JSON Assertor

This topic covers the JSON Assertor tool, which lets you place assertions on different elements in a JSON message. This tool requires a validate license.

Sections include:

- Understanding JSON Assertor
- Configuring JSON Assertor
- Tool Options
- Video Tutoral

Migration Note

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

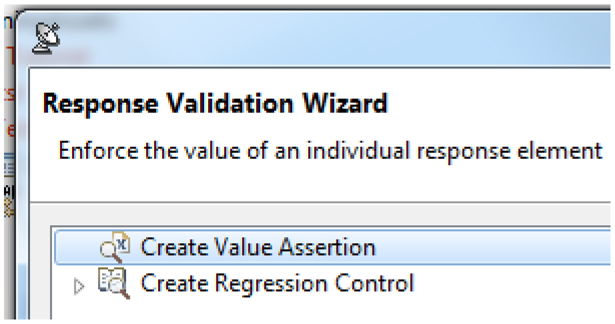
This topic focuses on the current JSON Assertor implementation.For details on the deprecated JSON Assertor, see JSON Assertor Deprecated.

Understanding JSON Assertor

JSON Assertor is used to enforce the correctness of data in a JSON message. It enables you to introspect into individual elements in JSON messages and check whether they meet expectations.

JSON Assertor can be chained against any tool that communicates a JSON message, It is most commonly connected with the Messaging Client and REST Client tools.

This tool is often added from the Create/Update Regression Control dialog for JSON messages (by choosing the **Create Value Assertion** option). For details on adding JSON Assertors in this manner, see Validating the Value of an Individual Response Element.


Response Validation Wizard
Enforce the value of an individual response element
Create Value Assertion
Create Regression Control

It can also be added via the Add Output wizard, which is described in Adding Test Outputs.

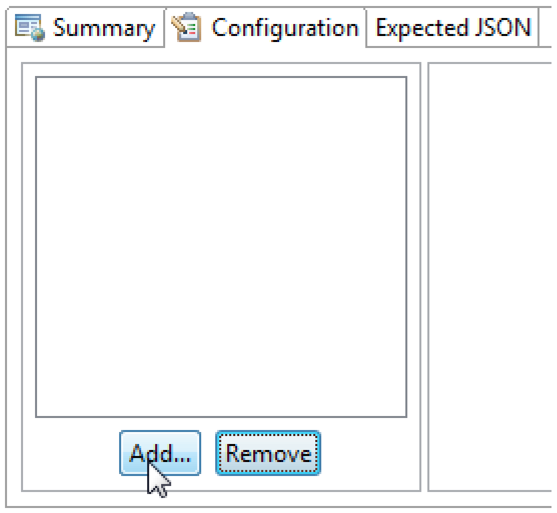
Configuring JSON Assertor

JSON Assertor consists of three main tabs:

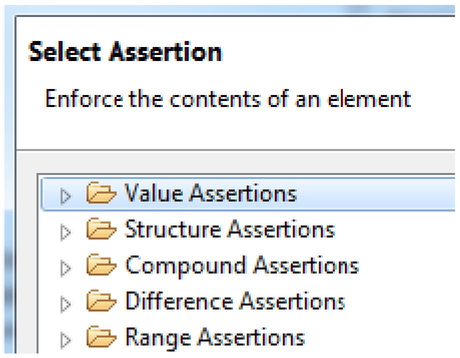
- **Summary:** This tab contains a table showing the details of the JSON assertions that have been configured.
- **Configuration:** This tab is used to create and configure new and existing JSON assertions.
- **Expected JSON:** Specifies the expected JSON response, creating a template from which you can select elements. If JSON Assertor receives a valid JSON message (e.g., from traffic or from the 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.

To configure JSON Assertor:

1. Click the **Add** button in the JSON Assertor’s Configuration tab.

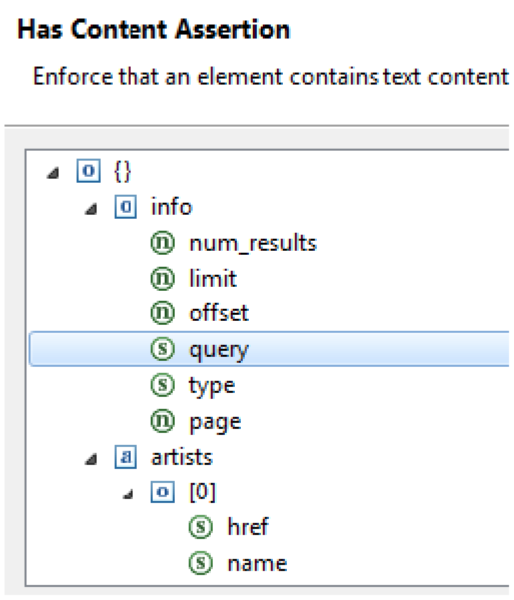


The **Select Assertion** wizard displays.


Select Assertion

Enforce the contents of an element

2. Select an assertion type. The following is a brief summary of the available types of assertions.
   - **Value Assertions:** The following value assertions are available:
     - **Value Assertion:** Enforce the value of a particular element.
     - **Value Occurrence Assertion:** Enforce a certain number of occurrences of an element with a given value (e.g., that the document must have n matches on both the XPath selector and the specified value string).
     - **Numeric Assertion:** Enforce the numeric value of an element.
     - **String Comparison Assertion:** Enforce the value of the text content of a given element.
     - **Regular Expression Assertion:** Enforce that an element matches a regular expression.
     - **Expression Assertion:** Enforce the value of an expression composed of elements.
     - **Custom Assertion:** Enforce the value of an element by scripting custom logic.
   - **Structure Assertions:** The following structure assertions are available:
     - **Occurrence Assertion:** Enforce the number of occurrences of an element.
     - **Has Content Assertion:** Enforce that an element has text content (i.e., length of text > 0).
     - **Has Children Assertion:** Enforce that an element has one or more child elements.
     - **Type Assertion:** Enforce the type of an element.
   - **Compound Assertions:** The following compound assertions are available:
     - **AND Assertion:** Group multiple assertions that all must succeed.
     - **OR Assertion:** Group multiple assertions where at least one must succeed.
     - **Conditional Assertion:** Enforce an assertion only if a condition is met (where the condition is a combination of previously-defined assertions).
   - **Difference Assertions:**Th e following difference assertions are available:
     - **Numeric Difference Assertion:** Enforce a numeric difference on a value of a particular element. Assert that the numeric value of an element differs from a user-specified base value by a user-specified value. For example, in order to assert that a value in degrees Fahrenheit is 3 degrees below freezing, you would set the Base Value to 32 and the Difference Value to -3.
     - **Date Difference Assertion:** Enforce a date difference on a value of a particular element.
     - **DateTime Difference Assertion:** Enforce a date time difference on a value of a particular element.
   - **Range Assertions:**Th e following difference assertions are available:
     - **Numeric Range Assertion:** Enforce that the numeric value of an element is within the inclusive bounds of a numeric range.
     - **Date Range Assertion:** Enforce a date range on a value of a particular element.
     - **Date Time Range Assertion:** Enforce a date time range on a value of a particular element.
3. Click the **Next** button. A tree view displays.
4. From the tree view, select the element that you want this assertion to check, then click the **Finish** button.


Has Content Assertion

Enforce that an element contains text content

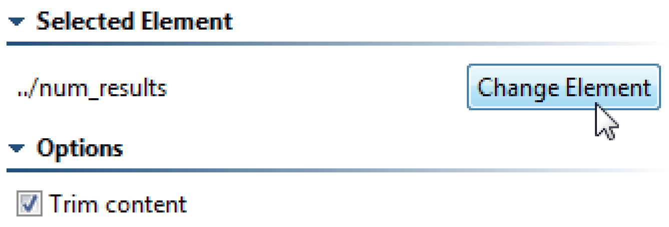
Note that you can edit the structure of this tree in the tool's **Expected JSON** tab (described above).

You may add additional assertions to apply to the message by clicking the **Add** button in the **Configuration** tab.

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 to an existing variable) —or if you want to modify the referenced element—then click the **Change Element** button which is at the bottom right of the **Configuration** tab. This opens a dialog that lets you graphically or manually edit the given element.

For details on how to use this dialog to configure additional options, see JSON Selector Reference.

Tool Options



The **Trim content** option will remove any white space from the start and end of the extracted string before comparing it to the expected text. For example, if " bar " was extracted (ignore quotes in all examples; they are used to show white spaces), it would become "bar"; this would match "bar" (and fail to match " bar ") if the **Trim content** option was not enabled.

Video Tutoral

In this video, you'll learn how to add targeted assertions for values in JSON responses.

</p><p /> </div> </div> </body> </html>