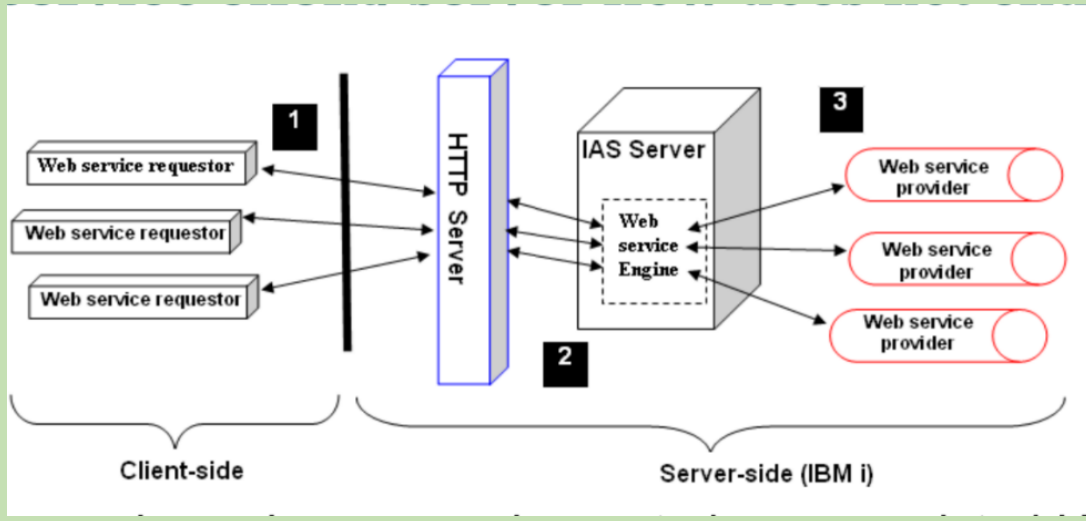


Core-i™ RST

REST/Webservice Framework (middleware and API engine)

v 1.01 – Created 05/28/2020



Label 1	<p>Label 1 consists of...</p> <ol style="list-style-type: none"> 1.) External business partners that exist beyond your vpn (ie. Firewall) and will require a secure https (TLS/encrypted) protocol to be able to access your http server to access your backend API's. 2.) Resources in your organization such as front end developers that would only require an http (generally with user/password auth) protocol to access/utilize your backend API's.
Label 2	<p>Label 2 consists of...</p> <ol style="list-style-type: none"> 1.) Apache HTTP server which is the gateway for the outside world into your IBMi. It will generally be secured with certificate authentication to ensure absolute authorized access. Core-i RST allows for extremely simple creation and deployment of this server configured for your needs whether it be http or https secured. 2.) IBMi which houses the DB2 database.
Label 3	<p>Label 3 consists of...</p> <p>Your backend (IBMi housed) API library. Core-i RST provides a simplified 5250 interface to allow you to create, maintain, validate, and execute your API's before you even attempt to access them from off-platform. Core-i RST provides an automated compiler to generate the code and compile the source for the API *PGM. You just provided the critical dynamic parameters for the API.</p>

INITIAL SETUP

```
*****
RSTMAIN          Core-i RST REST/Webservice Framework      20-05-28
*****
COREIADM                                                13:13:42
```

----- Initial Setup -----

```
Tutorial Popups: N (Y/N)                                     1=Start
Enable SQL/Json: N (Y/N)                                     2=End
                  (*SECADM)
                  http: COREIHTTP      80      N      -
                  https: COREISSL     443      N      N      -
                              (Y/N) (Y/N)
                              (*SECADM)
```

----- Maintenance -----

1. Maintain API Library
2. View API Activity Log

F3=Exit F7=Work With Instances F12=Cancel

Option: █

MA █ A

23/011

This is the main screen for the Core-i RST framework. From here, everything you will need to configure to work for the http framework can be achieved.

This guide should be given to your system administrator who possesses *SECADM privileges.

- **Tutorial Popus**
 - This feature is a simple Y/N toggle.
 - “Y”es = Display help text for each screen navigated to in order to give insight to the product.
 - “N”o = Do not show the help text.
- **Enable SQL/JSON**
 - This is an option the Core-i RST product highly recommends using. Though not required to utilize the framework, the demo API’s shipped with the product will indeed need them enabled if you plan to execute them to demonstrate the product.
 - Not every IBMi has this option enabled. This will allow the new JSON sql functions to be utilized.
 - PTF’s required
 - 7.2 (SF99702 Level 14)
 - 7.3 (SF99703 Level 3)
 - You will need *SECADM privileges to execute this option.
 - This is required to be executed only 1 time.

- If you are not sure if it is loaded, it is a “no harm – no foul” to request the option to enable a 2nd time.

- **Instance**

- **http**

- Instance name
 - Defaults to COREIHTTP – you can change this if desired
- Port
 - Defaults to port 80
 - If you require a different port, simply specify that here
- Certificate
 - A certificate cannot be provided for http protocol
 - Authentication will only be userProfile/Password.
 - NOTE: recommend use within VPN only, though https is always a better option.
- Deploy
 - You can have Core-i RST deploy the instance for you by simply selecting “Y” for Deploy
 - “Deploy” means Core-i RST will do ALL the steps necessary to create a functional http server instance to utilize with the Core-i RST framework. No manual configuration actions will be required.
 - You will need *SECADM privileges to execute this option.
- Start/End
 - Simply enter “1” to start the server instance after it has been created.
 - Simply enter “2” to end the server instance after it has been started
- NOTE:
 - http instance will always be configured to require an IBMi userProfile/password for authentication per any http request.
 - http is an acceptable choice for allowing internal resources (ie. Front end developers, or anyone within your VPN/Firewall) to access the IBMi API’s. The http request/response is no encrypted and security is minimal, however since this is deployed for resources within your organization, the configuration should be sufficient.

- **https**

- Instance name
 - Defaults to COREISSL – you can change this if desired
- Port
 - Defaults to port 443
 - If you require a different port, simply specify that here
- Certificate
 - A certificate can be provided for this protocol.
 - By selecting “Y”, the instance will be configured to REQUIRE a certificate authentication, per any https request. No userProfile/Password is required.

- By selecting “N”, the instance will NOT require a certificate authentication, per any https request. UserProfile/Password is required. Connection will be ssl/tls encrypted.
 - Deploy
 - You can have Core-i RST deploy the instance for you by simply selecting “Y” for Deploy
 - “Deploy” means Core-i RST will do ALL the steps necessary to create a functional http server instance to utilize with the Core-i RST framework. No manual configuration actions will be required.
 - You will need *SECADM privileges to execute this option.
 - Start/End
 - Simply enter “1” to start the server instance after it has been created.
 - Simply enter “2” to end the server instance after it has been started
 - NOTE:
 - https instance will secure the http data transmission with encryption. The level of encryption is based on the IBMi release.
 - IBM i Release QSSLPL *OPSYS definition
 - 7.4 *TLSV1.3 *TLSV1.2
 - 7.3 *TLSV1.2 *TLSV1.1 *TLSV1
 - 7.2 *TLSV1.2 *TLSV1.1 *TLSV1
 - Core-i RST assumes at least a 7.2 release and therefore will disable obsolete SSL protocols SSLv2 SSLv3
- **F7=Work With Instances**
 - Pressing Function Key F7 simply allows you to quickly view and work with your *started* server instances.