

How do I parse a JSON response for validation or to extract a value?

You will need to write a Script that plays after your message and extracts the value.

Step 1

Insert a Script following the Message whose response you wish to validate or extract a value from.

Step 2

Ensure that the Script doesn't play until the Message has completed.

If you are using a Sequenced Test Clip, this will always be the case and there is nothing special to do.

If you are using a Timed Test Clip, then you must either put the Message and the Script together into a Chain or put a Checkpoint between the Message and the Script. You could also insert a Delay between the Message and Script that you know will always be long enough, but this may not be consistently reliable if the response times to your Message vary.

Note that your Script could also put the extracted value into a property, insert it into another message, or do anything with that value that a Script can do.

Step 3

```
// This Script looks at the response to the Message that
```

Your script should be something like the following:

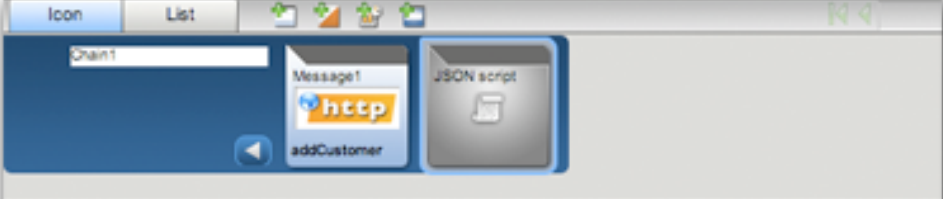
```
// precedes it (it assumes that a Message precedes it).
// The response is assumed to be JSON containing an
// array of values, in a form such as the following:
//   {"ResultSet":{"Result":"lincoln"}}
// It extracts all of the values from the array and logs
// them in the Result.

// Get the Message that immediately precedes this
// Script.
var msg = $context.currentItem.previousItem;

// Extract the array from the JSON response. Note
// that for XPath purposes, the JSON is always
// considered to have a "JSON" node at the root.
var matches = msg.getResponse
    (msg.RESPONSE_HTTP_BODY_AS_JSON,
     "/JSON/ResultSet/Result");

// Was the array found in the response?
if (matches == null)
{
    // The array was not found in the response.
    // Abort the Test Composition with an error.
    $context.composition.abort("Response for
        " + msg.name + ": Result array not found.");
}
else
{
    // The array was found. Format the array
    // value nicely, and then post an entry to
    // the Result with the value.
    var details = "";
    for (var i = 0; i < matches.length; i++)
    {
        details += "Match " + i + ": " + matches[i] + "\n";
    }
    $context.result.postMessage($context.result.LEVEL_INFO,
        "Extracted value from JSON response for
            " + msg.name, details);
}
```

Here is an example of the Script following the Message in a Chain:



The screenshot shows a test chain editor interface. At the top, there are tabs for 'Icon', 'List', and 'JSON script'. Below these, there are three main components: a 'Chain 1' label, a 'Message1' component with an 'http' icon and 'addCustomer' text, and a 'JSON script' component with a document icon. Below the main components, there is a 'Clip' section with a dropdown menu showing 'Selected: JSON script'. The main area contains a JavaScript script with the following content:

```
1
2 // This Script looks at the response to the Message that precedes
3 // it (it assumes that a Message precedes it). The response is
4 // assumed to be JSON containing an array of values, in a form
5 // such as the following:
6 // {"ResultSet":{"Result":"lincoln"}}
7 // It extracts all of the values from the array and logs them
8 // in the Result.
9
10 // Get the Message that immediately precedes this Script.
11 var msg = $context.currentItem.previousItem;
12
13 // Extract the array from the JSON response.
14 var matches = msg.getResponse(msg.RESPONSE_HTTP_BODY_AS_JSON,
15                               "/ResultSet/Result");
16
17 // Was the array found in the response?
18 if (matches == null)
19 {
20     // The array was not found in the response.
21     // Abort the Composition with an error.
22     $context.composition.abort("Response for " + msg.name +
23                               ": Result array not found.");
24 }
25 else
26 {
27     // The array was found. Format the array value nicely, and
28     // then post an entry to the Result with the value.
29     var details = "";
30     for (var i = 0; i < matches.length; i++)
31     {
32         details += "Match " + i + ": " + matches[i] + "\n";
33     }
34     $context.result.postMessage($context.result.LEVEL_INFO,
35                               "Extracted value from JSON response for " + msg.name, details);
36 }
37
```