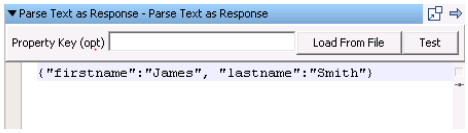
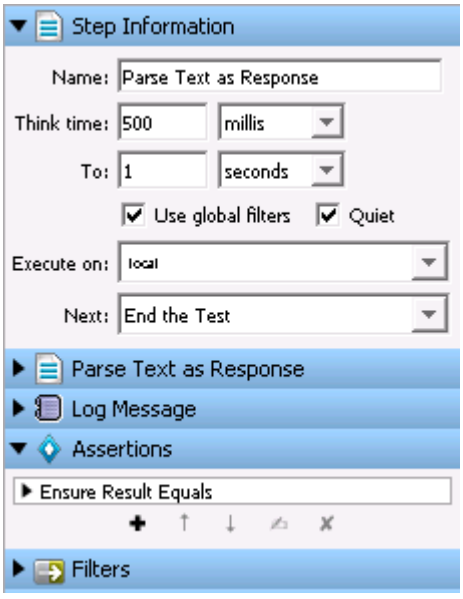
DevTest 8.1 – Application Test

Enhanced feature – Improved JSON assertion

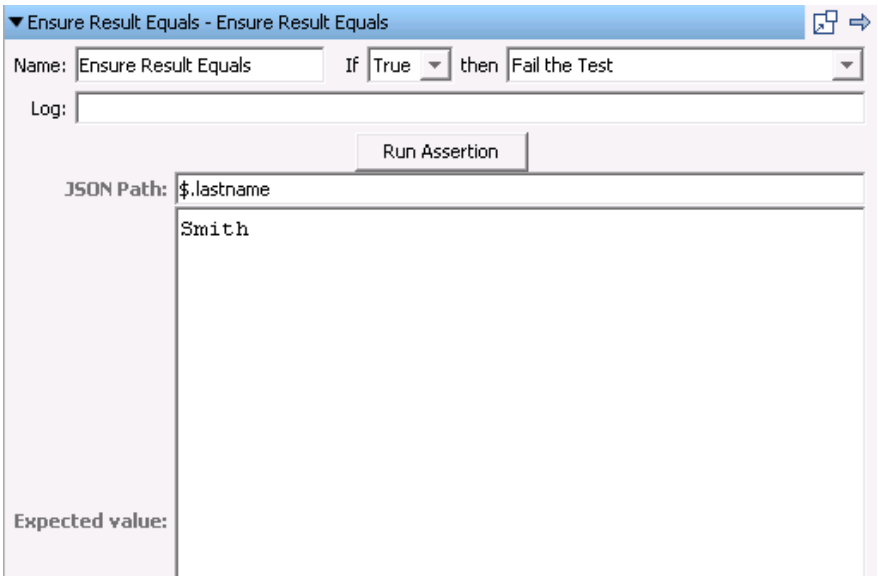
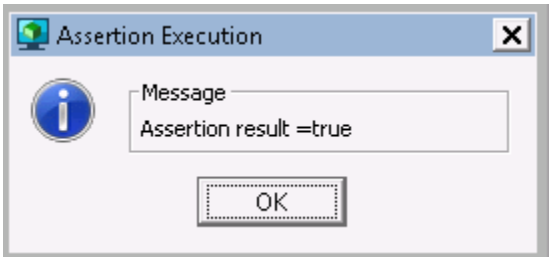
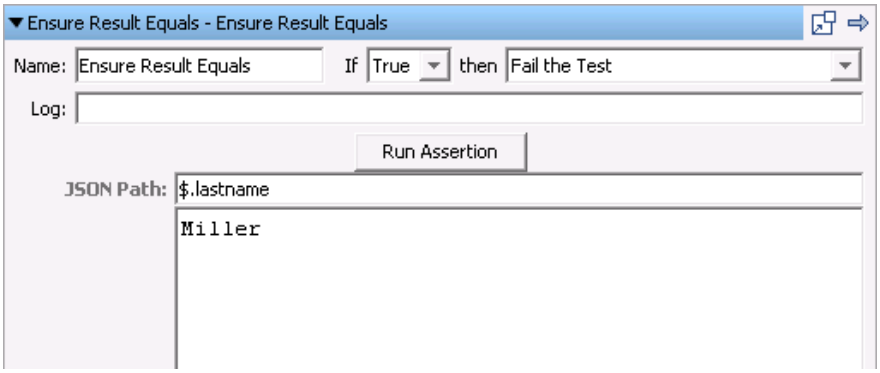


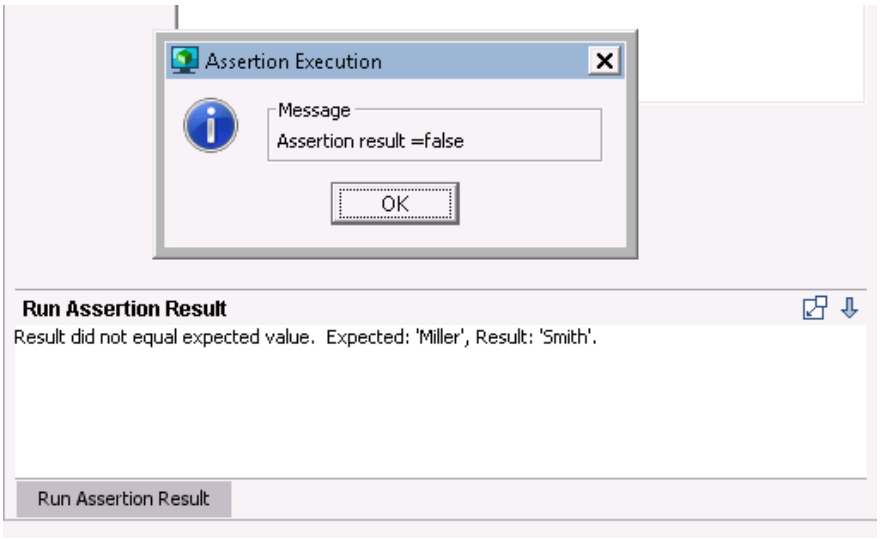
Starting with 8.1, the JSON assertion has been enhanced to show detailed information about the assertion result when the result does not match the expected value.

Step called Parse Text as Response in the Workstation UI contains the JSON data as shown

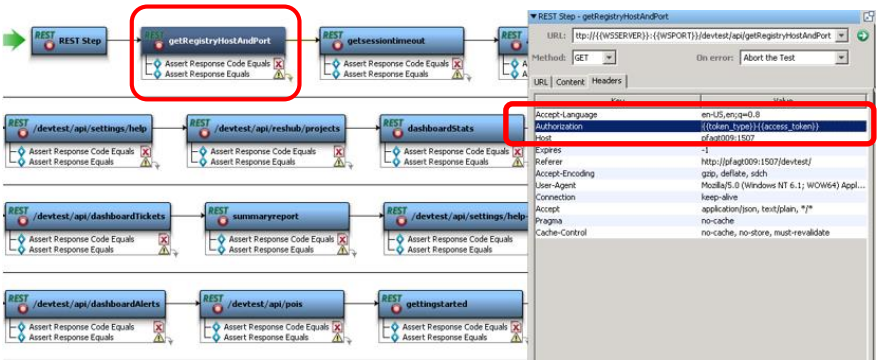
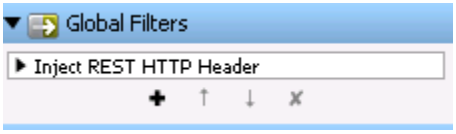


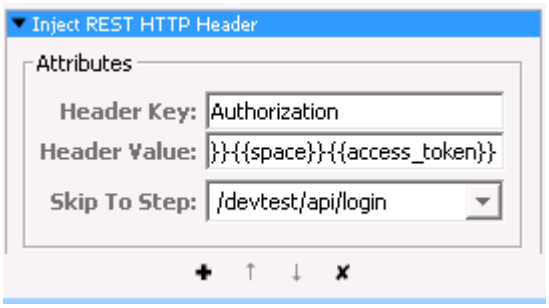
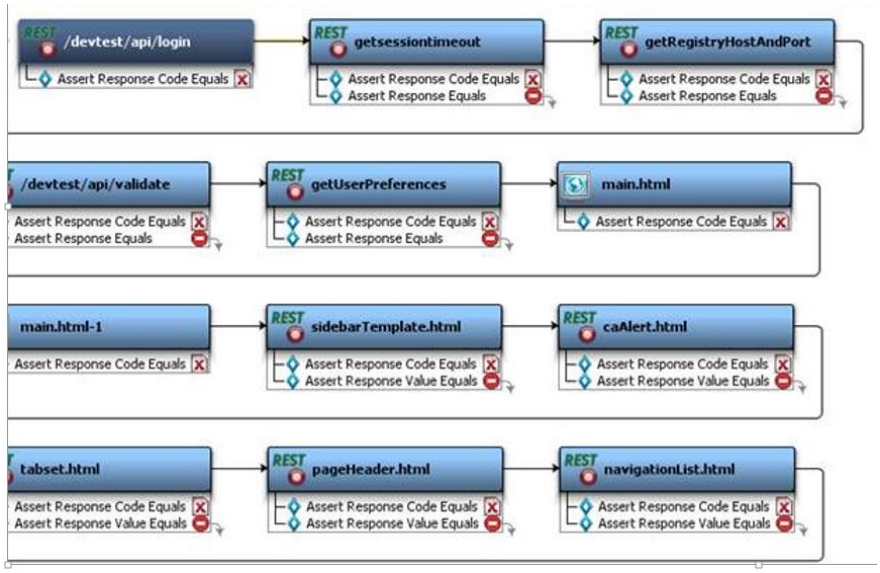
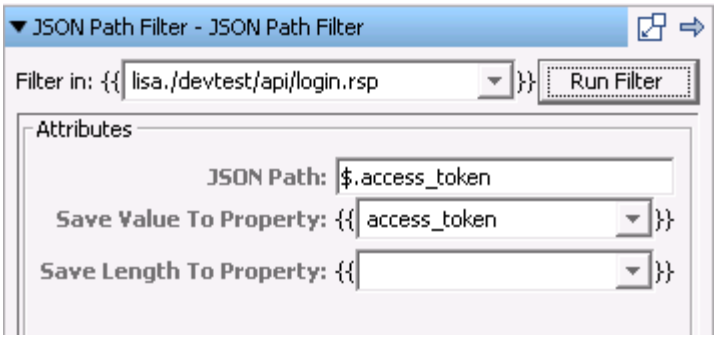
Created a JSON assertion 'Ensure Result Equals' for this step as shown

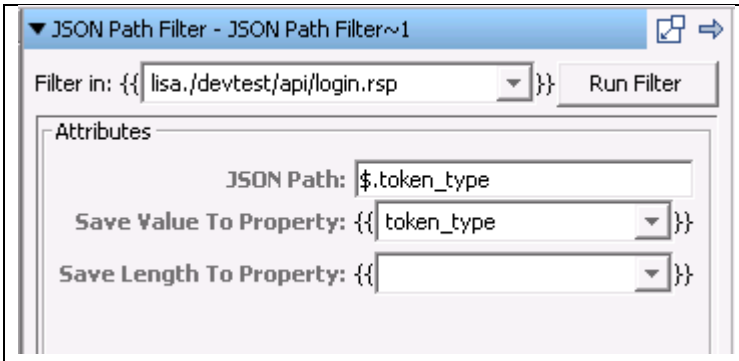
	A look at the Assertion data JSON Path - $.lastname Expected value: Smith Click on Run Assertion button
	This is the result that ensures that the Results are equal for what was Expected: Smith and Result: Smith
	Click OK and now change the Expected value field to Miller Click Run Assertion

	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.

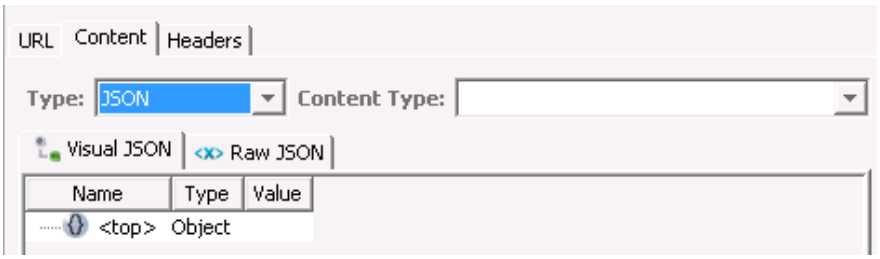
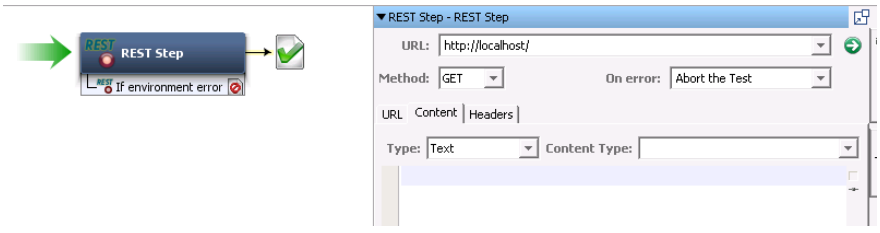
New feature – Inject REST HTTP Header

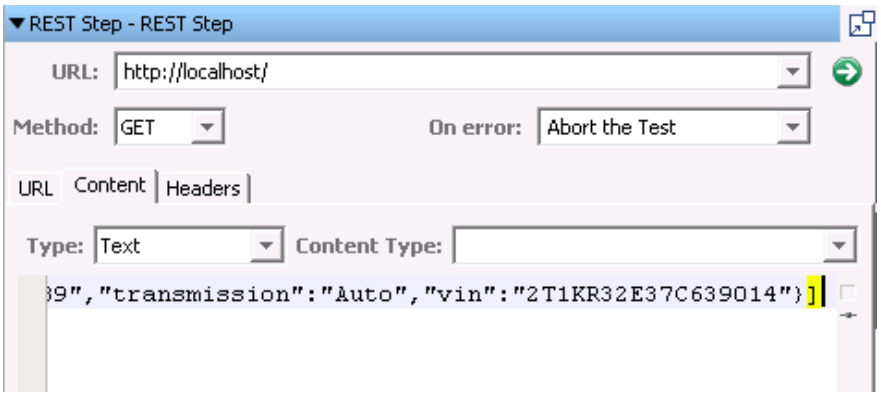
	Prior to 8.1, when the target application required extra security tokens to execute, the user had to add an extra step to login to the application and add filters to parse the response to retrieve the runtime tokens. Then, they would have to manually update the Authorization key-value in the test with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

	A Global Filter example here.
	Consider the REST step for login
	Execution of this call sets the required values for token_type and access_token in a filter.

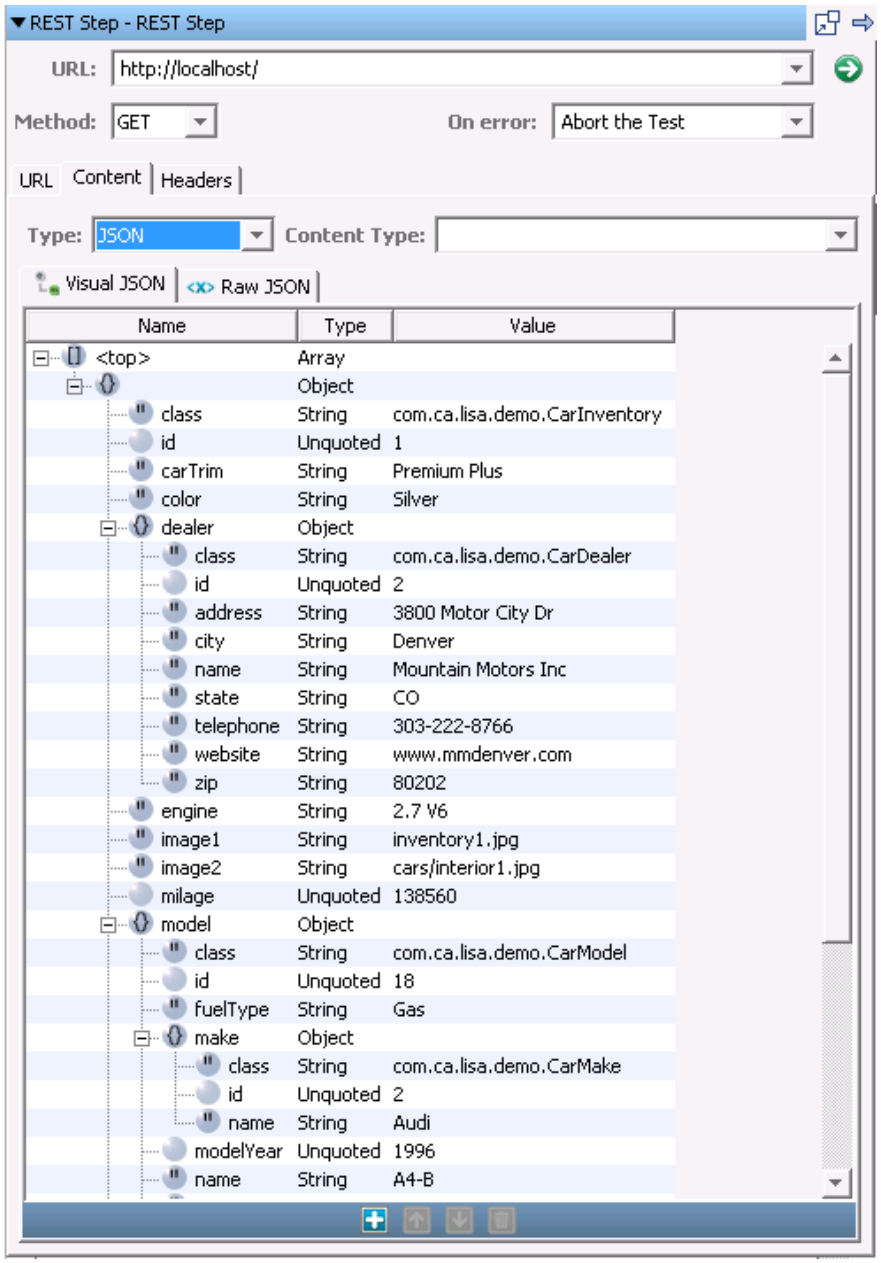
	
	For every subsequent REST step, these values will be added in the header.

Enhanced feature – Improved REST API testing using Visual JSON editor

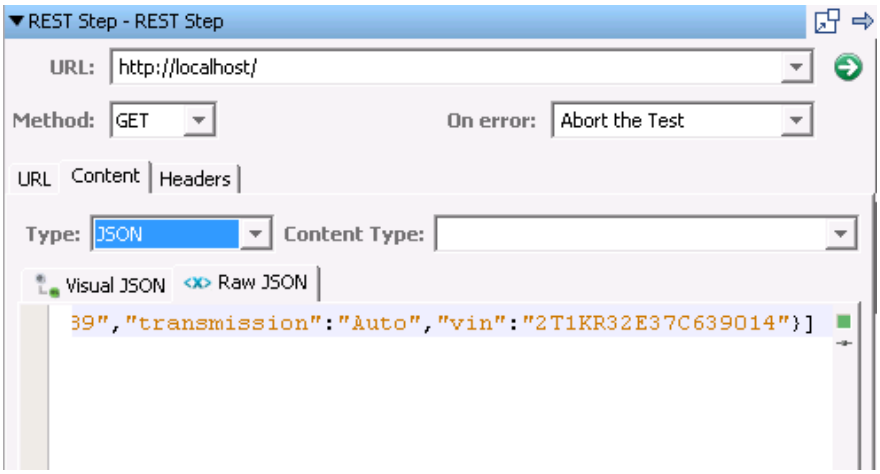
	Starting with 8.1, we now have a Visual JSON and Raw JSON editors as a part of the REST step Opened a REST step and observed that in the Content tab there is a new entry for JSON in the Type dropdown box.
	We will use the following sample data. Copy this sample data - [{"class":"com.ca.lisa.demo.CarInventory","id":1,"carTrim":"Premium 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"}]
	Open a new Test Case and add a REST step to it. Click on the Content tab to see the default Type as Text



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



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



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