JavaScript REST Client (v.2)

The **JavaScript REST Client (v.2)** communication point enables users to make REST calls against configured REST clients[rest-clients.html] by executing a user-configured script to handle the connection and response. The connection itself is established by the Rhapsody REST client framework, and the custom JavaScript code performs the HTTP calls.

The JavaScript REST Client communication point can have a maximum of five connections to the same server and a maximum of 50 connections in total.

> **Note**
>
> As of Rhapsody 7.4, the JavaScript REST Client communication point (now known as the JavaScript REST Client (v.1) communication point) has been deprecated and replaced by the JavaScript REST Client (v.2) communication point. RLCs containing the JavaScript REST Client (v.1) communication point will continue to load into Rhapsody and the communication point will still operate and be configurable. However, you will no longer be able to add new JavaScript REST Client (v.1) communication points. Therefore, it is highly recommended you update your RLC to work with the JavaScript REST Client (v.2) communication point as soon as is practicable.
>
> The main difference between the two versions of the communication point is that version 2:
>
> - Employs the GraalJS JavaScript engine. Refer to JavaScript Scripting Engine[javascript-scripting-engine.html] for details.
> - Offers OAuth2 to connect to REST clients. Refer to OAuth2 for REST Clients[oauth2-for-rest-clients.html] for a summary.

Supported Operational Modes: *Output*, *Out->In*.

Configuration Properties

Output Mode

Property	Description
REST Client	The user-selected REST client that is bound to the JavaScript code.
OAuth2 REST Client	A REST client for connecting to the OAuth 2.0 Authorization Server. This property is only required if the Authorization Server has a different hostname from the primary REST client. **Note** Ensure you use HTTPS to connect when using OAuth 2.0.
Request JavaScript	The JavaScript code that is executed when the communication point is run. Refer to JavaScript REST Objects[javascript-rest-objects.html] for details on the objects associated with the scripts used by the JavaScript REST Client communication point.
Read Timeout (ms)	How long to wait (in milliseconds) when trying to read a response from the server. This property applies to each request made in the script. A value of zero or less is treated as an infinite timeout.

Out->In Mode

The configuration properties for the JavaScript Rest Client communication point in *Out->In* mode is the same as *Output* mode. Refer to *Out->In* and *In->Out* Properties[general.html] for general details on the *Out->In* mode for communication points.

When in this mode, an additional output object is available for use within JavaScript. Through the output object, you can set the message body and properties based on the response received.

Converting JavaScript REST Client (v.1) Communication Points

Rhapsody provides an option to automatically convert JavaScript REST Client (v.1) communication points to JavaScript REST Client (v.2) communication points.

> **Note**
>
> All communication point functionality and configuration are retained unaltered in the converted communication point. This

> includes JavaScript code. It is therefore important you verify the communication points operate as expected after conversion.

To convert one or more JavaScript REST Client (v.1) communication points, either:

- In Route View or from the Workspace, select a JavaScript REST Client (v.1) filter to convert and select **Convert to JavaScript REST Client (v.2) Communication Point** from the right-click menu, or
- In the Object Browser, set the **Components** field to include communication points, select the JavaScript REST Client (v.1) communication points to bulk convert and select **Convert to JavaScript REST Client (v.2) Communication Points** from the right-click menu.

In this section

Last modified: 2024-11-12