

C. FIRING A REQUEST FOR SWAGGER UI

Users can use the swagger UI to test the API endpoints and to know the exact URL which is being fired with what Query Parameters.

Following Picture <Figure 5> shows how can a user use swagger ui to fire a URL Request

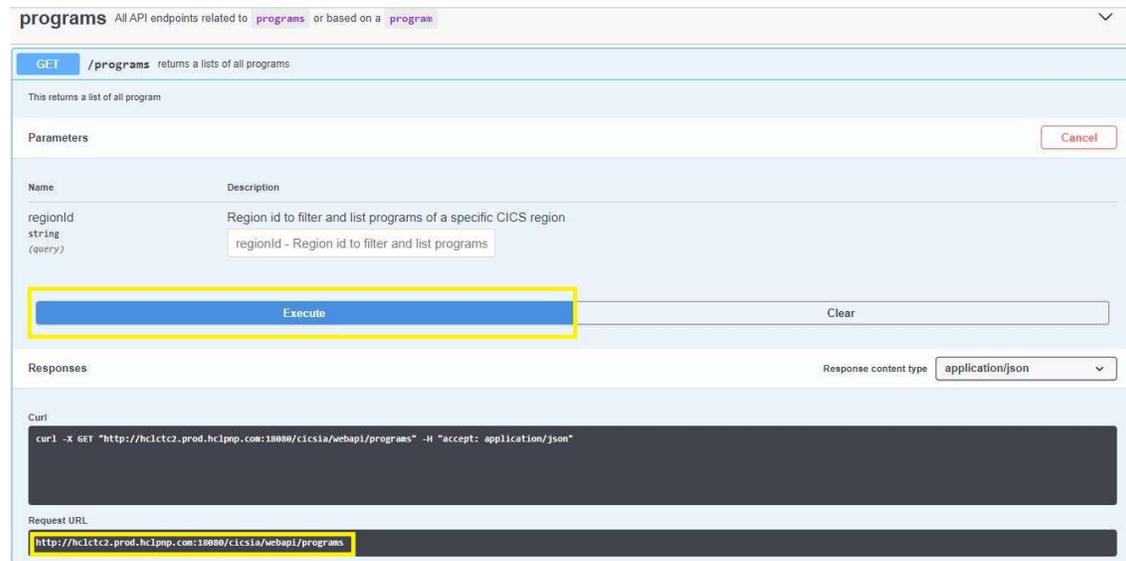


Figure 5

Once you click on "Try it out" link (as shown in figure 3). You see the screen similar to the one in picture above <Figure 5>.

As Highlighted, Execute Will initiate an API request to the specified API Endpoints.

Request URL displays the exact URL which is being fired as part of this request. Which can useful when users want to integrate this API project with their existing web-project.

D. API RESPONSE

Users can verify and download the JSON response from the Swagger UI itself for further processing. Following picture <Figure 6> the API in the swagger UI

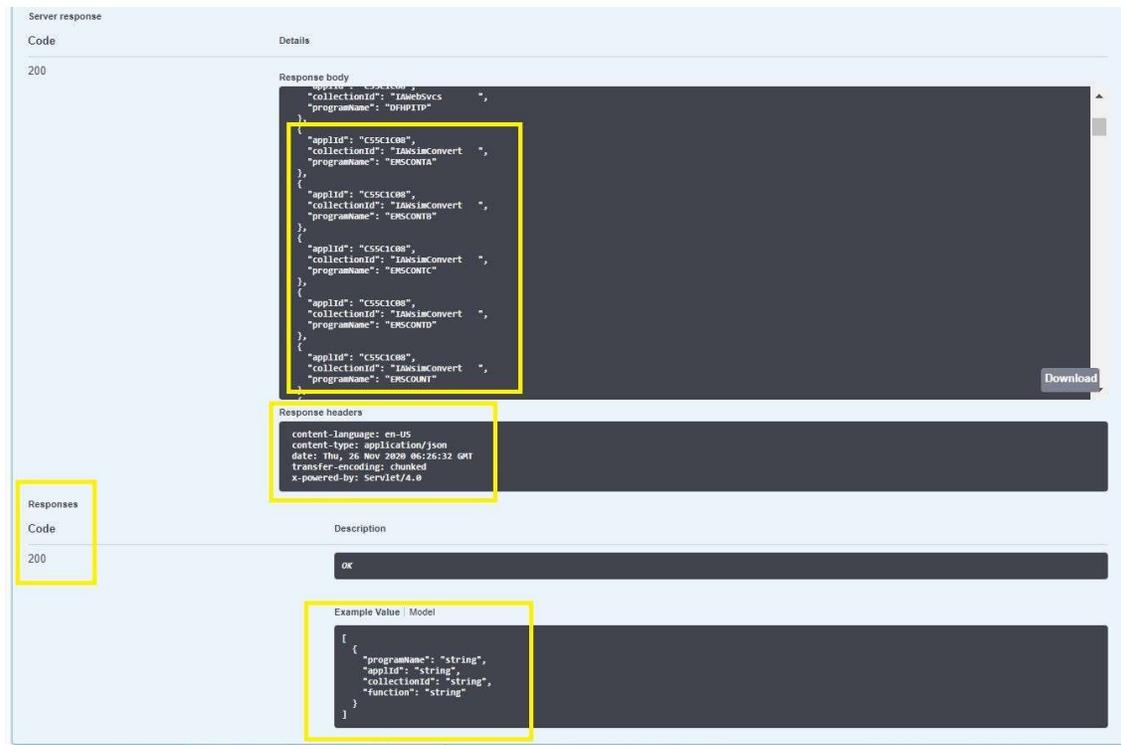


Figure 6

As Highlighted below are some response components which can be useful

Response Body: JSON response of the requested URL

Response Headers: Response Headers like language, content type and data time etc

Response Code: Response Code indicating the whether the request was successful or not

Model: Data Model of the Response

Users can download by clicking on the

E. SAMPLE COMPARISON OF CICS IA PLUGIN WITH CICS IA API

Below section shows how visualization in the CICS IA Plugin compares to the JSON response from the CICS IA API

Following Pictures shows the result of computing a programs usage of a particular program in other programs in CICS IA plugin and API respectively.

CICS IA Plugin:

From the Programs view in the CICS IA plugin, right clicking on the Program “LGDPOL01” and select “Used By Programs” from the context menu as shown in the picture <Figure 7>

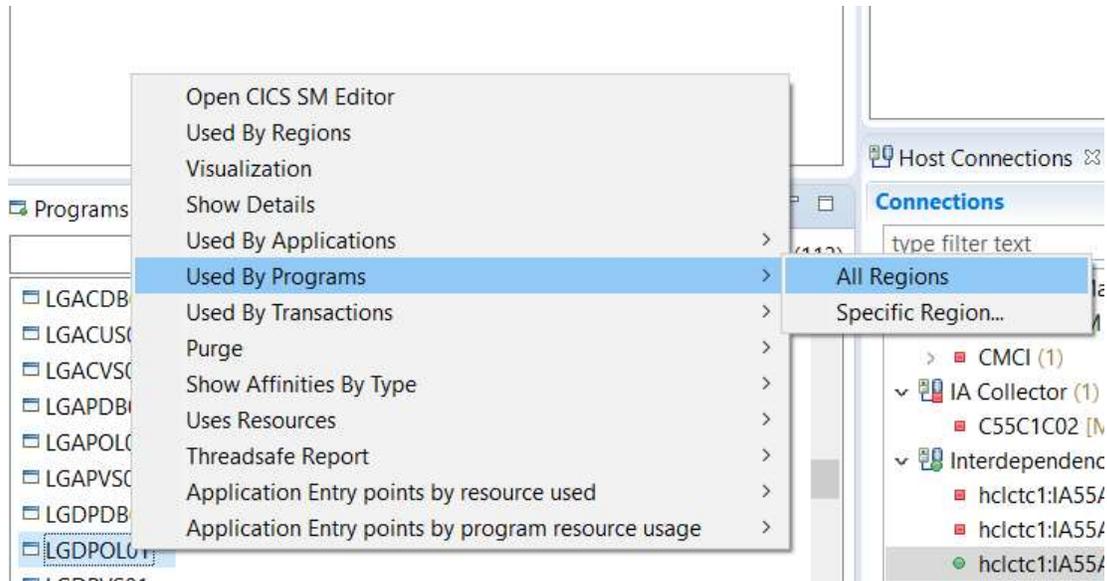


Figure 7

The Result View will be displayed as shown in picture <Figure 8>. View displays the List of Programs using a program “LGDPOL01”, it has 4 programs namely LGTESTP1, LGTESTP2, LGTESTP3 and LGTESTP4

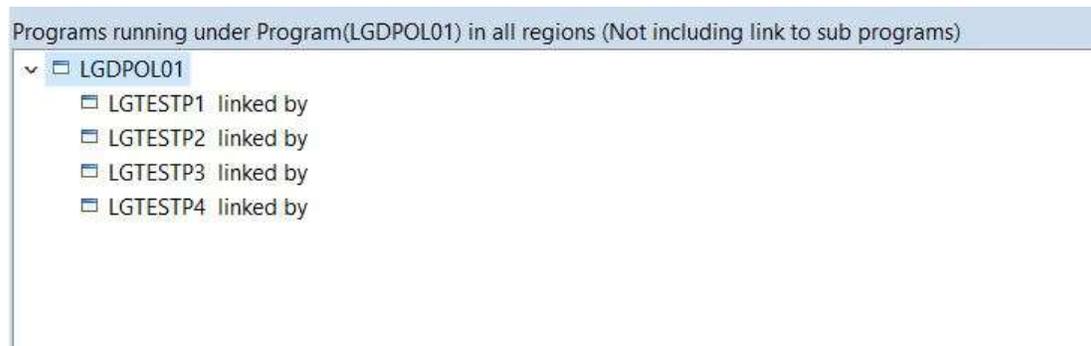


Figure 8

CICS IA API:

From the Swagger, go to the programs-by-program API as shown in the below picture <Figure 9> and enter the value "LGDPOL01" in programId parameter textbox and click on Execute

GET /programs/programs-by-program API to get the programs using the given program

API to get the programs using the given program. If regionId is null then returns all the programs using the given program. If region is not null then returns programs using the given program in specific region.

Parameters

Name	Description
programId * required string (query)	Id of the program <input type="text" value="LGDPOL01"/>
regionId string (query)	Region id to filter and list programs of a specific CICS region <input type="text" value="regionId - Region id to filter and list programs"/>

Execute

Figure 9

The Response as shown in the picture <Figure 10> will be displayed with 4 programs LGTESTP1, LGTESTP2, LGTESTP3 and LGTESTP4 in the list

Response body

```
[
  {
    "applId": "IYDZT328",
    "function": "LINK",
    "programName": "LGTESTP4"
  },
  {
    "applId": "IYDZT328",
    "function": "LINK",
    "programName": "LGTESTP2"
  },
  {
    "applId": "IYDZT328",
    "function": "LINK",
    "programName": "LGTESTP3"
  },
  {
    "applId": "IYDZT328",
    "function": "LINK",
    "programName": "LGTESTP1"
  }
]
```

Download

Figure 10

From the above comparison, in both plugin and API, the data displayed is same in different ways. The API displays the result in JSON response consumable from browser thus can be used easily for integrating with different application for future evaluation.

9. IMPORTANT NOTES

- The SupportPac CICS IA Rest APIs does not implement explicit Authentication and Authorization of its own. The Users if they require, need to configure the API URL to work with their own authentication system.
- The DB2 Connect Jars are not bundled with the product and User need to reference the Jars which comes as part the DB2 installation on Zos

