

Enhanced feature – Improved JSON assertion

	Parse Text as Response - Parse Text as Response operty Key (opt) Load From File Test ("firstname":"James", "lastname":"Smith")	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. Step called Parse Text as Response in the Workstation UI contains the JSON data as shown
 Step Information Name: Parse Text as Response Think time: 500 millis To: 1 seconds Use global filters Quiet Execute on: Iccal Next: End the Test Parse Text as Response Dog Message Assertions Ensure Result Equals Filters 		Created a JSON assertion 'Ensure Result Equals' for this step as shown

Ensure Result Equals - Ensure Result Equals Name: Ensure Result Equals If True then Fail the Test Image: Run Assertion JSON Path: \$.lastname Smith Expected value:	A look at the Assertion data JSON Path - \$.lastname Expected value: Smith Click on Run Assertion button
Assertion Execution	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: Ensure Result Equals If True then Fail the Test Image: Insure Result Equals Run Assertion JSON Path: \$.lastname Miller	Click OK and now change the Expected value field to Miller Click Run Assertion

Assertion Execution Message Assertion result =false OK	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.
Result did not equal expected value. Expected: 'Miller', Result: 'Smith'.	
Run Assertion Result	

New feature – Inject REST HTTP Header

PST REST /devtest/api/reshuh/project Ausert Response Code Equals Ausert Response Code Equals Ausert Response Code Equ	Star optilegity/Holdwite Star optilegity/Holdwite Int: [tp://(WSCRTHE):(WSCRT)]devted/as/gelogethyHoldwite(WSCRTHE):(WSCRTH	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
Global Filters Inject REST HTTP Header		New Global Filter called Inject REST HTTP Header is provided. Here's what this new filter would do –
+ t ↓ ×		 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: Authorization Header Value: }}{{space}}{{access_token}} Skip To Step: /devtest/api/login	A Global Filter example here.
#EST getsessiontimeout #EST getsessiontimeout Assert Response Code Equals Assert Response Code Equals Assert Response Equals /devtest/api/validate #EST getUserPreferences main.html Assert Response Code Equals Assert Response Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Assert Response Code Equals Main.html-1 #EST sidebarTemplate.html #EST caAlert.html Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals Assert Response Code Equals #Assert Response Value Equals #Assert Response Code Equals #Assert Response Code Equals #Assert Response Code Equals #Assert Response Value Equals #Assert Response Valu	Consider the REST step for login
▼ JSON Path Filter - JSON Path Filter Filter in: {{ lisa./devtest/api/login.rsp •}} Run Filter Attributes JSON Path: \$.access_token Save Value To Property: {{ access_token •}} Save Length To Property: {{ access_token •}}	Execution of this call sets the required values for token_type and access_token in a filter.

▼ JSON Path Filter - JSON Path Filter~1	
Filter in: {{ lisa./devtest/api/login.rsp }} Run Filter	
Attributes	
JSON Path: \$.token_type	
Save Value To Property: {{ token_type >}}	
Save Length To Property: {{	
	For every subsequent REST step, these
	values will be added in the header.

URL Content Headers Type: JSON Content Type: Visual JSON Raw JSON Name Type Value Content Type Value Conten	Starting with 8.1, we now have a Visual JSON and Raw JSON editors as a part of the REST step Opened a REST step and observed that in the Content tab there is a new entry for JSON in the Type dropdown box.
	We will use the following sample data. Copy this sample data - [{"class":"com.ca.lisa.demo.CarInvento ry","id":1,"carTrim":"Premium Plus","color":"Silver","dealer":{"class":" com.ca.lisa.demo.CarDealer","id":2,"ad dress":"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.Car Model","id":18,"fuelType":"Gas","make" :{"class":"com.ca.lisa.demo.CarMake"," id":2,"name":"Audi"},"modelYear":199 6,"name":"A4- B","subName":"Base","type":{"class":"c om.ca.lisa.demo.CarType","id":7,"name ":"Sedan"}},"modelYear":2011,"options ":"Leather, Navigation, Rear Air","owners":6,"price":3995.0,"stockN umber":"NAD8989","transmission":"Aut o","vin":"2T1KR32E37C639014"}]
REST Step VREST Step VREST Step URL: http://localhost/ V URL: http://localhost/ V URL: fttp://localhost/ V URL: Content Headers URL: Type: Type: Type: Type: V	Open a new Test Case and add a REST step to it. Click on the Content tab to see the default Type as Text

Enhanced feature – Improved REST API testing using Visual JSON editor

▼ REST Step - REST Step URL: http://localhost/ Method: GET ▼ URL Content Headers Type: Text ▼ \$9", "transmission	Content ":"Auto	On error: Abort the Test Type: ", "vin": "2T1KR32E37	_	Paste the above sample data in the empty text box to see the data as shown
	Content Ty	On error: Abort the Tes		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
Name	N Type	Value		
⊡0 <top></top>	Array			
E 0	Object			
class	String	com.ca.lisa.demo.CarInventory		
did	Unquoted			
arTrim	String	- Premium Plus		
color	String	Silver		
⊡	Object			
- Class	String	com.ca.lisa.demo.CarDealer		
id	Unquoted			
	String	- 3800 Motor City Dr		
city	String	Denver		
name	String	Mountain Motors Inc		
	String	со		
	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	138260		
⊡O model	Object String	com.ca.lisa.demo.CarModel		
id	Unquoted			
fuelType	String	Gas		
⊡ Ø make	Object			
	String	com.ca.lisa.demo.CarMake		
ð id	Unquoted			
name	String	Audi		
modelYear	Unquoted	1996		
- 🕛 name	String	A4-B	_	
	+			

▼ REST Step - REST Step URL: http://localhost/		⇒ 1 <u>3</u> ▼ €	Now change the tab to Raw JSON and you will see the display change from the Visual JSON to Raw JSON
Method: GET	On error: Abort the Test	Ŧ	
URL Content Headers			
Type: JSON Content Typ	e:	-	
🐛 Visual JSON 🛛 👁 Raw JSON			
39","transmission":"Auto	","vin":"2T1KR32E37C639	014")]	