regoUniversity 2018

GEL Scripts | Advanced

Your Guide: James Gille and Ben Rimmasch

Introductions

- Take 5 Minutes
- Turn to a Person Near You
- Introduce Yourself
- Business Cards

Agenda

- File Operations
- Error Handling
- JAVA in GEL
- REST Web Services
- Exercise

File Operations

regoUniversity 2018

Let Rego be your guide.

File Operations – Overview

- The CA provided GEL tag libraries can
  - Read files and write to a file
  - Perform a set of FTP operations
  - Parse nodes and attributes of XML or comma-delimited files with some limitations
- The CA provided GEL tag libraries cannot
  - Create a directory to put files in
  - Move files around
  - Delete files after it is done with them
- Java can almost always fill in the gaps

File Operations – Read File

- Below is an example script to read a pipe delimited file
  - 'embedded' attribute determines if data is surrounded by quotes or not. Default value is false.

```
<gel:script xmlns:core="jelly:core"
   xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
   xmlns:files="jelly:com.niku.union.gel.FileTagLibrary">

   <gel:parameter var="vFileName" default="/fs0/CA PPM1/share/RESOURCES.CSV"/>

   <files:readFile fileName="${vFileName}" delimiter="\|" var="vResourceData" embedded="false"/>

   <core:forEach items="${vResourceData.rows}" var="row" begin="1" end="10">
      <gel:log level="INFO"> Resource Last Name: ${row[0]} </gel:log>
      <gel:log level="INFO"> Resource First Name: ${row[1]} </gel:log>
   </core:forEach>

</gel:script>
```

*Note: The only delimiter that works for tab is: &#x9

File Operations – Write File

- Writing files is a way to export data, perhaps to be used in an integration by another application

```
<file:writeFile delimiter="," embedded="false" fileName="Resources.csv">
  <sql:query dataSource="${CA_PPMDS}" escapeText="false" var="result">
    <![CDATA[
    SELECT  u.first_name firstName,
            u.last_name lastName,
            u.user_name userName
    FROM    cmn_sec_users u
    WHERE   u.user_status_id = 200
    ]]>
  </sql:query>
  <core:forEach items="${result.rows}" trim="true" var="row">
    <file:line>
      <file:column value="${row.userName}"/>
      <file:column value="${row.lastName}"/>
      <file:column value="${row.firstName}"/>
    </file:line>
  </core:forEach>
</file:writeFile>
```

- Use serialize tag to write XML content to an XML file

```
<gel:serialize fileName="C:\XOG\output\lookup_mapping.xml" var="${loadContent}"/>
```

File Operations – FTP

- By including the namespace for the FTPTagLibrary, GEL has the ability to read and write files on an FTP Server
  - This library does not support SFTP, FTPS or passive mode
  - Namespace

    <gel:script **xmlns:ftp="jelly:com.niku.union.gel.FTPTagLibrary"**>
  - Tags
    - ftp:open
    - ftp:put
    - ftp:get

File Operations – FTP Cont'd

- Below is an example of reading a file from an FTP location

```
<ftp:open hostName="myclarityserver" user="niku" password="clarity">
   <ftp:get localDir="c:/temp" fileName="app-ca.log" remoteDir="/niku/clarity/logs"/>
</ftp:open>
```

- Below is an example of writing a file to an FTP location

```
<ftp:open hostName="localhost" user="niku" password="clarity">
   <ftp:put localDir="/home/niku/xog/bin" fileName="gel.bat" remoteDir="/tmp"/>
</ftp:open>
```

Error Handling

regoUniversity 2018

Let Rego be your guide.

Error Handling

- The <core:catch> tag can be used to capture and handle exceptions that occur.

```
<core:catch var="exception">
   <gel:set select="$bad/text()" var="mynode"/>
</core:catch>

<core:if test="${!empty(exception)}">
   <gel:log level="ERROR">Exception: ${exception}</gel:log>
</core:if>
```

  - In addition, several tags support specifying an exception variable specific to that tag

```
<core:invoke method="delete" on="${output}" var="void" exceptionVar="exception"/>
```

Java In GEL

regoUniversity 2018

Let Rego be your guide.

Java In GEL

- Instantiate Java Classes and Call Java Methods
  - **<core:new>** Use this GEL tag to instantiate Java classes

```
<core:new className="java.net.URL" var="myUrl">
   <core:arg type="java.lang.String" value="${myUrlString}"/>
</core:new>
```

  - **<core:invoke>** Use this GEL tag to call a method on an instantiated object

```
<core:invoke method="openConnection" on="${myUrl}" var="myConnection"/>
```

Java In GEL

- Instantiate Java Classes and Call Java Methods
  - **<core:expr>** Use this GEL tag to call a method on an instantiated Java object where you do not require access to the result of the operation.
    - In the REST calls, this tag can be used to set request headers

```
<core:expr value='${connection.setRequestMethod("POST")}'/>
```

  - **<core:invokeStatic>** Use this GEL tag to call a static method of a Java class

```
<core:invokeStatic className="com.niku.union.utility.Base64" method="encode"
var=“encodedString">
   <core:arg type="java.lang.String" value="${valueToEncode}"/>
</core:invokeStatic>
```

Java In GEL – Common Uses

- File Operations
  - Using Java instead of the FileTagLibrary to read a file gives the developer more flexibility. The following snippet shows how to get the values in the first line of the file (the headers).

```
<core:new className="java.io.FileReader" var="thisFile">
   <core:arg type="java.lang.String" value="${tskFile}" />
</core:new>
<core:new className="java.io.BufferedReader" var="csvFile">
   <core:arg type="java.io.FileReader" value="${thisFile}" />
</core:new>
<core:set var="dataRow" value="${csvFile.readLine()}" />
<core:invoke method="split" on="${dataRow}" var="headers">
   <core:arg type="java.lang.String" value=","/>
</core:invoke>
```

Java In GEL – Common Uses

- Moving, copying, and deleting files or directories.

```
<core:new className="java.io.File" var="output" >
   <core:arg type="java.lang.String" value="${fullFilePath}"/>
</core:new>
<core:invoke method="delete" on="${output}" var="void" />
```

REST Web Services

regoUniversity 2018

Let Rego be your guide.

REST Web Services

- A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.
- REST technology is generally preferred to the more robust SOAP technology because REST leverages less bandwidth, making it more suitable for internet usage.
- Results typically JSON (JavaScript Object Notation) response format
- Currently CA PPM REST APIs are not supported for customer or partner use
- The steps described here can be used to call REST APIs of any application from CA PPM

REST Web Services – Authentication

- Below are the most commonly used authentication methods for REST web services
  - Basic Authentication
    - Authenticate using Base64 encoded string of Username and Password

```
<core:invokeStatic className="com.niku.union.utility.Base64" method="encode" var="encodedString">
   <core:arg type="java.lang.String" value="${userName}:${password}"/>
</core:invokeStatic>
<core:set var="basicAuth" value="Basic ${encodedString}"/>
```

  - Token Based Authentication
    - Token from Login and Logout API
    - OAuth 2.0 tokens
    - Application generated tokens

REST Web Services – REST Call

- Maintain Session
  - CA PPM REST API requires to maintain the session. Use the below snippet to maintain the CA PPM REST session

```
<core:new className="java.net.CookieManager" var="cManager"/>
<core:getStatic var="cPollicy" className="java.net.CookiePolicy" field="ACCEPT_ALL"/>
<core:invoke var="setCPolicy" method="setCookiePolicy" on="${cManager}">
   <core:arg value="${cPolicy}"/>
</core:invoke>
<core:invokeStatic className="java.net.CookieHandler" method="setDefault">
   <core:arg type="java.net.CookieHandler" value="${cManager}"/>
</core:invokeStatic>
```

REST Web Services – REST Call

- Below are the steps to call CA PPM REST API
  - Form Input JSON

```
<core:set var="requestJSON" escapeText="false">
  {
    "code": "REST01",
    "isOpenForTimeEntry": "true",
    "description": "Project Created via REST",
    "isActive": "true",
    "name": "REST Project"
  }
</core:set>
```

  - Open Connection to CA PPM REST URI

```
<core:set var="restEndPoint" value="http://ppm.example.com/ppm/rest/v1/projects"/>
<core:new className="java.net.URL" var="restUrl">
  <core:arg type="java.lang.String" value="${restEndPoint}"/>
</core:new>
<core:invoke var="connection" method="openConnection" on="${restUrl}">
```

REST Web Services – REST Call

- Set Request Headers

```
<core:expr value='${connection.setRequestMethod("POST")}'/>
<core:expr value='${connection.setRequestProperty("Authorization", authKey)}'/>
<core:expr value='${connection.setRequestProperty("Content-type", "application/json")}'/>
<core:expr value='${connection.setRequestProperty("Accept", "application/json")}'/>
<core:expr value='${connection.setRequestProperty("Connection", "keep-alive")}'/>
<core:expr value="${connection.setDoOutput(true)}"/>
```

- Write Input JSON to REST API

```
<core:invoke var="outputStream" method="getOutputStream" on="${connection}">
<core:new className="java.io.OutputStreamWriter" var="outputStreamWriter">
   <core:arg type="java.io.OutputStream" value="${outputStream}"/>
</core:new>
<core:expr value="${outputStreamWriter.write(requestJSON)}"/>
<core:expr value="${outputStreamWriter.close()}"/>
```

REST Web Services – REST Call

- Process REST Output

```
<core:invoke var="restOutput" method="getInputStream" on="${connection}">
<core:invoke var="restResponseCode" method="getResponseCode" on="${connection}">
<core:choose>
  <core:when test="${restResponseCode == '200'}">
    <gel:log> Successfully created CA PPM Project </gel:log>
    <!-- Convert REST output to String -->
    <core:invokeStatic className="org.apache.cxf.helpers.IOUtils" method="toString" var="projectOutputString">
      <core:arg type="java.io.InputStream" value="${restOutput}"/>
      <core:arg value="UTF-8"/>
    </core:invokeStatic>
    <core:expr value="${restOutput.close()}"/>
    <core:new className="org.json.JSONObject" var="projectJsonObject">
      <core:arg type="java.lang.String" value='${projectOutputString}'/>
    </core:new>
    <core:set var="prjInternalId" value="${projectJsonObject.get('_internalId')}"/>
    <gel:log> Project ID: ${prjInternalId} </gel:log>
  </core:when>
  <core:otherwise>
    <gel:log> Failed to create CA PPM Project </gel:log>
  </core:otherwise>
</core:choose>
```

Exercise

regoUniversity 2018

Let Rego be your guide.

Exercise

- Create a process to import a Project Request as a new project and email the PM with the new project ID. The ID should get the next auto-numbered project ID.
  - Example for retrieving the next auto-numbered ID:

```
<core:invokeStatic className="com.niku.odf.object.autonumbering.AutoNumberGenerator"
      method="getNextNumber" var="nextId" >
   <core:arg type="java.lang.String" value="project"/>
   <core:arg type="java.lang.String" value="unique_code"/>
   <core:arg type="java.lang.String" value="NIKU.ROOT"/>
   <core:arg type="java.lang.Number" value="${null}"/>
   <core:arg type="com.niku.union.security.SecurityIdentifier"
      value="${context.getSecurityIdentifier()}"/>
   <core:arg type="java.sql.Connection" value="${null}"/>
</core:invokeStatic>
```

Questions?

regoUniversity 2018

Let Rego be your guide.

Thank You For Attending regoUniversity

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Name = **regoUniversity**
- Course Number = **Session Number**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**

Let us know how we can improve!
Don't forget to fill out the class survey.

Phone
888.813.0444

Email
info@regouniversity.com

Website
www.regouniversity.com