



Sponsored by
ValueOps®
by Broadcom

Clarity®
by Broadcom

Rally®
by Broadcom

ConnectALL
by Broadcom

Insights
by Broadcom

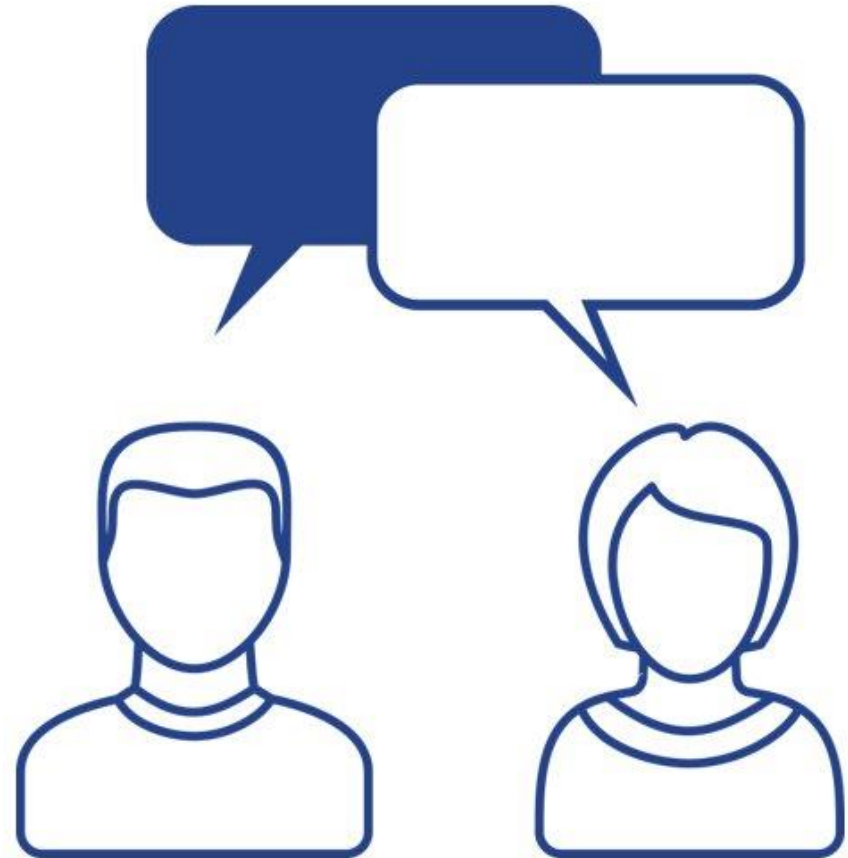
Best Practices in GEL

Your Guides:
James Gille and Rajini Mamidi



Introductions

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



Agenda

- What do you want to discuss?
 - What challenges have you had in GEL?
 - Have you learned any best practices first hand?
- Potential Topics

Potential Topics

- Code Formatting
- Comments
- Unused Code
- Environment Specific Information
- Email
- Exception Handling / Debugging
- REST
- XOG
- SQL

Code Formatting

Code Formatting

- Maintain the same code style throughout the script
- Properly format and indent your code
 - Use an auto-formatter
- Left adjust everything inside CDATA tags
- Options for complex scripts

Comments

Comments

- When to comment?
- How much to comment?
- Keep a change log at the beginning of the script
- Place all code within the `<gel:script>` tags, even comments
- Do not use comments to disable chunks of code (use source control instead)

Unused Code

Unused Code

Remove all unused or unnecessary code.

Example:

```
<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="username" value="{secId.getUsername()}" />
<core:set var="sessionId" value="{secId.getSessionId()}" />
```

Multiple lines are unnecessary

```
<core:invokeStatic var="userSessionCtrl"
className="com.niku.union.security.UserSessionControllerFactory" method="getInstance" />
<core:set var="secId" value="{userSessionCtrl.init(username)}" />
```

<core:set var="sessionId" value="{secId.getSessionId()}" />

Let Rego be your guide.

```
<gel:script xmlns:core="jelly:core" xmlns:gel="jelly:com.niku.union.gel.GELTagLibrary"
xmlns:rem="jelly:com.niku.union.gel.RemedyTagLibrary"
xmlns:soap="jelly:com.niku.union.gel.SOAPTagLibrary"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sql="jelly:sql"
xmlns:util="jelly:util" xmlns:xog="http://www.niku.com/xog" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:file="jelly:com.niku.union.gel.FileTagLibrary"
xmlns:email="jelly:email">
...
</gel:script>
```

Environment Specific Information

Environment Specific Information

- Pull info from Configuration Manager for sender address, app URL, etc.

- Configuration Manager

```
<core:invokeStatic var="config" className="com.niku.union.config.ConfigurationManager"
method="getInstance"/>
```

- App URL

```
<core:set var="appURL" value="${config.getApplicationEntryUrl()}" />
```

- REST URL

```
<core:set var="restUrl" value="${config.getRestAPIEntryURL()}" />
```

- Default Sender Address

```
<core:set var="senderEmailAddress"
value="${config.getProperties().getMailServer().getDefaultSenderAddress()}" />
```

Environment Specific Information

- Custom property file

- Sample File

```
#DEV custom property file
```

```
environment=dev
```

```
jira-connection=jira_dev
```

```
process-error-send-emails=false
```

- Accessing properties in GEL

```
<util:properties var="envProps" uri=file:///fs0/clarity1/share/env.properties />
```

```
<core:set var="sendEmails" value="${envProps.getProperty('process-error-send-emails')}" />
```

Environment Specific Information

- Connection Manager

Connection Manager List

> Filter: None ▾ 					
☐	Name ▲	ID	Connection Type	Clarity Environment	Is Active
☐	AIR	air	Action Item Responder	https://clarity.moventus.com	✓
☐	AIR - DEV	air_dev	Action Item Responder	https://clarity-dev.moventus.com	✓

Email

Deep Links

```
<core:invokeStatic var="config" className="com.niku.union.config.ConfigurationManager" method="getInstance"/>
<core:set var="appURL" value="{config.getApplicationEntryUrl()}" />

<!-- Create the link -->
<core:set var="emailLink" encode="false">
    <![CDATA[{appURL}/pm/#/idea/5009027/details]]>
</core:set>

<core:set var="emailLink" value="{config.getExpandedExternalApplicationUrl(emailLink)}" />
<core:if test="{debug}">
    <gel:log level="INFO">Email Link: {emailLink}</gel:log>
</core:if>
```

Email

- Sender address
- Email templates
 - Include file
 - Script object
- Include a way to identify the environment in the email
- Mask emails in non-prod environments or disable mail server
- Test Email parameter to override recipient emails when testing

Exception Handling / Debugging



Exception Handling / Debugging

- Verbose logging should be controlled with a debug flag
- Debug flags should be set up as gel parameters and defaulted to false

```
<gel:parameter var="debug" default="{false}"/>
```

```
<core:if test="{debug}">
```

```
<gel:log level="INFO">Email Link: {emailLink}</gel:log>
```

```
</core:if>
```

Exception Handling / Debugging

- Monitor the bg-ca.log for silent exceptions

- Calling methods in an expression

```
<core:set var="secId" value="{userSessionCtrl.init(username)}" />
```

- Calling methods using the invoke tag

```
<core:invoke var="secId" on="{userSessionCtrl}" method="init">
```

```
  <core:arg type="java.lang.String" value="{username}" />
```

```
</core:invoke>
```

Exception Handling / Debugging

- Catch exceptions!
 - If you don't catch the exception, it will suppress logs that were supposed to print
- Understand that logging an "ERROR" doesn't stop the script
 - Break Tag
 - Throw Tag - Jam Tags

REST

REST

- Jam Tags (REST API Advanced Topics class)
- Create JSON in a readable way
- If you want the blueprint rules to be triggered, include the following header:
 - x-api-rule-engine: ui

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>
```

Check both the Status and the Failure Record count

```
<!-- 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="ERROR">User XOG Stats: ${xogStats} </gel:log>
```

```
<gel:log level="ERROR"><gel:expr select="$xogState" /> </gel:log>
```

SQL

- ```
<sql:query dataSource="${clarityDS}" escapeText="false" var="result">
 <![CDATA[
SELECT r.full_name resName,
 r.email resEmail
FROM srm_resources r
WHERE r.id = ?
]]>
 <sql:param value="${resourceId}"/>
</sql:query>
```

- SQL Updates

- Avoid Insert statements - these are best suited for REST/XOG
- Direct database updates will not trigger a process to start - REST/XOG updates typically will trigger processes to start
- Avoid updating OOTB tables when possible - using REST/XOG will ensure all Clarity business rules are followed
- Generally safe to update custom attributes with SQL - tables beginning with odf\_ca\_
- Extra caution should also be used within On-Demand environments
- Update the last\_updated\_date and last\_updated\_by columns
- **SQL Updates will not be picked up by the instantaneous DWH sync**



# 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.



# Questions?



# Thank You for Attending Rego University

## Instructions for PMI credits

- Access your account at [pmi.org](https://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](mailto:info@regoconsulting.com)



### Website

[www.regouniversity.com](http://www.regouniversity.com)

# Continue to Get Resources and Stay Connected

- 1 Use [RegoXchange.com](https://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](https://RegoGroups.com).
- 4 Sign up for webinars and join in-person Rego groups near you through at [RegoConsulting.com](https://RegoConsulting.com)
- 5 Join us for the next [Rego University](#)!

