z/OSMF 2.2 Advanced Programming

Joey Zhu (zhuxiaoz@cn.ibm.com)
IBM Corporation

Aug 13, 2015
Session Number 17446
Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area. All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.


Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Application Server routing service
  - Topology service
  - Multisystem routing service
  - z/OSMF workflow service
  - Software Management service
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler

- Summary
IBM z/OS Management Facility (z/OSMF) delivers on IBM’s strategy for mainframe simplification and modernization

- The IBM z/OS Management Facility is now a part of z/OS V2R2 that provides support for a modern, Web-browser based management console for z/OS.
- It helps system programmers more easily manage and administer a mainframe system by simplifying day to day operations and administration of a z/OS system.
- More than just a graphical user interface, the z/OS Management Facility is intelligent, addressing the needs of a diversified skilled workforce and maximizing their productivity.
  - Automated tasks can help reduce the learning curve and improve productivity.
  - Embedded active user assistance (such as wizards) guide you through tasks and helps provide simplified operations.
What is z/OSMF

- How does z/OSMF function in the z/OS environment?
  - z/OSMF runs on the z/OS enables managing z/OS from z/OS
    - UI is rendering in browser on a PC
    - No client install required
What is z/OSMF

• Gain simplification and modernization through z/OSMF plugins
What is z/OSMF

What's more?
Agenda

• What is z/OSMF
• What is z/OSMF Advanced Programming
• Using z/OSMF RESTful services
  – z/OS jobs service
  – z/OS data set and file service
  – Application Linking Manager interface service
  – TSO/E address space service
  – Data persistence service
  – Topology service
  – Multisystem routing service
  – z/OSMF workflow service
  – Software Management service
  – Using z/OSMF RESTful services – simple demo
  – Using z/OSMF RESTful services – Simpler than Simpler

• Summary
What is z/OSMF Advanced Programming

• In addition to z/OSMF plugins with modern UI and simplified task, z/OSMF also provides services and facilities to help you write programs.

• z/OSMF Advanced Programming includes:
  – Using z/OSMF RESTful services
  – Develop workflow
  – Create your own z/OSMF plugins

• This session focus on “Using z/OSMF RESTful services”. Please refer to <IBM z/OS Management Facility Programming Guide> for more details about “Develop workflow” and “Create your own z/OSMF plugins”.
What is z/OSMF Advanced Programming

- Why would I need “z/OSMF RESTful services”
  - Representational State Transfer (REST) is a software architecture style. It defines constraints for designing services:
    - Client-Server (Scalability, Simplicity)
    - Stateless (Scalability, Reliability)
    - Uniform interface (Simplicity, Visibility)
      - Identification of resources
      - Manipulation of resources through representation
      - Hypermedia as the engine of application state
  - RESTful web service is based on HTTP channel and it’s lightweight:
    - Has gained widespread acceptance across the Web. (Amazon, Twitter)
    - Easy to call
    - Could be driven remotely (via HTTPS) and securely
    - Language and platform independent
  - z/OSMF RESTful services makes z/OS and z/OSMF more approachable
Why would I need to “develop workflow”

- z/OSMF Workflows application provides a framework supports user (workflow provider) to define a guided flow (a.k.a. workflow) through steps to accomplish a task in z/OS.

- z/OSMF Workflows application is useful to:
  - Assist people unfamiliar with how to perform a given task, or a task that they perform rarely
  - Ensure that all tasks are performed in the right order and only when their dependencies have been met
  - Monitor and track progress toward the completion of the task in the centralized place
  - Provide a history (audit trail) of the steps performed for a task
  - Automate a workflow without the need of interaction
  - Adjust step status dynamically according to various pre-defined condition
  - Work with workflow through REST service instead of having to logon to z/OSMF UI

- Exploiters of workflow
  - zEDC workflow to enable zEDC on multiple systems
  - z/OS Migration workflow
  - z/OSMF Configure workflow
  - z/OSMF CA workflows
What is z/OSMF Advanced Programming

• Why would I need to “create my own z/OSMF plugins”
  – z/OSMF V2R1 provides “External application support” solution enables non-z/OSMF web application (plugin) – html and/or JavaScript applications to render their UI and run in the z/OSMF browser.
  – “External application support” solution provides:
    • z/OS TSO/E address space RESTful service allows “external application” to communicate with their back-end TSO/E application which facilitates reusing existing TSO application to serve web application.
    • Data persistence RESTful service helps “external application” to manage (read/write) their persistent data in z/OS side.
    • Import Manager plugin provides the UI interface to import “external application” into z/OSMF.
  – “Create your own z/OSMF plugins” builds centralized z/OS management portal and makes developing and deploying web application more easy.
Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Application Server Routing service
  - Topology service
  - Multisystem routing service
  - z/OSMF workflow service
  - Software Management service
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler

- Summary
Using z/OSMF RESTful services

- z/OSMF provides a set of RESTful services to make z/OS and z/OSMF more approachable.

- Process overview
  - z/OSMF RESTful services can be invoked by any HTTP client application running on the z/OS local system or remote system.
  - Your application (client) issues HTTP request to the target system (z/OS).
  - z/OSMF determines if the request is valid or not. If it’s valid, z/OSMF performs the requested service and returns the result.

- Security consideration
  - Authenticate to z/OSMF is required. (Certificate authentication is also supported)
  - Other authorization may be required for different RESTful services.
Using z/OSMF RESTful services

- z/OSMF provides below RESTful services:
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Application Server routing service
  - Topology service
  - Multisystem routing service
  - z/OSMF workflow service
  - Software Management service
**z/OS jobs service – API list**

- z/OS jobs RESTful service is provided for working with batch jobs on a z/OS system.
- z/OS jobs RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obtain the status of a job.</td>
<td>GET</td>
</tr>
<tr>
<td>List the jobs for an owner, prefix, or job ID.</td>
<td>GET</td>
</tr>
<tr>
<td>List the spool files for a job.</td>
<td>GET</td>
</tr>
<tr>
<td>Retrieve the contents of a job spool file.</td>
<td>GET</td>
</tr>
<tr>
<td>Submit a job to run on z/OS.</td>
<td>PUT</td>
</tr>
<tr>
<td>Hold a job.</td>
<td>PUT</td>
</tr>
<tr>
<td>Release a job.</td>
<td>PUT</td>
</tr>
<tr>
<td>Change the job class.</td>
<td>PUT</td>
</tr>
<tr>
<td>Cancel a job.</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete a job (cancel a job and purge its output).</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
List the jobs for an owner, prefix or job ID

- **URL format**
  
  There are various formats of the URL for this operation:
  
  - https://host:port/zosmf/restjobs/jobs
  - https://host:port/zosmf/restjobs/jobs?owner=owner
  - https://host:port/zosmf/restjobs/jobs?prefix=prefix*
  - https://host:port/zosmf/restjobs/jobs?owner=owner&prefix=prefix*
  - https://host:port/zosmf/restjobs/jobs?user-correlator=correlator

- **Expected response**

  On completion, the z/OS jobs REST service returns an HTTP response with an Array of matching jobs, each as a JSON job document.
List the jobs for an owner, prefix or job ID

Example request

In the following example, the GET method is used to list the jobs that are owned by “IBMUSER” and have a job name prefix beginning with “TESTJOB”:

```
GET /zosmf/restjobs/jobs?owner=IBMUSER&prefix=TESTJOB* HTTP/1.1
```

Example response

```
HTTP/1.1 200 OK
Date: Fri, 17 Jan 2014 05:39:28 +0000 GMT
Content-Type: application/json
Connection: close

[
  {
  {
]
```
z/OS jobs service – API example

• Submit a job
  – URL format
    https://host:port/zosmf/restjobs/jobs/-JESB
    • To submit a job to secondary JES, use “-JESB” in the URL
    • The job to be submitted could be included in the request body OR resided in a data set or unix file in the host z/OS system.
  – Expected response
    On completion, the z/OS jobs service returns an HTTP response with a JSON job document. The document contains information about the submitted job such as jobid, status, type, etc.
Submit a job

- Example request

  The following request submits a job “TESTJOBX” to run on z/OS. The content of the job to be submitted is included in the request.

  ```plaintext
  PUT /zosmf/restjobs/jobs HTTP/1.1
  Content-Type: text/plain
  X-IBM-Intrdr-Class: A
  X-IBM-Intrdr-Recfm: F
  X-IBM-Intrdr-Lrecl: 80
  X-IBM-Intrdr-Mode: TEXT

  //TESTJOBX JOB (),MSGCLASS=H
  // EXEC PGM=IEFBR14
  ```

- Example response

  ```plaintext
  HTTP/1.1 201 Created
  Date: Fri, 17 Jan 2014 05:39:28 +0000 GMT
  Content-Type: application/json
  Connection: close

  {
    "jobid":"JOB000025","jobname":"TESTJOBX","subsystem":null,"owner":"IBMUSER","status":"INPUT","type":"JOB","class":"A","retcode":null,"url":"https://\//host.port/zosmf/restjobs/jobs\//TESTJOBX\//JOB000025","files-url":"https://\//host.port/zosmf/restjobs/jobs\//TESTJOBX\//JOB000025\//files"
  }
  ```
z/OS jobs service – API exploiters

• By exploiting z/OS jobs RESTful service:
  – application could focus on implementing their own logic and be released from taking care about how to work with z/OS jobs remotely or locally
  – application can easily call these jobs API.

• Current exploiters:
  – Explorer family such as CICS Explorer, z/OS Explorer.
  – z/OSMF Software Management
  – z/OSMF SDSF UI
  – z/OSMF Workflows
  – Customer’s self-developed web application
- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (1/3)
z/OS jobs service – API exploiters

- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (2/3)
z/OS jobs service – API exploiters

- z/OSMF Software Management plugin easily gets job management capability to manage deploy jobs (3/3)
z/OS jobs service – API exploiters

- z/OSMF SDSF UI exploits “z/OS jobs RESTful service” to submit job.
z/OS data set and file service – API list

- z/OS data set and file service is provided for working with data sets and Unix files on a z/OS system.
- z/OS data set and file RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the data sets on a z/OS system. You can filter the returned list of data set names through the specification of high-level qualifiers and wildcards.</td>
<td>GET</td>
</tr>
<tr>
<td>List the members of a z/OS partitioned data set (PDS or PDSE). You can filter the returned list of member names through the specification of wildcards.</td>
<td>GET</td>
</tr>
<tr>
<td>List the files and directories in a UNIX file path on a z/OS system.</td>
<td>GET</td>
</tr>
<tr>
<td>Retrieve the contents of a sequential data set, or a member of a PDS or PDSE.</td>
<td>GET</td>
</tr>
<tr>
<td>Retrieve the contents of a z/OS UNIX file.</td>
<td>GET</td>
</tr>
<tr>
<td>Write data to a sequential data set or a member of a PDS or PDSE.</td>
<td>PUT</td>
</tr>
<tr>
<td>Write data to a z/OS UNIX file.</td>
<td>PUT</td>
</tr>
</tbody>
</table>
List the z/OS data sets on a system

- URL format
  - https://host:port/zosmf/restfiles/ds/?dslevel=filter-criteria

- Expected response
  On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred.

For a successful request, the HTTP response includes an array of matching data sets, each as a JSON list document.
**z/OS data set and file service – API example**

- **List the z/OS data sets on a system**
  - **Example request**
    The following request is used to list all of the cataloged data sets that match the partial name "sys1.*lib". That is, the data sets that have a name beginning with "sys1". and a last qualifier that ends in "lib".

    ```
    GET https://zosmf1.yourco.com/zosmf/restfiles/ds/?dslevel=sys1.*lib HTTP/1.1
    ```

  - **Example response**

    ```
    HTTP/1.1 200 OK
    Date: Wed, 23 Oct 2013 00:42:49 GMT
    Content-Type: application/json
    Connection: close
    
    { "items": [ { "dsname": "SYS1.AUXLIB" }, { "dsname": "SYS1.BDTLIB" }, { "dsname": "SYS1.CHSLIB" }, { "dsname": "SYS1.COMLIB" }, { "dsname": "SYS1.COBLIB" }, { "dsname": "SYS1.CSSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" }, { "dsname": "SYS1.PDSLIB" } ], "returnedRows": 43 }
    ```
• Retrieve the contents of a z/OS data set or member
  – URL format
    For a request to retrieve data from a sequential data set:
      https://host:port/zosmf/restfiles/ds/<data-set-name>
    For a request to retrieve data from a member of a PDS or PDSE:
      https://host:port/zosmf/restfiles/ds/<data-set-name>(<member-name>)
    For a request to retrieve data from an uncataloged sequential data set:
    For a request to retrieve data from a member of an uncataloged PDS or PDSE:
      https://host:port/zosmf/restfiles/ds/-<volser>/ds/<data-set-name>(<member-name>)
  – Expected response
    On completion, the service returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success.
z/OS data set and file service – API example

- Retrieve the contents of a z/OS data set or member
  - Example request
  The following request is used to retrieve the contents of the member SMFPRM00 in data set SYS1.PARMLIB:

  ```
  GET https://zosmf1.yourco.com/zosmf/restfiles/ds/SYS1.PARMLIB(SMFPRM00) HTTP/1.1
  ```

  - Example response

  ```
  200 OK
  Etag: B5C6484F7B5590A4B3EC15BB88E29EA53
  Content-Type: text/plain; charset=UTF-8
  Content-Language: en-US
  Content-Length: 1944
  Date: Fri, 07 Nov 2014 02:13:07 GMT
  Connection: close
  ```

  ```
  ... (some content of the file) ...
  ```
z/OS data set and file service – API exploiters

- By exploiting z/OS data set and file RESTful service, application could easily get the capability of managing data set and unix files remotely or locally.

- Current exploiters:
  - z/OSMF Software Management
  - Customer’s self-developed web application
z/OS data set and file service – API exploiters

- z/OSMF Software Management plugin simplifies the process of adding data set into software instance.
- Previously, user has to enter data set name manually, and, therefore, one data set at a time.

![Add Data Set GUI](image)
With exploiting “z/OS data set and file service”, Software Management plugin provides searching data set function. User can select multiple data sets returned by the search operation and add them once.
z/OS data set and file service – API exploiters

- Build something cool by exploiting “z/OS data set and file service”:

Search for and list z/OS data sets and files.

Resource type:
- [ ] z/OS data sets
- [ ] z/OS UNIX files

Name:

- IBMUSER
- ▲ Back
- IBMUSER.HOSTS.ADDRINFO
- IBMUSER.HOSTS.SITEINFO
- IBMUSER.ISPFWEB.EXEC
- IBMUSER.ISPFWEB.LOAD.OLD
- IBMUSER.ISPFWEB.LOAD.OLD2
- IBMUSER.ISPFWEB.LOAD.OLD3
- IBMUSER.SPFWEB.MSGTYPES.LOAD
- IBMUSER.SPFWEB.PANELS
- IBMUSER.SPFWEB.SISPMENU
- ▼ Next

Volume:
- [ ] Select or type. ▼
- [ ] Search

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

© Copyright IBM Corporation 2015 8/1/15
Application Linking Manager service

- To perform traditional system management tasks in z/OS, you might interact with several different interfaces.
- In z/OSMF, it is possible to link or connect some of these tasks and external applications together for a smoother user experience via the Application linking Manager service.
- Key components:
  - Event Requestor: z/OSMF task or external application
  - Event: Action requested by the “Event Requestor”
  - Event type: Object that contains an “Event Requestor” to an “Event Handler”
  - Event handler: z/OSMF task or external application
Application Linking Manager service – API list

• Application Linking Manager service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register an event type.</td>
<td>POST</td>
</tr>
<tr>
<td>Register a handler for an event type</td>
<td>POST</td>
</tr>
<tr>
<td>List all tasks that are eligible to be handlers</td>
<td>GET</td>
</tr>
<tr>
<td>List the registered handlers for an event type</td>
<td>GET</td>
</tr>
<tr>
<td>Unregister a handler</td>
<td>DELETE</td>
</tr>
<tr>
<td>Unregister an event type</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
Register a handler for an event type

- URL format
  https://{host}:{port}/zosmf/izual/rest/handler?eventTypeId={eventTypeId}

Request content:
  type: handler type. “INTERNAL” for z/OSMF plug-in, “EXTERNAL” for external application.
  Id: Unique identifier for a launch point within the handler task or application.
  applID: Identifier assigned to z/OSMF plugin
  displayName: handler name
  URL: URL to be used for accessing the handler.
  options: indicates how the handler will be displayed when it process events.

- Expected response
  On completion, the Application Linking Manager interface returns an HTTP response, which includes a status code indicating whether your request completed. Status code 200 indicates success. A status code of 4nn or 5nn indicates that an error has occurred.
• z/OSMF WLM plugin is used to define performance policy. z/OSMF RMF plugin is used to monitor performance. By exploiting “Application Linking Manager service”, WLM and RMF provide a smooth user experience when there is need to switch between these two plugins.

• Example of linkage between z/OSMF RMF plugin and z/OSMF WLM plugin (1/2)
  – Event Requestor: z/OSMF RMF plugin
  – Event type: IBM.ZOSMF.VIEW_ACTIVE_WLM_SERVICE_DEFINITION.SERVICE_CLASS
  – Event Handler: z/OSMF WLM plugin
Application Linking Manager service – API exploiters

- Example of linkage between z/OSMF RMF plugin and z/OSMF WLM plugin (2/2)
**TSO/E address space service – API list**

- TSO/E address space service is provided for web application (especially for external web application) to communicate with their back-end TSO/E application running on the z/OS system.
- TSO/E address space RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start or reconnect to a TSO/E address space</td>
<td>POST</td>
</tr>
<tr>
<td>Start an application on a TSO/E address space</td>
<td>POST</td>
</tr>
<tr>
<td>Receive messages from a TSO/E address space</td>
<td>GET</td>
</tr>
<tr>
<td>Receive messages from an application running in a TSO/E address space</td>
<td>GET</td>
</tr>
<tr>
<td>Send messages to a TSO/E address space</td>
<td>PUT</td>
</tr>
<tr>
<td>Send messages to an application running in a TSO/E address space</td>
<td>PUT</td>
</tr>
<tr>
<td>Ping a TSO/E address space</td>
<td>PUT</td>
</tr>
<tr>
<td>End a TSO/E address space</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
Data persistence service – API list

- Data persistence service is provided for web application (especially for external web application) to manage their persistent data in z/OS system.
- Data persistence service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persist user-specific and global application data</td>
<td>PUT</td>
</tr>
<tr>
<td>Retrieve user-specific and global application data</td>
<td>GET</td>
</tr>
<tr>
<td>Delete user-specific and global application data</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
Exploiters (not only web application) could easily get the capability of communicating with TSO/E application and managing persistence data in the z/OS system such that they can focus on implementation of business logic.

z/OS SDSF created their own z/OSMF plugin (a.k.a. external plugin) “SDSF UI” by exploiting these RESTful services.
Questions:
- What if my server-end code needs to be run in an application server?
- Can I have my java code running in z/OSMF server?
Application server routing service – Overview

- Application server routing service supports routing requests and responses between client-side and server-side code for any z/OSMF plug-ins you created where the server-side code is hosted on an application server other than the z/OSMF server.
Application server routing service – API list & example

• API list

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve data from an application server</td>
<td>GET</td>
</tr>
<tr>
<td>Update data for an application server</td>
<td>POST or PUT</td>
</tr>
<tr>
<td>Delete data from an application server</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

• API example

You client-side code which is rendered in z/OSMF window is about to add objectC on the application server identified in system entry “appServer1”, which is defined in the z/OSMF Systems task, submit the following request:

```
POST /zosmf/externalgateway/system HTTP/1.1
Host: appname.yourco.com

{"target":"appServer1","resourcePath":"/testApp/objectC","content":{"attribute1":"value11","attribute2":"value12","attribute3":"value13","attribute4":"value14","attribute5":"value15"}}
```
Topology service – API list

- Topology RESTful service is provided for working with the groups, sysplexes and the systems that are defined to z/OSMF.

- Topology RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the systems defined to z/OSMF</td>
<td>GET</td>
</tr>
<tr>
<td>List the groups defined to z/OSMF</td>
<td>GET</td>
</tr>
<tr>
<td>List the systems included in a group</td>
<td>GET</td>
</tr>
<tr>
<td>List the sysplexes defined to z/OSMF</td>
<td>GET</td>
</tr>
<tr>
<td>List the systems included in a sysplex</td>
<td>GET</td>
</tr>
</tbody>
</table>
Multisystem routing service – API list

- To communicate with and transfer data between systems within your enterprise, z/OSMF uses z/OSMF-to-z/OSMF communication. Multisystem routing service plays key role in the z/OSMF-to-z/OSMF model. It has the capability of:
  - Forwarding request to single remote z/OSMF instance and return the response (Basic-Proxy mode)
  - Forwarding request to a group of remote z/OSMF instances, packaging the responses and return it. (Aggregation mode)
- Multisystem routing service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieve data from one system, a list of systems, or all the systems in a group.</td>
<td>GET</td>
</tr>
<tr>
<td>Update data for one system, a list of systems, or all the systems in a group.</td>
<td>POST or PUT</td>
</tr>
<tr>
<td>Delete data from one system, a list of systems, or all the systems in a group.</td>
<td>DELETE</td>
</tr>
<tr>
<td>Authenticate with a secondary z/OSMF instance.</td>
<td>POST</td>
</tr>
</tbody>
</table>
• Exploiters could:
  – Retrieve topology information through “Topology service”
  – Manage multiple systems with only connected to primary z/OSMF which runs “Multisystem routing service”.
  – Get the aggregation capability by specifying target systems/groups when accessing the “Multisystem routing service”.

• z/OS V2.2 z/OSMF allows you to obtain an aggregated display of incidents across z/OSMF instances within your enterprise
Workflow service – API list

- Workflow RESTful service allows user to create, start and manage workflow in z/OS through programmatic way instead of having to operate in z/OSMF UI.

- Workflow RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a workflow</td>
<td>POST</td>
</tr>
<tr>
<td>Start a workflow</td>
<td>PUT</td>
</tr>
<tr>
<td>Lists workflows by search criteria</td>
<td>GET</td>
</tr>
<tr>
<td>Delete a workflow</td>
<td>DELETE</td>
</tr>
<tr>
<td>Cancel workflow</td>
<td>PUT</td>
</tr>
<tr>
<td>Retrieve workflow definition</td>
<td>GET</td>
</tr>
</tbody>
</table>
Workflow service – API exploiter

• User scenario
  – Product (workflow provider) ships workflow which intends to accomplish a task in z/OS system.
  – End user of the product (or even the product itself) could build application which (remotely or locally) calls workflow RESTful service to:
    • Create workflow on demand
    • Perform the workflow to accomplish the task in an automatic way
    • Activities of performing workflow is recorded by z/OSMF workflows application for future review
Software management service – API list

- Software management RESTful service allows a client application to interact with the z/OSMF Software Management task.
- Software management RESTful service provides below operations (APIs):

<table>
<thead>
<tr>
<th>Operation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>List the software instances defined to z/OSMF.</td>
<td>GET</td>
</tr>
<tr>
<td>Retrieve the properties of a software instance.</td>
<td>GET</td>
</tr>
<tr>
<td>List the data sets included in a software instance</td>
<td>GET</td>
</tr>
<tr>
<td>Load the products, features, and FMIDs for a software instance</td>
<td>GET</td>
</tr>
<tr>
<td>Add a new software instance.</td>
<td>POST</td>
</tr>
<tr>
<td>Modify the properties of a software instance</td>
<td>PUT</td>
</tr>
<tr>
<td>Delete a software instance</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
Agenda

• What is z/OSMF
• What is z/OSMF Advanced Programming
• Using z/OSMF RESTful services
  – z/OS jobs service
  – z/OS data set and file service
  – Application Linking Manager interface service
  – TSO/E address space service
  – Data persistence service
  – Application Server routing service
  – Topology service
  – Multisystem routing service
  – z/OSMF workflow service
  – Software Management service
  – Using z/OSMF RESTful services – simple demo
  – Using z/OSMF RESTful services – Simpler than Simpler

• Summary
Using z/OSMF RESTful services – simple demo

- Experience z/OSMF RESTful services with a simple html page (1/3)

```html
<body>
  <div>
    <button id="getMyJobs" onclick="getMyJobs()">List my jobs</button>
  </div>
  <hr />
  <div id="submitArea">
    <textarea id="jclArea" style="width: 400px">//TESTJOBW JOB (),MSGCLASS=H
      EXEC PGM=IEFBR14</textarea>
  </div>
  <button id="subbtn" onclick="submit()">Submit a job</button>
  <hr />
  <div id="getJobArea">
    <label>Job Name:</label><input id="jobname" />
    <label>Job ID:</label><input id="jobid" />
  </div>
  <button id="getbtn" onclick="getJobStatus()">Get job status</button>
  <div id="resultArea">
    <div><h4>Result:</h4></div>
    <div><div id="result"></div></div>
  </div>
</body>
</html>
```
Using z/OSMF RESTful services – simple demo

• Experience z/OSMF RESTful services with a simple html page (2/3)

```html
<head>
  <title>Demo of REST zOS Jobs service</title>
  <script type="text/javascript">
    function submit() {
      var jcl = document.getElementById("jclArea").value;
      var xhrs = new XMLHttpRequest();
      xhrs.open("PUT", "https://aws1.centers.ihost.com/zosmf/restjobs/jobs", true);
      xhrs.setRequestHeader("Content-Type", "text/plain");
      xhrs.onreadystatechange = function () {
        if(xhrs.readyState == 4) {
          document.getElementById("result").innerHTML = xhrs.responseText;
        }
      }
      xhrs.send(jcl);
    }
    function sendGetRequest(url) {
      var xhrget = new XMLHttpRequest();
      xhrget.open("GET", url, true);
      xhrget.onreadystatechange = function () {
        if(xhrget.readyState == 4) {
          document.getElementById("result").innerHTML = xhrget.responseText;
        }
      }
      xhrget.send();
    }
    function getJobsStatus() {
      var xhrg = new XMLHttpRequest();
      var jobname = document.getElementById("jobname").value;
      var jobid = document.getElementById("jobid").value;
      var url = "https://aws1.centers.ihost.com/zosmf/restjobs/jobs/" + jobname + "/" + jobid;
      sendGetRequest(url);
    }
    function getMyJobs() {
      var url = "https://aws1.centers.ihost.com/zosmf/restjobs/jobs/";
      sendGetRequest(url);
    }
  </script>
</head>
```

Construct the URL

Access the RESTful service with JCL to be submitted

Construct the URL

Access the RESTful service

Construct the URL

Access the RESTful service
Using z/OSMF RESTful services – simple demo

- Experience z/OSMF RESTful services with a simple html page (3/3)

```javascript
//TESTJOBW JOB { }, MSGCLASS=K
// EXEC PGM=IEFBR14

Submit a job

Job Name: TESTJOBW  Job ID: JO827785

Get job status

Result:

```json
{
    "jobname": "TESTJOBW",
    "retcode": "CC0000",
    "subsystem": "JES2",
    "status": "OUTPUT",
    "owner": "SHARA01",
    "jobid": "JOB27785",
    "class": "A",
    "job-correlator": "J0027785N1.....CE96D4ED.....",
    "type": "JOB",
    "phase": 20,
    "phase-name": "Job is on the hard copy queue"
}
```
Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Application Server routing service
  - Topology service
  - Multisystem routing service
  - z/OSMF workflow service
  - Software Management service
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler

- Summary

** available for z/OSMF V2R1 with APAR PI32148

Complete your session evaluations online at www.SHARE.org/Orlando-Eval
Experience z/OSMF RESTful services without the need to write any code: What you need are just:

- Valid z/OSMF user id and password
- Firefox browser with “HttpRequester” plugin installed

Example 1 – list jobs owned by your logon user
Using z/OSMF RESTful services - Simpler than Simpler

- Example 2 – list data set names matches the criteria of “SYS1.*lib”
Agenda

- What is z/OSMF
- What is z/OSMF Advanced Programming
- Using z/OSMF RESTful services
  - z/OS jobs service
  - z/OS data set and file service
  - Application Linking Manager interface service
  - TSO/E address space service
  - Data persistence service
  - Application Server routing service
  - Topology service
  - Multisystem routing service
  - z/OSMF workflow service
  - Software Management service
  - Using z/OSMF RESTful services – simple demo
  - Using z/OSMF RESTful services – Simpler than Simpler

- Summary
Summary

• In addition to z/OSMF plugins with modern UI and simplified task, z/OSMF also provides services and facilities to help you write programs.
• z/OSMF Advanced Programming consists of:
  – Using z/OSMF RESTful services
  – Develop workflow
  – Create your own z/OSMF plugins
• z/OSMF RESTful services make z/OS and z/OSMF more approachable:
  – z/OS jobs service
  – z/OS data set and file service
  – Application Linking Manager interface service
  – TSO/E address space service
  – Data persistence service
  – Application Server routing service
  – Topology service
  – Multisystem routing service
  – z/OSMF workflow service
  – Software Management service
Advertisements

- Possible missed opportunities (handouts available for download):
  - Session 17795 What's new in z/OSMF 2.2?
  - Session 17236 z/OSMF 2.2 Implementation and Configuration
  - Session 17841 z/OSMF Roundtable
  - Session 17422 z/OSMF Hands-on Labs - Choose Your Own I
  - Session 17909 z/OSMF Hands-on Labs - Choose Your Own - II
Thank You

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

© Copyright IBM Corporation 2015
Appendix

- **z/OSMF Home Page**
  http://www-03.ibm.com/systems/z/os/zos/features/zosmf/index.html

- **z/OSMF V2R1 Programming Guide**
  http://www-01.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.izua700/toc.htm