

JSON Selector Reference

This topic provides information on selecting JSON elements in the [JSON Asserter](#) and [JSON Data Bank](#) tools in SOAtest.

Sections include:

- [Changing Elements](#)
- [Modifying Selected XPath Settings](#)
- [Parameterizing XPaths](#)

Changing Elements

For working with JSON and selecting nodes, SOAtest construct an XML DOM "under the hood" in order to use XPath for node selection. The XML structure very closely matches the JSON structure. All names and values are equivalent, except for the following cases:

- The root node of the document is called `root` in the XPath, but is an object or array reference in the JSON document. It appears as `{}` or `[]` in SOAtest JSON trees.
- Array items are called `item` in the XPath, but are unnamed in the JSON document (by nature of being array elements). They appear as `[0]`, `[1]`, etc. in SOAtest JSON trees.
- JSON names that contain characters that are illegal in XML element names are transformed to create legal XML element names for use in XPaths. SOAtest JSON trees will show the original names as they appear in the JSON document.

When you select a JSON tree node, SOAtest also create a corresponding XPath.

If you need to manually construct an XPath and you aren't sure what node names to use in your XPath, we suggest that you use the JSON tree to select a node. This will cause an XPath to be generated that you can then review and modify to meet your specific needs.

Modifying Selected XPath Settings

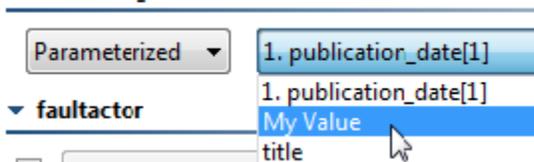
The following options are available in the dialog that opens when you click **Change Element** (for JSON Asserter) or **Modify** (for JSON Data Bank).

Selected XPath Options

- **XPath:** Displays the XPath indicating the value to extract. If you are looking for a more general XPath, you can easily type in a different number into the list indices. For example `[1]` can be changed to `[2]` if you are interested in the 2nd occurrence only. After editing the XPath text, click the **Evaluate XPath** button to validate the XPath format
- **Extract:** Allows you to specify exactly what is extracted.
 - **Entire Element:** Selecting **Entire Element** will output the entire XPath. For example, `XPath/Parent` will output `<parent>VALUE</parent>`. You can configure **Index to extract**. Default value is `1`. This controls which element is extracted in the case where the element occurs more than one time.
 - **Content Only:** Selecting **Content Only** will output only the value. For example, `XPath/Parent` will output `VALUE`. You can configure **Text Content**, which extracts the text content of the element selected, or **All Child Nodes**, which extracts all child nodes of the element selected.
- **XPath Evaluation:** Clicking the **Evaluate XPath** button displays the result of applying the XPath expression against the expected message.

Data Source Column Options (Data Bank Only)

- **Custom column name:** Specifies the name of the data source column in which to store the value. Values are stored in an internal data source unless you specify otherwise (e.g., if you select **Writable data source column or Variable**). This is the name you will use to reference the value in other places. For example, if it is stored in a data source column named **My Value**, you would choose **My Value** as the parameterized value. You could also reference it as `#{My Value}` in literal or multiple response views.



- **Writable data source column:** Tells SOAtest to store the value in a writable data source column (see [Configuring a Writable Data Source](#) for details). This allows you to store an array of values. Other tools can then iterate over the stored values.
- **Write to all columns that match:** Enables you to store values in all columns with names that contain the specified string. When extracting multiple values from a message, each value will be written across all the columns that match. In contrast, if you pick a single writable data source column (the above option), then the values will be written down the column across multiple rows.
- **Variable:** Enables you to save the value in the specified variable so it can be reused across the current test, Responder, or Action suite. The variable must already be added to the current suite as described in [Defining Variables for SOAtest](#). Any values set in this manner will override any local variable values specified in the suite properties panel.

Parameterizing XPath

You can parameterize XPath to reference test or Responder suite variables, environment variables, and data source values. The syntax to reference variables is `${myVariableName}`. The syntax to reference Data Bank values and Data Source Values is: `${myColumnName}`.

For example, if `${XPath Key}` is a data source column name, you could use the following:

```
/*[local-name(.)="bookstore" and namespace-uri(.)=""]/*[local-name(.)="book" and namespace-uri(.)=""][child::node()[local-name(.)="title" and text()="${XPath Key}"]]]
```