



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Composer Help

HTTP Rest Block

Contents

- 1 HTTP Rest Block
 - 1.1 Name Property
 - 1.2 Block Notes Property
 - 1.3 Exceptions Property
 - 1.4 Guarantee HTTP Execution Property
 - 1.5 Retries Property
 - 1.6 Retry Interval Property
 - 1.7 User Name Property
 - 1.8 Password Property
 - 1.9 Uri Property
 - 1.10 Condition Property
 - 1.11 Logging Details Property
 - 1.12 Log Level Property
 - 1.13 ORS Extensions Property
 - 1.14 Input Parameters Property
 - 1.15 JSON Payload Property
 - 1.16 Use JSON Payload Property
 - 1.17 Encoding Type Property
 - 1.18 Request Method Property
 - 1.19 Custom HTTP Headers Property
 - 1.20 Max Age Property
 - 1.21 Max Stale Property
 - 1.22 Timeout Property
 - 1.23 Fetch Data Type Property
 - 1.24 Fetch Audio Property
 - 1.25 Fetch Audio Delay Property
 - 1.26 Fetch Audio Minimum Property
 - 1.27 Fetch Hint Property
 - 1.28 Fetch Timeout Property
 - 1.29 Output Result
 - 1.30 Response Headers

-
- [1.31 Response Status](#)
 - [1.32 Status Code](#)
 - [1.33 Enable Status](#)

HTTP Rest Block

A new block, HTTP Rest, is introduced in 8.1.450.08. The HTTP Rest block can be used in both routing and voice applications.

In routing applications, the HTTP Rest block supports the native HTTP Fetch functionality in Orchestration Server and uses the ORS SCXML <session:fetch> element to fetch arbitrary XML and JSON data from a server. The data fetched by the <session:fetch> element is bound to an ECMAScript object through a named variable.

In voice applications, the HTTP Rest block supports the native HTTP Fetch functionality in GVP and uses the GVP VoiceXML <data> element to fetch arbitrary XML and JSON data from a document server. The data fetched by the <data> element is bound to an ECMAScript object through a named variable.

The HTTP Rest block has the following properties:

Name Property

Find this property's details under [Common Properties for Workflow Blocks](#) or [Common Properties for Callflow Blocks](#).

Block Notes Property

Find this property's details under [Common Properties for Workflow Blocks](#) or [Common Properties for Callflow Blocks](#).

Exceptions Property

Find this property's details under [Common Properties for Workflow Blocks](#) or [Common Properties for Callflow Blocks](#). You can also define [custom events](#).

The Web Service block Exceptions dialog box has the following pre-set exceptions:

- Callflows: `error.badfetch`, `error.semantic`, and `error`.
- Workflows: `error.session.fetch`, `error.script`, and `error`.

Guarantee HTTP Execution Property

Use this property (applicable for workflows only) to indicate whether the platform can guarantee the execution of the <fetch> action.

1. Select the **Guarantee HTTP Execution** row in the block's property table.
2. In the **Value** field, select **true** or **false**.

Retries Property

Specify an integer value in this property (applicable for workflows only) to indicate the number of times the platform will try to successfully deliver the associated HTTP request to the defined destination.

Retry Interval Property

Specify an integer value in this property (applicable for workflows only) to indicate the number of seconds to wait between each retry.

User Name Property

Specify the user name for HTTP basic authentication through a literal or a variable.

Important

When trying to access an anonymous HTTP URL, you need not provide any user credentials.

Password Property

Specify the password for the user name specified in the User Name property for HTTP basic authentication, through a literal or a variable.

Uri Property

Use this property to specify the URL destination or location of the XML and JSON data to retrieve.

1. Select the **Uri** row in the block's property table.
2. In the **Value** field, click the down arrow and select the variable that contains the URL or specify the URL (literal).

Condition Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

Logging Details Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

Log Level Property

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

ORS Extensions Property

Use this property to add custom attributes into any/all states and sub-states for any block they are configured in. For more information on this property, see [ORS Extensions](#).

Input Parameters Property

Use the **Input Parameters** property to specify a list of required Name/Value pairs to pass as parameters to the http:// page. To specify input parameters:

1. Click the Input Parameters row in the block's property table.
2. Click the ... button to open the Input Parameters dialog box.

Add Button

Use the **Add** button to enter parameter details.

1. Click **Add** to add a parameter.
2. In the **Parameter Name** field, leave the default name as is or change as required.

3. The **Parameter Type** field displays **input** to indicate that the parameter being defined is an input parameter
4. In the **Expression** field, click the ... button to use the **Expression Builder**.
5. In the **Description** field, type a description for the parameter.
6. Click **Add** again to enter another parameter, or click **OK** to finish.

Important

To reorder the priority of the input parameters, use the **Up** and **Down** buttons to rearrange the parameters as desired in the dialog box.

Remove Button

To remove a parameter:

1. Select an entry from the list.
2. Click **Remove**.

JSON Payload Property

Select the appropriate variable from the **Value** drop-down to specify the JSON object that will be sent for requesting data.

Use JSON Payload Property

Use this property to specify if parameters must be sent individually or collectively as an object.

1. Select the **Use JSON Payload** row in the block's property table.
2. In the **Value** field, select **true** or **false**.

Encoding Type Property

The Encoding Type property indicates the media encoding type of the submitted document.

GVP supports the following encoding types:

- application/x-www-form-urlencoded
- application/json
- multipart/form-data

ORS supports the following encoding types:

- application/x-www-form-urlencoded
- application/json

To select a value for the **Encoding Type** property:

1. Select the **Encoding Type** row in the block's property table.
2. In the **Value** field, select the required value or specify a literal value.

Request Method Property

This property indicates the method for invoking the web request:

- **get** - Invoked using HTTP Get.
- **post** - Invoked using HTTP Post.
- **put** - Invoked using HTTP Put.
- **delete** - Invoked using HTTP Delete.

To specify a value for the Request Method property:

1. Select the **Request Method** row in the block's property table.
2. In the **Value** field, specify or select **get**, **post**, **put**, or **delete** from the drop-down list.

Important

When the HTTP Rest block is used in a callflow, the value specified in this property is not case sensitive.

Custom HTTP Headers Property

Use this property (applicable for workflows only) to add Custom HTTP headers to be sent along with the HTTP request during runtime execution of the Server Side block.

1. Click the **Custom HTTP Headers** row in the block's property table.
2. Click the **...** button to open the Custom HTTP Headers dialog box.
3. Click **Add** to add a header entry.
4. Select the required variable that contains the header from the **Value** drop-down or specify a literal value.

Max Age Property

Use this property to specify the maximum duration (as an integer) for which the cached copy of the retrieved data is considered, in seconds.

Max Stale Property

Use this property to specify the maximum duration (as an integer) for which the cached copy of the retrieved data is considered after expiration, in seconds.

Timeout Property

Use this property (applicable for workflows only) to specify the duration in seconds (as an integer) before throwing a timeout error and sending the `error.session.fetch` event when the content is not returned.

Fetch Data Type Property

Use this property (applicable for workflows only) to select the type of data that will be retrieved by the `<fetch>` action. The following data types are supported:

- **JSON**
- **application/xml**

1. Click the **Fetch Data Type** row in the block's property table.
2. Select the required fetch data type from the **Value** drop-down.

Fetch Audio Property

Use this property (applicable for callflows only) to specify the URI of the audio to play while waiting for the data to be retrieved.

Fetch Audio Delay Property

Use this property (applicable for callflows only) to specify the time in seconds (as an integer) to wait before playing the retrieved audio.

Fetch Audio Minimum Property

Use this property (applicable for callflows only) to specify the minimum duration (as an integer) for which the retrieved audio must be played after being triggered.

Fetch Hint Property

Use this property (applicable for callflows only) to select when the audio content must be retrieved from the server.

- `safe` - file is downloaded only when needed.

Fetch Timeout Property

Use this property (applicable for callflows only) to specify the duration in seconds (as an integer) before throwing a timeout error and sending the `error.badfetch` event when content is not returned.

Output Result

Select one of the available variables from the drop-down to indicate the variable that will contain the result of the `<fetch>` request.

Response Headers

Use this property (applicable for workflows only) to indicate the variable that will contain the response headers for the HTTP request.

Response Status

Select one of the available variables from the drop-down to indicate the variable that will contain the response output status of the HTTP request.

Status Code

Select one of the available variables from the drop-down to indicate the variable that will contain the

output status code of the HTTP request.

Important

If any errors are encountered, the status code is updated only if the block handles exceptions.

Enable Status

Find this property's details under [Common Properties for Callflow Blocks](#) or [Common Properties for Workflow Blocks](#).

[+] HTTP Rest Workflow Block Code Sample

```
<state id="HTTPRest1">
  <onentry>
    <log expr="_sessionid + ': Inside HTTPRest Block: HTTPRest1'" />
    <session:fetch requestid="App_HTTPRest1['requestid']"
      srcexpr="user_uri" method="get" enctype="application/x-www-form-
urlencoded" username="username" password="password" maxage="40" maxstale="80"
type="application/json" gdelivery="false" gd_retries="1" gd_retry_interval="0" timeout="60">
    </session:fetch>
  </onentry>
  <transition event="session.fetch.done"
cond="_event.data.requestid==App_HTTPRest1['requestid']" target="$$_MY_PREFIX_$.Exit1">
    <log expr="Session FETCH DONE" />
    <log expr="'Composer Application:subroutine Block:
HTTPRest1'" />
    <log expr="'Data Fetched:' + _event.data.content" />
    <script>storeEvent("HTTPRest1", _event);</script>
    <script>App_HTTPRest1['data'] = eval('(' +
_event.data.content + ')');</script>
    <assign location="headers" expr="App_HTTPRest1['data']" />
    <log expr="'Data Assigned: ' +
headers.toSource()"/>
  </transition>
</state>
```

[+] HTTP Rest Callflow Block Code Sample

```
<form id="HTTPRest1">
  <block>
    <assign name="AppState.var0" expr="''"/>
    <assign name="AppState.var1" expr="false"/>
    <data name="HTTPRest1Response" srcexpr="AppState.var3" enctype="application/json"
method="get"
      gvp:usernameexpr="'null'" gvp:passwordexpr="'null'"
fetchaudio="http://10.31.12.109:4040/Final_Test/Resources/Prompts/A.wav"
gvp:fetchaudiodelay="1s" gvp:fetchaudiominimum="10s" fetchhint="safe" fetchtimeout="120s"
maxage="50" maxstale="40" gvp:contentexpr="AppState.var4"/>
  </block>
</form>
```

```
<if cond="typeof HTTPRest1Response != 'undefined'">
<assign name="AppState.var0" expr="HTTPRest1Response"/>
    <assign name="AppState.var1" expr="true"/>
    <assign name="AppState.var2" expr="200"/>
    <goto next="#Log1" />
<else/>
    <assign name="AppState.var2" expr="204"/>
</if>
</block>
<catch event="error">
    <assign name="AppState.var2" expr="extractVxmlHttpErrorCode(_event)"/>
    <goto next="#Copy_1_Log1" />
</catch>
</form>
```