

★ LAS VEGAS ★

regoUniversity

2025

Sponsored by
ValueOps
by Broadcom

Clarity
by Broadcom

Rally
by Broadcom

ConnectALL
by Broadcom

Insights
by Broadcom

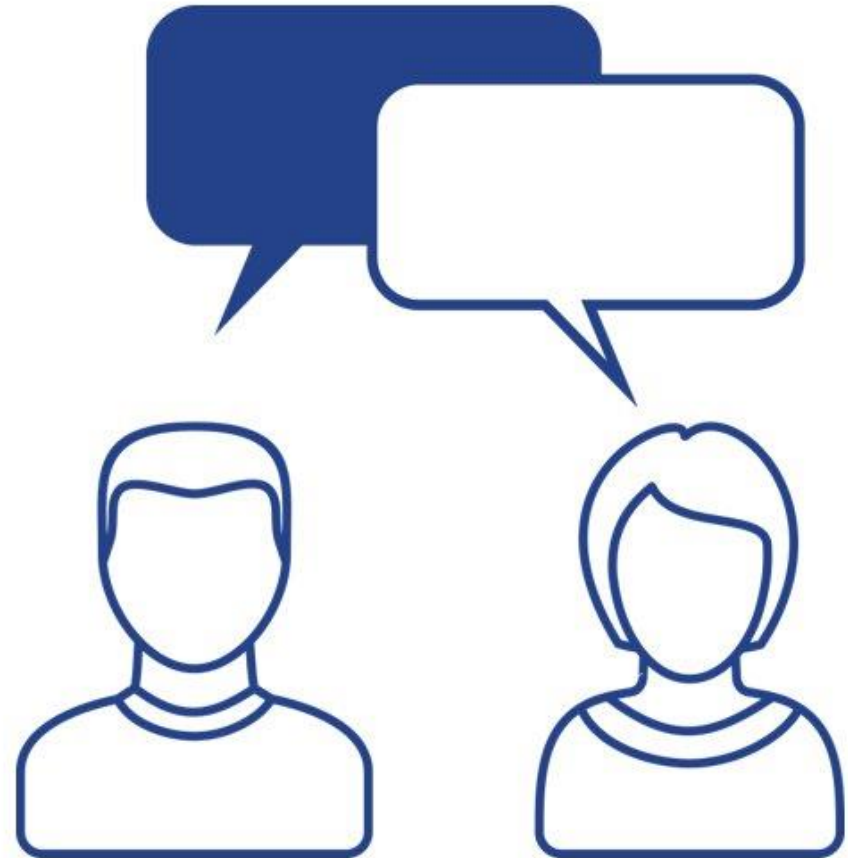
Code Reuse

Your Guides:
James Gille and Shekhar Kanade

Gold Sponsor
aws

Introductions

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



Agenda

- Jelly Includes
- Jam Tags
 - Jam Tag Library
 - Clarity Tag Library
 - File Tag Library
 - Rest Tag Library
- See the Jam Tags in Action



Jelly Includes

- **What:** Snippets of GEL scripts on the file system that can be included anywhere in another GEL script
- **Why:** Code reuse, code maintenance, code control, all the best practices
- **When:** Useful now!
- **Where:** SFTP home directory
- **Who:** Customers with specific repeatable needs not met by Jam
- **How:** Create a single purpose GEL script. Include it into any other script with jelly's core:include tag

Jelly Includes Example

Include File GetFooter.xml Example 1

```
<gel:script
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
  xmlns:core="jelly:core">
  <core:set var="emailFooter">
    <![CDATA[
```

Please raise a Remedy Request to report any issues or questions.

If you have trouble accessing Remedy, you may also send an e-mail to the IT Service Desk.

```
    ]]>
  </core:set>
</gel:script>
```

Jelly Includes Example

Include File GetFooter.xml Example 2

```
<gel:script
  xmlns:gel="jelly.com.niku.union.gel.GELTagLibrary"
  xmlns:core="jelly:core">
  <core:new var="footerFile" className="java.io.File">
    <core:arg type="java.lang.String" value="/fs0/clarity1/share/includes/foot.html" />
  </core:new>
  <core:invokeStatic var="emailFooter" className="org.apache.commons.io.FileUtils" method="readFileToString">
    <core:arg type="java.io.File" value="{footerFile}" />
    <core:arg type="java.lang.String" value="UTF-8" />
  </core:invokeStatic>
</gel:script>
```

Jelly Includes Example

Usage of GetFooter.xml

```
<gel:script  
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"  
  xmlns:core="jelly:core">  
  <!--do things-->  
  <!--send an email-->  
  <core:include file="/fs0/clarity1/share/includes/GetFooter.xml" />  
  <clarity:email from="${from}" subject="${subject}" to="${to}" escapeText="true" >  
    ${messageText}  
    ${emailFooter}  
  </clarity:email>  
</gel:script>
```

Jam Tags



Jam Tags

- **What:** new Jelly (GEL) tags
- **Why:** Code reuse, code maintenance, code control, implementation speed, all the best practices
- **When:** Now
- **Where:** customLib
- **Who:** Everybody
- **How:** Deploy the tags (done for you in SaaS), include the new namespaces and write the script

Jam Tags

- New tags means new namespaces!
 - `xmlns:jam="jelly:com.rego.jam.JamTagLibrary"`
 - `xmlns:clarity="jelly:com.rego.jam.ClarityTagLibrary"`
 - `xmlns:rest="jelly:com.rego.jam.RestTagLibrary"`
 - `xmlns:array="jelly:com.rego.jam.ArrayTagLibrary"`
 - `xmlns:io="jelly:com.rego.jam.FileTagLibrary"`
 - `xmlns:sftp="jelly:com.rego.jam.SFTPTagLibrary"`
 - `xmlns:security="jelly:com.rego.jam.SecurityTagLibrary"`

Jam Tag Library

Jam Tag Library

- jam:throw

```
<jam:throw message="Error creating project." />
```

- jam:concat

```
<core:new var="list" className="java.util.ArrayList" />
```

```
<core:expr value="{list.add('burt@reynolds.com')}" />
```

```
<core:expr value="{list.add('frank@sinatra.org')}" />
```

```
<core:expr value="{list.add('timmy@limmish.com')}" />
```

```
<core:set var="delimiter" trim="false"><![CDATA[;]]></core:set>
```

```
<jam:concat var="listed" collection="{list}" delimiter="{delimiter}" />
```

Jam Tag Library

- Contains better versions of core tags and general-purpose utility tags
- Base64Encode Tag (and Base64Decode Tag)

```
<jam:base64Encode var="basicAuth" value="{username}:{password}" />
```

- Arg Tag (and Array Tag)

<!--The Data Extractor currently has 56 lines of code to call the createTempDirectory on nio. Can we do it in 6? Yes, we can!-->

```
<array:array var="faArr" length="0" className="java.nio.file.attribute.FileAttribute" />
```

```
<core:invokeStatic var="tmpDir" className="java.nio.file.Files" method="createTempDirectory">
```

```
  <jam:arg type="java.nio.file.Path" value="{testDir.toPath()}" />
```

```
  <jam:arg type="java.lang.String" value="deleteme_" />
```

```
  <jam:arg type="[Ljava.nio.file.attribute.FileAttribute;" value="{faArr}" />
```

```
</core:invokeStatic>
```

Jam Tag Library

- Block Tag – Allows the developer to define a block of XML that can be included later in the script.

```
<jam:block var="SET_VARS">
```

```
  <core:set var="WATERMELON">WATERMELON</core:set>
```

```
</jam:block>
```

```
<jam:block xml="${SET_VARS}" />
```

- Use Cases – Resetting or reinitializing state, processing datasets of different types of things, repeating something that is complex. It's core:include but local to the script.

Clarity Tag Library

Clarity Tag Library

- Contains tags that do things specific to Clarity or must be Clarity aware
- Watch this fancy nonsense! (RestAuth and DestroyRestAuth Tags)

```
<gel:script
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
  xmlns:core="jelly:core"
  xmlns:clarity="jelly:com.rego.jam.ClarityTagLibrary"
  xmlns:rest="jelly:com.rego.jam.RestTagLibrary">
  <gel:parameter var="username" default="admin" />
  <gel:parameter var="projectCode" default="PRJ001" />
  <!--Get the REST base URL-->
  <core:invokeStatic var="config" className="com.niku.union.config.ConfigurationManager" method="getInstance"/>
  <core:set var="envURL" value="${config.getRestAPIEntryURL()}" />
  <core:set var="filter" encode="false" escapeText="false">
    <![CDATA[(code = '${projectCode}')]>
  </core:set>
  <rest:encodeUrl var="encodedFilter" value="${filter}" />
  <core:set var="restUrl" encode="false" escapeText="false">
    <![CDATA[${envURL}/rest/v1/projects?limit=1&filter=${encodedFilter}]]>
  </core:set>
  <!--Get a REST session-->
  <clarity:restAuth var="restAuth" userName="${username}" />
  <rest:get var="results" url="${restUrl}" headers="${restAuth}" />
  <gel:log>Result: ${results}</gel:log>
  <clarity:destroyRestAuth map="${restAuth}" />
</gel:script>
```

Clarity Tag Library

- DecryptString Tag and GetConMgrData Tag

```
<!--Get authentication from database-->
```

```
<gel:setDataSource var="clarityDS" dbId="niku"/>
```

```
<clarity:getConMgrData var="clarityAuth" code="{connMgrCode}" dataSource="{clarityDS}" />
```

```
<clarity:decryptString var="decryptedKey" value="{clarityAuth.rego_token_e}" />
```

```
<core:new var="headers" className="java.util.HashMap" />
```

```
<core:expr value="{headers.put('Authorization', 'Bearer ' + decryptedKey)}" />
```

```
<core:expr value="{headers.put('x-api-ppm-client', clarityAuth.rego_client_id)}" />
```

```
<core:expr value="{headers.put('Content-Type', 'application/json')}" />
```

```
<core:expr value="{headers.put('Accept', 'application/json')}" />
```

```
<rest:get var="results" url="{restUrl}" headers="{headers}" />
```

Clarity Tag Library

• Email Tag

```
<gel:script
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
  xmlns:core="jelly:core"
  xmlns:clarity="jelly:com.rego.jam.ClarityTagLibrary">
  <core:set var="subject">Hello, World!</core:set>
  <core:set var="address">ben@fake.com.jelly</core:set>
  <core:set var="attach">C:/files/readme.txt</core:set>
  <core:set var="messageText" encode="false" escapeText="false">
    <![CDATA[<html>
      <head>
        <title>${subject}</title>
      </head>
      <body>
        <h1>Dear, World,</h1>
        <pre>
          This is an HTML email with a CC, BCC, and attachment. Isn't it about time?
        </pre>
      </body>
    </html>]]>
  </core:set>
  <clarity:email from="clarity@mailserver.com" subject="${subject}" to="${address}" cc="${address}" bcc="${address}" attach="${attach}" escapeText="true" >
    ${messageText}
  </clarity:email>
</gel:script>
```

Clarity Tag Library

- RunScript Tag

```
<gel:script
  xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
  xmlns:core="jelly:core"
  xmlns:clarity="jelly:com.rego.jam.ClarityTagLibrary">
  <gel:setDataSource dbId="niku" var="clarityDB"/>
  <core:set var="emailSubject">Hello, World!</core:set>
  <core:set var="emailToRecipients">ben@fake.com.jelly</core:set>
  <core:set var="emailBody" encode="false" escapeText="false">
    <![CDATA[<h1>Dear, World,</h1>
    <pre>
      This is an HTML email with a CC, BCC, and attachment. Isn't it about time?
    </pre>]]>
  </core:set>
  <clarity:runScript code="sendEmail" dataSource="${clarityDB}" />
</gel:script>
```

File Tag Library

File Tag Library

- io:deleteFiles

```
<io:deleteFiles var="numDeleted" files="{folderName}/Users*.xml" />
```

```
<io:deleteFiles var="numDeleted" files="{folderName}/Use*.xml" olderThan="0" />
```

- io:copyFile

```
<io:copyFile source="{folderName}\\Users.txt" destination="{folderName}\\other_file2.txt" allowOverwrite="false"  
deleteOriginal="false" />
```

Rest Tag Library



Rest Tag Library

- Contains tags for each of the REST verbs
- Contain a tag to create a Google GSON object from a String
- Contains utility tags to escape and unescaped JSON Strings and encode and decode URL Strings
- Covered in more detail in the REST API Advanced Topics class

Learn More

- Jam Tag Documentation

Questions?





Master Clarity with Rego University

Earn Certifications in
Administration, Leadership,
and Technical Proficiency

Let Rego be your guide.



Elevate Your Professional Expertise with Rego University Certifications

Rego is excited to continue our **certification programs**, designed to enhance your expertise in Clarity administration, leadership, and technical skills. These certifications provide hands-on experience and knowledge to excel in your career.



Certification Requirements:

- ✓ **Completion:** 12 units per certification track
- ✓ **Eligibility:** Open to all Rego University attendees



Important Reminder:

To have your certification **credits tracked**, ensure you **complete the class surveys in the app** after each session. This step is critical for certification progress.

Surveys

Please take a few moments to fill out the class survey.
Your feedback is extremely important for future events.



Thank You for Attending Rego University

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 = **Rego University**
- 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

Continue to Get Resources and Stay Connected

- 1 Use RegoXchange.com for instructions and how-tos.
- 2 Talk with your account managers and your Rego consultants.
- 3 Connect with each other and Clarity experts at RegoGroups.com.
- 4 Sign up for webinars and join in-person Rego groups near you through at RegoConsulting.com
- 5 Join us for the next [Rego University](#)!

