Dinesh. C

MAIL : DINESHMCA.C@gmail.com

INTRoduction To WEB SERVICE using SOAPUI

**Web Service:**

In the below example Red bus needs to communicate with all the travels to make sure seats are available – so to communicated we need common language – there comes the webservice

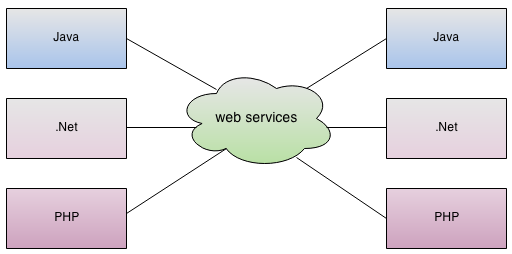
**XML, JSON ::** Praveen Travels (designed using PHP)

**XML, JSON** :**:** KPN Travels (designed using DOTNET)

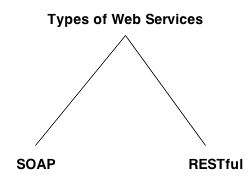
Red bus (designed using Java) **XML, JSON** **XML, JSON ::** ABT Travels (designed using HTML)

**XML, JSON ::** XXX Travels (designed using HTML5)

* Compatibility – Every travels has their own design and language for designing web pages but we use common language like XML to communicate



* In the modern era of technology if you want to build a software application you don't need to build each and everything from scratch. There are lots of readymade services available which you can plug into your application and you can start providing those services in your application.



**SOAP:**

SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.

**Advantages of Soap Web Services**

* WS Security: SOAP defines its own security known as WS Security.
* Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.

**Disadvantages of Soap Web Services**

* Slow: SOAP uses XML format that must be parsed to be read.
* WSDL dependent

**REST:**

REST stands for Representational State Transfer. REST is web standards based architecture and uses HTTP Protocol for data communication.

**Advantages**

* Request sent using by URIs/ global IDs.
* REST uses various representations to represent a resource like text, JSON and XML. (Now a days JSON is the most popular format being used in web services)

**HTTP Methods**

Following well known HTTP methods are commonly used in REST based architecture.

**GET -** Provides a read only access to a resource.

**PUT -** Used to create a new resource.

**DELETE -** Used to remove a resource.

**POST -** Used to update a existing resource or create a new resource.

**OPTIONS -** Used to get the supported operations on a resource.

**Tools:**

* **SOAPUI (Automating using Groovy Script)**
* **SOAPUI Pro**
* **Using JAVA**

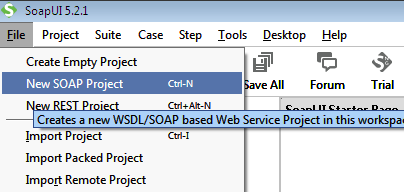
**(REST jar, JSON jar and junit jar) or (HTTP component) or (Integrating SOAPUI using JAVA)**

* **POSTMAP (Manual Testing)**

**SOAPUI Manual Testing:**

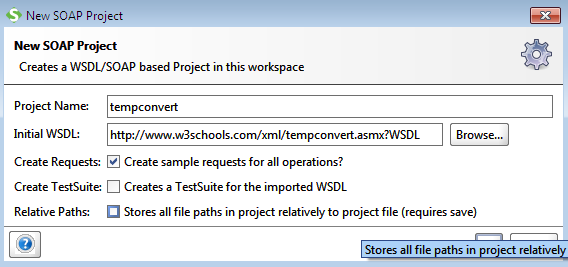
1. Need to install SOAPUI tool – Firewall need to be off
2. WSDL (Web Services Description Language) given by Dev Team/ Service Team

WSDL contains what services can be called, i.e what input value we have to provide and what will be the format of the response it is going to generate for each kind of service.

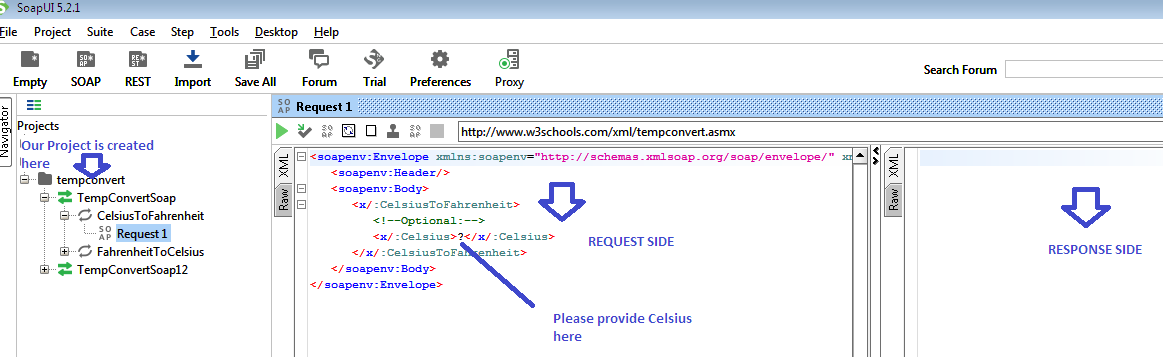
****

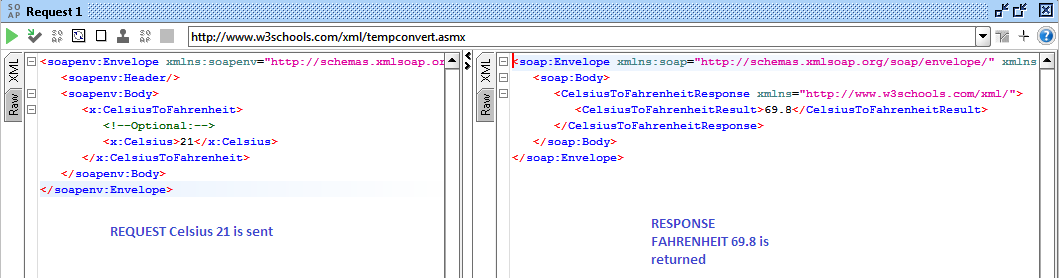
WSDL – I am taking free WSDL file temperature converter from w3c school website

<http://www.w3schools.com/xml/tempconvert.asmx?WSDL>

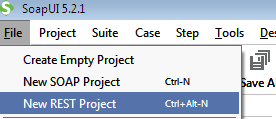


Give wsdl and project name and click ok



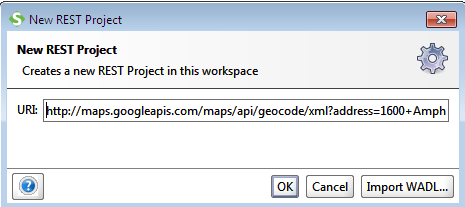
****

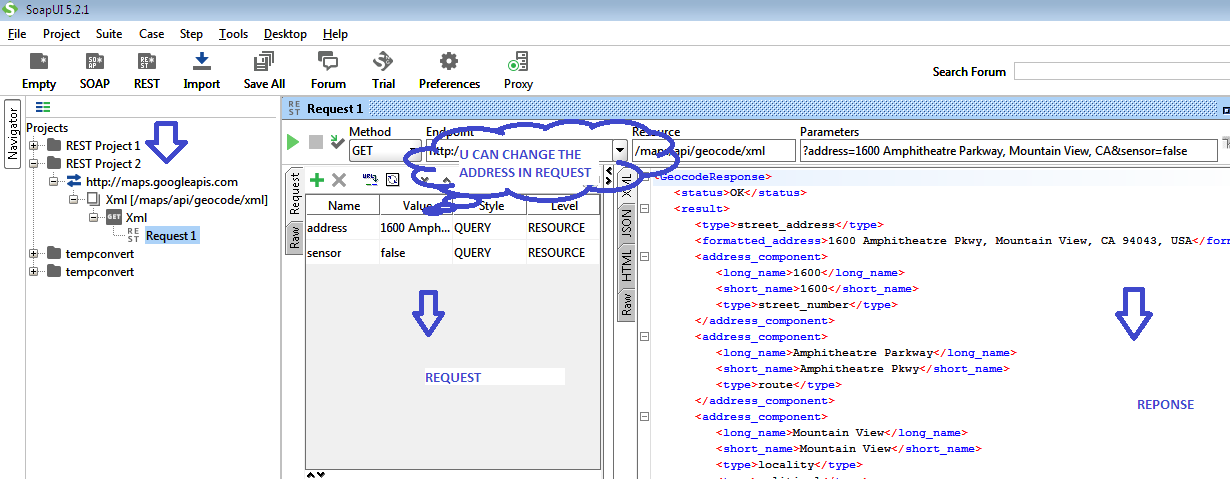
**REST** **Manual Testing:**



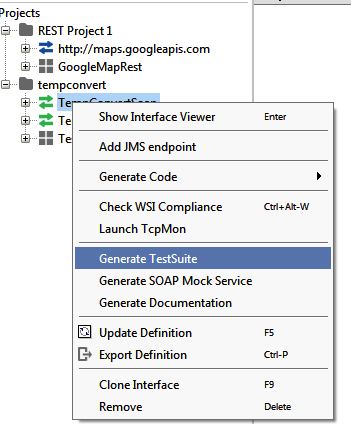
URI – is required where we can send the request using URI and we will be getting the response as XML, JSON based on the dev team design

URI [**http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false**](http://maps.googleapis.com/maps/api/geocode/xml?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false)





**Using Properties and property transfer**



**Level Properties**

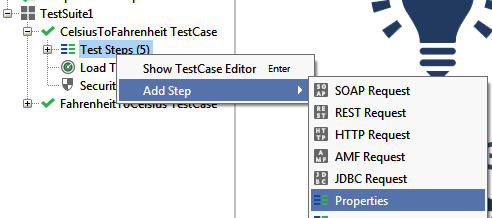
${Property-name} refers to a **Global Property**

${#Project#Property-name}refers to a **Project Property**

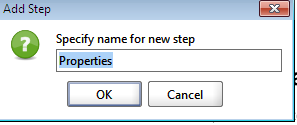
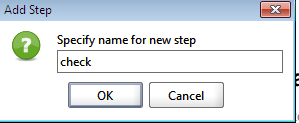
${#TestSuite#Property-name} refers to a **TestSuite Property**

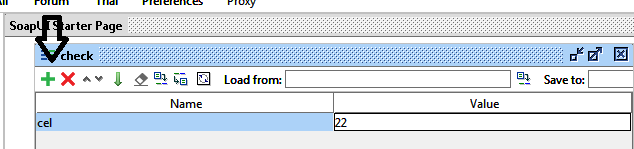
${#TestCase#Property-name} refers to a **TestCase Property**

${TestStep-name#Property-name} refers to a property in a named **TestStep.**



${TestStep-name#Property-name} refers to a property in a named **TestStep.**

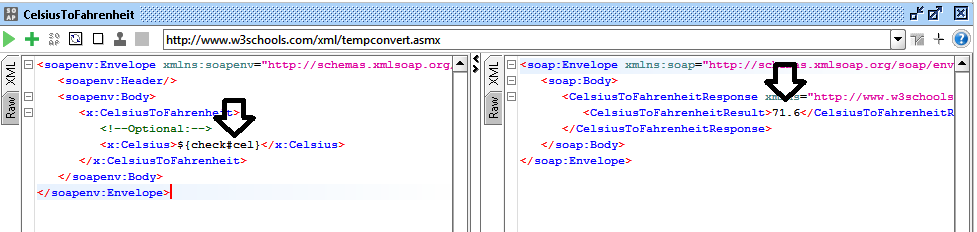


**Syntax**

${TestStep-name#Property-name} refers to a property in a named **TestStep.**

**Our Coding**

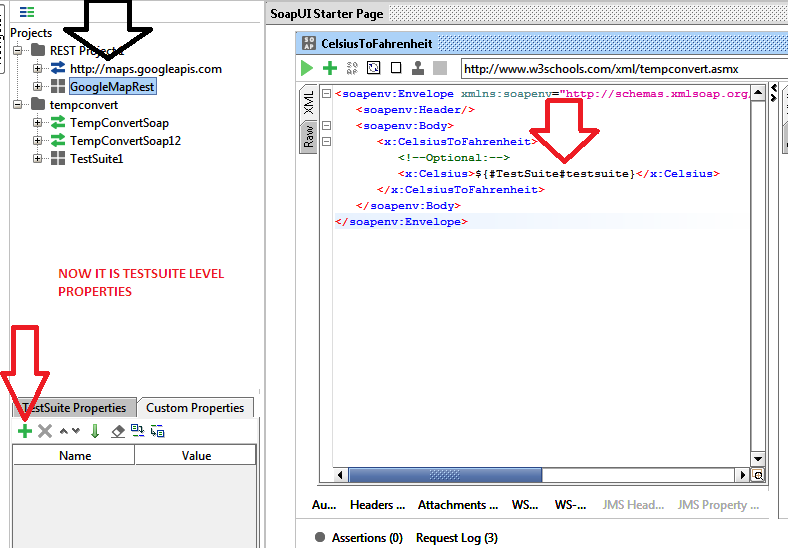
${check#cel}



${#Project#Property-name}refers to a **Project Property**

${#TestSuite#Property-name} refers to a **TestSuite Property**

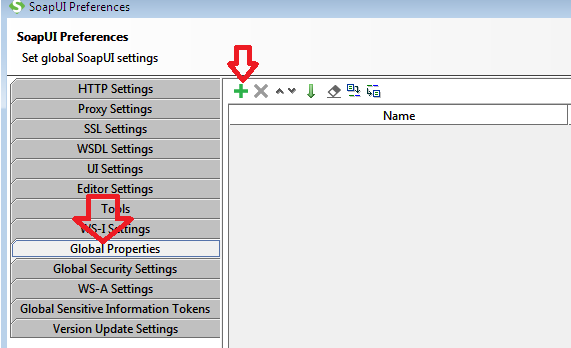
${#TestCase#Property-name} refers to a **TestCase Property**



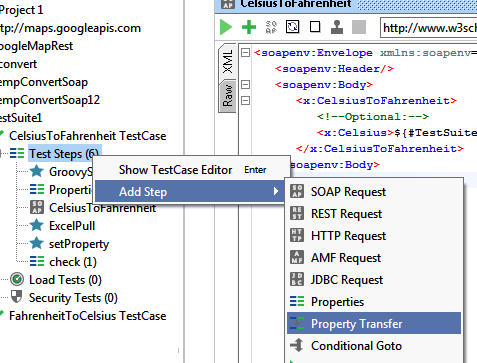
So we have to click test case, project or testsuite and add properties for the corresponding requirement.

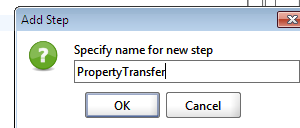
For Global LEVEL Properties::

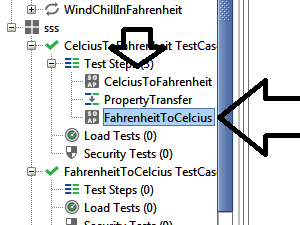
File –> Preference



Property Transfer







**Drag and drop the request of ‘fahrenheitToCelcius’ to property transfer Test Step**