

JSON Data Bank

This topic explains how to configure and apply the JSON Data Bank tool in SOAtest and Virtualize. This tool extracts JSON values (e.g., from a request or response message) so that they can be used in another place. Data can also be sent to a Writable Data Source and accessed in the Extension Tool or it can be sent to variables for easy reuse across the test, Responder, or Action suite.

Sections include:

- [Understanding JSON Data Bank](#)
- [Configuring JSON Data Bank Using the Data Source Wizard](#)
- [Configuring the JSON Data Bank Manually](#)
- [Tool Options](#)
- [Viewing the Data Bank Variables Used During Test Execution](#)
- [Handling Empty/Missing Elements to Maintain the Integrity of the JSON Response](#)



Migration Note

The JSON Data Bank tool was re-implemented in version 9.7. The previous implementations are deprecated: any existing tools will continue to work, but all new JSON Data Banks you add will use the new implementation.

This topic focuses on the current JSON Data Bank implementation. For details on the deprecated JSON Data Bank, see [JSON Data Bank Deprecated](#).

Understanding JSON Data Bank

This tool enables you to extract certain JSON values (e.g., from a request or response message) so that they can be used in another place. The JSON Data Bank tool can be chained to any other tool that outputs JSON. It can extract any information from the JSON and make that information available for later use.

For example, you can configure a test suite that tests a bank's service transactions. Test 1 of that test suite can log on to the service using a User ID, then the JSON response would return a session ID back to Test 1. Test 2 of that test suite can be configured to use the session ID from Test 1 to perform transactions. You can configure any of the tests in a test suite to use JSON response parameters as JSON request parameters.

Users typically configure a JSON Data Bank in SOAtest by accessing the "Use Data Source Wizard" while parameterizing a value in a tool such as the REST Client or Messaging Client tool. This provides a quick, intuitive, and largely automated way to extract data from one tool and use it another. You simply go to the tool where you want to insert extracted data, then use a wizard to specify what data (e.g., from what tool) you want to extract. This is the usage model demonstrated in the [Storing Results to Be Used in Subsequent Tests](#) tutorial. This same method can be used to extract data that is used to set a variable. Alternatively, you can manually configure a JSON Data Bank tool to extract data from one tool, then manually configure other tools to use the extracted values.

You can use a JSON Data Bank in Virtualize, for example, to extract a value from an incoming request and use it to populate an element of the response to be sent when that request is received. You can configure JSON Data Banks automatically when using wizards to create virtual assets from traffic. In addition, you can use the "Use Data Source Wizard" to extract a value and use it as a parameterized value in a response. For Message Responders, you can extract values from the incoming request (body or header); for other tools (e.g., tools used in action suites), you can extract values from another tool in the suite. Another option is to manually add a JSON Data Bank tool (as an output to an existing tool) that extracts the desired data, then configure other tools to use the extracted data.

Video Tutorial

In this video, you'll learn how to extract values from JSON responses and reuse them in other tests.

[Configuring JSON Data Bank Using the Data Source Wizard](#)

Configuring the Extraction

To use the "Use Data Source Wizard" wizard to configure a JSON Data Bank:

- (Not applicable for use with Message Responders) Ensure that you have a test suite or action set with at least two tools.
- In the configuration panel for the tool that you want use the extracted value, select one of the available Form views.
- Go to the message element that you want to use the extracted value, then select **Parameterized** and **Use Data Source Wizard** from the available drop-down menus.

In the wizard that opens:

- For test and action suite tools, select the tool you want to extract a value from. The drop-down menu at the top of the panel will contain all tools in the test or responder suite that occur before the current tool you are configuring. For example, if you are configuring tool 4, tools 1, 2, and 3 will display in this menu along with any data sources that may be available.
- For Message Responders, select the incoming request message that you want to extract a value from, then specify whether you want to extract values from the message body or message header. Under **Expected Message**, indicate what element(s) you want to extract, then click **Add**. The right panel lists the values you have configured for extraction and shows the name of the data source column where each value will be stored (if you keep the default setting).

For test and action suite tools, select the tool you want to extract a value from. The drop-down menu at the top of the panel will contain all tools in the test or responder suite that occur before the current tool you are configuring. For example, if you are configuring tool 4, tools 1, 2, and 3 will display in this menu along with any data sources that may be available.

For Message Responders, select the incoming request message that you want to extract a value from, then specify whether you want to extract values from the message body or message header. Under **Expected Message**, indicate what element(s) you want to extract, then click **Add**. The right panel lists the values you have configured for extraction and shows the name of the data source column where each value will be stored (if you keep the default setting).

jpg" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/jpeg" data-linked-resource-container-id="77035236" data-linked-resource-container-version="35"></p><p>If you later want to specify additional options (e.g., if you want to change the name of the column used to store the value, you want the value saved to a writable data source, or you want the value stored in an existing variable) —or if you want to modify the referenced element—then select the appropriate element in table on the right and click **Modify**. Next, configure the options as needed, then click **OK**.</p><p>For details on how to use this dialog to configure additional options, see [/JSON+Selector+Reference](/display/SVC20202/JSON+Selector+Reference)></p><h1 id="JSONDataBank-ConfiguringtheJSONDataBankManually">Configuring the JSON Data Bank Manually</h1><p>You can also manually chain the JSON Data Bank tool to a tool within the test, Responder, or Action suite. To configure the JSON Data Bank as a chained tool, complete the following:(Not applicable for use with Message Responders)</p><p>Ensure that you have a test suite or action set with at least two tools.</p><p>Right-click the node for the tool associated with the data you want to extract(e.g., if you want to extract a value from an incoming request or outgoing response, choose the Message Responder that handles those messages), then choose **Add Output**.</p><p>In the **Add Output** wizard, indicate where you want to extract the value from(e.g., Incoming Request, Transport Header, Incoming Attachment, Outgoing Response, etc), select **JSON Data Bank** from the list of tools, then click the **Finish** button. A **JSON Data Bank** node displays below the tool.</p><p>Configure the tool as follows:</p><p>Use the available controls to specify what element you want to extract. To extract an element, select a value from the JSON tree and click the **Extract Element** button. The value you added displays in the **Selected Element** list with a Data Source Column name containing the name of the tool the value came from, as well as the extracted value.

</p><p>The left panel displays the expected JSON response used to create a template from which you can select elements. If JSON Data Bank receives a valid JSON message (e.g., from traffic or as defined in the client tool it is attached to), this panel will be populated automatically. Alternatively, you can copy a sample message into the Literal or Tree tabs. Note that the expected JSON does not get saved by default; if you want to save it, enable the **Save Expected JSON** option.</p><p>If you later want to modify the element referenced by the extraction, click **Modify**, then modify it as desired. See [/JSON+Selector+Reference](/display/SVC20202/JSON+Selector+Reference) for details.</p><p>Repeat steps a and b as needed to configure any additional extractions you want performed.</p><p>In the bottom area of the JSON Data Bank configuration panel, customize the options as desired. See [for details.</p><p>Using the **Extracted Value**<p>After adding and/or modifying the extraction, configure the tool that you want to use the extracted data.</p><p>Set the value to **Parameterized**, and choose the appropriate item from the drop-down. For example, if you saved the value to the **"title"** data source column, you would select it as follows.</p>
</p><h1 id="JSONDataBank-ToolOptions">Tool Options</h1><p>The following options can be configured in the lower portion of the JSON Data Bank tool configuration panel.</p>
</p>Save expected JSON: Specifies whether or not to save the expected JSON.Allow alteration: Specifies whether to allow the alteration of an extraction. When this option is selected, an **Extract** tab and an **Alter** tab display beneath the **Selected Element** list. To alter an extraction, select the **Allow alteration** check box, select the **Alter** tab, add an extraction by clicking the **Extract Element** button, and then modify the extraction by clicking the **Modify** button. The **Modify** dialog displays and contains the following options:XPath: Displays the selected XPath. See \[for details.Alteration Type: Allows you to select how the **Value** you enter alters the JSON. Selecting **Append** will add the altered value to the end of the extraction. Selecting **Prepend** will add the altered value to the beginning of the extraction. Selecting **Replace With** will replace the entire extraction with the altered value you specify.Alteration Value: Allows you to specify either a fixed or parameterized value using a data source.Extract empty elements as: Specifies whether or not empty elements will be extracted. When this option is enabled, you can use the adjacent text field to specify a text string that indicates what **"placeholder"** value should be added for every empty extracted element.This option applies when the extracted element is an empty string. For example, JSON { **"**name**"**; **"**; **"**;name**"**; } has an empty string for **"**name**"**.Note that additional configuration is required if you use this option; see \\[Extract missing elements as: Specifies whether or not missing elements will be extracted. When this option is enabled, you can use the adjacent text field to specify a text string that indicates what **"placeholder"** value should be added for every missing extracted element.This option applies when the extraction fails to locate any matching nodes.Note that additional configuration is required if you use this option; see \\\[
<h1 id="JSONDataBank-ViewingtheDataBankVariablesUsedDuringTestExecution">Viewing the Data Bank Variables Used During Test Execution</h1><p>You can configure the Console view \\\\(**Window**; **Show View**; **Console**\\\\) to display the data bank variables used during test execution. For details, see \\\\[</p><p></p><h1 id="JSONDataBank-HandlingEmpty/MissingElementsstoMaintaintheIntegrityoftheJSONResponse">Handling Empty/Missing Elements to Maintain the Integrity of the JSON Response</h1><p>By default, empty and missing elements will not be extracted. This could impact the integrity of the JSON response for use in a Writable Data Source.</p><p>For instance, assume you have the following JSON:</p><pre><code> {
 "e": 5,
 "e": "",
 "e": "",
 "e": 6</code></pre><p>Also assume that you want to create an extraction for all `e` elements. To do this, select the first element, then click **Extract Element**.</p>
</p></p></div>\\\\]\\\\(/pages/viewpage.action?pageId=77032134#ConfiguringTestSuitePropertiesTestFlowLogic,Variables,etc.-MonitoringVariableUsage\\\\)\\\]\\\(#JSONDataBank-HandlingEmpty/MissingElementsstoMaintaintheIntegrityoftheJSONResponse\\\)\\]\\(#JSONDataBank-HandlingEmpty/MissingElementsstoMaintaintheIntegrityoftheJSONResponse\\)\]\(/display/SVC20202/JSON+Selector+Reference\)](#JSONDataBank-ToolOptions)

Next, select the new extraction, then click **Modify**. In the Modify dialog, change the XPath to extract from `/root/e[1]/text()` to `/root/e/text()`. This will extract all three text nodes. Next, click **Data source column** and select a writable data source column. When the Data Bank is executed, the writable data source will only contain two rows because the second element is missing text:
ROW 1 = 5
ROW 2 = 6
If you want to write the empty value, change the XPath from `/root/e/text()` to `/root/e/` and ensure that **Content Only** is enabled. This way, all three elements are extracted, including their content. You can optionally enable the **Extract empty element** option to change the value extracted for the empty value. The Data Bank will now append the empty value to the writable data source:
ROW 1 = 5
ROW 2 =
ROW 3 = 6