

WebData Tools Reference

Test REST, GraphQL, and MCP HTTP endpoints from the Designer, then export a working request into the solution.

[Reference](#) [Solution](#) [Designer](#) [Data Explorer](#) [MQTT](#) | [OPC](#) | [PLC](#) | [SQL](#) | [WebData](#)

The WebData Tools tab exposes a request builder and a response viewer for three API styles, REST, GraphQL, and MCP HTTP. Use it to validate endpoints, inspect payloads, and export a working request into the solution as a **ReportsWebData** entry.

[Connection Target](#)
[Request Builder](#)
[Actions](#)
[MCP Session Handshake](#)
[Response Viewer](#)
[Export to Solution](#)
[MCP Tab Actions](#)

Connection Target

Choose where the request points at the top of the tab.

Mode	Description
Local	Load a request from an existing ReportsWebData entry in the solution. Use this mode to replay a configured request.
Remote	Enter a URL and request parameters directly. The request is not tied to a solution object.

In Local mode, credentials and headers come from the selected **ReportsWebData** row, not from the Remote-mode Authorization and Headers fields. This matters for MCP execution: the session handshake uses the row's Authorization and Headers, so a token typed into the Remote-mode UI while a Local row is selected does not apply to the executed request.

Request Builder

API

Select the API style in the **API** drop-down.

Value	Body editor
REST	Raw request body. The Method drop-down controls the HTTP verb.
GraphQL	GraphQL query text. Variables go in the body as JSON. The Operation Name field selects a named operation when the query document declares more than one.
McpHttp	MCP tool arguments as JSON. The tool name is set through Select Tool .

Method

For REST, choose a verb, GET, POST, PUT, PATCH, or DELETE. GraphQL and McpHttp ignore the Method field.

URL

Enter the full endpoint URL. Tag substitution is supported. Write `{{Tag.Name}}` to inject a tag value. In the Designer tester, tokens resolve to the tag's **InitialValue** at design time. At runtime, the same tokens resolve against the live RTDB value.

Authorization

The **Authorization** field is multi-line. The first line selects the scheme, following lines supply the credentials.

Scheme	Lines
NoAuth	Single line, <code>NoAuth</code> .
BearerToken	Line 1, <code>BearerToken</code> . Line 2, the token string.
BasicAuth	Line 1, <code>BasicAuth</code> . Line 2, the username. Line 3, the password. The client sends a Base64-encoded header.

Headers

Enter custom headers as `key=value;key=value` on a single line. Use a semicolon between pairs inside the field.

Body

The body editor label reflects the selected API, **REST body**, **GraphQL query**, or **MCP arguments**. Tag substitution is supported in all three editors.

Actions

Button	Description
Execute	Send the request and populate the Response viewer.
Stop	Cancel an in-progress request.
Refresh Tools	McpHttp mode. Initializes an MCP session at the current URL and lists the available tools.
Select Tool	McpHttp mode. Pick a tool from the discovered list. The input schema is shown below the selector.
Export	Open the export dialog. Creates a new ReportsWebData row with the current API, URL, method, authorization, headers, and body.

MCP Session Handshake

MCP servers that follow the streamable-HTTP spec (2025-03-26) require a session before any `tools/list` or `tools/call`. The tester performs the full handshake automatically:

1. POST `initialize` to the endpoint.
2. Capture `Mcp-Session-Id` from the response headers.
3. POST `notifications/initialized`.

Only then does the tester send the actual request. All three execution paths, **Refresh Tools**, Remote **Execute**, and Local **Execute**, run the handshake. Strict MCP servers would otherwise reject a cold `tools/call`.

Session Sharing

The session ID and the initialized flag are keyed on the endpoint URL and shared across the three execution paths. A handshake performed by **Refresh Tools** benefits a subsequent **Execute** against the same URL. Changing the URL triggers a fresh handshake against the new endpoint.

Idle and Re-init

The tester mirrors runtime `McpHttpSession` behavior. An idle session is closed after five minutes. A server returning HTTP 404 on a tool call triggers automatic session re-init and one retry. Any other HTTP error surfaces in the Response viewer as received.

Local-mode Credentials

When Local **Execute** runs against a **ReportsWebData** row configured as `McpHttp`, the handshake and the tool call both use the row's Authorization and Headers. The Remote-mode Authorization and Headers fields on the tab are ignored for that execution. Change credentials by editing the row, not the tab.

Response Viewer

Responses appear in three tabs.

- **Pretty JSON**. Parsed and indented when the response is JSON.
 - **Raw**. The response body as received. Truncated at 8000 characters for display. The export dialog stores the full body.
 - **Headers**. HTTP status line and response headers.
-

Export to Solution

Click **Export** to open the **Export WebData** dialog. Choose a name, confirm, and a new entry is added under **Reports WebData**. The solution-side runtime, [Reports WebData Reference](#), handles execution at runtime.

The exported entry carries the API, URL, method, authorization, headers, and body from the tab. Edit the entry in the WebData grid to refine it afterwards.

Encoding Mapping

The export derives the **Encoding** column on the new row from the response `Content-Type` header, not from a hardcoded value.

Response Content-Type	Exported Encoding
application/json, +json variants	JSON
application/xml, text/xml	XML
text/html	HTML
text/plain, other text/*	Text
Empty or missing	JSON (default)

GraphQL Operation Name

When the tab is in GraphQL mode, the **Operation Name** field persists to the row's `OperationName` column, which is new in 10.1.5. It is not packed into `ToolName`. See the **GraphQL OperationName** section of [Reports WebData - GraphQL and MCP Client](#) for runtime semantics.

MCP Tab Actions

The WebData Tools tab exposes **designer_action** controls for MCP clients. Each action drives the same tab state as the UI controls.

Action	Description
<code>set_mode</code>	Switch between Local and Remote.
<code>set_api</code>	Set REST, GraphQL, or McpHttp. Invalid values return an error with the accepted list.
<code>set_url</code>	Set the URL field.
<code>set_method</code>	Set the REST verb.
<code>set_authorization</code>	Set the multi-line Authorization field. The action hint shows the per-line format.
<code>set_headers</code>	Set headers as <code>key=value;key=value</code> . The action hint shows the expected format.
<code>set_body</code>	Set the body editor. The response echoes the target editor name, REST body, GraphQL query, or MCP arguments.
<code>set_operation_name</code>	GraphQL mode. Set the Operation Name field. Persists to the row's <code>OperationName</code> column on export.
<code>execute</code>	Send the request.
<code>stop</code>	Cancel a pending request.
<code>get_response</code>	Return status, headers, and body, with up to 8000 characters of response text.
<code>get_state</code>	Return the full tab state, including authorization, headers, body editor, and MCP input schema.
<code>list_webdata</code>	List the ReportsWebData entries in the solution.
<code>select_webdata</code>	Load a saved entry into the tab. Invalid names return a warning with the available list.
<code>refresh_tools</code>	McpHttp mode. Initialize the session and list tools.
<code>select_tool</code>	McpHttp mode. Select a tool by name. Invalid names return a warning with the available list.
<code>export</code>	Run the export. Returns a success or failure message, including a duplicate-name error.

In this section...
