

What Is An Assertion?

- Assertion means act of affirming or stating something. It can also be interpreted as check point or a validation point.
- Once a request is sent to a web server a response is received. We need to validate if the response contains the data that we expect. In order to validate the response, we need to use assertions.

Types Of Assertion

There are various ways of asserting a response; however we will focus on the commonly used Assertions types while validating a response. Below are the ones that are available in Open Source version of SoapUI.

- Property Content
- Compliance Status Standard
- Script
- SLA
- JMS
- Security

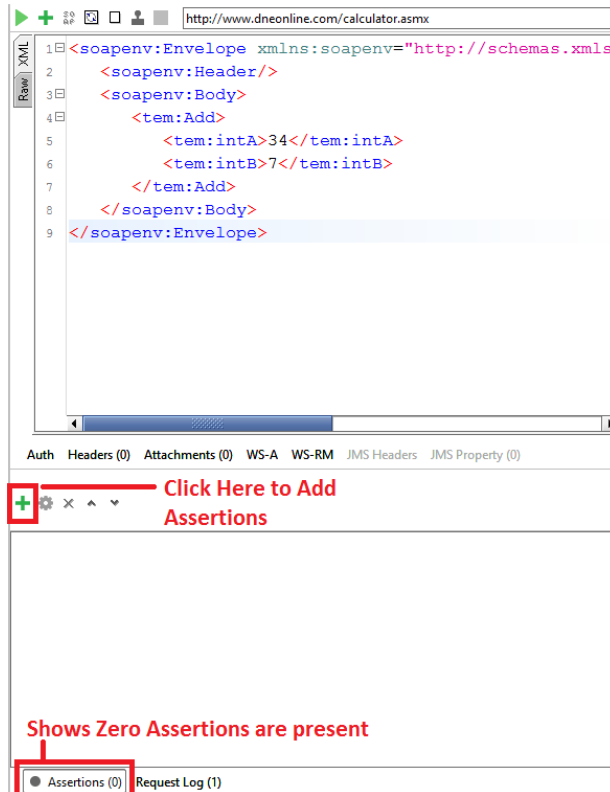
PRO version also has an inbuilt **JDBC Assertion** using which we can assert if the web service has updated the database correctly.

CONTAINS ASSERTION

Searches for the existence of the specified string. It also supports regular expression.

Step 1: By Default there are no assertions.

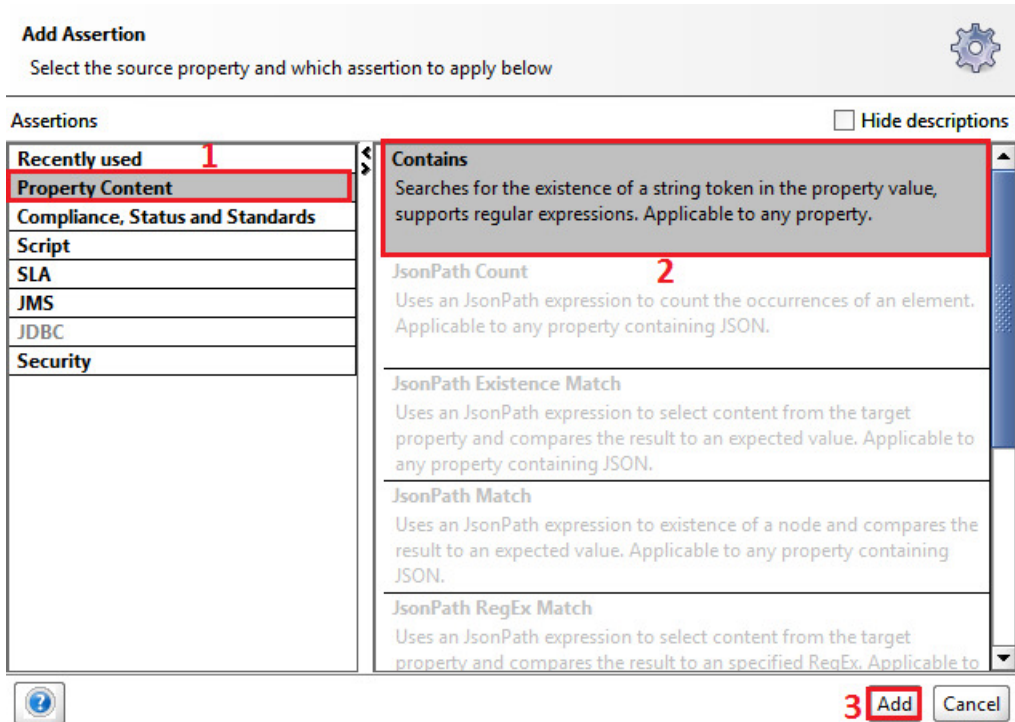
- The Number of Assertions are shown in the Assertions Tab.
- To add a new as



sertion, click on 'Add New Assertion' button.

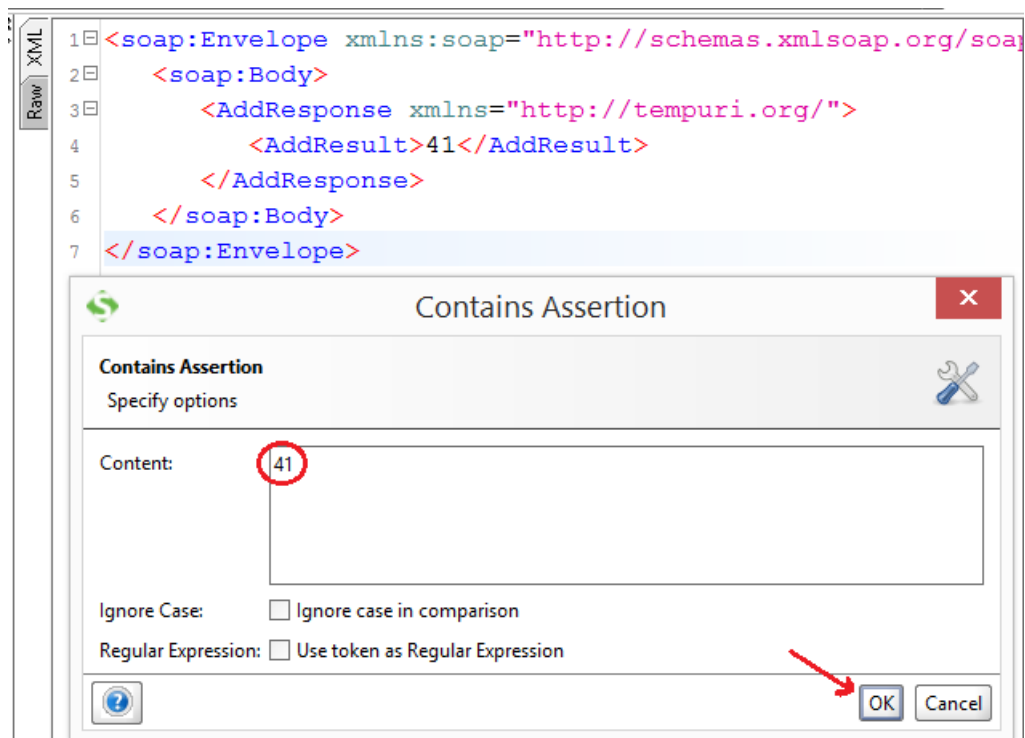
Step 2: Now,

1. Select the Assertion Category.
2. Select the Assertion Type.
3. Click 'Add'

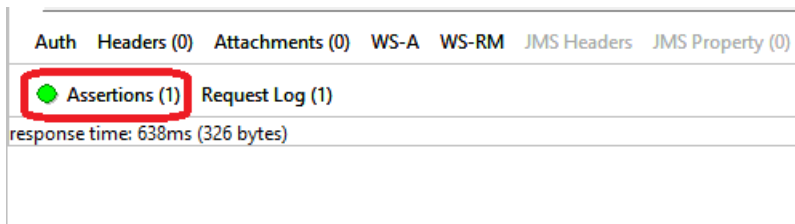


Step 3: Let us validate if the string '63' exist in the response. Click 'OK'

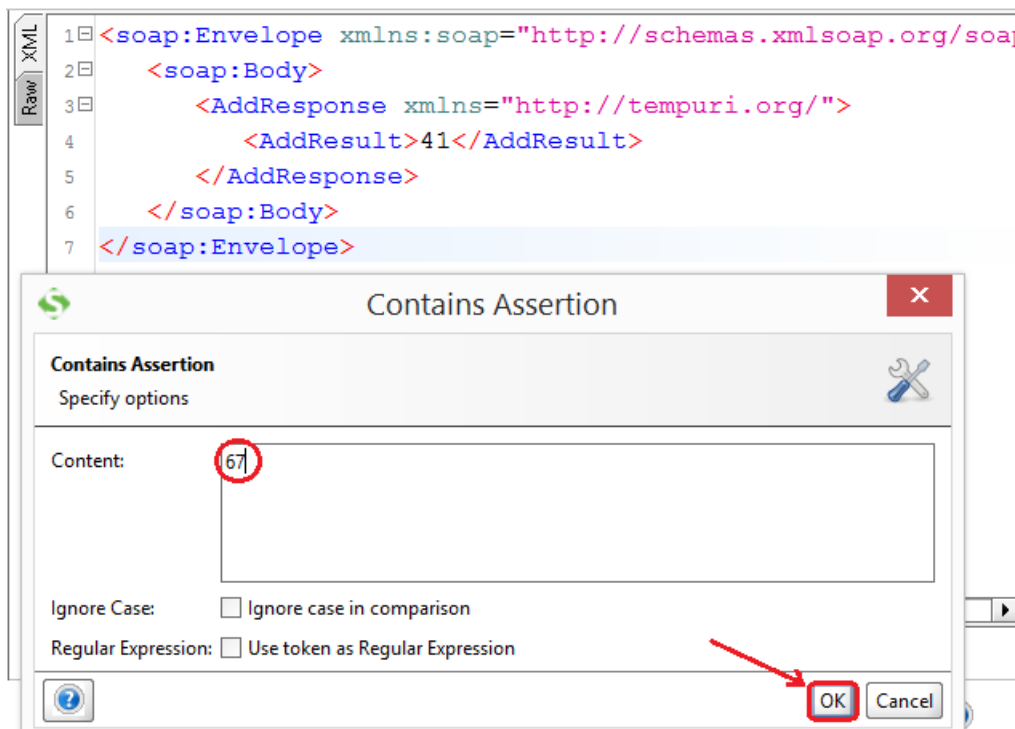
Note : We can also ignore case and add regular expression.



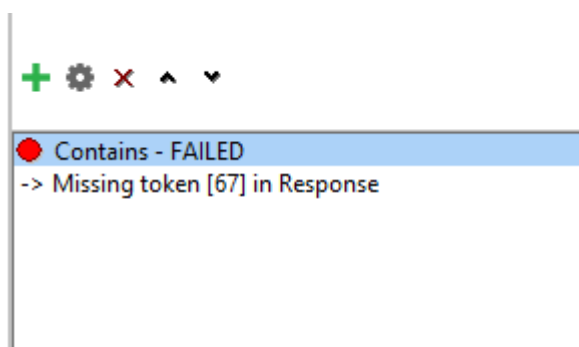
Step 4: Upon adding it, immediately assertion is executed and shows if VALID or INVALID.



Step 5: Now Let us say we change the content of 'Contains Assertion' to '64' and see what happens.



Step 6: The Assertion is executed and the result is thrown to the user. Since we don't have the string "64" within the response, the assertion has failed.



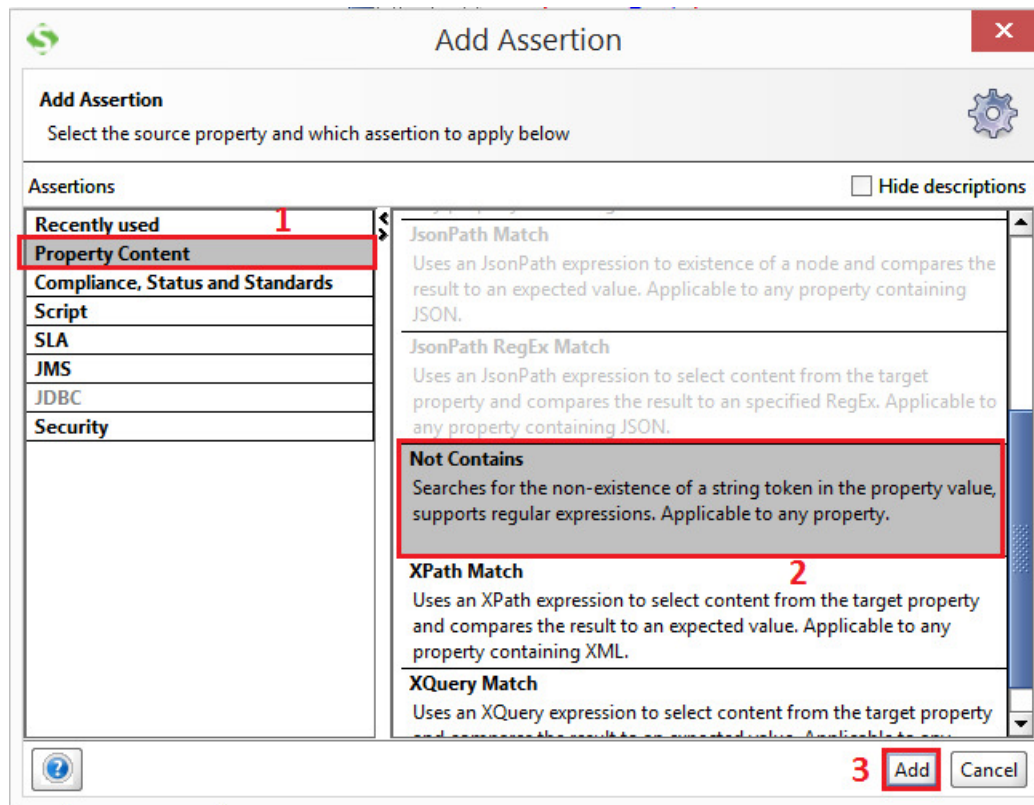
NOT CONTAINS ASSERTION

Searches for the Non-existence of the specified string. It also supports regular expression.

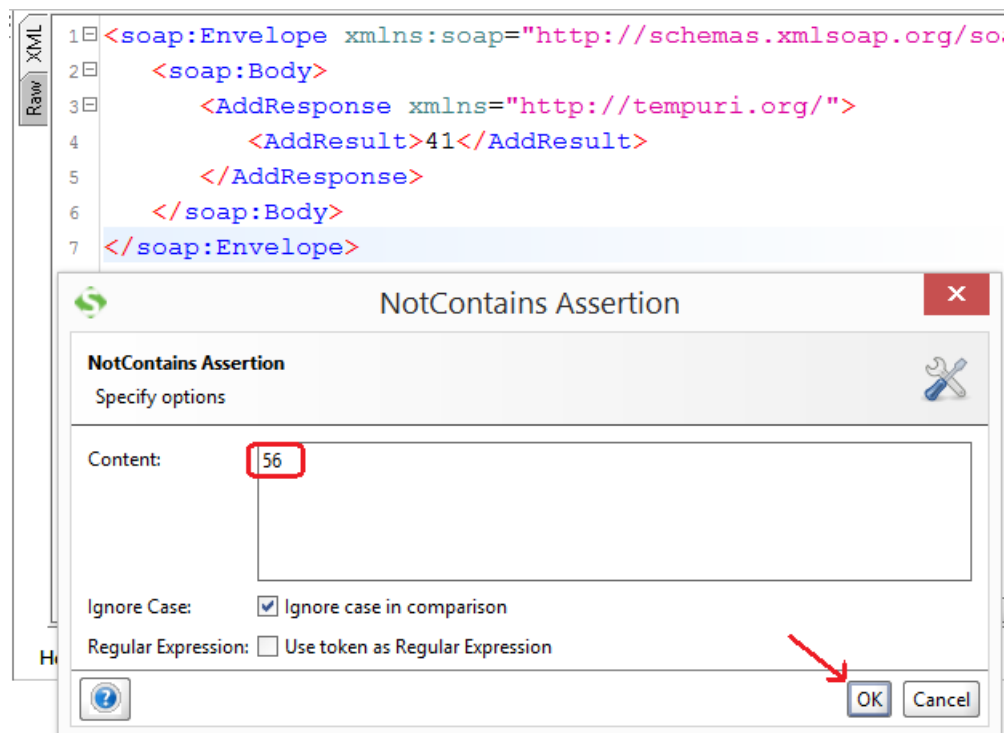
Step 1: Now after clicking on 'add new assertions' button,

1. Select the Assertion Category.

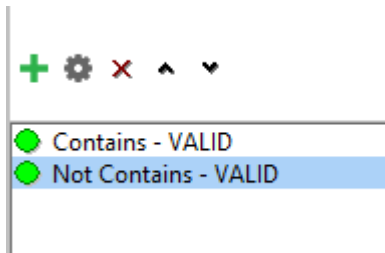
2. Select the Assertion Type – In this case 'NOT Contains'
3. Click 'Add'



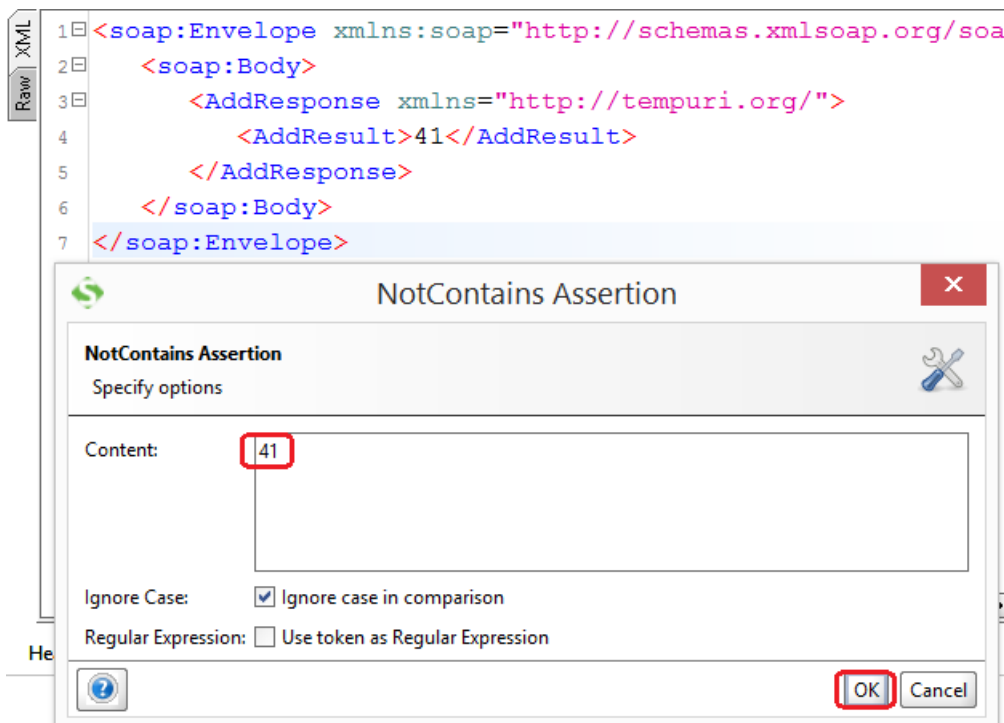
Step 2: Let us validate if the string 'FromCurrency' exist in the response. Enter the string 'FromCurrency' and Click 'OK'



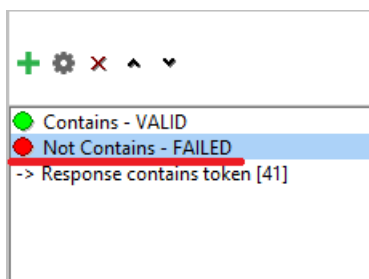
Step 3: As soon as an assertion is added, it executes and displays the result. So far we have added two assertions hence both the assertions are executed and displayed the result.



Step 4: Now let us change the contents of the 'Not Contains Assertion' and see what happens. We will check for the non-existence of the string "ConversionRateResponse".



Step 5 : The string 'ConversionRateResponse' is actually present in the response, hence the 'NOT Contains' assertion will fail as shown below.

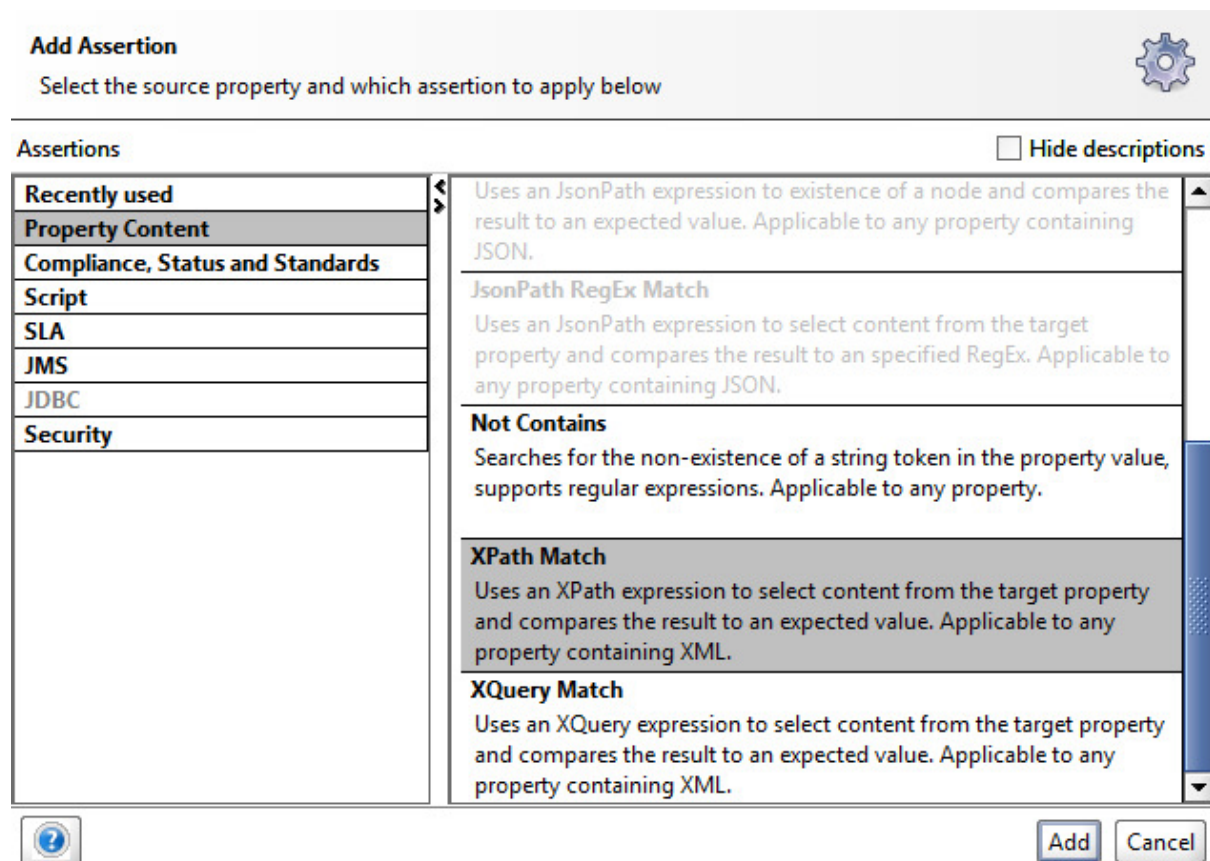


XPATH MATCH ASSERTION

Uses [XPath](#) expression to select the target node and its values. XPath, is an XML query language for selecting nodes from an XML document.

Step 1: Now after clicking on 'Add New Assertions' button,

1. Select the Assertion Category.
2. Select the Assertion Type – In this case 'XPath Match'
3. Click 'Add'



Step 2: Add XPath Window opens.

- Before Adding XPath, we need to declare the Namespace. An XML namespace is a collection of names, identified by a Uniform Resource Identifier (URI) reference, which are used in XML documents as element and attribute names. The same is used in SOAP UI XPath Assertion.
- For declaring XML Namespace, we just need to click on 'Declare' button which would do the job for us else we can also manually declare a namespace ourselves.
- After declaring the namespace we need to refer the XPath using the created namespace.
- Upon clicking the 'Declare' button, two namespaces will pop up as we have two URI's. One of them is the schema URL and the other one corresponds to the actual

web service URL. We need to use the actual namespace where the web service is located and NOT the schema namespace while referencing XPath.

Step 3: Now we need to enter the XPath of the XML node that we need to validate.

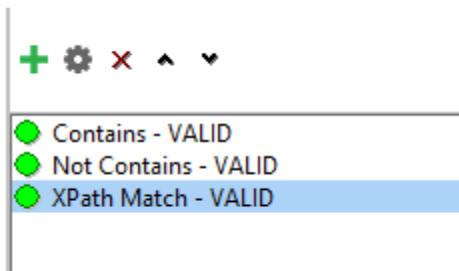
//ns1:AddResult Gives us the Value of the node enclosed between **<AddResult>** & **</AddResult >** and ns1 corresponds to the declared namespace which is pointing to 'http://tempura.org/'

After entering the XML, we need to click on 'Select from current' so that value from the current response would be picked up for comparison going forward.

Step 4: So far,

1. After declaring the namespaces, we have entered the XPath of XML node that we need to Validate.
2. We Need to click 'Select from Current' to make the current value as the expected value.
3. The current value is shown to the user which we can modify if required.
4. Click 'Save'.

Step 5: The added Assertion will be displayed as shown below



Raw XML

```
1 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
2   <soap:Body>
3     <AddResponse xmlns="http://tempuri.org/">
4       <AddResult>41</AddResult>
5     </AddResponse>
6   </soap:Body>
7 </soap:Envelope>
```

XPath Match Configuration

Specify XPath expression and expected result
declare namespaces with declare namespace <prefix>=<namespace>;

XPath Expression

Declare 1

declare namespace soap='http://schemas.xmlsoap.org/soap/envelope/';
declare namespace ns1='http://tempuri.org/';

//ns1:AddResult **xPath**

Expected Result 2

Select from current Test ☐ Allow Wildcards ☐ Ignore namespace prefixes ☐ Ignore XML Comm

41

3 **Save** Cancel