



Sponsored by

ValueOps®
by Broadcom

Clarity®
by Broadcom

Rally®
by Broadcom

ConnectALL
by Broadcom

Insights
by Broadcom

Rest API Advanced Topics

Your Guides:
Jair Flores and Luis Palacios

Gold Sponsor



Introductions

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



Agenda

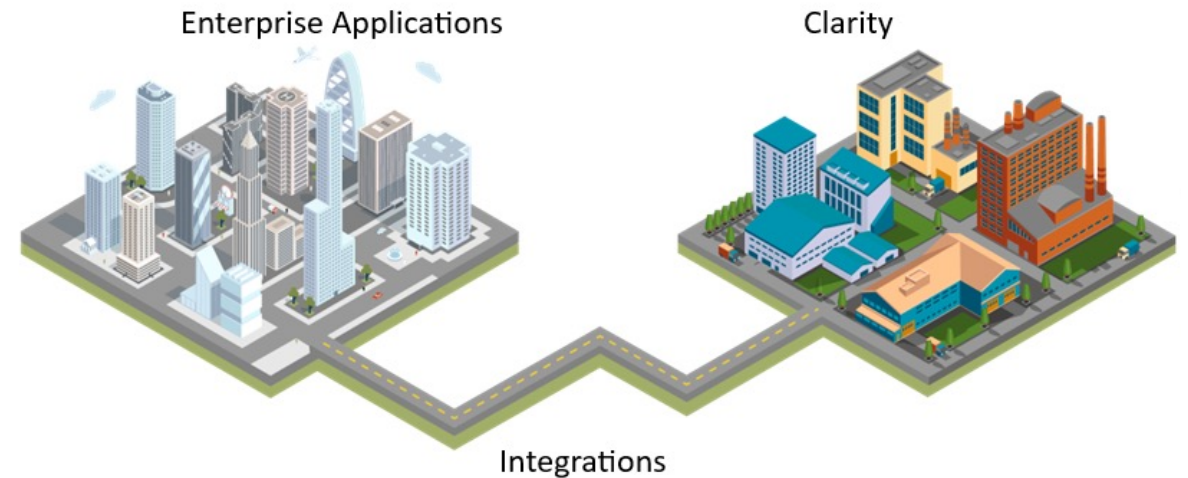
- Why use APIs
- Clarity APIs
 - XOG
 - Rest
 - ODATA
- Rest APIs
- Considerations and Best Practices
 - Considerations
 - Tools
 - Developer Tools
 - Postman
 - VSCode Extension
 - JAM Tags

Why Use APIs



Why Use APIs?

- Increasing number of applications with specialized functionality
- Organizations are allowing teams to use their own applications
- Facilitating business process
- Eliminating administrative activities
- Eliminate double entry
- Data consistency
- Proving a one-stop-shop for automations
- Reduction of license costs
- Data is a valuable asset



Clarity APIs



Clarity APIs

	REST	XOG / SOAP	ODATA
Relevance	Industry Standard	Legacy API	Industry Standard
MUX Support	Fully supported in MUX	Not supported in MUX	
Message Payload	JSON	XML	JSON
Authentication	Basic / API Key	Basic	Basic, API Key
Data Volume	Low	Low	Medium
Performance / Scalability	Average	Slow	Average
Payload Precision	Good	Poor	Good
Record Selection Capability	Good Limited	Poor	Good Limited
Error Messaging	Good	Poor Limited	Good
Ease of Development	Average	Poor	Average
Documentation	Good	Poor	Good

Clarity APIs – What Is XOG

- Original Clarity API
- SOAP Based
- Payloads are XML based
- Used to migrate Studio items (Processes, security groups, objects, etc.)
- MUX not supported (no Blueprints, CIT, etc.)

Clarity APIs – XOG

<https://clarity-dev.yourdomain.moventus.com/niku/xog>

Request Payload

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
  xmlns:xog="http://www.niku.com/xog">  
  <soapenv:Header>  
    <Auth xmlns="http://www.niku.com/xog/Object">  
      <Username>jflores</Username>  
      <Password>P@ssword123!</Password>  
    </Auth>  
  </soapenv:Header>  
  <soapenv:Body>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Response Payload

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <SessionID xmlns="http://www.niku.com/xog/Object">58274630__7418CB43-C265</SessionID>  
  </soapenv:Body>  
</soapenv:Envelope>
```

Clarity APIs – XOG

<https://clarity-dev.yourdomain.moventus.com/niku/xog>

Request Payload

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xog="http://www.niku.com/xog">
  <soapenv:Header>
    <Auth xmlns="http://www.niku.com/xog/Object">
      <Username>user1</Username>
      <Password>P@ssword123!</Password>
    </Auth>
  </soapenv:Header>
  <soapenv:Body>
    <NikuDataBus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="../../xsd/nikuxog_read.xsd">
      <Header version="6.0.13" action="read" objectType="resource" externalSource="NIKU">
        <args name="include_contact" value="false"/>
        <args name="include_management" value="false"/>
        <args name="include_custom" value="false"/>
        <args name="include_financial" value="false"/>
      </Header>
      <Query>
        <Filter name="resourceID" criteria="EQUALS">resource123</Filter>
      </Query>
    </NikuDataBus>
  </soapenv:Body>
</soapenv:Envelope>
```


Clarity APIs – XOG

<https://clarity-dev.yourdomain.moventus.com/niku/xog>

Response Payload

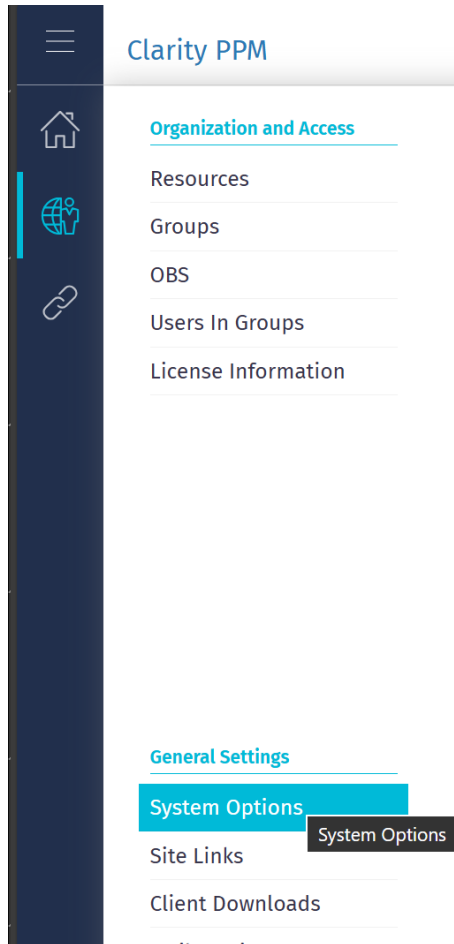
```
<NikuDataBus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../xsd/nikuxog_resource.xsd">
  <Header action="write" externalSource="NIKU" objectType="resource" version="16.2.2.1317"/>
  <Resources>
    <Resource employmentType="EMPLOYEE" externalId=" " includeInDatamart="true" isActive="true"
isExternal="false" managerUserName="admin" resourceId="resource123"
resourceType="LABOR" username="resource123">
      <PersonalInformation displayName="User, Clarity" emailAddress="clarity.user@regoconsulting.com"
firstName="Clarity" lastName="User"/>
      <General addedBy="admin" addedDate="2025-09-14"/>
      <OBSAssocs complete="false"/>
      <SkillAssocs/>
      <Calendar baseCalendar="CLR" resetCalendar="false"/>
    </Resource>
  </Resources>
</NikuDataBus>
```

Clarity APIs – Rest

- Industry standard
- Fully MUX focused.
- New Clarity features only supported through REST.
- Payload is JSON based

`https://[hostname]/niku/rest/describe/index.html`

Clarity APIs – Rest



- Navigate to : ADMIN > System Settings > System Options.
- Scroll down to API section.

API

REST API STATUS

Enabled

API URL

<https://clarity-test.regoconsulting.moventus.com/ppm/rest/v1>

API DOCUMENTATION URL

<https://clarity-test.regoconsulting.moventus.com/niku/rest/describe/index.html>

Clarity APIs – Rest

Clarity PPM REST API

Clarity PPM only supports REST APIs that are listed below.

Auth : Login, Logout, Tokens	Show/Hide	List Operations	Expand Operations
Discovery : Describe Attributes	Show/Hide	List Operations	Expand Operations
Financials : Cost Plans, Fiscal Time Periods	Show/Hide	List Operations	Expand Operations
Other Resources	Show/Hide	List Operations	Expand Operations
Project : Projects, Tasks, Teams, Assignments, Risks, Issues, Changes, Status Reports, Task Dependency	Show/Hide	List Operations	Expand Operations
Roadmap : Roadmap, Roadmap Dependencies, Roadmap Items	Show/Hide	List Operations	Expand Operations
Timesheet : Timesheets, Time Entries, Time Periods, Notes	Show/Hide	List Operations	Expand Operations

[BASE URL: /ppm/rest/v1 , API VERSION: 1.0]

Clarity APIs – Rest

Endpoint: `https://clarity-url.yourdomain.com/ppm/rest/v1/resources/<<INTERNAL_ID>>`

Verb: GET

```
{
  "firstName": "John",
  "lastName": "Doe",
  "uniqueName": "JD1234",
  "dateOfHire": "2022-08-22T00:00:00",
  "dateOfTermination": "2023-01-25T00:00:00",
  "email": "no-reply@moventus.com",
  "fullName": "Doe, John",
  "openForTimeEntry": false
}
```


Considerations and Best Practices



Clarity REST URL

- Programmatically retrieve the base URL with a ConfigurationManager.

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

- Use the Rest API Entry URL

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

- Define URL for REST call

```
<core:set var="restResource" encode="false" escapeText="false">  
<![CDATA[${baseUrl}/rest/v1/projects]]>  
</core:set>
```

External REST Endpoints

- Applications handle authentication in several different ways:
 - Basic Auth: Username and Password
 - Token
 - Oauth
- Credentials should be stored in a way that is secure
 - Rego's Connection Manager encrypts passwords, tokens, and certificates

External REST Endpoints

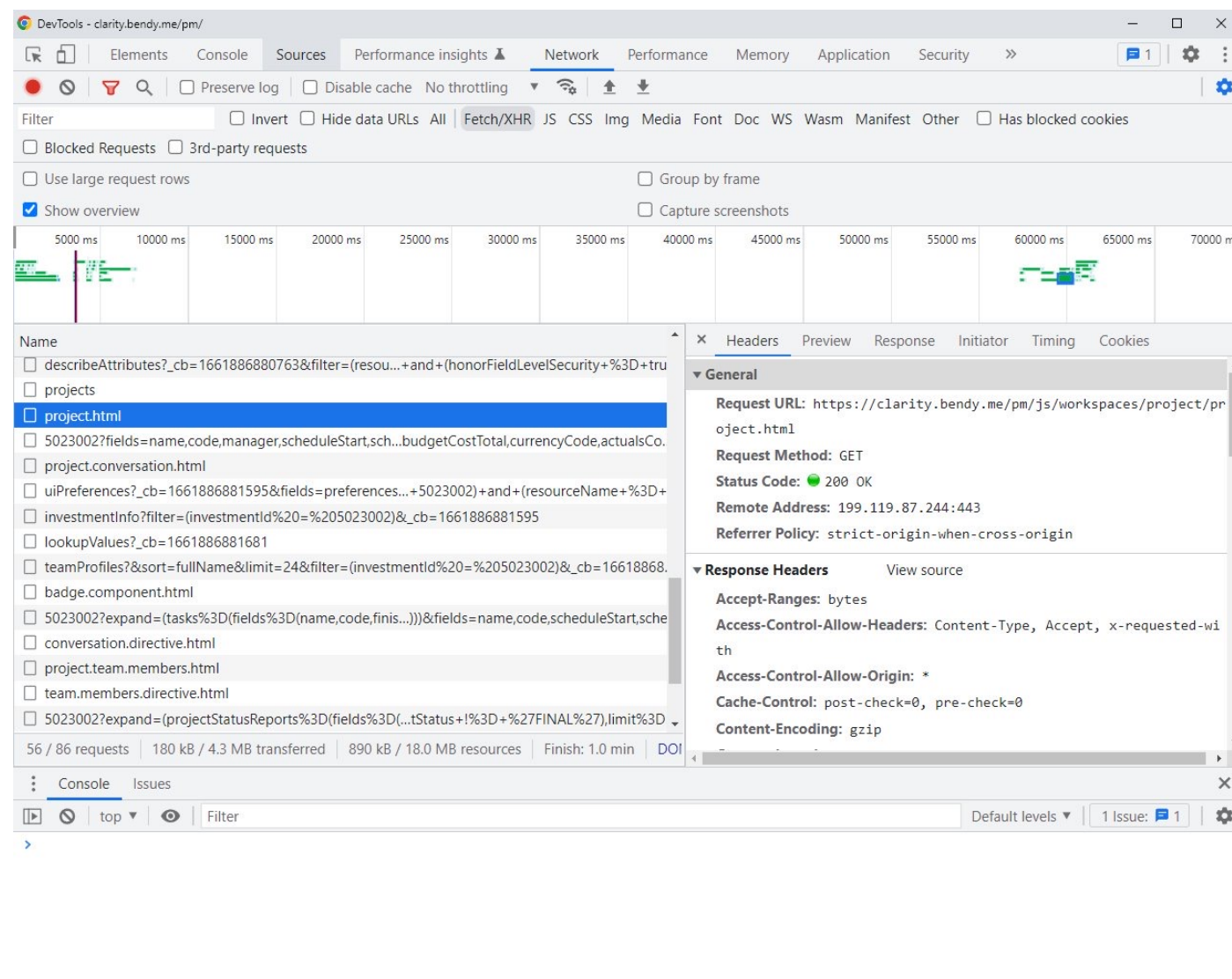
- Three common methods for calling REST endpoints
 - java.net.URL
 - Apache Commons HTTP Client.
 - Jam Tags
- Applications handle the HTTP PATCH verb in different ways. A common implementation for many applications is to use a PUT HTTP verb with the following header.

```
<core:expr value="${restConnection.setRequestProperty('x-http-method-override', 'PATCH')}" />
```


Clarity APIs – What Is ODATA

Chrome Developer Tools

- Allows you to view all REST API calls that the MUX performs when navigating and saving



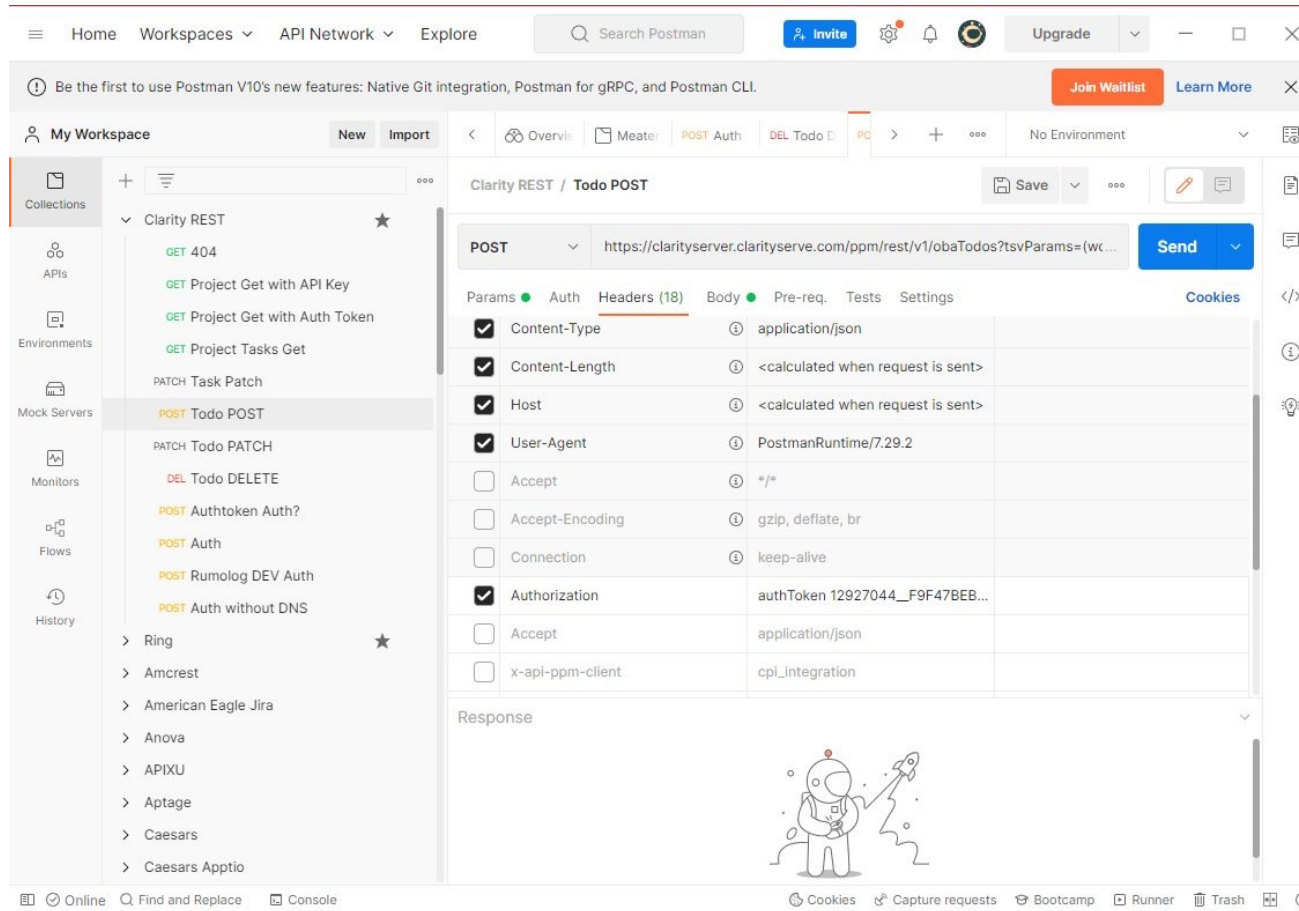
Postman

Postman is a popular platform for API related development. It provides tools for designing, testing, and managing APIs. Here are some key aspects of Postman API:

- Robust API Testing capabilities, detailed functionality to configure requests
- Highly collaborative
- Easy to organize through **Collections** and **Environments**

Postman

- Allows you to view, mock-up, and test API calls



VS Code Extension

- Rego Clarity Developer Extension

<https://marketplace.visualstudio.com/items?itemName=Rego-Consulting.clarity-vscode-extension>

Visual Studio | Marketplace

Visual Studio Code > Snippets > Rego Clarity Developer Extension



Rego Clarity Developer Extension

Rego Consulting |  230 installs | ★★★★★ (0) | Free

Snippets for Gel Scripting and XOG in Broadcom Clarity

[Install](#) [Trouble Installing?](#)

GELSCRIPT SNIPPET EDITOR

-  Search Snippets
-  Table Relationships
-  Sync Snippets from Cloud
- >  Import/Export Snippets
- >  Favorites
- >  Custom Snippets
- >  Quick Start
- ✓  Gels
 - >  Built-in Parameters
 - >  Common
 - >  Conditionals and Loops
 - >  Date
 - >  Dynamic URLs
 - >  Email
 - >  Files
 - >  Jam
 - >  Java Methods & Classes
 - >  Logging and Process Messages
 - >  Notifications
 - >  REST
 - >  SQL
 - >  System Information
 - >  XML
 - >  XOG
 - >  NSQL
 - >  abc XOGS

Jam Tags – Namespaces

- Rego set of custom libraries
- Expand GEL Script capabilities
- Free to use

Jam Tags – Namespaces

- New namespace for Clarity specific tags

```
<gel:script xmlns:clarity="jelly:com.rego.jam.ClarityTagLibrary" >
```

- New namespace for REST specific tags

- ```
<gel:script xmlns:rest="jelly:com.rego.jam.RestTagLibrary" >
```

# Jam Tags – REST Auth

- clarity:restAuth

```
<clarity:restAuth var="restAuth" userName="admin" />
```

OR

```
<clarity:restAuth var="restAuth" userId="${gel_processInitiator}" />
```

- clarity:destroyRestAuth

```
<clarity:destroyRestAuth var="result" map="${restAuth}" />
```

# Jam Tags – REST Actions

- rest:get
- rest:post
- rest:delete
- rest:patch
- rest:put

```
<rest:post var="restResponse" url="${restResource}" headers="${restAuth}"
content="${restBody}" />
```

# Jam Tags – REST Actions

- **var (Optional)**
  - The result of the request.
  - Return Type: `com.rego.net.clients.rest.impl.TagHttpResponse`
- **url (Required)**
  - The URL of the resource where the action will be performed.
  - Type: `java.lang.String`
- **headers (Optional)**
  - A container of headers that will be passed with the request. Two types of containers are supported. Any other type will throw an exception.
    - `java.util.HashMap<String, String>`
    - `java.util.ArrayList<org.apache.http.NameValuePair>`
  - Type: `java.lang.Object`



# Jam Tags – REST Actions

- **params (Optional)**
  - A container of URL query parameters that will be passed with the Request. Two types of containers are supported. Any other type will throw an exception.
    - `java.util.HashMap<String, String>`
    - `java.util.ArrayList<org.apache.http.NameValuePair>`
  - Type: `java.lang.Object`
- **content (Optional)**
  - The request's body. Either contents or json can be used but not both.
  - Type: `java.lang.String`
- **json (Optional)**
  - The request's body. Either json or contents can be used but not both.
  - Type: `com.google.gson.JsonElement`

- Result of all REST actions is a container class (`com.rego.net.clients.rest.impl.TagHttpResponse`) with the following fields:
  - **code**: the HTTP status code (int)
  - **success**: Whether or not the request succeeded. This is calculated by checking if the code is greater than or equal to 200 and less than 300. (boolean)
  - **json**: The response to the request. When the json field is accessed, the response is attempted to be parsed into a Google GSON object which can fail and throw an exception if the JSON was malformed. (`com.google.gson.JsonElement`)
  - **body**: The raw response to the request as a String. (`java.lang.String`)
  - **headers**: A map with the response headers. (`java.util.Map<String, String>`)

# Jam Tags – JSON

- rest:json
  - Creates a com.google.gson.JsonElement from the provided input

```
<rest:json var="restBody" encode="false" escapeText="false">
<![CDATA[
 "name": "Test Project",
 "code": "PRJ-TEST"
]]>
</rest:json>
```

- Allows you to reference values in the JSON later

```
<gel:log>Project Name: ${restBody.name.getAsString()}</gel:log>
```

# Jam Tags – JSON

- rest:escapeJson
  - Escapes a string to be safe in JSON

```
<core:set var="inputString"><![CDATA[They didn't say, "Hamburger!"]]></core:set>
<rest:escapeJson var="result" value="{inputString}"/>
```

result: They didn't say, \"Hamburger!\"

- rest:unescapeJson
  - Unescapes a JSON string

```
<core:set var="inputString"><![CDATA[They didn't say, \"Hamburger!\"]]></core:set>
<rest:unescapeJson var="result" value="{inputString}"/>
```

result: They didn't say, "Hamburger!"



# Jam Tags – URL Encoding

- rest:decodeUrl
  - Decodes URL encoded string

```
<core:set var="fields" encode="false"><![CDATA[clientID%2CownerID]]></core:set>
<rest:decodeUrl var="decodedFields" value="{fields}"/>
```

decodedFields: clientID,ownerID

- rest:encodeUrl
  - Encodes a string for use in a URL

```
<core:set var="fields" encode="false">ClientID,ownerID</core:set>
<rest:encodeUrl var="encodedFields" value="{fields}"/>
```

encodedFields: clientID%2CownerID



# 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.



# Questions?





# Surveys

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



# Sample Table

| Post Time Sheets         | 6:00 PM | 6:00 PM |
|--------------------------|---------|---------|
| Post Transactions        | 6:30 PM | 6:30 PM |
| Post to WIP              | 6:45 PM | 6:45 PM |
| Import Financial Actuals | 7:00 PM | 7:00 PM |

# 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 = **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](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](https://RegoUniversity.com)!

