regoUniversity

NASHVILLE • 2022

GEL Scripts | Intermediate

Your Guides:

Ben Rimmasch & James Gille

Introductions

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

Agenda

- XML Manipulation
- SOAP Web Services (XOG)
- Email
- Code Reuse
- Best Practices
- Exercise

Part I:
XML Manipulation

XML Manipulation – Create XML Document

- **<gel:parse>** Used to create an XML document in memory. The tag can be used to generate an entire XML document, or specific nodes that can later be attached into an existing XML document.

```
<gel:parse var="userXML">
 <NikuDataBus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../xsd/nikuxog_user.xsd">
  <Header action="write" externalSource="NIKU" objectType="user" version="13.2.0.472"/>
  <Users>
   <User externalId=" " isLDAP="false" uiThemeDefaultPartitionCode="" userLanguage="English" userLocale="en_US"
    userName="${row.userName}" userStatus="${row.userStatus}" userTimezone="America/Los_Angeles" userType="INTERNAL">
    <PersonalInformation emailAddress="${row.email}" firstName="${row.firstName}" lastName="${row.lastName}"/>
    <Resource resourceId="${row.resourceId}"/>
    <Groups>
     <Group id="${row.groupId}"/>
    </Groups>
   </User>
  </Users>
 </NikuDataBus>
</gel:parse>
```

XML Manipulation – Read XML Node

- **<gel:set>** Used to retrieve content of an XML node or document to a variable. This tag can also be used to change content or add elements.
    - Creates a variable called "textContent" with the text inside the User node. If using the XML in the parse above as an example the value of "textContent" would be blank since the User node has no text body.

      <gel:set var="textContent" select="$userXML/NikuDataBus/Users/User/text()" asString="true" />
    - Creates a variable called "userName" with the value inside the "userName" attribute. If using the XML in the parse above the value would be ${row.userName}.

      <gel:set var="userName" select="$userXML/NikuDataBus/Users/User/@userName" asString="true" />
    - Creates a variable called "userGroups" which contains the Groups node, including all sub-nodes

      <gel:set asString="true" select="$userXML/NikuDataBus/Users/User/Groups" var="userGroups"/>

- **<gel:expr>** Used to evaluate an expression as text.
    - Logging User XML to process logs:

      <gel:log level="WARN">**<gel:expr select="$userXML/"/>**</gel:log>

- **<gel:serialize>** Used to save the XML document.

  <gel:serialize var="${userXML}" fileName="/fs0/clarity1/share/userInput.xml" />

XML Manipulation – Update XML Node

- **<gel:set>** This tag can also be used to change content or add elements.
  - Updates the text content of a node to the value specified

    ```
    <gel:set asString="true" select="$userXML/NikuDataBus/Users/User/text()" value="New Text"/>
    ```

  - Updates the value of an attribute (User Name) to the specified value (jsmith)

    ```
    <gel:set asString="true" select="$userXML/NikuDataBus/Users/User/@userName" value="jsmith"/>
    ```

  - Creates an XML node for a group and inserts it into the existing userXML document under the Groups node

    ```
    <gel:parse var="groupXML">
      <Group id="${groupRow.groupCode}"/>
    </gel:parse>
    <gel:set insert="true" value="${groupXML}" select="$userXML/NikuDataBus/Users/User/Groups"/>
    ```

XML Manipulation – Delete Nodes

- Delete Node
  - The following can be used to delete a group node from userXML.

```
<gel:set var="groupToDelete" select="$userXML//Users/User/Groups/Group[@id='pm']" />
<core:set var="parent" value="${groupToDelete.getParentNode()}" />
<core:expr value="${parent.removeChild(groupToDelete)}" />
```

- XSLT Transformations can also be used to delete or heavily modify XML.

Part II:
SOAP Web Services (XOG)

SOAP Web Services (XOG) – Namespaces

- Think of XML namespaces as a way to organize tag names (and attribute names) into groups
- The namespaces below are required to call Clarity SOAP Web Services (XOG)

```
<gel:script
   xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
   xmlns:core="jelly:core"
   xmlns:xog="http://www.niku.com/xog"
   xmlns:soap="jelly:com.niku.union.gel.SOAPTagLibrary"
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

SOAP Web Services (XOG) – Obtain Session ID

- The following snippet shows the recommended way to retrieve a session ID. This approach cannot be used for cross-environment SOAP calls.

```
<!-- Get sessionId by username -->
<gel:parameter var="username" default="admin"/>
<core:new var="secId" className="com.niku.union.security.DefaultSecurityIdentifier" />
<core:invokeStatic var="userSessionCtrl" className="com.niku.union.security.UserSessionControllerFactory" method="getInstance" />
<core:set var="secId" value="${userSessionCtrl.init(username, secId)}"/>
<core:set var="sessionID" value="${secId.getSessionId()}"/>

<core:choose>
   <core:when test="${sessionID == null}">
      <gel:log level="ERROR"> Unable to obtain a Session ID. </gel:log>
   </core:when>
   <core:otherwise>
      <!-- Execute XOG -->
   </core:otherwise>
</core:choose>
```

SOAP Web Services (XOG) – Build XML

- Use the <gel:parse> tag to build a XOG input XML

```
<gel:parse var="userXML">
  <NikuDataBus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="../xsd/nikuxog_user.xsd">
  <Header action="write" externalSource="NIKU" objectType="user" version="13.3.0.286"/>
    <Users>
      <User externalId="" isLDAP="false" uiThemeDefaultPartitionCode="" userLanguage="English"
        userLocale="en_US" userName="${row.userName}" userStatus="${row.userStatus}"
        userTimezone="America/Los_Angeles" userType="INTERNAL">
        <PersonalInformation emailAddress="${row.email}" firstName="${row.firstName}" lastName="${row.lastName}"/>
        <Resource resourceId="${row.resourceId}"/>
        <Groups/>
      </User>
    </Users>
  </NikuDataBus>
</gel:parse>
```

SOAP Web Services (XOG) – Execute XOG

- Use the below snippet to execute a XOG SOAP web service call

```
<!-- Execute XOG -->
<soap:invoke endpoint="internal" var="result">
  <soap:message>
    <soapenv:Envelope>
      <soapenv:Header>
        <xog:Auth>
          <xog:SessionID>${sessionID}</xog:SessionID>
        </xog:Auth>
      </soapenv:Header>
      <soapenv:Body>
        <gel:include select="$userXML"/>
      </soapenv:Body>
    </soapenv:Envelope>
  </soap:message>
</soap:invoke>
```

SOAP Web Services (XOG) – Execute XOG

- Obtaining XOG endpoint
  - endpoint="internal"
    - Quick way to set up SOAP call to the same instance

```
<!-- Execute XOG -->
<soap:invoke endpoint="internal" var="result">
  …
  …
</soap:invoke>
```

  - From properties.xml using JAVA Classes

```
<core:invokeStatic var="config“ className="com.niku.union.config.ConfigurationManager" method="getInstance" />
<core:set var="URL" value="${config.getProperties().getWebServer().getSchedulerUrl()}"/>
<gel:log level="INFO">Environment is: ${URL}</gel:log>
```

SOAP Web Services (XOG) – Execute XOG

- Obtaining XOG endpoint
  - From properties.xml using GEL tags
    - If the schedulerUrl doesn’t work, the entryUrl or sslEntryUrl can be pulled

```
<core:invokeStatic var="nikuHome" className="java.lang.System" method="getenv" >
   <core:arg value="NIKU_HOME"/>
</core:invokeStatic>
<gel:parse var="propertiesXML" file="${nikuHome}/config/properties.xml" />
<gel:set var="host_url“
   select="$propertiesXML/properties/webServer/webServerInstance[@id='app']/@schedulerUrl"
   asString="true" />
<gel:log level="INFO">Host URL: ${hostUrl}</gel:log>
<gel:log>${hostUrl}/niku/xog</gel:log>
```

SOAP Web Services (XOG) – Parse Output

- To ensure success and as part of error handling the following snippet demonstrates how to parse through the XML response returned by the SOAP call to the XOG interface.

```
<!-- Parse Results -->
<gel:set var="XOGState" select="$result//XOGOutput/Status/@state" asString="true" />
<gel:set var="xogStats" select="$result//XOGOutput/Statistics" asString="true"  />
<gel:set var="XOGFailures" select="$result//XOGOutput/Statistics/@failureRecords" asString="true" />

<core:choose>
  <!-- Success -->
  <core:when test="${XOGState == 'SUCCESS' and XOGFailures == 0}">
    <gel:log level="INFO">User XOG Stats: ${xogStats} </gel:log>
  </core:when>
  <!-- Failure -->
  <core:otherwise>
    <gel:log level="INFO">User XOG Stats: ${xogStats} </gel:log>
    <gel:log level="WARN"><gel:expr select="$userXML/"/></gel:log>
    <gel:log level="ERROR"><gel:expr select="$result/"/></gel:log>
  </core:otherwise>
</core:choose>
```

SOAP Web Services (XOG) – Logout

- Use the below snippet to end the XOG session

```
<!-- Logout XOG-->
<soap:invoke var="result" endpoint="internal" >
   <soap:message>
      <soapenv:Envelope>
         <soapenv:Header>
            <xog:Auth>
               <xog:SessionID><![CDATA[${sessionID}]]></xog:SessionID>
            </xog:Auth>
         </soapenv:Header>
         <soapenv:Body>
            <xog:Logout/>
         </soapenv:Body>
      </soapenv:Envelope>
   </soap:message>
</soap:invoke>
```

OR

```
<core:set var="sessionID" value="${userSessionCtrl.delete(sessionID)}" />
```

Part III: Email

Email

- **<gel:email>**
  - Email server information is derived from the properties.xml of the installation
  - Supports HTML
  - Separate multiple email addresses with a ‘;’

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

   <core:invokeStatic var="configManager" className="com.niku.union.config.ConfigurationManager" method="getInstance"/>
   <core:set var="emailFrom" value="${configManager.getProperties().getMailServer().getDefaultSenderAddress()}"/>

   <gel:email from="${emailFrom}" subject="Clarity - Test Email" to="john@gmail.com">
        <![CDATA[
        This is a test email.
        <br/><br/>--------------------------------------------------------------------<br/>
        This is an automated message, please do not reply.
        ]]>
   </gel:email>

</gel:script>
```

Email

- **<email:email> WARNING THAT THIS TAG IS DEPRECATED IN CLARITY 16 DUE TO AUTHENTICATION ISSUES**
  - Mail Server must be specified within the tag
  - Does not support HTML
  - Supports Attachments

```
<gel:script xmlns:core="jelly:core" xmlns:email="jelly:email">

  <core:invokeStatic className="java.lang.System" method="getenv" var="NIKU_HOME">
    <core:arg value="NIKU_HOME"/>
  </core:invokeStatic>
  <core:invokeStatic var="configManager" className="com.niku.union.config.ConfigurationManager" method="getInstance"/>
  <core:set var="emailFrom" value="${configManager.getProperties().getMailServer().getDefaultSenderAddress()}"/>
  <core:set var="mailServer" value="${configManager.getProperties().getMailServer().getHost()}"/>

  <email:email to="john@gmail.com" from="${emailFrom}" subject="app-ca.log file" server="${mailServer}"
        attach="${NIKU_HOME}/logs/app-ca.log">
  App-ca.log File
  </email:email>

</gel:script>
```

Email

- **Java - This is the only way to send emails with attachments and CC and BCC recipients.**
- **Since this requires Java, it will not be covered in this class, but we will gladly cover it in the labs.**

Part IV:
Code Reuse

Email

- **<core:include>**
  - This interesting, little-used tag in-lines a snippet of XML to run in a GEL script.
  - Common functions can be defined in these snippets. For example, Rego maintains an internal library of include files to email, interact with SFTP and REST and more.

```
<gel:script xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary" xmlns:core="jelly:core">
  <gel:parameter var="includesDir" default="c:\files\includes" />
  <core:include file="${includesDir}/GetXOGSession.xml" />
  <gel:log level="DEBUG">Session ID: ${sessionId}</gel:log>
  <core:include file="${includesDir}/DestroyXOGSession.xml" />
</gel:script>
```

Email

- **Sample GetXOGSession.xml**

```
<gel:script
  xmlns:core="jelly:core"
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary">
  <core:new className="com.niku.union.security.DefaultSecurityIdentifier" var="secId" />
  <core:invokeStatic var="userSessionCtrl"
    className="com.niku.union.security.UserSessionControllerFactory"
    method="getInstance" />
  <core:set var="secId" value="${userSessionCtrl.init('admin', secId)}"/>
  <core:set var="sessionId" value="${secId.getSessionId()}"/>
</gel:script>
```

Email

- **Sample DestroyXOGSession.xml**

```
<gel:script
  xmlns:core="jelly:core"
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary">
  <!--Delete returns void so by doing this in set instead of expr sessionID is cleared-->
  <core:set var="sessionId" var="${userSessionCtrl.delete(sessionId)}" />
</gel:script>
```

Part V:
Best Practices

Best Practices

- When possible, pull info from properties.xml or Configuration Manager for mail server, sender address, app URL, etc.
- Avoid sending emails in non-prod environments by either disabling mail server or masking emails. A debug flag, debugs email and logic to prevent or send email regardless of the environment can also be helpful.
- GEL XML tags load the entire XML document into memory. Use caution when dealing with large XML files.
- Include a small and discrete footer in emails to identify the environment and process that is sending it, where applicable. Find some means to identify the environment as Dev, Test, etc., without basing it on the database content, such as the properties.xml or the database schema name.

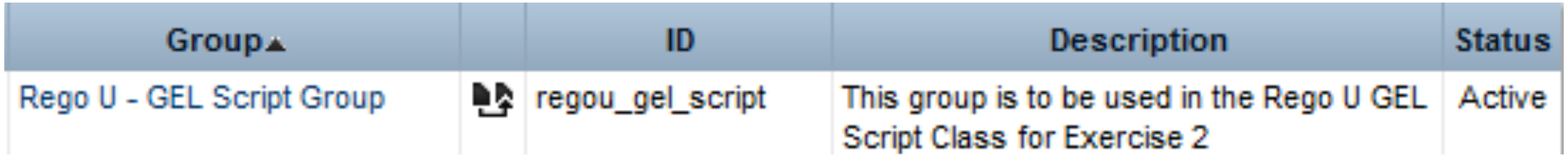
Best Practices (Continued)

- Format and comment code.
- Keep a change log at the beginning of the script.
- Maintain the same code style throughout the script.
- Put unnecessary, additional debugging output behind a parameter.

Part VI: Exercise

Exercise

- Write a script that will email a security group a list of all projects that have inactive project managers.



Group▲		ID	Description	Status
Rego U - GEL Script Group		regou_gel_script	This group is to be used in the Rego U GEL Script Class for Exercise 2	Active

Exercise – SQL Hints

- **SQL to get Resources in Group**:

```
SELECT   r.first_name || ' ' || r.last_name resName,
         r.email resEmail
FROM     cmn_sec_user_groups ug
         JOIN cmn_sec_groups g ON ug.group_id = g.id AND g.group_code = 'regou_gel_script‘
         JOIN srm_resources r ON r.user_id = ug.user_id
```

- **SQL to get Projects**:

```
SELECT   i.id prjId,
         i.code prjCode,
         i.name prjName,
         r.full_name prjMgr
FROM     inv_investments I
         JOIN inv_projects ip ON ip.prid = i.id AND ip.is_template = 0 AND ip.is_program = 0 AND (i.purge_flag = 0 OR i.purge_flag IS NULL)
         JOIN odf_ca_project ocp ON i.id = ocp.id
         JOIN cmn_sec_users u ON i.manager_id = u.id AND u.user_status_id != 200
         JOIN srm_resources r ON u.id = r.user_id
```

Questions?

regoUniversity
NASHVILLE • 2022

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 Provider = **Rego Consulting**
- Class Name = **regoUniversity**
- Course **Description**
- 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@regoconsulting.com

Website
www.regouniversity.com