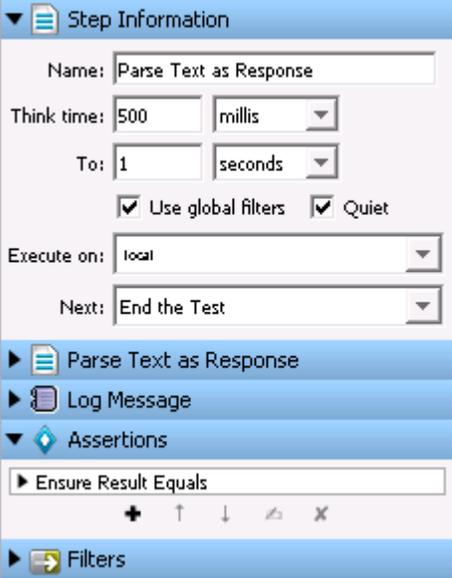




# DevTest 8.1 – Application Test

## Enhanced feature – Improved JSON assertion

	<p>Starting with 8.1, the JSON assertion has been enhanced to show detailed information about the assertion result when the result does not match the expected value.</p> <p>Step called Parse Text as Response in the Workstation UI contains the JSON data as shown</p>
	<p>Created a JSON assertion 'Ensure Result Equals' for this step as shown</p>

▼ Ensure Result Equals - Ensure Result Equals

Name:  If  then

Log:

JSON Path:

Expected value:

A look at the Assertion data  
 JSON Path - \$.lastname  
 Expected value: Smith  
 Click on Run Assertion button

Assertion Execution

Message  
 Assertion result =true

This is the result that ensures that the Results are equal for what was Expected: Smith and Result: Smith

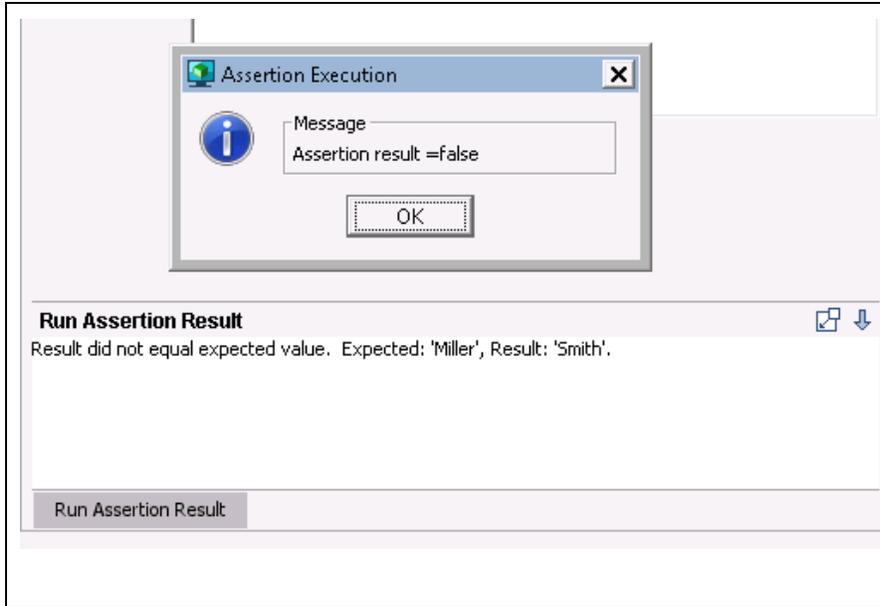
▼ Ensure Result Equals - Ensure Result Equals

Name:  If  then

Log:

JSON Path:

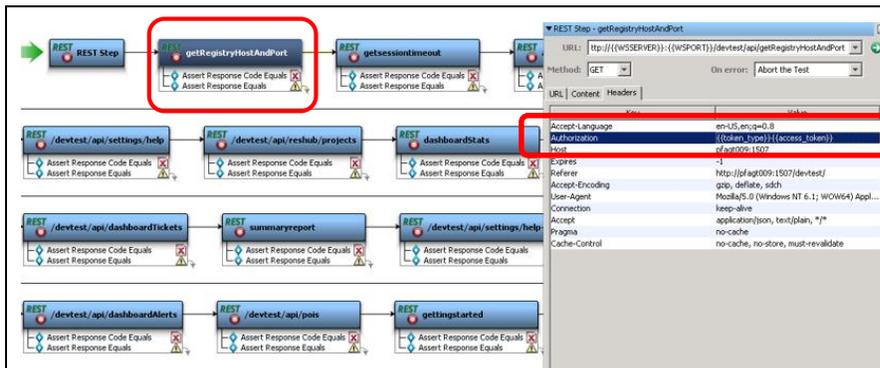
Click OK and now change the Expected value field to Miller  
 Click Run Assertion



You will see the Assertion Result = false

Also verify that a new field is seen at the bottom of that frame called Run Assertion Results. That provides information on what the Expected value was and what the Result was.

## New feature – Inject REST HTTP Header



Prior to 8.1, when the target application required extra security tokens to execute, the user had to add an extra step to login to the application and add filters to parse the response to retrieve the runtime tokens. Then, they would have to manually update the Authorization key-value in the test with the new parameters defined in every step



New Global Filter called Inject REST HTTP Header is provided. Here's what this new filter would do –

- Replace the value in all the HTTP steps if the header and the value exists in the request header of the HTTP steps
- Add the new header and value if the header and value are not in the request header of the HTTP steps
- Skip step. The step that grabs the tokens should not apply authorization to itself

**Inject REST HTTP Header**

Attributes

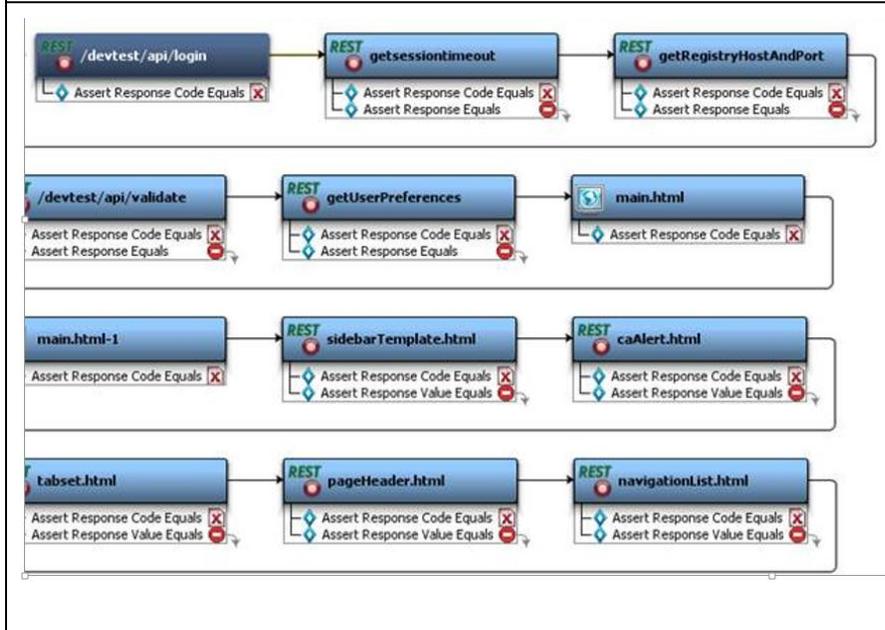
Header Key:

Header Value:

Skip To Step:

+ ↑ ↓ ×

A Global Filter example here.



Consider the REST step for login

**JSON Path Filter - JSON Path Filter**

Filter in:

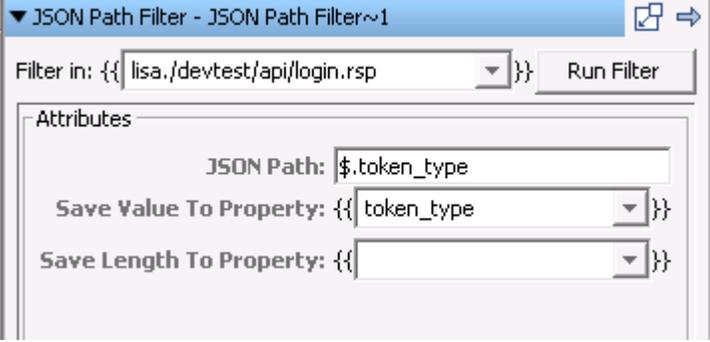
Attributes

JSON Path:

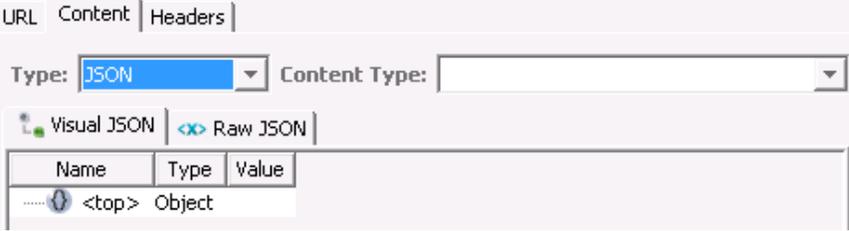
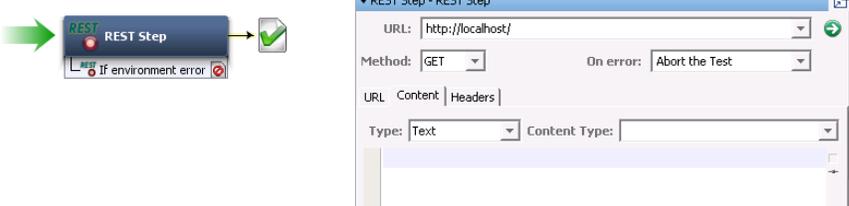
Save Value To Property:

Save Length To Property:

Execution of this call sets the required values for token\_type and access\_token in a filter.

	
	<p>For every subsequent REST step, these values will be added in the header.</p>

## Enhanced feature – Improved REST API testing using Visual JSON editor

 <p>URL   Content   Headers</p> <p>Type: <b>JSON</b> Content Type: [ ]</p> <p>Visual JSON   Raw JSON</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>..... &lt;top&gt;</td> <td>Object</td> <td></td> </tr> </tbody> </table>	Name	Type	Value	..... <top>	Object		<p>Starting with 8.1, we now have a Visual JSON and Raw JSON editors as a part of the REST step</p> <p>Opened a REST step and observed that in the Content tab there is a new entry for JSON in the Type dropdown box.</p>
Name	Type	Value					
..... <top>	Object						
	<p>We will use the following sample data. Copy this sample data -</p> <pre>[{"class": "com.ca.lisa.demo.CarInventory", "id": 1, "carTrim": "Premium Plus", "color": "Silver", "dealer": {"class": "com.ca.lisa.demo.CarDealer", "id": 2, "address": "3800 Motor City Dr", "city": "Denver", "name": "Mountain Motors Inc", "state": "CO", "telephone": "303-222-8766", "website": "www.mmdenver.com", "zip": "80202"}, "engine": "2.7 V6", "image1": "inventory1.jpg", "image2": "cars/interior1.jpg", "milage": 138560, "model": {"class": "com.ca.lisa.demo.CarModel", "id": 18, "fuelType": "Gas", "make": {"class": "com.ca.lisa.demo.CarMake", "id": 2, "name": "Audi"}, "modelYear": 1996, "name": "A4-B", "subName": "Base", "type": {"class": "com.ca.lisa.demo.CarType", "id": 7, "name": "Sedan"}}, "modelYear": 2011, "options": "Leather, Navigation, Rear Air", "owners": 6, "price": 3995.0, "stockNumber": "NAD8989", "transmission": "Auto", "vin": "2T1KR32E37C639014"}]</pre>						
 <p>REST Step</p> <p>REST Step - REST Step</p> <p>URL: http://localhost</p> <p>Method: GET On error: Abort the Test</p> <p>URL   Content   Headers</p> <p>Type: Text Content Type: [ ]</p>	<p>Open a new Test Case and add a REST step to it. Click on the Content tab to see the default Type as Text</p>						

▼ REST Step - REST Step

URL:

Method:  On error:

URL | Content | Headers

Type:  Content Type:

```

    "9", "transmission": "Auto", "vin": "2T1KR32E37C639014"}
  
```

Paste the above sample data in the empty text box to see the data as shown

▼ REST Step - REST Step

URL:

Method:  On error:

URL | Content | Headers

Type:  Content Type:

Visual JSON | Raw JSON

Name	Type	Value
<top>	Array	
Object	Object	
class	String	com.ca.lisa.demo.CarInventory
id	Unquoted	1
carTrim	String	Premium Plus
color	String	Silver
dealer	Object	
class	String	com.ca.lisa.demo.CarDealer
id	Unquoted	2
address	String	3800 Motor City Dr
city	String	Denver
name	String	Mountain Motors Inc
state	String	CO
telephone	String	303-222-8766
website	String	www.mmdenver.com
zip	String	80202
engine	String	2.7 V6
image1	String	inventory1.jpg
image2	String	cars/interior1.jpg
milage	Unquoted	138560
model	Object	
class	String	com.ca.lisa.demo.CarModel
id	Unquoted	18
fuelType	String	Gas
make	Object	
class	String	com.ca.lisa.demo.CarMake
id	Unquoted	2
name	String	Audi
modelYear	Unquoted	1996
name	String	A4-B

Change the Type from the default Text to the (new for 8.1) JSON type. By default you will see the Visual JSON tab with the nicely formatted visual JSON data

▼ REST Step - REST Step

URL:

Method:  On error:

URL | Content | Headers

Type:  Content Type:

Visual JSON <> Raw JSON

```
39", "transmission": "Auto", "vin": "2T1KR32E37C639014"}]
```

Now change the tab to Raw JSON and you will see the display change from the Visual JSON to Raw JSON