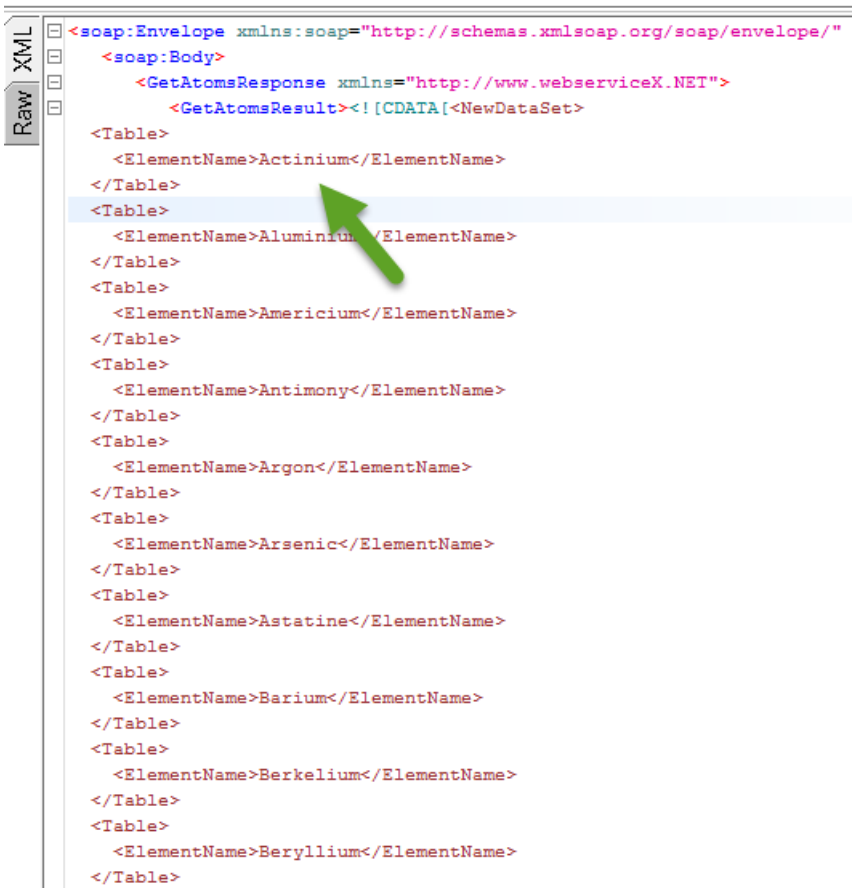
SOPAUI Assignments

1) SOAP

1) Open Soapui Tool
2) File -> Import Project (periodictable-soapui-project.xml)
3) Create “TestSuite” as “PeriodicTable_TestSuite”
4) Create “TestCase” as “PeriodicTable_TestCase”
5) Create Soap Request as “GetAtoms”
    a. Select “periodictableSoap -> GetAtoms” method
    b. click checkbox as “Create optional elements” then click “ok”
    c. Run the “Test Step” and Verify the response



6) Set Assertion

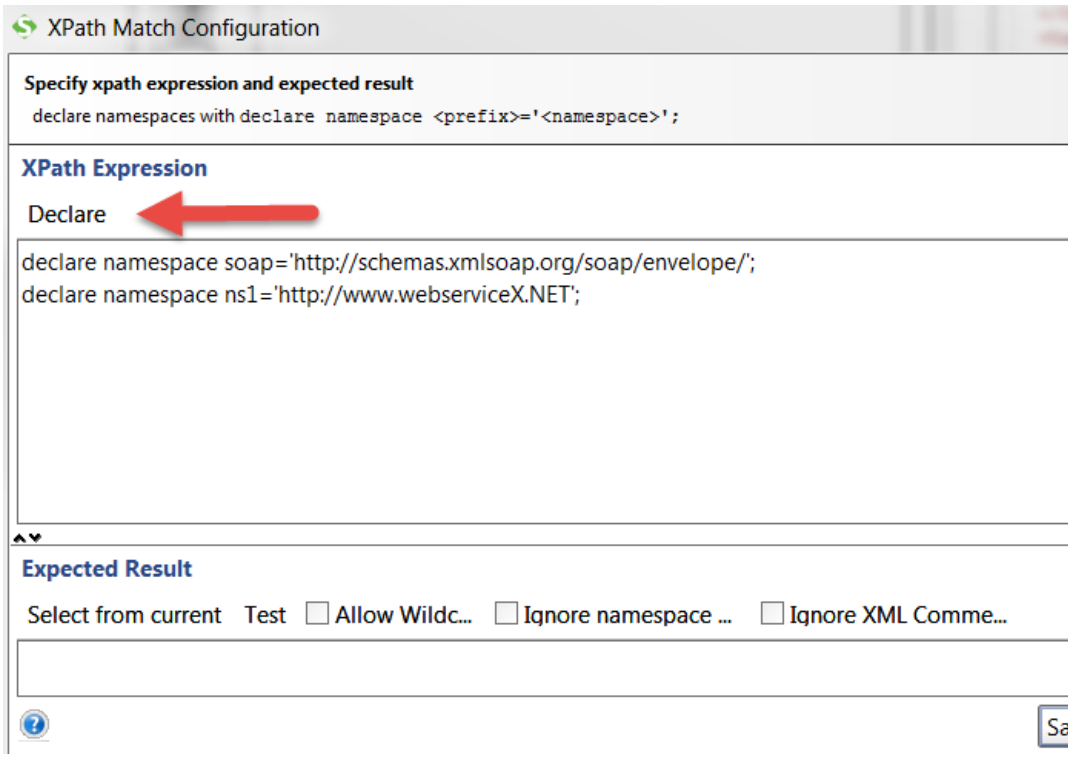
a. Valid http status code as “200”

b. Contains “Actinium”

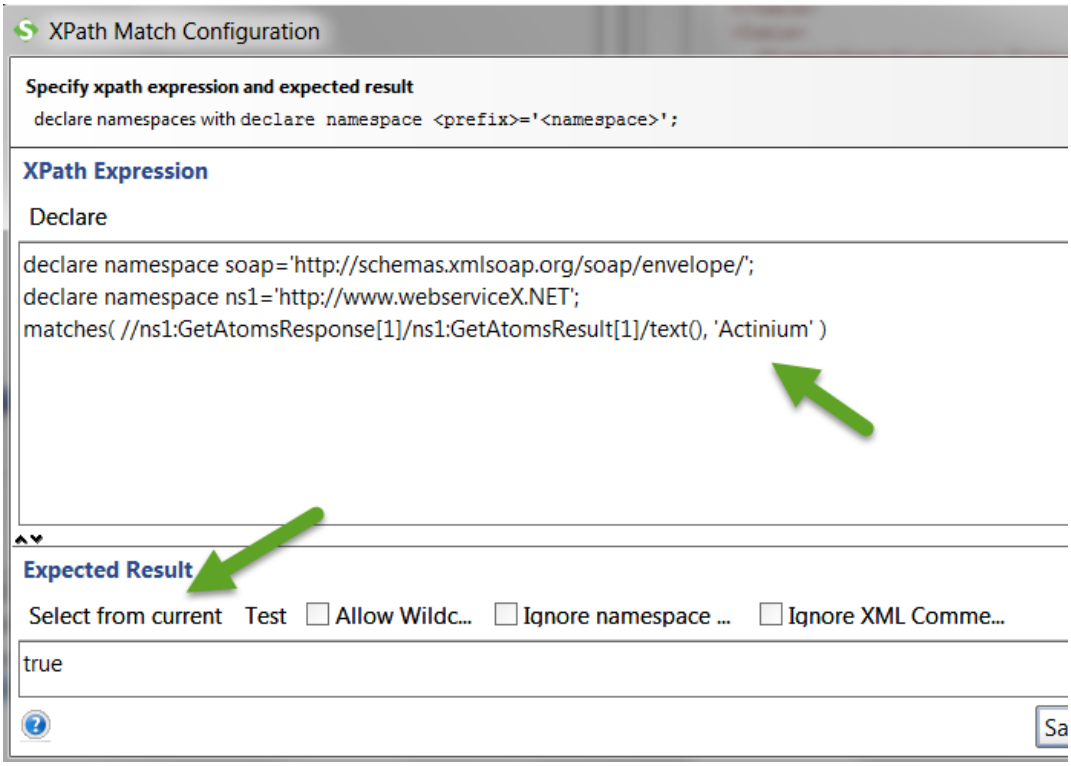
c. Not contains “Sweden”

d. XPath Match

a. Click Declare



b. Write the Expression



c. Click on "Select from current"

---

7) Create Soap Request as "GetAtomicNumber"

a. Select "periodictableSoap -> GetAtomicNumber" method

b. Set "Actinium" in Request

c. Run the "Test Step" and Verify the response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xm
   <soap:Body>
      <GetAtomicNumberResponse xmlns="http://www.webserviceX.NET">
         <GetAtomicNumberResult><![CDATA[<NewDataSet>
  <Table>
    <AtomicNumber>89</AtomicNumber>
    <ElementName>Actinium</ElementName>
    <Symbol>Ac</Symbol>
    <AtomicWeight>227</AtomicWeight>
    <BoilingPoint>3470</BoilingPoint>
    <IonisationPotential>6.94</IonisationPotential>
    <EletroNegativity>1</EletroNegativity>
    <AtomicRadius>1.8800000000000001</AtomicRadius>
    <MeltingPoint>1323</MeltingPoint>
    <Density>10070</Density>
  </Table>
</NewDataSet>]]></GetAtomicNumberResult>
      </GetAtomicNumberResponse>
   </soap:Body>
</soap:Envelope>
```


8) Set Assertion

   a. Valid http status code

   b. Contains "Ac"

   c. Not contains "AF"

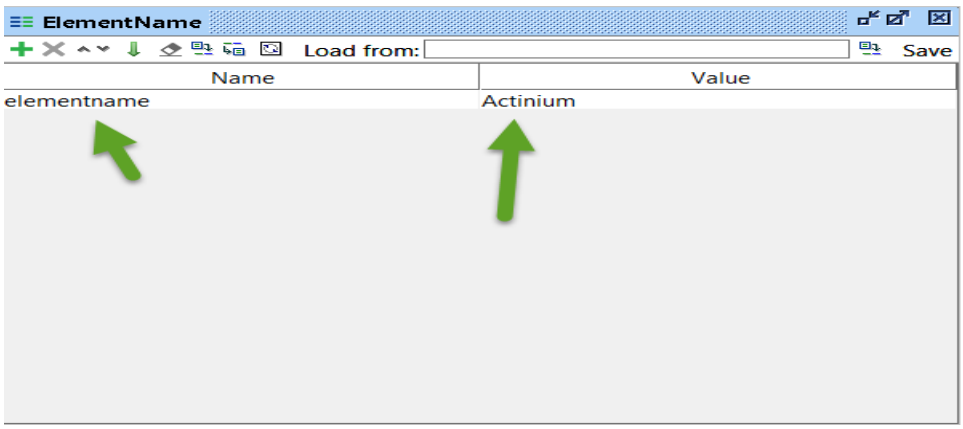
---

9) Properties

   a. Right click Insert Step -> Properties

   b. Set Properties as "ElementName"

   c. Set name as "elementname"and value as "Actinium"



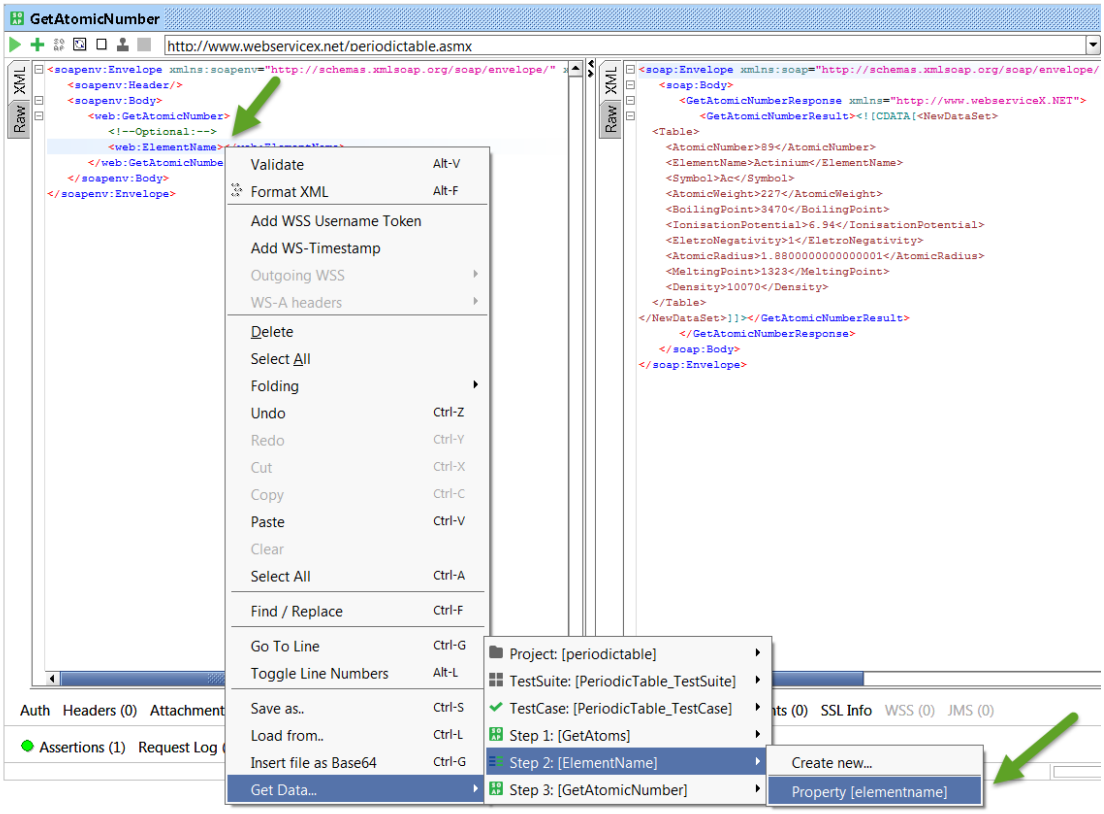
---

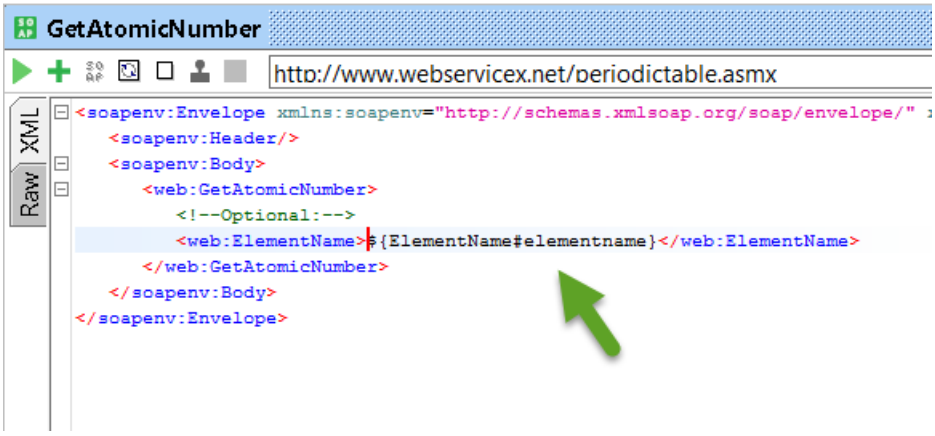
10) Set Properties in “GetAtomicNumber” Request

a. Remove “Actinium” from Request

b. Set Property value in Request

c. Right Click Get Data->ElementName-> Property[elementname]





d. Verify the response

---

10) Set Global Properties

a. Right click Insert Step -> Groovy Script

b. Set name as “Weight”

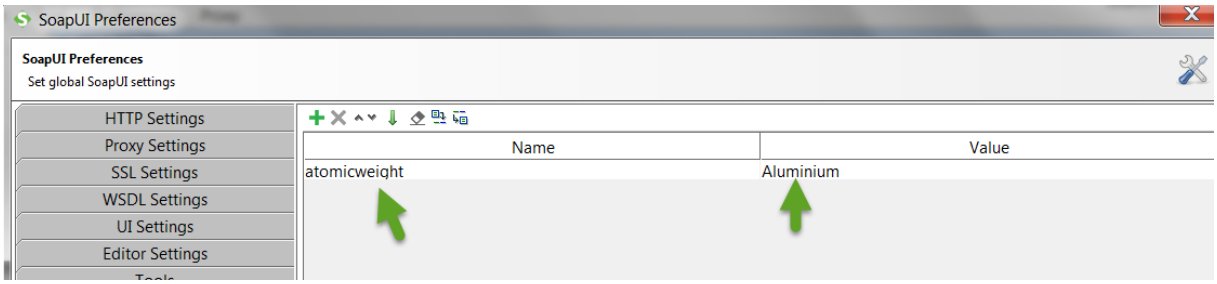
c. Write groovy script

Weight

```
import groovy.xml.*

def atomicweight = "Aluminium"

log.info(atomicweight)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "atomicweight", atomicweight)
```

d. Run the groovy script

e. Verify in “Global Property” File-> Preferences -> Global Properties



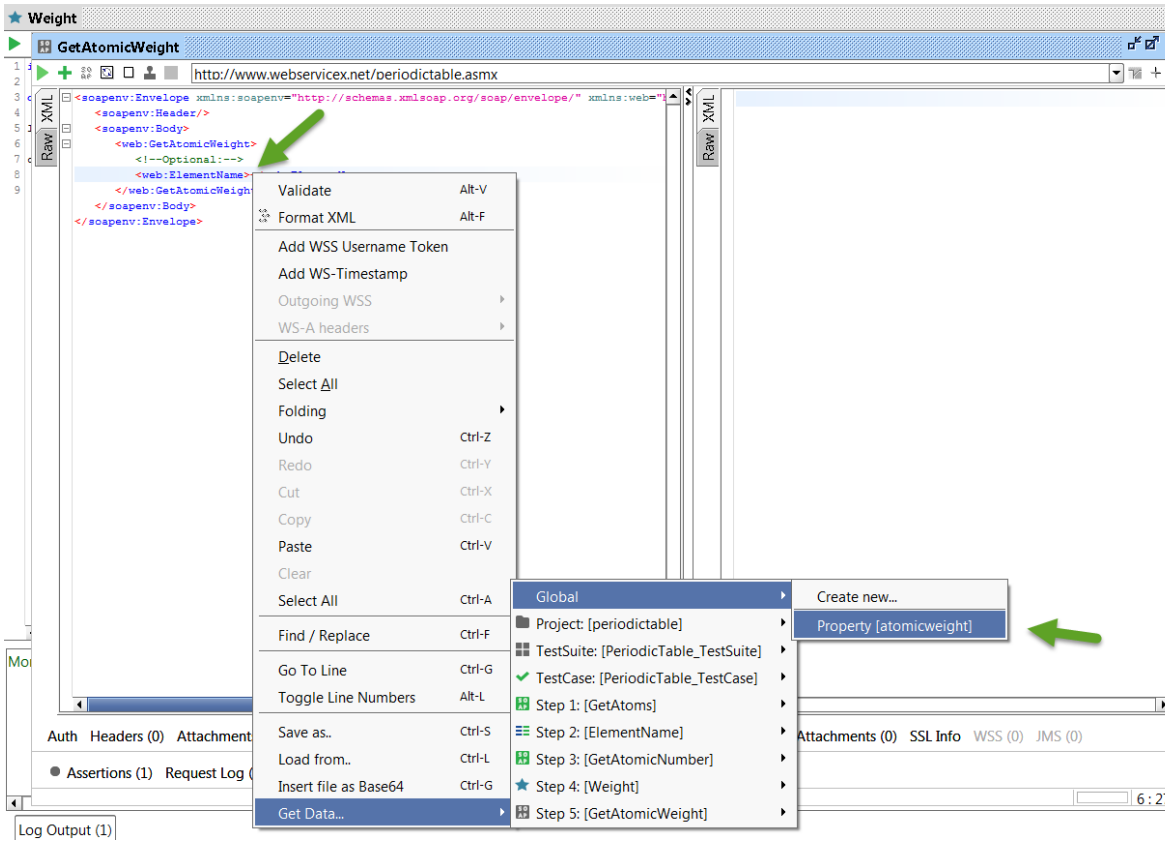
---

11) Create Soap Request as “GetAtomicWeight”

a. Select “periodictableSoap -> GetAtomicWeight” method

b. Set Property value in Request

c. Right Click Get Data->Global-> Property[atomicweight]



d. Run the "Test Step" and Verify the response

12) Set Assertion

a. Valid http status code

b. Contains "26.9815"

c. Not contains "29.00"

---

13) Fetch value from "GetAtoms" Response

a. Right click Insert Step -> Groovy Script

b. Set name as "FetchResponse_ Atoms"

c. Fetch ElementName as "Antimony"

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
  <soap:Body>
    <GetAtomsResponse xmlns="http://www.webserviceX.NET">
      <GetAtomsResult><![CDATA[<NewDataSet>
<Table>
  <ElementName>Actinium</ElementName>
</Table>
<Table>
  <ElementName>Aluminium</ElementName>
</Table>
<Table>
  <ElementName>Americium</ElementName>
</Table>
<Table>
  <ElementName>Antimony</ElementName>
</Table>
```

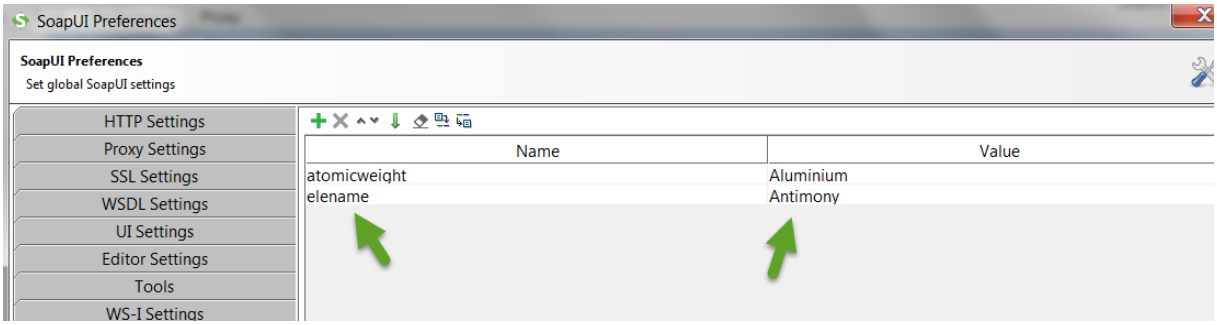
d. Write groovy script

FetchResponse_ Atoms

```
import groovy.xml.*
import groovy.util.XmlParser
import groovy.util.XmlSlurper

def responseContent = testRunner.testCase.getTestStepByName("GetAtoms").getPropertyValue("Response")

def response = new XmlSlurper().parseText(responseContent).Body.GetAtomsResponse.GetAtomsResult.text()

def elename = new XmlSlurper().parseText(response).Table.ElementName[3]

log.info(elename)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "elename", elename.toString())
```

e. Run the groovy script

f. Verify in "Global Property" File-> Preferences -> Global Properties



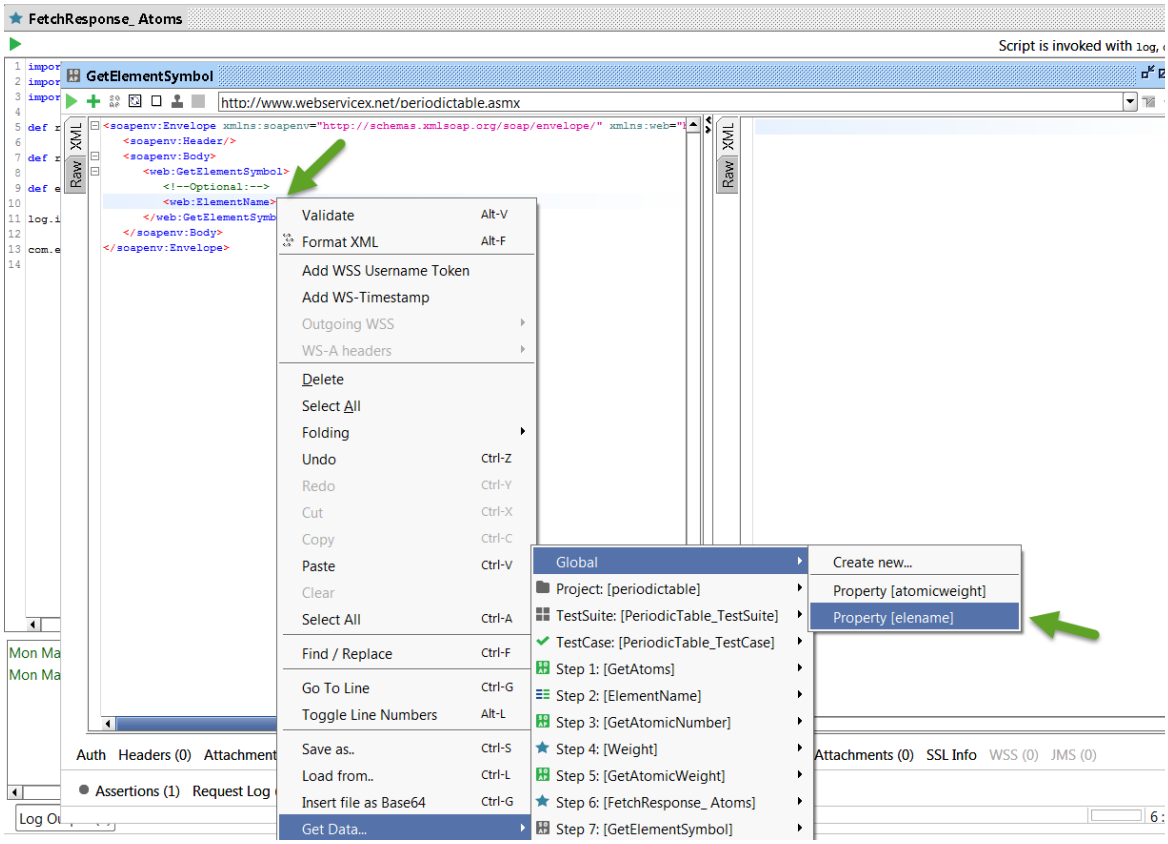
---

14) Create Soap Request as “GetElementSymbol”

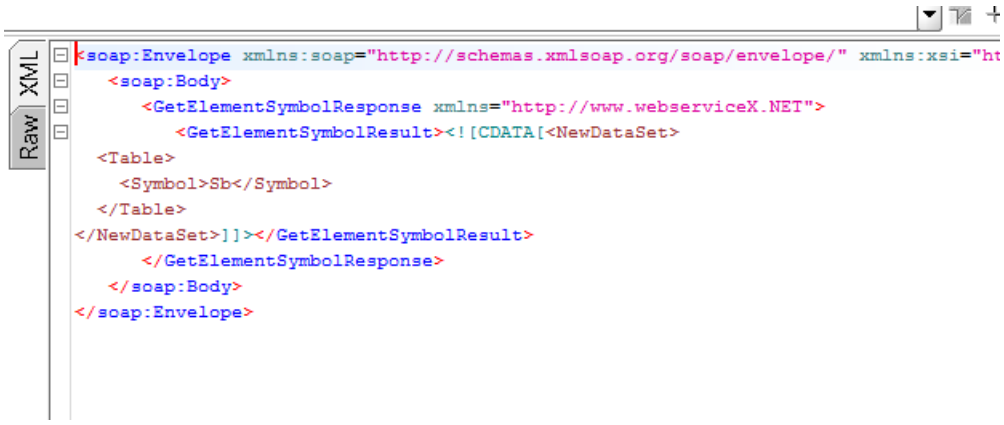
a. Select “periodictableSoap -> GetElementSymbol” method

b. Set Property value in Request

c. Right Click Get Data->Global-> Property[elename]



d. Run the “Test Step” and Verify the response



15) Set Assertion

a. Valid http status code

b. Contains “Sb”

c. Not contains “AF”

----------End of SOAP Project----------

2) REST

1) Open Soapui Tool

2) File -> Import Project (google-maps-soapui-project.xml)

3) Create “TestSuite” as “Geocoding API _TestSuite”

4) Create “TestCase” as “Geocoding API _TestCase”

5) Right click add “REST” Test Step as “GET - Sample Request”

   a. Select “Geocoding API -> GET-> Sample Request” method

   b. Run and Verify the response


```
{
   "results": [   {
      "address_components":       [
                  {
            "long_name": "1600",
            "short_name": "1600",
            "types": ["street_number"]
         },
                  {
            "long_name": "Amphitheatre Parkway",
            "short_name": "Amphitheatre Pkwy",
            "types": ["route"]
         },
                  {
            "long_name": "Mountain View",
            "short_name": "Mountain View",
            "types":              [
               "locality",
               "political"
            ]
         },
                  {
            "long_name": "Santa Clara County",
            "short_name": "Santa Clara County",
            "types":              [
               "administrative_area_level_2",
               "political"
            ]
         },
                  {
            "long_name": "California",
            "short_name": "CA",
            "types":              [
               "administrative_area_level_1",
               "political"
            ]
         },
```

6) Set Assertion

a. Valid http status code as “200”

b. Contains “California”

c. Not contains “Sweden”

---

7) Right click add “REST” Test Step as “GET - Region Biasing Sample”

a. Select “Geocoding API -> GET-> Region Biasing Sample” method

b. Run and Verify the response


```
{
   "results": [   {
      "address_components":       [
                  {
            "long_name": "Toledo",
            "short_name": "Toledo",
            "types":             [
               "locality",
               "political"
            ]
         },
                  {
            "long_name": "Toledo",
            "short_name": "Toledo",
            "types":             [
               "administrative_area_level_4",
               "political"
            ]
         },
                  {
            "long_name": "Vega de Toledo",
            "short_name": "Vega de Toledo",
            "types":             [
               "administrative_area_level_3",
               "political"
            ]
         },
                  {
            "long_name": "Toledo",
            "short_name": "TO",
            "types":             [
               "administrative_area_level_2",
               "political"
            ]
         },
```


8. Set Assertion

a. Valid http status code as “200”

b. Contains “Spain”

c. Not contains “Sweden”

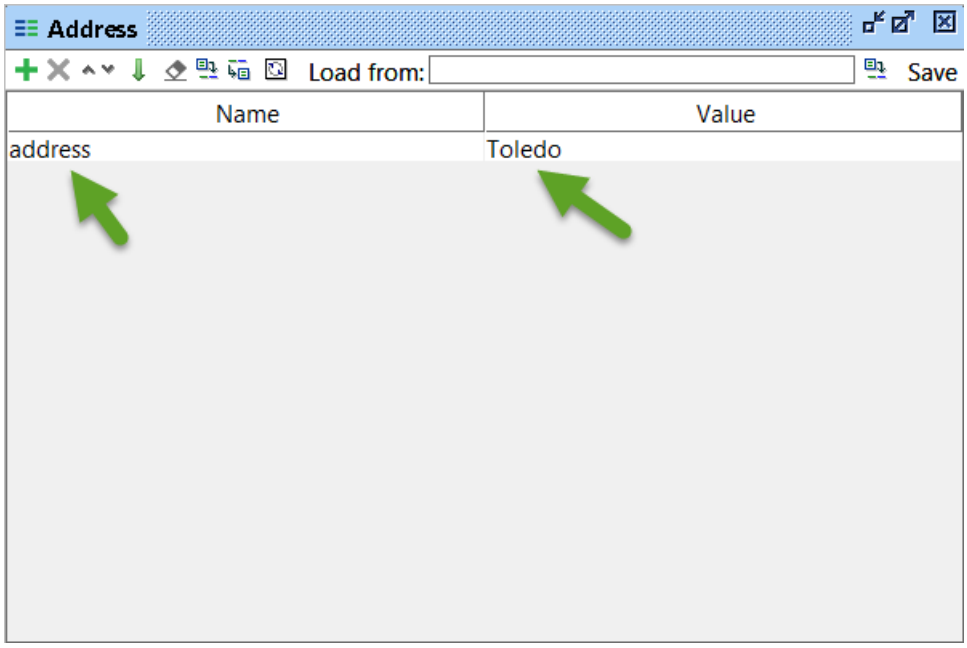
---

9. Properties

a. Right click Insert Step -> Properties

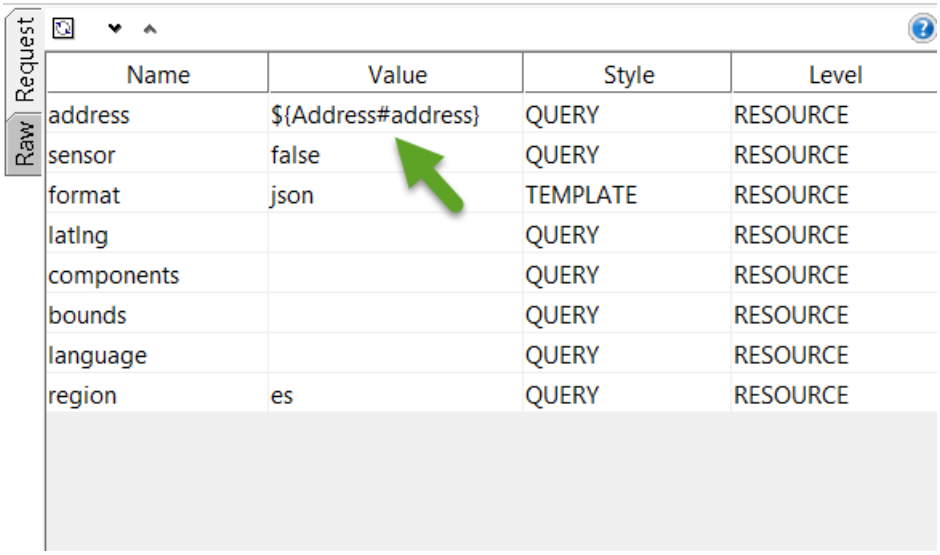
b. Set Properties name as “Address”

c. Inside the Properties Set name as “address” and value as “Toledo”



---

11) Set Properties in “GET - Region Biasing Sample”



---

12) Right click add REST Test Step as “GET - Reverse Lookup Sample”

a. Select “Direction API -> GET-> GET - Reverse Lookup Sample” method

b. Run and Verify the response


```
{
   "results":    [
            {
         "address_components":          [
                        {
               "long_name": "277",
               "short_name": "277",
               "types": ["street_number"]
            },
                        {
               "long_name": "Bedford Avenue",
               "short_name": "Bedford Ave",
               "types": ["route"]
            },
                        {
               "long_name": "Williamsburg",
               "short_name": "Williamsburg",
               "types":                [
                  "neighborhood",
                  "political"
               ]
            },
                        {
               "long_name": "Brooklyn",
               "short_name": "Brooklyn",
               "types":                [
                  "sublocality_level_1",
                  "sublocality",
                  "political"
               ]
            },
```


13) Set Assertion

a. Valid http status code as “200”

b. Contains “Brooklyn”

c. Not contains “Stockholm”

---

14) Set Global Properties

a. Right click Insert Step -> Groovy Script

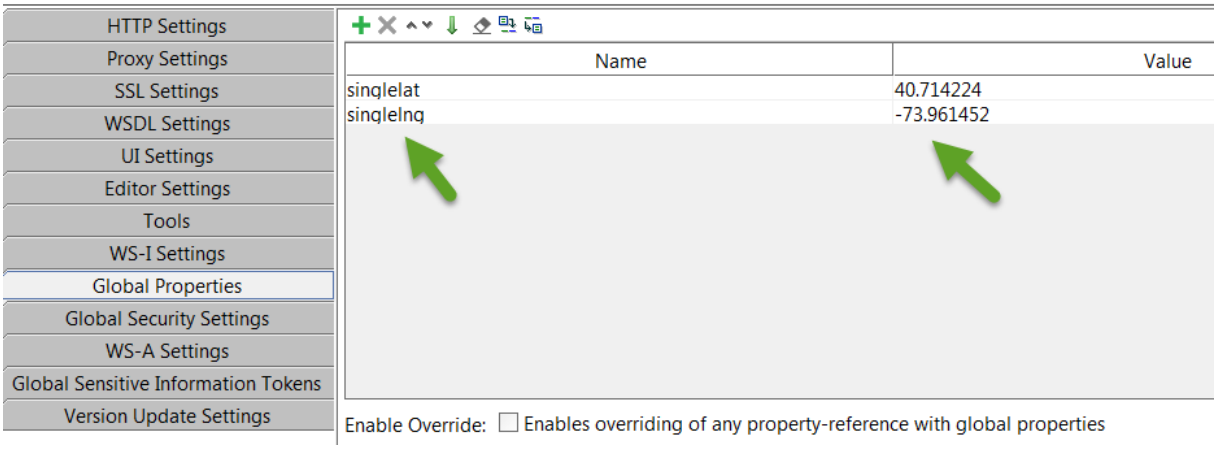
b. Set name as “Geocoding API”

c. Write groovy script

```
import groovy.xml.*

def singlelat = "40.714224"

log.info(singlelat)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "singlelat", singlelat)

def singlelng = "-73.961452"

log.info(singlelat)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue( "singlelng", singlelng)
```

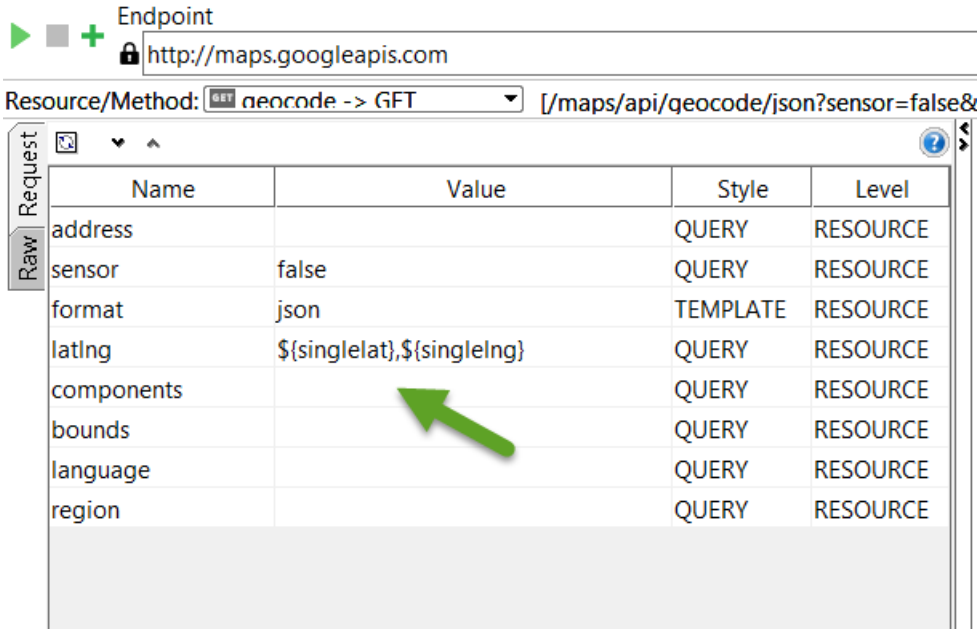
d. Run the groovy script

e. Verify in “Global Property” File-> Preferences -> Global Properties



---

15) Set Properties in “GET - Reverse Lookup Sample” Test Step



---

16) Right click add REST Test Step as “GET - Viewport Biasing Sample”

a. Select “Direction API -> GET -> Viewport Biasing Sample” method

b. Run and Verify the response

```
{
   "results": [   {
      "address_components":      [
                  {
            "long_name": "Winnetka",
            "short_name": "Winnetka",
            "types":              [
               "neighborhood",
               "political"
            ]
         },
                  {
            "long_name": "Los Angeles",
            "short_name": "Los Angeles",
            "types":              [
               "locality",
               "political"
            ]
         },
                  {
            "long_name": "Los Angeles County",
            "short_name": "Los Angeles County",
            "types":              [
               "administrative_area_level_2",
               "political"
            ]
         },
                  {
            "long_name": "California",
            "short_name": "CA",
            "types":              [
               "administrative_area_level_1",
               "political"
            ]
         },
```


17) Set Assertion

a. Valid http status code as “200”

b. Contains “Winnetka”

c. Not contains “Stockholm”

---

18) Fetch value from “GET - Reverse Lookup Sample” Response

a. Right click Insert Step -> Groovy Script

b. Set name as “FetchResponse_Reverse”

c. Fetch lat and lng value for both “northeast” and “southewst”


```
            "geometry":          {
               "location":             {
                  "lat": 40.714232,
                  "lng": -73.9612889
               },
```

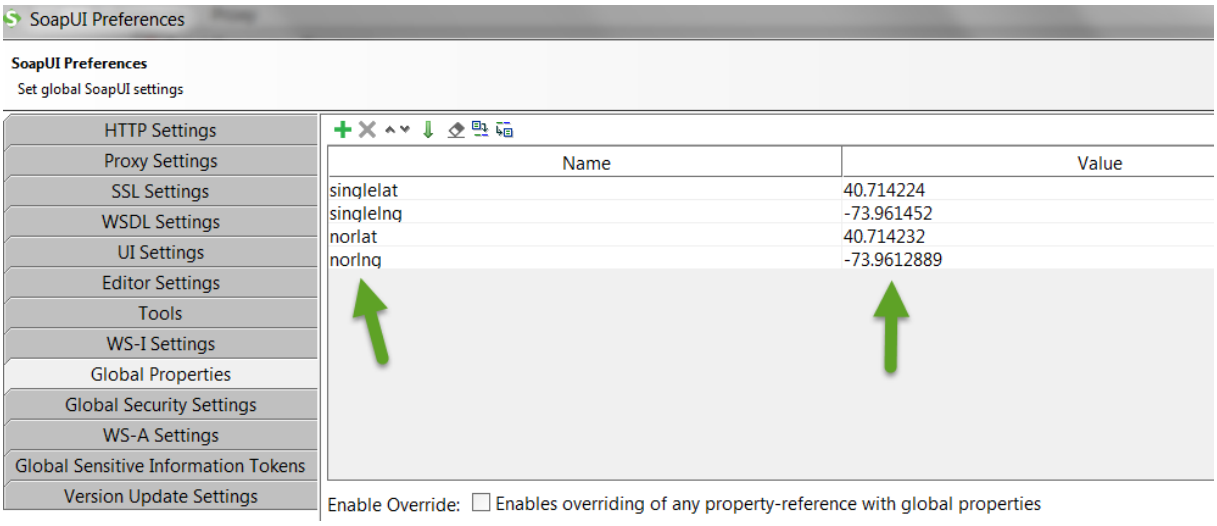

d. Write groovy script

FetchResponse_Reverse

```
import groovy.json.JsonSlurper
import groovy.json.JsonBuilder
import groovy.json.JsonOutput

def responseContent = testRunner.testCase.getTestStepByName("GET - Reverse Lookup Sample").getPropertyValue("response")
def response = new JsonSlurper().parseText(responseContent)
log.info(response)

def norlat = new JsonOutput().toJson(response.results.geometry.location.lat.get(0));

log.info (norlat)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue('norlat', norlat);


def norlng = new JsonOutput().toJson(response.results.geometry.location.lng.get(0));

log.info (norlng)

com.eviware.soapui.SoapUI.globalProperties.setPropertyValue('norlng', norlng);
```

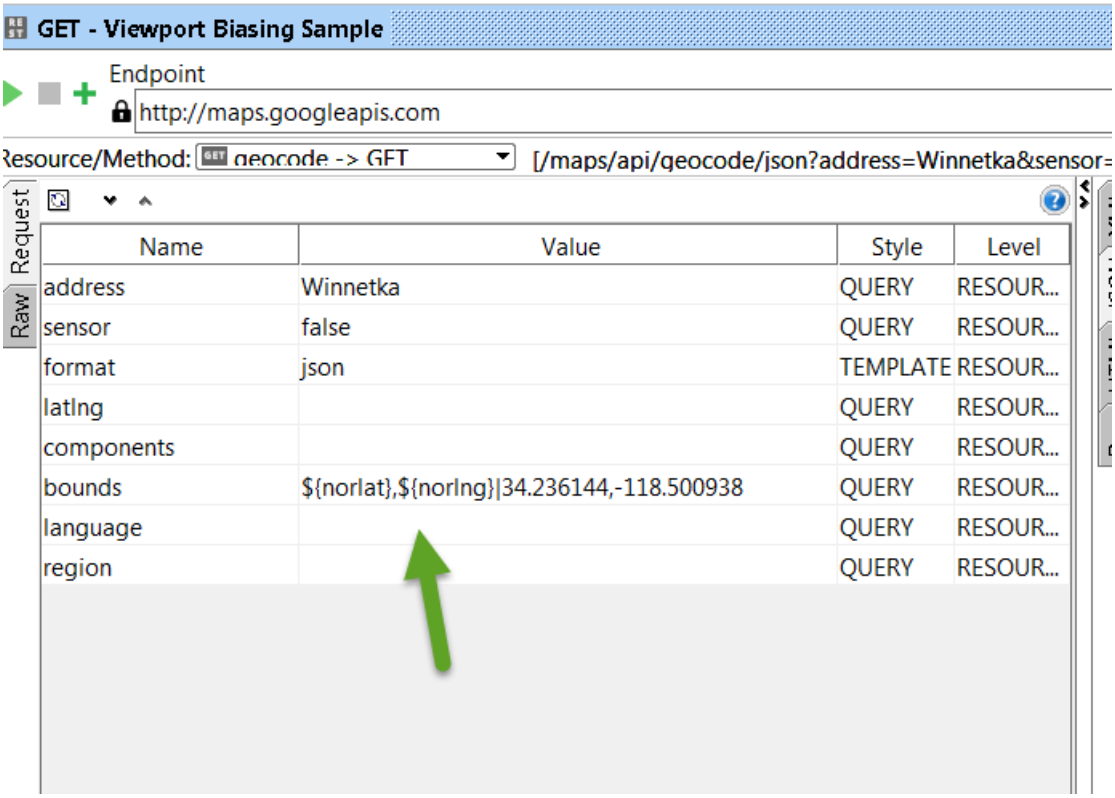
e. Run the groovy script

f. Verify in “Global Property” File-> Preferences -> Global Properties



---

19) Set Properties in “GET - Viewport Biasing Sample” Test Step



----------End of REST Project----------

3) GitHub

1) Upload both SOAP and REST Project in existing repository from last class assignment. Check in the repository should contains POM file

a. periodictable-soap-project.xml

b. google-maps-rest-project.xml

c. git status

d. git add .

e. git commit –m “new file”

f. git push

---

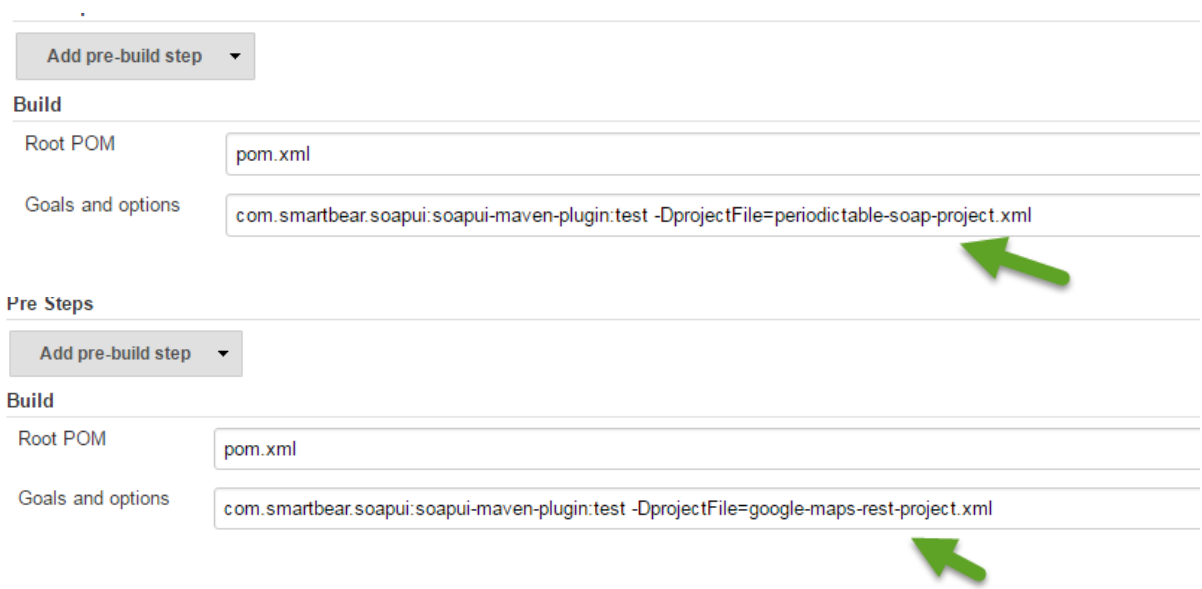
4) Jenkins

2) Create a two project

a. Set Title “Soap – PeriodicTable” for “periodictable-soap-project.xml”

b. Set Title “Rest – GoogleMaps” for “google-maps-rest-project.xml”

c. Change the file name in config according to the project



d. Run the jobs

e. Check Build is Success

f. Go to Console page and click view text

---

5) Save your SOAP and REST Project also Jenkins logs in text format in to the folder and set the folder name as your “Full Name”. Zip the folder before upload in to student portal

6) Upload your Zip folder in to student portal before 16.pm on Monday (23-05-2016)

http://studentportal.nackademin.se/mod/assign/view.php?id=6913