QUALITEST
software testing experts

# Introduction to SoapUI

QualiTest

Think Globally ▪ Test Locally

# Follow Along with the Slides

Slides are located on the Shared Drive at: \\nsx0862-dm08\AED-FTD\

QUALITEST
software testing experts

# Who are We?

Jamie Mitchell

- Senior Test Specialist and QA Manager
- Architected, Developed and Managed a number of projects with SoapUI
- Over 20+ years experience in Testing – eCommerce, finance, healthcare, entertainment, defense, legal, travel, agriculture, education, state registries & databases, utilities, etc…

# Introductions (ctd…)

Brian Van Stone

- Senior Test Specialist
- Three years experience automating business processes across varied infrastructure
  - Load Balancers, Web Infrastructure, Windows/Unix/Linux platforms, Desktop and Web Applications, Source Control Systems

# About QualiTest

- **World's 2nd largest pure play independent testing company**
  (Nelson Hall - 2012)
- **QA & Testing focus**
  QA and testing is all that we do!
- **SLA based Managed Testing services (MTS)**
  Reducing costs, Raising quality, Driving continuous improvement
- **Right Shore! - Global Service delivery Model**
  Onsite, Onshore, Near-Shore and Offshore
- **Client centric solutions - Outcome-based testing and pricing**
  Client only pays for testing delivered

QUALITEST software testing experts

# Partial Customer List

# Training Agenda

- Day 1 – Intro to Protocols & Tool
- Day 2 – Data Driven Testing & Assertions
- Day 3 – Groovy scripting & Refactoring
- Day 4 & 5 – Advanced Topics & Testing Multiple Protocols

# Getting to Know You

- Who are you?
- What's your background?
- What do you do?
- What is your experience with SoapUI?
- What is your experience with testing web services?

# Course Objectives

- Strong understanding of Web Services (SOAP & REST)
- Imparting the importance of Test Structure, Test Verification and Test Project Management (inputs, environments, results)
- A firm understanding of SoapUI.

# Key Lessons

- Basic Test Creation
- Transformation of Keyword Tests to scripted tests.
- Basic SoapUI Functionality
- Basic Managing a Suite of Tests
- A solid understanding of testing web services.

# Course Format

- This is meant to be a learn along type environment.
- The class structure will consist of introductory lessons coupled with on screen demonstration
- Individual Exercises will be interspersed.
- Questions are ***Strongly Encouraged!!***

Introduction to Web Services

# GETTING STARTED

# What are Web services

- A method of communicating between two devices
- A software function provided at a network address over the web with the service always on
- It has an interface described in a machine-processable format

# Some Examples

Google

how to file a patent

Web Videos News Shopping Images More Search tools

About 112,000,000 results (0.25 seconds)

**File** Your **Patent** Today - **Patent** Your Work in 3 Easy Steps
Ad www.legalzoom.com/
Featured by Popular Science and CNN
LegalZoom has 1,123 followers on Google+
Patents Made Easy
3 Easy Steps
Online Patent Search
TMs, Patents & Copyrights

How To **Patent** Your Idea - InventionHome.com
Ad www.**patents**.inventionhome.com/
Get A Free **Patent** & Invention Kit. 3 Easy Steps To **Patent** & **Make** Money

**File a Patent** - Flat $149 - Get "**Patent** Pending" In 1 Day
Ad www.thoughtstopaper.com/
Full Service. No Fill-In-The-Blanks
Provisional Patent App - Client Reviews - Patent Search

Ads

How To **File A Patent**
www.ipprocurement.com/**FilePatent**
Experienced **Patent** Attorney For **Patent** Search, Drafting & Filing.

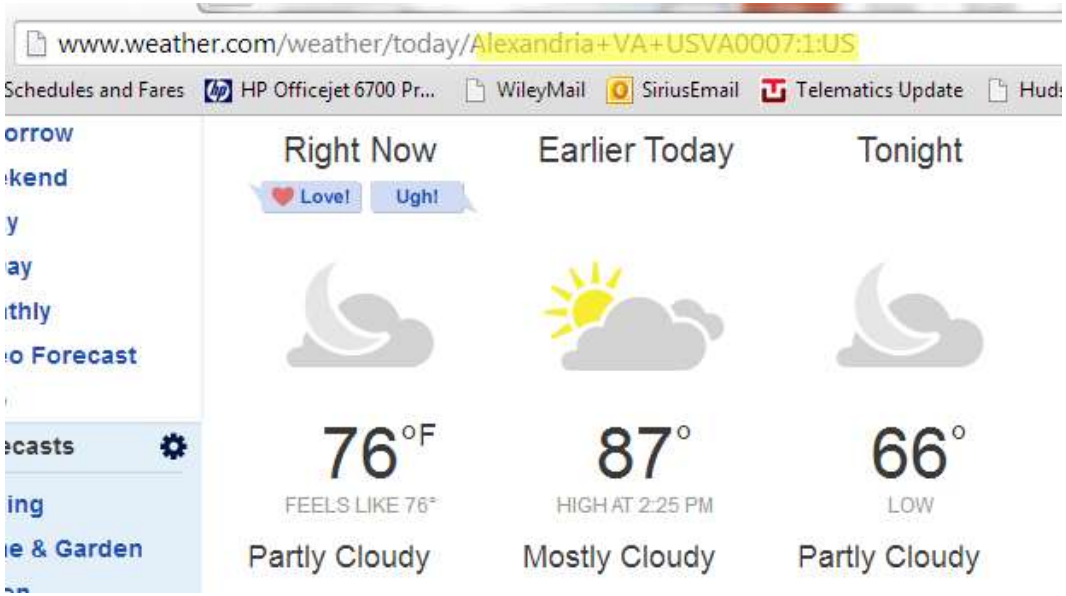
**Patent** Your Invention
www.idea2invention.com/
Learn How to Get Started with Our Free **Patent** & Invention Kit

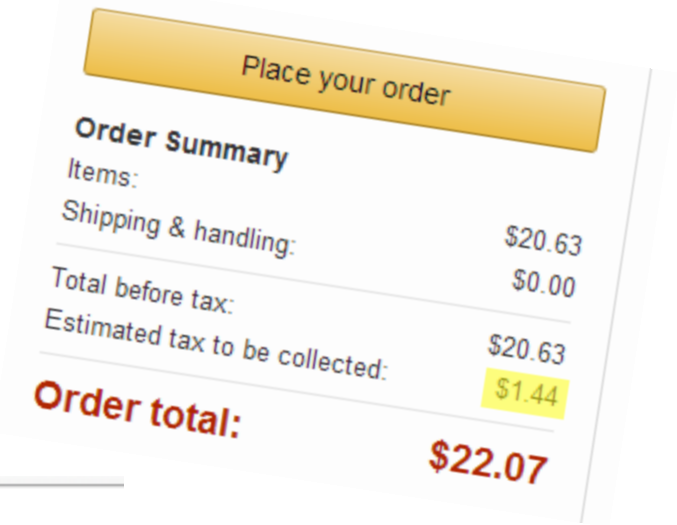
How To **File A Patent**
www.inventionideas.org/
Have an Idea that Needs a **Patent**? Request Free Invention **Patent** Kit!

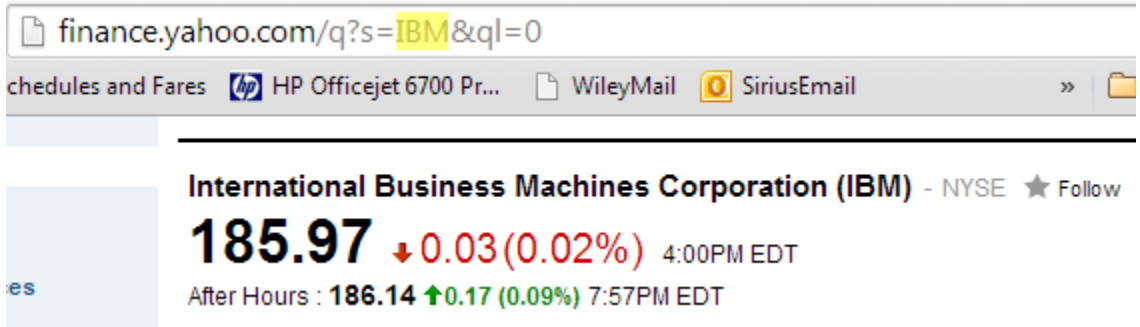


QUALITEST software testing experts

# More examples



# Protocols

- There are multiple protocols and transport mechanisms for web services- this class will focus on:
  - SOAP
  - REST
- Before looking at these web services, we need a basic understanding of XML and XML Namespaces

# XML

- XML stands for eXtensible Markup Language
- XML is designed to transport and store data, not to display data
- XML tags are not predefined
- XML is designed to be self-descriptive
- XML does not DO anything.
  - It is just information wrapped in tags
  - Need software to send, receive or display it.

# XML examples

```
▼<CATALOG>
 ▼<CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
 ▼<CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
 ▼<CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
 ▼<CD>
    <TITLE>Still got the blues</TITLE>
    <ARTIST>Gary Moore</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>Virgin records</COMPANY>
    <PRICE>10.20</PRICE>
    <YEAR>1990</YEAR>
  </CD>
```

```
▼<CATALOG>
 ▼<PLANT>
    <COMMON>Bloodroot</COMMON>
    <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$2.44</PRICE>
    <AVAILABILITY>031599</AVAILABILITY>
  </PLANT>
 ▼<PLANT>
    <COMMON>Columbine</COMMON>
    <BOTANICAL>Aquilegia canadensis</BOTANICAL>
    <ZONE>3</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$9.37</PRICE>
    <AVAILABILITY>030699</AVAILABILITY>
  </PLANT>
 ▼<PLANT>
    <COMMON>Marsh Marigold</COMMON>
    <BOTANICAL>Caltha palustris</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Sunny</LIGHT>
    <PRICE>$6.81</PRICE>
    <AVAILABILITY>051799</AVAILABILITY>
  </PLANT>
```

# Another Example

Login is parent

```
<soapenv:Envelope xmlns:soapenv="http://schemas.
    <soapenv:Header/>
    <soapenv:Body>
        <urn:Login>
            <urn:UserName>?</urn:UserName>
            <urn:Password>?</urn:Password>
        </urn:Login>
    </soapenv:Body>
</soapenv:Envelope>
```

start tag

end tag

# XML Syntax Rules

- All XML Elements Must Have a Closing Tag
  - This is incorrect.
  - This is a correct.
  -  &  are also correct
- XML Tags are Case Sensitive
  - <**M**essage>This is incorrect</**m**essage>
  - <message>This is correct</message>
- XML Elements Must be Properly Nested
  - This is incorrect
  - This is bold correct

# XML Rules (cont)

- XML Documents Must Have a Root Element

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

- Only the characters "<" and "&" are strictly illegal in XML
- <!-- This is a comment -->
- White-space is Preserved in XML

# Elements and Attributes

- Bookstore has an element book
- Book has an attribute – category and elements: title, author, year, price

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

# XML Namespaces

- XML Namespaces provide a method to avoid element name conflicts and to provide for reusability of elements

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:us:gov:uspto:patent:palm:utility:v1.0"
xmlns:urn1="urn:us:gov:uspto:enterprise"
xmlns:urn2="urn:us:gov:uspto:patent:palm:utility"
xmlns:urn3="urn:us:gov:uspto:patent:palm">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:ProcessIDSInformationRequest>
         <urn1:Authentication>
            <urn1:Password>cid:438827620531</urn1:Password>
            <!--Optional:-->
            <urn1:RequestorName>?</urn1:RequestorName>
            <urn1:Login>?</urn1:Login>
         </urn1:Authentication>
         <urn1:ServiceHeader>
            <urn1:ServiceName>?</urn1:ServiceName>
            <urn1:ServiceVersionNumber>?</urn1:ServiceVersionNumber>
            <urn1:RequestorSystemIdentifier>?</urn1:RequestorSystemIdentifier>
            <!--Optional:-->
            <urn1:RequestorName>?</urn1:RequestorName>
         </urn1:ServiceHeader>
         <!--1 or more repetitions:-->
         <urn:IdsInformation>
            <urn2:IDSIdentifierSeqNo>?</urn2:IDSIdentifierSeqNo>
            <urn3:ApplicationNumber>?</urn3:ApplicationNumber>
         </urn:IdsInformation>
      </urn:ProcessIDSInformationRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

# Why do we care ?

- Errors in XML documents will stop your XML applications
- Many of the errors that we find in testing have to do with
  - Namespace issues
  - Schema data types
  - Case
  - Order
  - Whitespaces

# XPath

- XPath (the XML Path language) is a language for finding information in an XML document
- XPath is a syntax for defining parts of an XML document
- XPath uses path expressions to navigate in XML documents
- XPath contains a library of standard functions

# XPath examples

```
<bookstore>

<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>

<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>

<book category="WEB">
  <title lang="en">XQuery K
  <author>James McGovern</a
  <author>Per Bothner</auth
  <author>Kurt Cagle</autho
  <author>James Linn</autho
  <author>Vaidyanathan Naga
  <year>2003</year>
  <price>49.99</price>
</book>

<book category="WEB">
  <title lang="en">Learning
  <author>Erik T. Ray</auth
  <year>2003</year>
  <price>39.95</price>
</book>

</bookstore>
```

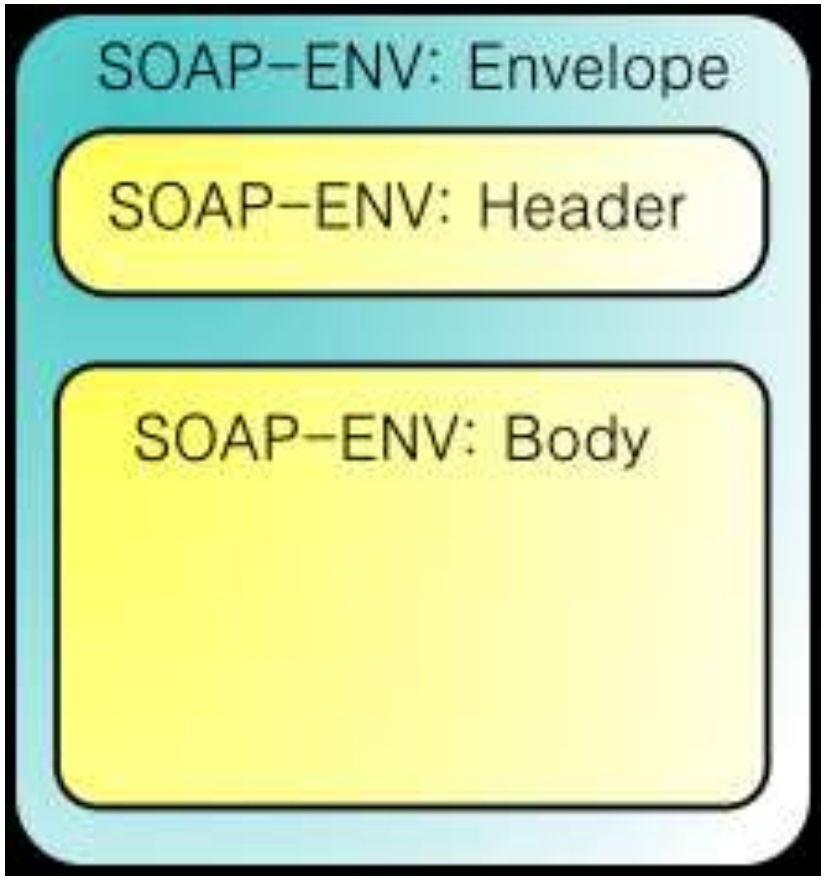
| XPath Expression | Result |
|---|---|
| /bookstore/book[1] | Selects the first book element that is the child of the bookstore element |
| /bookstore/book[last()] | Selects the last book element that is the child of the bookstore element |
| /bookstore/book[last()-1] | Selects the last but one book element that is the child of the bookstore element |
| /bookstore/book[position()<3] | Selects the first two book elements that are children of the bookstore element |
| //title[@lang] | Selects all the title elements that have an attribute named lang |
| //title[@lang='eng'] | Selects all the title elements that have an attribute named lang with a value of 'eng' |
| /bookstore/book[price>35.00] | Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00 |

# SOAP

- SOAP stands for Simple Object Access Protocol
- SOAP is an XML based protocol for accessing Web Services.
- SOAP is based on XML

# SOAP

- A SOAP message is an ordinary XML document containing the following elements:



# SOAP Elements

| Element | Description | Required |
|---|---|---|
| Envelope | Identifies the XML document as a SOAP message. | Yes |
| Header | Contains header information. | No |
| Body | Contains call, and response information. | Yes |
| Fault | Provides information about errors that occurred while processing the message. | No |

# Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" x
   <soapenv:Header/>
   <soapenv:Body>
      <urn:GetIDSInformationRequest>
         <urn1:Authentication>
            <!--Optional:-->
            <urn1:Password>cid:972012163448</urn1:Password>
            <!--Optional:-->
            <urn1:RequestorName>?</urn1:RequestorName>
            <!--Optional:-->
            <urn1:Login>?</urn1:Login>
         </urn1:Authentication>
         <urn1:ServiceHeader>
            <urn1:ServiceName>?</urn1:ServiceName>
            <urn1:ServiceVersionNumber>?</urn1:ServiceVersionNumber>
            <!--Optional:-->
            <urn1:RequestorSystemIdentifier>?</urn1:RequestorSystemIdentifier
            <!--Optional:-->
            <urn1:RequestorName>?</urn1:RequestorName>
            <urn1:RequestorSystemName>?</urn1:RequestorSystemName>
         </urn1:ServiceHeader>
         <urn2:ApplicationNumber>?</urn2:ApplicationNumber>
      </urn:GetIDSInformationRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

operation name

body

# SOAP Syntax Rules

- Must be encoded using XML
- Must use a SOAP envelope namespace
  - xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
- Must not contain XML processing instructions

# SOAP requests defined by WSDL

- WSDL stands for Web Services Description Language
- WSDL is an XML-based language for describing Web services.
- WSDL is written in XML
- WSDL is also used to locate Web services

# WSDL Document Structure

```
<definitions>

<types>
  data type definitions........
</types>

<message>
  definition of the data being communicated....
</message>

<portType>
  set of operations......
</portType>

<binding>
  protocol and data format specification....
</binding>

</definitions>
```

| Element | Description |
|---|---|
| <types> | A container for data type definitions used by the web service |
| <message> | A typed definition of the data being communicated |
| <portType> | A set of operations supported by one or more endpoints |
| <binding> | A protocol and data format specification for a particular port type |

# WSDL Example

```
<wsdl:definitions targetNamespace="http://ws.cdyne.com/WeatherWS/" xmlns:s="http://www.w3.org/2(
   <wsdl:types>
      <s:schema elementFormDefault="qualified" targetNamespace="http://ws.cdyne.com/WeatherWS/">
         <s:element name="GetWeatherInformation">
            <s:complexType/>
         </s:element>
         <s:element name="GetWeatherInformationResponse">
            <s:complexType>
               <s:sequence>
                  <s:element minOccurs="0" maxOccurs="1" name="GetWeatherInformationResult" type
               </s:sequence>
            </s:complexType>
         </s:element>
         <s:complexType name="ArrayOfWeatherDescription">
            <s:sequence>
               <s:element minOccurs="0" maxOccurs="unbounded" name="WeatherDescription" type="tr
            </s:sequence>
         </s:complexType>
         <s:complexType name="WeatherDescription">
            <s:sequence>
               <s:element minOccurs="1" maxOccurs="1" name="WeatherID" type="s:short"/>
               <s:element minOccurs="0" maxOccurs="1" name="Description" type="s:string"/>
               <s:element minOccurs="0" maxOccurs="1" name="PictureURL" type="s:string"/>
            </s:sequence>
         </s:complexType>
```

# Operation Types

| Type | Definition |
|---|---|
| One-way | The operation can receive a message but will not return a response |
| Request-response | The operation can receive a request and will return a response |
| Solicit-response | The operation can send a request and will wait for a response |
| Notification | The operation can send a message but will not wait for a response |

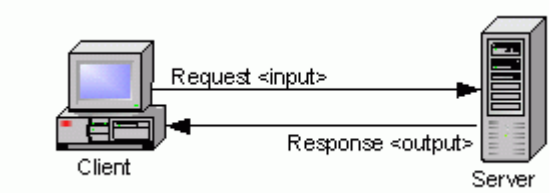
No output – one way

```
<message name="newTermValues">
  <part name="term" type="xs:string"/>
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="setTerm">
    <input name="newTerm" message="newTermValues"/>
  </operation>
</portType >
```

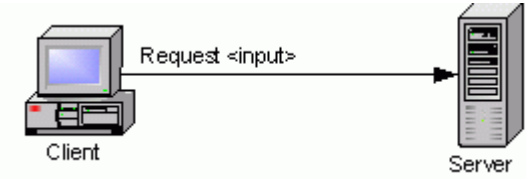
```
<message name="getTermRequest">
  <part name="term" type="xs:string"/>
</message>

<message name="getTermResponse">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTermRequest"/>
    <output message="getTermResponse"/>
  </operation>
</portType>
```

# Operation Types

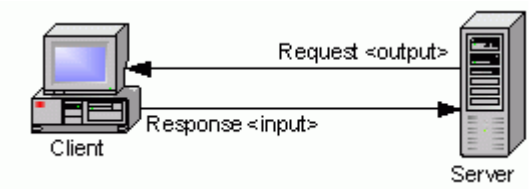
- Request <> Response




- One Way




- Solicit < > Response




- Notification




QUALITEST software testing experts

# REST

- REpresentational State Transfer (REST)
- Resources are manipulated using a fixed set of four operations -create, read, update, delete : PUT, GET, POST, and DELETE.
  - PUT creates or updates a new resource, which can be then deleted by using DELETE.
  - GET retrieves the current state of a resource (read)
  - POST transfers a new state onto a resource (create).

# Using PUT & POST

- Either PUT or POST can be used to create a new resource (depends on how your service is defined)
  - For example creating a new chapter of MPEP (Manual of Patent Examining Procedure)
  - POST /manual/MPEP/current
  - PUT /manual/MPEP/current/d0e55397.xml
- If you want to update an existing document on the server, then use PUT

# Examples

- URI: uniform resource identifiers (string of characters to identify resource)
- Customer # 18 info:
  - http://www.thomas-bayer.com/sqlrest/CUSTOMER/18/
- List of invoices
  - http://www.thomas-bayer.com/sqlrest/INVOICE/
- List of products
  - http://www.thomas-bayer.com/sqlrest/PRODUCT/

# REST RESPONSES

- PRODUCT/30

http://www.thomas-bayer.com/sqlrest/PRODUCT

```
<?xml version="1.0"?>
- <PRODUCT xmlns:xlink="http://www.w3.org/1999/xlink">
    <ID>30</ID>
    <NAME>Shoe Iron</NAME>
    <PRICE>23.2</PRICE>
  </PRODUCT>
```

- INVOICE/14

http://www.thomas-bayer.com/sqlrest/INVOICE/1 thomas-bayer.com

```
<?xml version="1.0"?>
- <INVOICE xmlns:xlink="http://www.w3.org/1999/xlink">
    <ID>14</ID>
    <CUSTOMERID xlink:href="http://www.thomas-bayer.com/sqlrest/CUSTOMER/35/">35</CUSTOMERID>
    <TOTAL>4940.40</TOTAL>
  </INVOICE>
```

# REST > WADL

- Sometimes a REST service has documented information in a WADL
- WADL = Web Application Description Language
- XML in format
- Not Standardized

# Sample WADL

**URI:**

https://tmog.uspto.gov/eOG/search/all/info.json?issues=2014-07-05&pubReason=OPPOSITION&limit=20&orderBy=SERIAL_NR&order=ASC

https://tmog.uspto.gov.wadl

```
1  <application xmlns="http://wadl.dev.java.net/2009/02">
2     <doc xml:lang="en" title="https://tmog.uspto.gov"/>
3     <resources base="https://tmog.uspto.gov">
4        <resource path="eOG/search/all/info.json" id="Info.json">
5           <doc xml:lang="en" title="Info.json"/>
6           <method name="GET" id="Info.json">
7              <doc xml:lang="en" title="Info.json"/>
8              <request>
9                 <param name="issues" type="xs:string" required="false" default="" style="query" xmlns:xs="
10                <param name="pubReason" type="xs:string" required="false" default="" style="query" xmlns:x
11                <param name="limit" type="xs:string" required="false" default="" style="query" xmlns:xs="h
12                <param name="orderBy" type="xs:string" required="false" default="" style="query" xmlns:xs=
13                <param name="order" type="xs:string" required="false" default="" style="query" xmlns:xs="h
14             </request>
15          </method>
16       </resource>
17    </resources>
18 </application>
```

# Differences

- SOAP REQUEST

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
 <soap:body pb="http://www.acme.com/phonebook">
  <pb:GetUserDetails>
   <pb:UserID>12345</pb:UserID>
  </pb:GetUserDetails>
 </soap:Body>
</soap:Envelope>
```

- SAME AS A REST REQUEST

```
http://www.acme.com/phonebook/UserDetails/12345
```

# Differences

| Attribute | SOAP | REST |
|---|---|---|
| Language/Platform | Language, platform, and transport independent | Only http |
| Environments | Works well in distributed enterprise environments | Requires point-to-point communication |
| Standardization | Standardized | Has some standards |
| Error handling | Built in error-handling | None |
| Format | Requires XML | Efficient – can use any format |
| Speed | Slower | Fast (no extensive processing required) |

# SOAP & REST Responses

- REST doesn’t have to use XML to provide the response. REST-based Web services can output the data in
  - Command Separated Value (CSV)
  - JavaScript Object Notation (JSON) and
  - Really Simple Syndication (RSS)
- Often when you find a REST service, one of the inputs is the requested format

# REST Example

- http://api.wunderground.com/api/*74e36bb7a98f6a55*/conditions/q/CA/San_Francisco.json
- Json Response


```
{
  "response": {
  "version": "0.1",
  "termsofService": "http://www.wunderground.com/weather/api/d/terms.h
  "features": {
  "conditions": 1
  }
  },
  "current_observation": {
  "image": {
  "url": "http://icons-ak.wxug.com/graphics/wu2/logo_130x80.png",
  "title": "Weather Underground",
  "link": "http://www.wunderground.com"
  },
  "display_location": {
  "full": "San Francisco, CA",
  "city": "San Francisco",
  "state": "CA",
  "state_name": "California",
  "country": "US",
  "country_iso3166": "US",
  "zip": "94101",
```


- XML response


```
▼<response>
  <version>0.1</version>
  ▼<termsofService>
    http://www.wunderground.com/weather/api/d/terms.html
  </termsofService>
  ▼<features>
    <feature>conditions</feature>
  </features>
  ▼<current_observation>
    ▼<image>
      <url>http://icons.wxug.com/graphics/wu2/logo_130x80.png</url>
      <title>Weather Underground</title>
      <link>http://www.wunderground.com</link>
    </image>
    ▼<display_location>
      <full>San Francisco, CA</full>
      <city>San Francisco</city>
      <state>CA</state>
      <state_name>California</state_name>
      <country>US</country>
      <country_iso3166>US</country_iso3166>
      <zip>94101</zip>
      <magic>1</magic>
      <wmo>99999</wmo>
      <latitude>37.77500916</latitude>
      <longitude>-122.41825867</longitude>
      <elevation>47.00000000</elevation>
    </display_location>
```

Introduction to SOAPUI

# USING SOAPUI

# Introduction to SOAPUI

- Create projects
- Add WSDL from sample web service
- Setup up security settings
- Walk through structure of request and response.
- Discuss faults from failed structure vs data errors.
- Request structure
- Request-level properties
- Validating request inputs
- Creating multiple requests
- Response structure
- Working with endpoints

SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Search Forum

Navigator

Projects

SoapUI Starter Page

**Quick Launch**

**New:**

Create new project

Discover REST APIs PRO

**New from template:**

Create data driven test PRO

**Sample projects**

**With features:**

- SOAP
- REST
- Authentication

**With popular APIs:**

- Amazon
- Facebook
- Flickr
- Google Maps
- Twitter
- YouTube

**Find It Fast**

More features

**Get Certified i SoapUI**

Read More

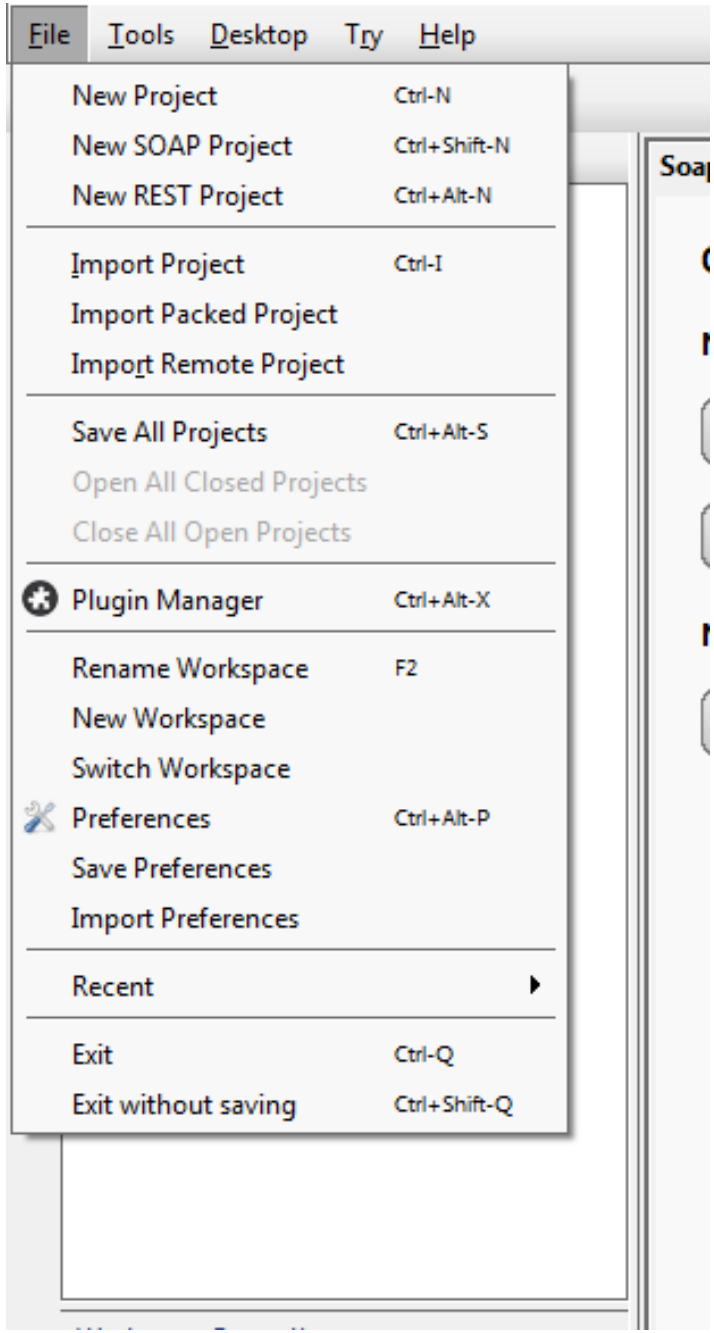
Properties

SoapUI log | http log | jetty log | error log | wsrm log | memory log



QUALITEST software testing experts

# File Menu






QUALITEST software testing experts

SoapUI Preferences

**SoapUI Preferences**

Set global SoapUI settings

- HTTP Settings
- Proxy Settings
- SSL Settings
- WSDL Settings
- UI Settings
- Editor Settings
- Tools
- WS-I Settings
- Global Properties
- Global Security Settings
- WS-A Settings
- Global Sensitive Information Tokens
- Version Update Settings
- SoapUI Pro
- Coverage Settings
- Code Templates
- JDBC Drivers Properties
- Debug Context Filter

| | |
|---|---|
| HTTP Version: | 1.1 |
| User-Agent Header: | |
| Request compression: | None |
| Response compression: | ☑ Accept compressed responses from hosts |
| Disable Response Decompression: | ☐ Disable decompression of compressed responses |
| Close connections after request: | ☐ Closes the HTTP connection after each SOAP request |
| Chunking Threshold: | |
| Authenticate Preemptively: | ☐ Adds authentication information to outgoing request |
| Expect-Continue: | ☐ Adds Expect-Continue header to outgoing request |
| Pre-encoded Endpoints: | ☐ URI contains encoded endpoints, don't try to re-encode |
| Normalize Forward Slashes: | ☐ Replaces duplicate forward slashes in HTTP request endpoints with a single slash |
| Bind Address: | |
| Include request in time taken: | ☑ Includes the time it took to write the request in time-taken |
| Include response in time taken: | ☑ Includes the time it took to read the entire response in time-taken |
| Socket Timeout (ms): | |
| Max response size: | |
| Max Connections Per Host: | |
| Max Total Connections: | 2000 |
| Leave MockEngine: | ☑ Leave MockEngine running when stopping MockServices |
| Enable Mock HTTP Log: | ☐ Logs wire content of all mock requests |
| Start REST MockService: | ☑ Start REST MockService after creation |

OK Cancel

# WSDL Settings

**SoapUI Preferences**

Set global SoapUI settings

- HTTP Settings
- Proxy Settings
- SSL Settings
- WSDL Settings
- UI Settings
- Editor Settings
- Tools
- WS-I Settings
- Global Properties
- Global Security Settings
- WS-A Settings
- Global Sensitive Information Tokens
- Version Update Settings
- SoapUI Pro
- Coverage Settings
- Code Templates
- JDBC Drivers Properties
- Debug Context Filter

| | |
|---|---|
| Cache WSDLs: | ☑ caches and associated WSDLs locally for offline access and improved performance |
| Sample Values: | ☐ generate example values in new requests |
| Type Comment: | ☐ generate comments with type information in new requests |
| Include Optional: | ☑ always include optional schema elements when creating requests |
| Pretty Print: | ☑ pretty print response messages |
| Attachment Parts: | ☐ generate rpc message parts for attachments |
| No Content-Type Validation: | ☐ allow incorrect content-types in mime-attachments |
| Schema Directory: | Browse... |
| Name with Binding: | ☑ uses the WSDL binding name (instead of portType) for imported Interfaces |
| Excluded types: | schema@http://www.w3.org/2001/XMLSchema — Add.. / Edit.. / Remove.. |
| Strict schema types: | ☐ fails schema imports if types/particles are redefined |
| Compression Limit: | |
| Pretty Print Project Files: | ☐ pretty prints project files |
| Trim WSDL: | ☐ Trims leading and trailing whitespaces from WSDL file (might not work on non 8 bit encoding) |

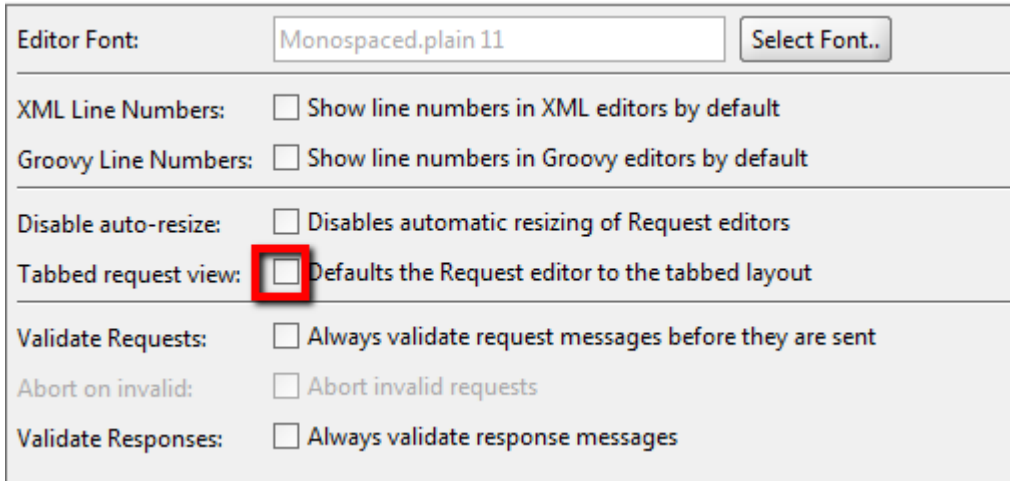
**SoapUI Preferences**

Set global SoapUI settings

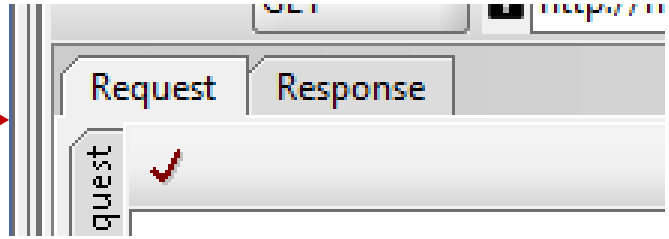
- HTTP Settings
- Proxy Settings
- SSL Settings
- WSDL Settings
- UI Settings
- Editor Settings
- Tools
- WS-I Settings
- Global Properties
- Global Security Settings
- WS-A Settings
- Global Sensitive Information Tokens
- Version Update Settings
- SoapUI Pro
- Coverage Settings
- Code Templates
- JDBC Drivers Properties
- Debug Context Filter

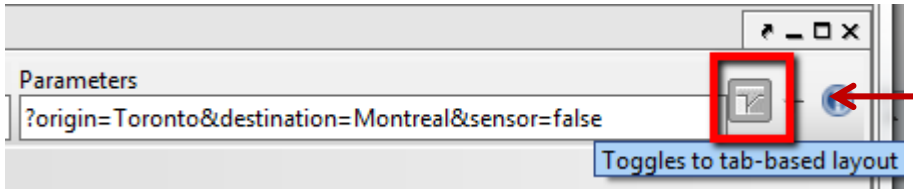
| | |
|---|---|
| Close Projects: | ☐ Close all projects on startup |
| Order Projects: | ☐ Order Projects alphabetically in tree |
| Order Services: | ☐ Order Services alphabetically in tree |
| Order Requests: | ☐ Order Requests alphabetically in tree |
| Show Descriptions: | ☑ Show description content when available |
| Save projects on exit: | ☑ Save all projects on exit |
| Create Backup: | ☐ Backup project files before they are saved |
| Backup Folder: | |
| AutoSave Interval: | 0 |
| Desktop Type: | Tabbed |
| Select most recently used desktop panel on close: | ☐ Show most recently used panel on close (requires restart) |
| Native LF: | ☐ Use native Look & Feel (requires restart) |
| Do not disable Groovy Log: | ☐ Do not disable the groovy log when running load tests |
| Show Log Tabs: | ☐ Shows log tabs when starting SoapUI |
| Show Startup Page: | ☑ Opens startup web page when starting SoapUI |
| Disable Tooltips: | ☐ Disables all tooltips |
| Normalize line-break: | ☐ Normalize line-breaks when saving project |
| Garbage Collection Interval (s): | 60 |
| Size of Raw Response Message to Show: | 10000 |

# Editor Settings




Tabbed request/response



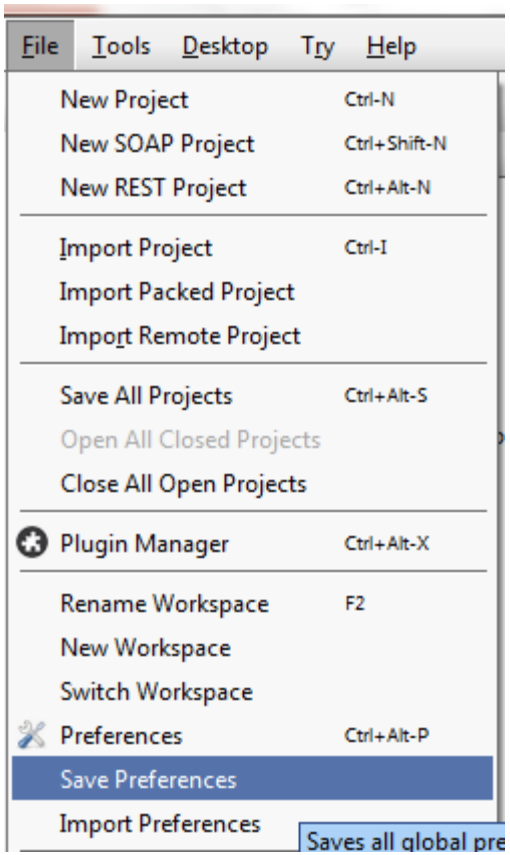

Toggle tabs on/off

# Preferences

- Once done with changes,

Save Preferences

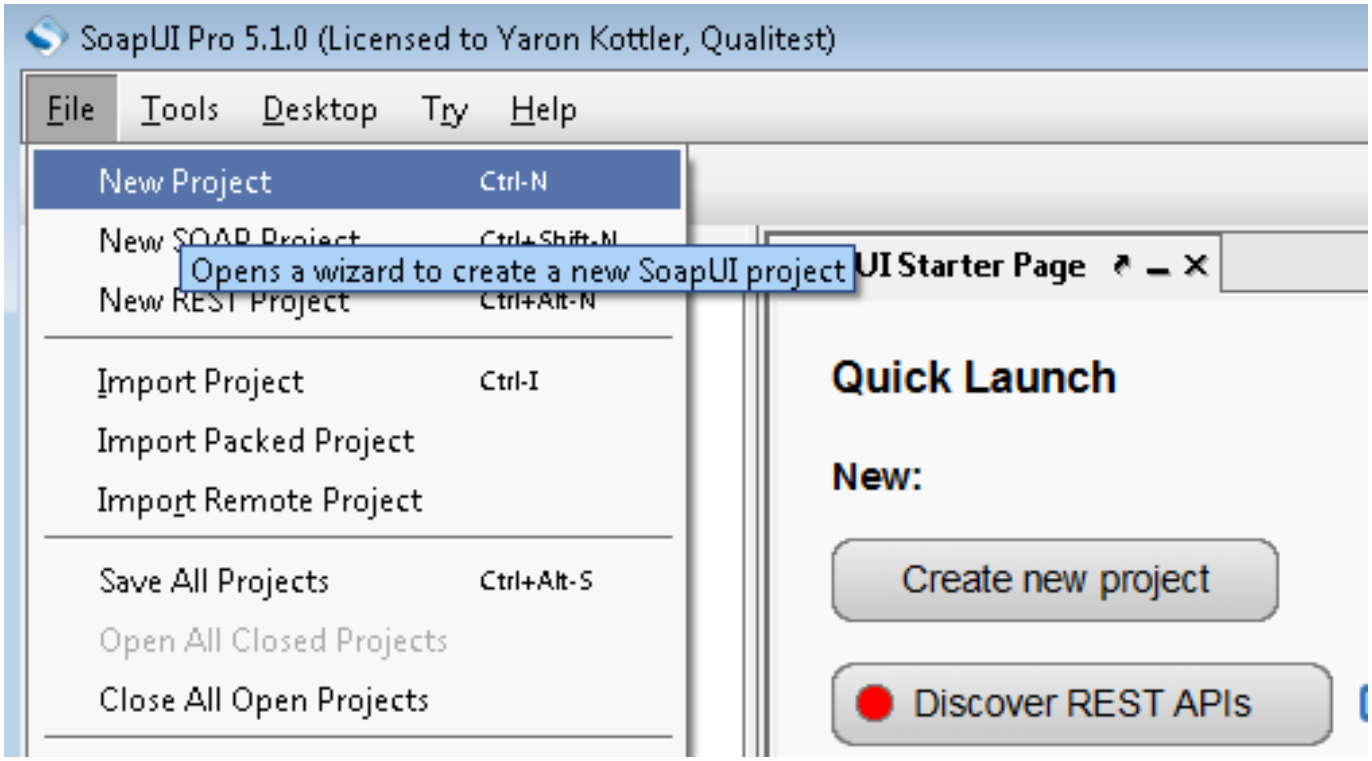
can also import preferences






QUALITEST software testing experts

# Create a New Project



SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Search Forum

Navigator

Projects

SoapUI Starter Page

## Quick Launch

## Sample projects

**With features:**

- SOAP
- REST
- Authentication

**With popular APIs:**

- Amazon
- Facebook
- Flickr
- Google Maps
- Twitter
- YouTube

New Project

**New Project**

Creates a new SoapUI project in this workspace

- (•) Create project from — WSDL definition (SOAP)
- ( ) Create REST project from URI
- ( ) Discover REST APIs using — Internal browser
- ( ) Create empty project

OK Cancel

SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Search Forum

Navigator

Projects

SoapUI Starter Page

**Quick Launch**

**New:**

Create new project

**Sample projects**

**With features:**

SOAP

New SOAP Project

**New SOAP Project**

Creates a WSDL/SOAP based Project in this workspace

Project Name:

Initial WSDL: Browse...

Create Requests: Create sample requests for all operations?

Create TestSuite: Creates a TestSuite for the imported WSDL

Relative Paths: Stores all file paths in project relatively to project file (requires save)

OK Cancel

# Can Create Empty Project at this point

SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Navigator

Projects

SoapUI Starter Page

**Quick Launch**

**New:**

Create new project

**Sample projects**

**With features:**

SOAP

New SOAP Project

**New SOAP Project**

Creates a WSDL/SOAP based Project in this workspace

Project Name: ClassProject_1_V1

Initial WSDL: Browse...

Create Requests: Create sample requests for all operations?

Create TestSuite: Creates a TestSuite for the imported WSDL

Relative Paths: Stores all file paths in project relatively to project file (requires save)

OK Cancel

# Or Add WSDL

http://www.webservicex.com/currencyconvertor.asmx?WSDL

SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Navigator

Projects

SoapUI Starter Page

**Quick Launch**

**New:**

Create new project

**Sample projects**

**With features:**

SOAP

New SOAP Project

**New SOAP Project**

Creates a WSDL/SOAP based Project in this workspace

Project Name: ClassProject_1_V1

Initial WSDL: http://www.webservicex.com/currencyconvertor.asmx?WSDL Browse...

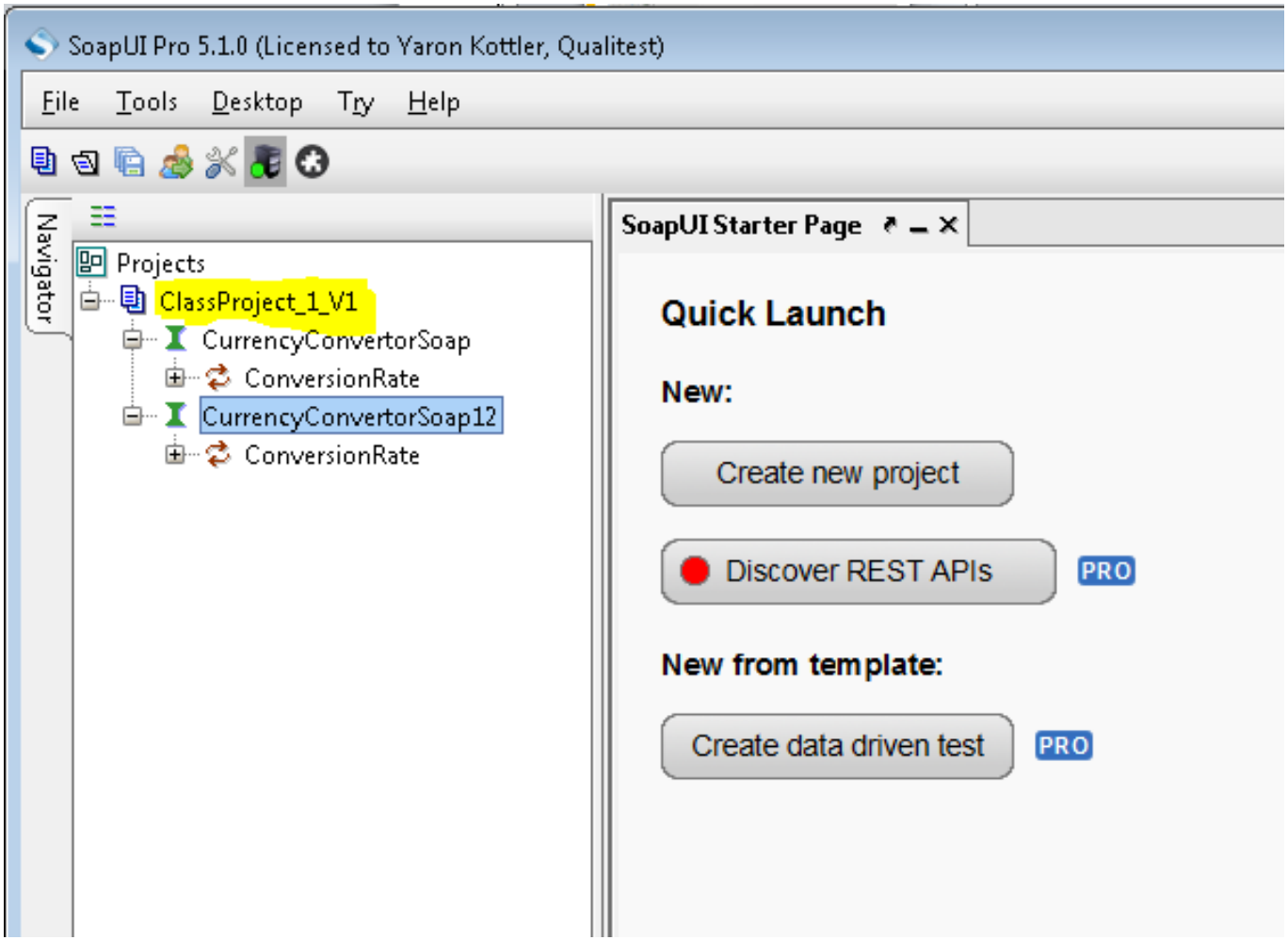
Create Requests: ☑ Create sample requests for all operations?

Create TestSuite: ☐ Creates a TestSuite for the imported WSDL

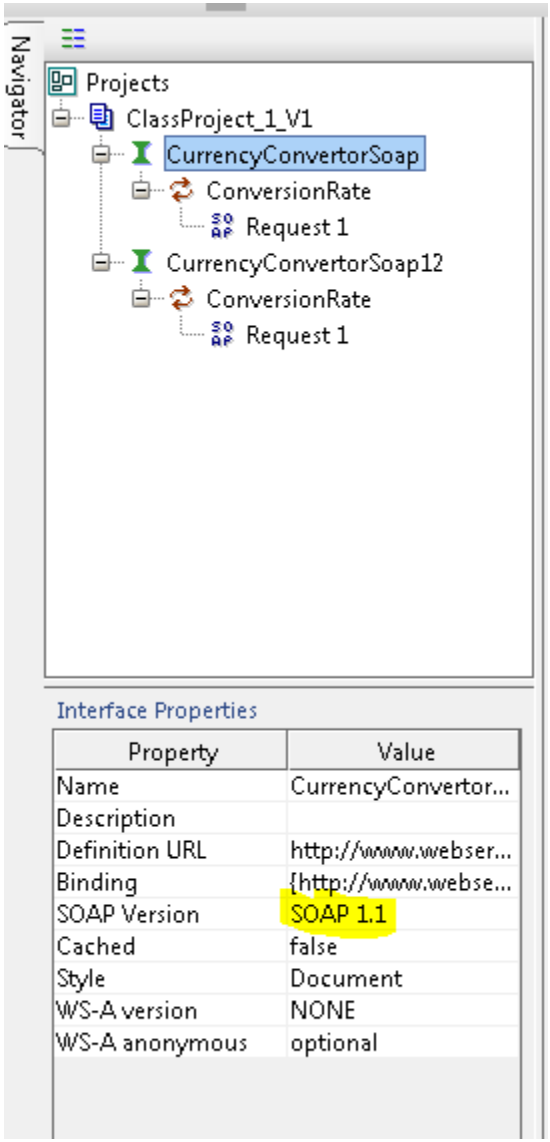
Relative Paths: ☐ Stores all file paths in project relatively to project file (requires save)

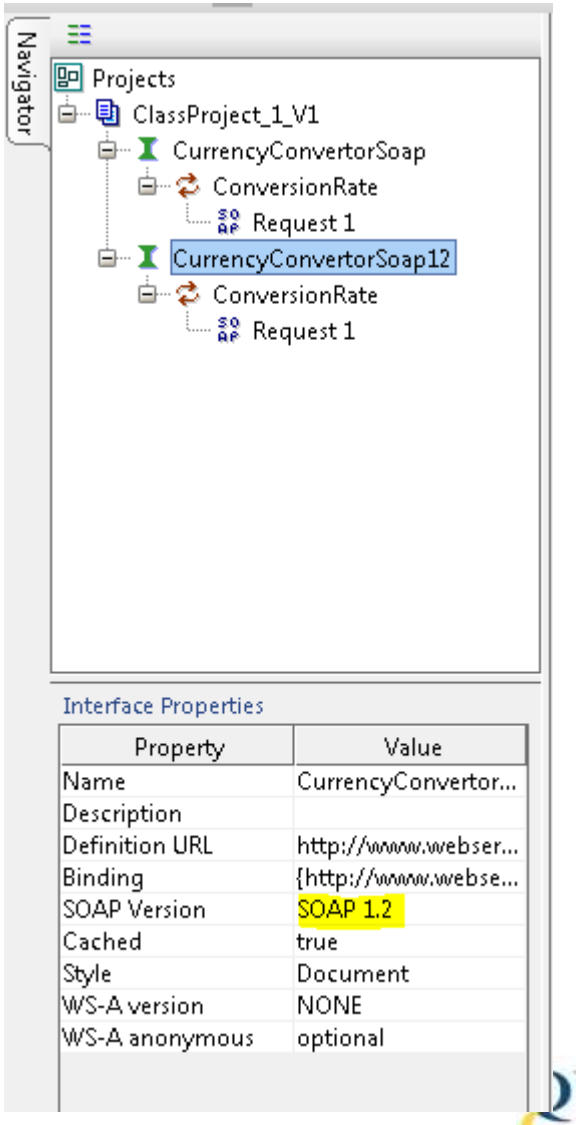
OK Cancel

# Project Created



# This Sample has 2 Versions



# SoapUI Parses WSDL

SOAPUI looks for port & operation name(s)

```
▼<wsdl:portType name="CurrencyConvertorSoap">
  ▼<wsdl:operation name="ConversionRate">
    ▶<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">...</wsdl:d
     <wsdl:input message="tns:ConversionRateSoapIn"/>
     <wsdl:output message="tns:ConversionRateSoapOut"/>
   </wsdl:operation>
```

Input Elements

```
▼<s:element name="ConversionRate">
  ▼<s:complexType>
    ▼<s:sequence>
       <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Currency"/>
       <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency"/>
     </s:sequence>
   </s:complexType>
 </s:element>
```

Enums

```
▼<s:simpleType name="Currency">
  ▼<s:restriction base="s:string">
     <s:enumeration value="AFA"/>
     <s:enumeration value="ALL"/>
     <s:enumeration value="DZD"/>
     <s:enumeration value="ARS"/>
     <s:enumeration value="AWG"/>
```

Response

```
▼<s:element name="ConversionRateResponse">
  ▼<s:complexType>
    ▼<s:sequence>
       <s:element minOccurs="1" maxOccurs="1" name="ConversionRateResult" type="s:double"/>
     </s:sequence>
   </s:complexType>
 </s:element>
```

# Components

SoapUI Pro 5.1.0 (Licensed to Yaron Kottler, Qualitest)

File Tools Desktop Try Help

Navigator

Projects
- ClassProject_1_V1
  - CurrencyConvertorSoap
    - ConversionRate
      - Request 1
  - CurrencyConvertorSoap12
    - ConversionRate
      - Request 1

Request 1 | Request 1

http://www.webservicex.com/currencyconvertor.asmx

View Type: All

XML Raw Outline Form

ConversionRate

FromCurrency *: ? (Currency)

ToCurrency *: ? (Currency)

Navigator

Projects
- ClassProject_1_V1
  - CurrencyConvertorSoap
    - ConversionRate
      - Request 1
  - CurrencyConvertorSoap12
    - ConversionRate
      - Request 1

Request 1 | Request 1

http://www.webservicex.com/currencyconvertor.asmx

XML Raw Outline Form

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:
    <soap:Header/>
    <soap:Body>
        <web:ConversionRate>
            <web:FromCurrency>?</web:FromCurrency>
            <web:ToCurrency>?</web:ToCurrency>
        </web:ConversionRate>
    </soap:Body>
</soap:Envelope>
```



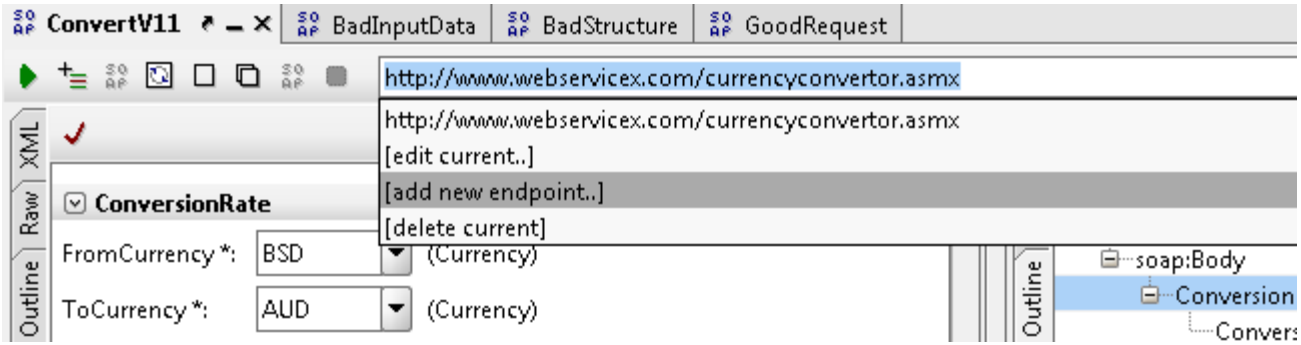
QUALITEST software testing experts

# Endpoints

- An endpoint allows you to point to different environments for testing
- Managing the endpoints can be difficult and it's very, very easy to test against the wrong endpoint
  - Manage endpoints within a project
  - Manage by maintaining Multiple projects
  - Use environments to manage endpoints
  - Use workspaces to manage endpoints

# Endpoints


ConvertV11
BadInputData
BadStructure
GoodRequest
http://www.webservicex.com/currencyconvertor.asmx
View Type: All
XML
XML Node
ConversionRate


ConvertV11
BadInputData
BadStructure
GoodRequest
http://www.webservicex.com/currencyconvertor.asmx
http://www.webservicex.com/currencyconvertor.asmx
[edit current..]
[add new endpoint..]
[delete current]
ConversionRate
FromCurrency *:
BSD
(Currency)
ToCurrency *:
AUD
(Currency)
XML
Raw
Outline
soap:Body
Conversion

# A Word About Requests

Navigator

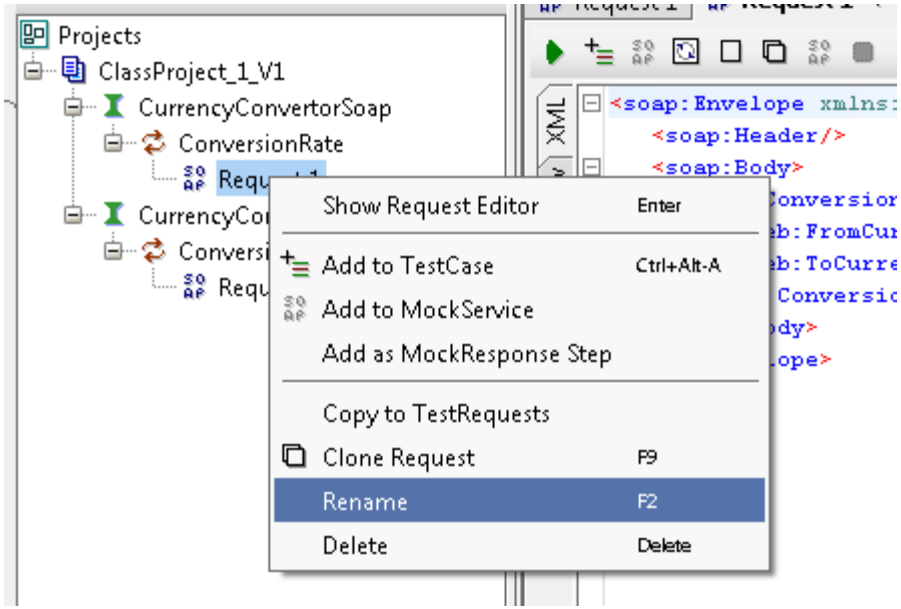
Projects
- ClassProject_1_V1
  - CurrencyConvertorSoap
    - ConversionRate
      - Request 1
  - CurrencyConvertorSoap12
    - ConversionRate
      - Request 1

Request 1 | Request 1

http://www.webservicex.com/currencyconvertor.asmx

XML | Raw | Outline | Form

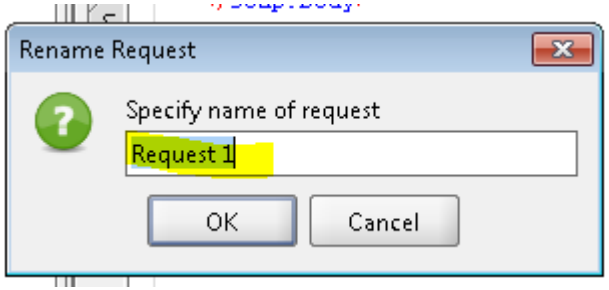
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>?</web:FromCurrency>
            <web:ToCurrency>?</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

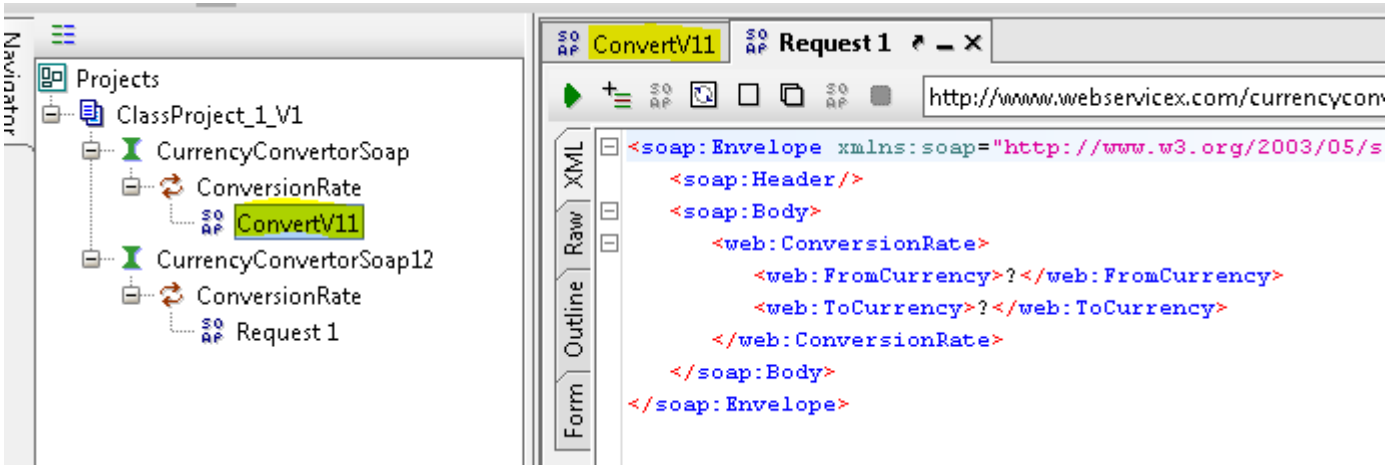





QUALITEST software testing experts

# Renamed

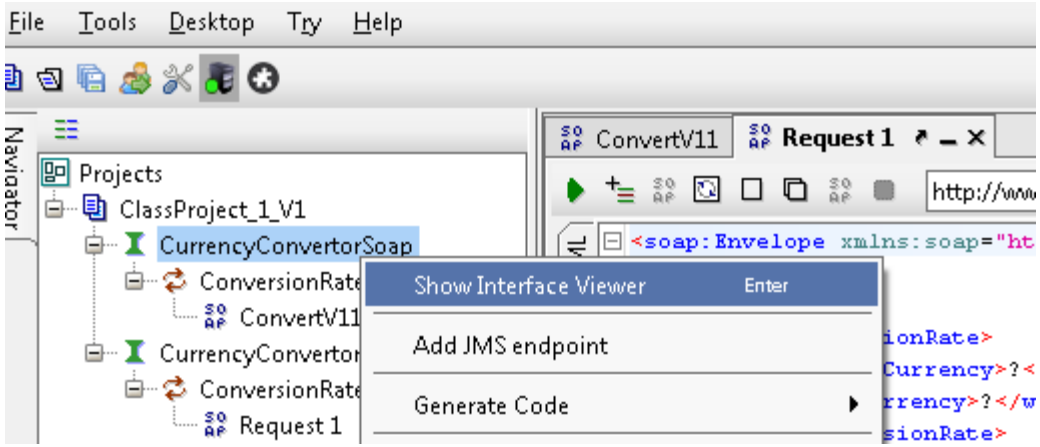


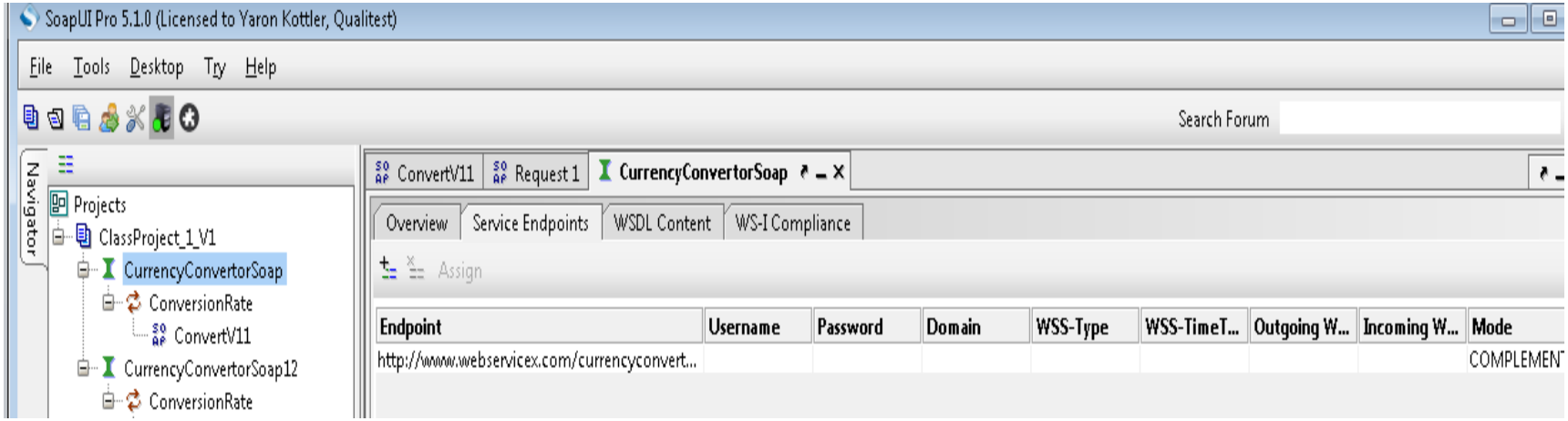



# A Word about Project Names

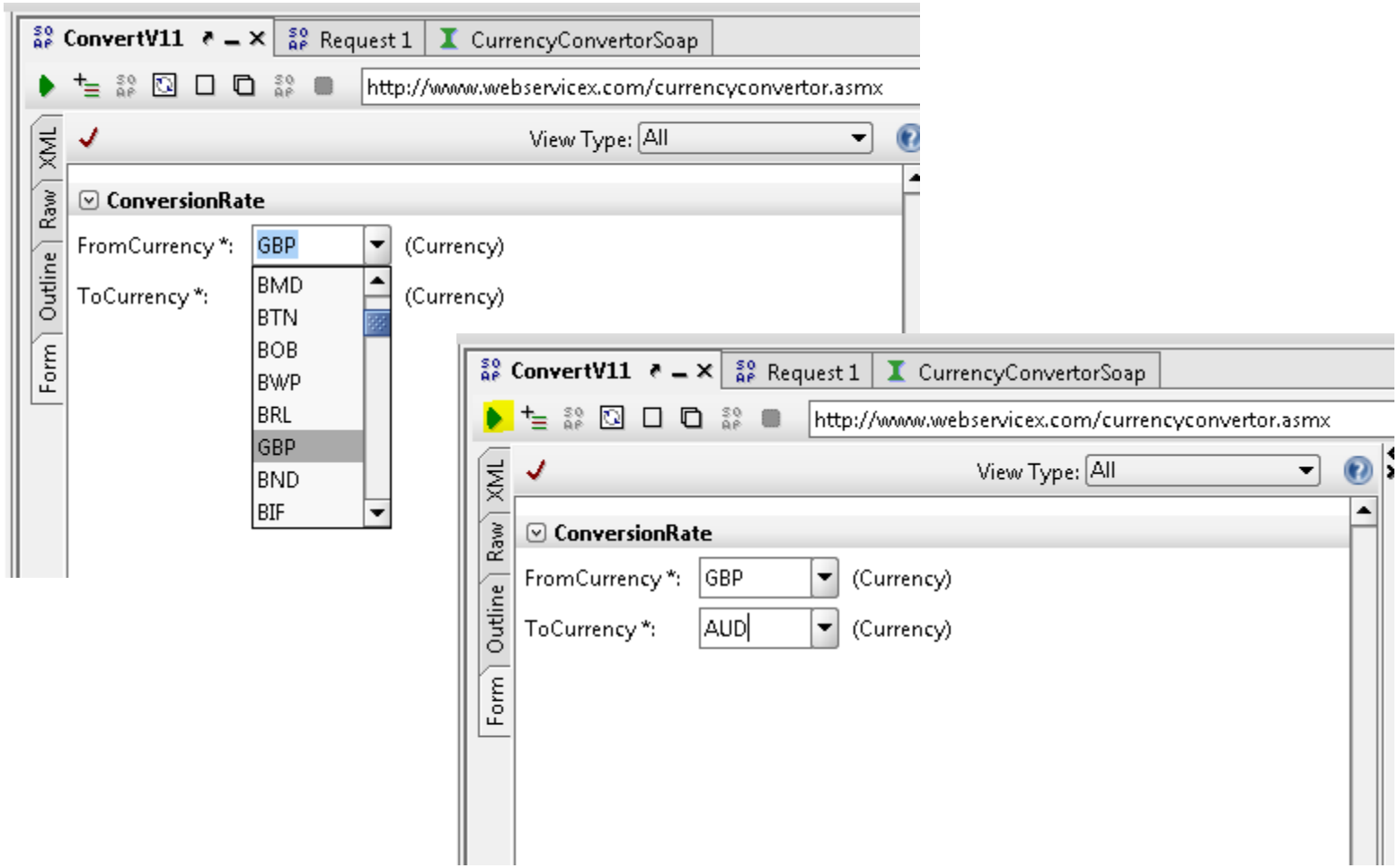
- Projects get sorted alphabetically
- You can name them anything you want
  - Having a version or date included in the project name can save you lots of time/frustration later
- By default, projects will be saved in My Documents
- A saved project is an xml file with the format of project name-soapui-project.xml

# Setting Passwords

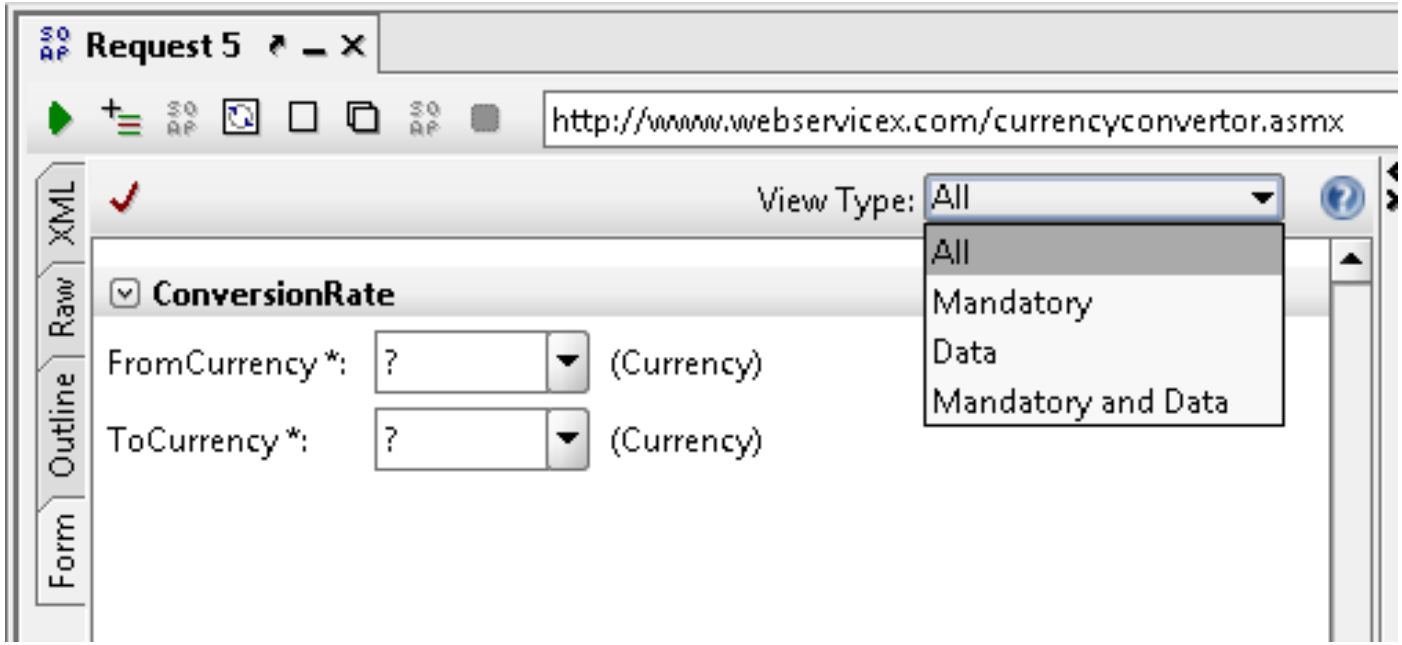


# Submitting a Request from the Form



# Form Request




In the form view, you can filter the fields that you want

# Response

ConvertV11

http://www.webservicex.com/currencyconvertor.asmx

View Type: All

**ConversionRate**

FromCurrency *: GBP (Currency)

ToCurrency *: AUD (Currency)

| XML Node | Value | |
|---|---|---|
| soap:Envelope | | (Envelope) |
| soap:Body | | (Body) |
| ConversionRateResponse | | |
| ConversionRateResult | 1.8063 | (xsd:double) |

ConvertV11

http://www.webservicex.com/currencyconvertor.asmx

View Type: All

**ConversionRate**

FromCurrency *: GBP (Currency)

ToCurrency *: AUD (Currency)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmln
  <soap:Body>
    <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
      <ConversionRateResult>1.8063</ConversionRateResult>
    </ConversionRateResponse>
  </soap:Body>
</soap:Envelope>
```

# Exercise

- Create a new soap project and add this wsdl http://www.webservicex.com/currencyconvertor.asmx?WSDL
- From the form view, select different currency types from the dropdown and submit the request
- View the response in different formats

# Structure of the Request

ConvertV11 | CurrencyConvertorSoap

Overview | Service Endpoints | WSDL Content | WS-I Compliance

- CurrencyConvertorSoap
  - Simple Types
  - Anonymous Complex Types
  - Global Elements
  - Schemas
  - Messages
  - PortTypes
  - Bindings
  - Services

currencyconvertor.asmx?WSDL

http://www.webservicex.com/currencyconvertor.asmx?WSDL

```
1  <wsdl:definitions targetNamespace="http://www.webserviceX.NET/" xmlns:tm="http://microsoft.com/wsdl/mime/te
2     <wsdl:types>
3        <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
4           <s:element name="ConversionRate">
5              <s:complexType>
6                 <s:sequence>
7                    <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Currency"/>
8                    <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency"/>
9                 </s:sequence>
10             </s:complexType>
11          </s:element>
12          <s:simpleType name="Currency">
13             <s:restriction base="s:string">
14                <s:enumeration value="AFA"/>
15                <s:enumeration value="ALL"/>
16                <s:enumeration value="DZD"/>
17                <s:enumeration value="ARS"/>
18                <s:enumeration value="AWG"/>
19                <s:enumeration value="AUD"/>
20                <s:enumeration value="BSD"/>
```

# Navigating thru the Structure

ConvertV11 | CurrencyConvertorSoap

Overview | Service Endpoints | WSDL Content | WS-I Compliance

- CurrencyConverto
  - Simple Types
    - Currency
  - Anonymous C
  - Global Element
  - Schemas
  - Messages
  - PortTypes
  - Bindings
  - Services

currencyconvertor.asmx?WSDL

http://www.webservicex.com/currencyconvertor.asmx?WSDL

```
1  <wsdl:definitions targetNamespace="http://www.webserviceX.NET/" xmlns:tm="http://microsoft.co
2     <wsdl:types>
3        <s:schema elementFormDefault="qualified" targetNamespace="http://www.webserviceX.NET/">
4           <s:element name="ConversionRate">
5              <s:complexType>
6                 <s:sequence>
7                    <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Curren
8                    <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency
9                 </s:sequence>
10             </s:complexType>
11          </s:element>
12          <s:simpleType name="Currency">
13             <s:restriction base="s:string">
14                <s:enumeration value="AFA"/>
15                <s:enumeration value="ALL"/>
16                <s:enumeration value="DZD"/>
17                <s:enumeration value="ARS"/>
18                <s:enumeration value="AWG"/>
19                <s:enumeration value="AUD"/>
```

# Submitting Bad Data

ConvertV11 | CurrencyConvertorSoap

http://www.webservicex.com/currencyconvertor.asmx

XML | Raw | Outline | Form

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlso
   <soapenv:Header/>
   <soapenv:Body>
      <web:ConversionRate>
         <web:FromCurrency>AAA</web:FromCurrency>
         <web:ToCurrency>AUD</web:ToCurrency>
      </web:ConversionRate>
   </soapenv:Body>
</soapenv:Envelope>
```

XML | Raw | Outline | Overview

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelop
   <soap:Body>
      <soap:Fault>
         <faultcode>soap:Client</faultcode>
         <faultstring>System.Web.Services.Protocols.SoapException:
   at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializati
   at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializati
   at Microsoft.Xml.Serialization.GeneratedAssembly.ArrayOfObjectS
   at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader
   --- End of inner exception stack trace ---
   at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader
   at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader
   at System.Web.Services.Protocols.SoapServerProtocol.ReadParamet
   --- End of inner exception stack trace ---
   at System.Web.Services.Protocols.SoapServerProtocol.ReadParamet
   at System.Web.Services.Protocols.WebServiceHandler.CoreProcessR
         <detail/>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

# Overview Results View

vicex.com/currencyconvertor.asmx

Overview | Outline | Raw | XML

**soap:Envelope**

**soap:Body**

**soap:Fault**

faultcode: soap:Client

faultstring: System.Web.Services.Protocols.SoapException: Server was unable to read request. ---> System.InvalidOperationExceptic
-> System.InvalidOperationException: Instance validation error: 'AAA' is not a valid value for Currency. at Microsoft.Xml
onReader1.Read1_Currency(String s) at Microsoft.Xml.Serialization.GeneratedAssembly.XmlSerializationReader1.Read2_
GeneratedAssembly.ArrayOfObjectSerializer.Deserialize(XmlSerializationReader reader) at System.Xml.Serialization.XmlS
encodingStyle, XmlDeserializationEvents events) --- End of inner exception stack trace --- at System.Xml.Serialization.X
ring encodingStyle, XmlDeserializationEvents events) at System.Xml.Serialization.XmlSerializer.Deserialize(XmlReader xm
ervices.Protocols.SoapServerProtocol.ReadParameters() --- End of inner exception stack trace --- at System.Web.Service
() at System.Web.Services.Protocols.WebServiceHandler.CoreProcessRequest()

# Complex Types

Overview | Service Endpoints | WSDL Content

- CurrencyConvertorSoap
  - Simple Types
    - Currency
  - Anonymous Complex Types
    - ConversionRate
    - ConversionRateResponse
  - Global Elements
  - Schemas
  - Messages
  - PortTypes
  - Bindings
  - Services

currencyconvertor.asmx?WSDL

http://www.webservicex.com/currencyconvertor.asmx?WSDL

```
4        <s:element name="ConversionRate">
5           <s:complexType>
6              <s:sequence>
7                 <s:element minOccurs="1" maxOccurs="1" name="FromCurrency" type="tns:Currency"/>
8                 <s:element minOccurs="1" maxOccurs="1" name="ToCurrency" type="tns:Currency"/>
9              </s:sequence>
10          </s:complexType>
11       </s:element>
12       <s:simpleType name="Currency">
```

# Faults Structure

- Remove one element

http://www.webservicex.com/currencyconvertor.asm

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/s
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <!-- <web:FromCurrency>AAA</web:FromCurrency>-->
            <web:ToCurrency>AUD</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xm
    <soap:Body>
        <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
            <ConversionRateResult>0</ConversionRateResult>
        </ConversionRateResponse>
    </soap:Body>
</soap:Envelope>
```

# Fault - Structure

- Duplicate first element

http://www.webservicex.com/currencyconvert

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlso
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>CAD</web:FromCurrency>
            <web:FromCurrency>BSD</web:FromCurrency>
            <web:ToCurrency>AUD</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:
    <soap:Body>
        <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
            <ConversionRateResult>0.9825</ConversionRateResult>
        </ConversionRateResponse>
    </soap:Body>
</soap:Envelope>
```

# Fault?

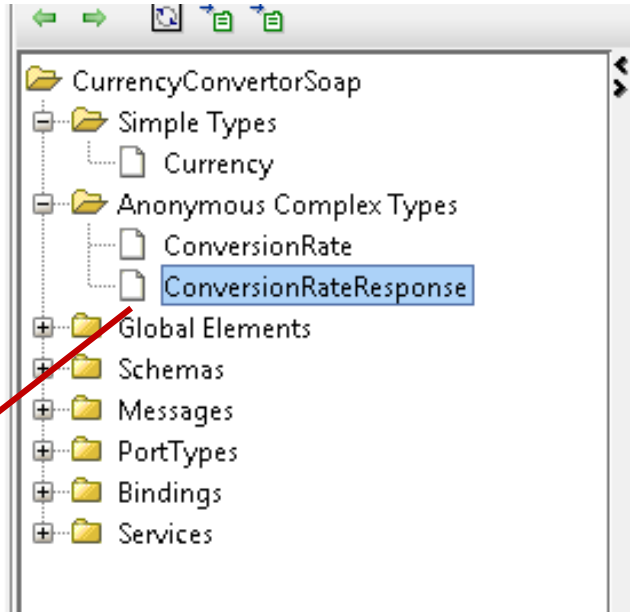
- Duplicate the entire complex object

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/s
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>BSD</web:FromCurrency>
            <web:ToCurrency>AUD</web:ToCurrency>
        </web:ConversionRate>
        <web:ConversionRate>
            <web:FromCurrency>CAD</web:FromCurrency>
            <web:ToCurrency>AUD</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xm
    <soap:Body>
        <ConversionRateResponse xmlns="http://www.webserviceX.NET/">
            <ConversionRateResult>0.9826</ConversionRateResult>
        </ConversionRateResponse>
    </soap:Body>
</soap:Envelope>
```

# Complex Type - Response

- Rules for both the request and response are provided in the wsdl




```
167        <s:element name="ConversionRateResponse">
168           <s:complexType>
169              <s:sequence>
170                 <s:element minOccurs="1" maxOccurs="1" name="ConversionRateResult" type="s:double"/>
171              </s:sequence>
172           </s:complexType>
```

# Quick Navigation

Overview | Service Endpoints | WSDL Content | WS-I Compliance

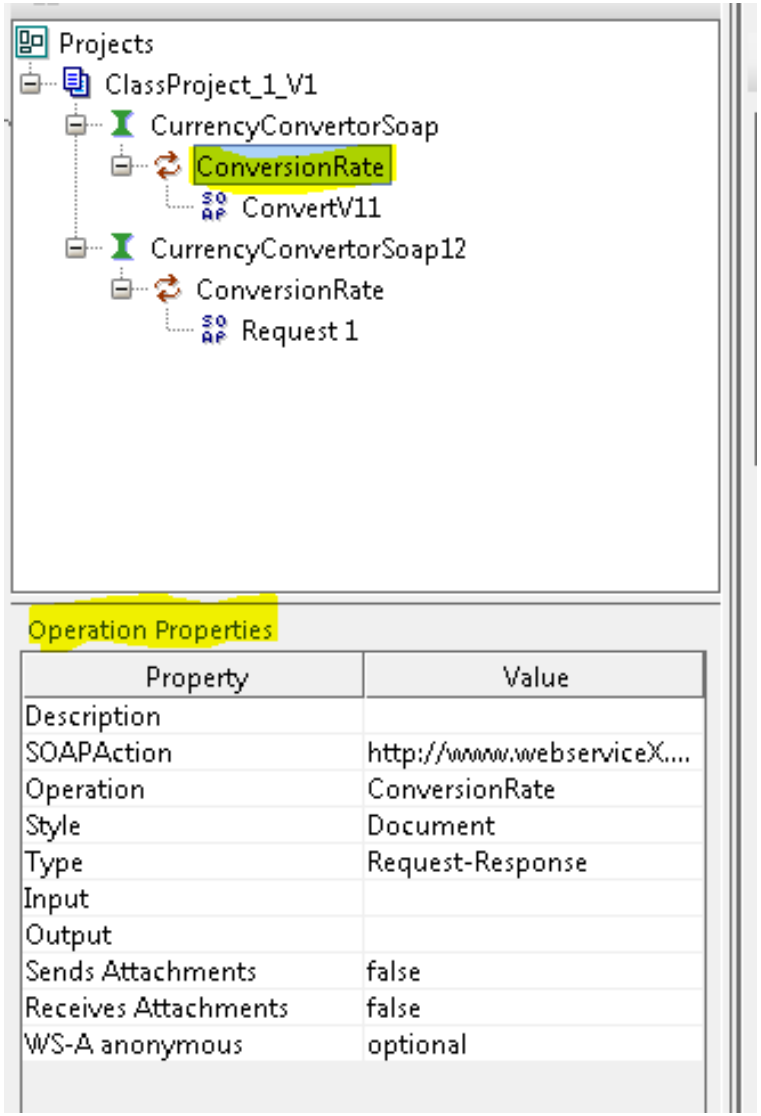
Exports the entire WSDL and included/imported files to a local directory

- CurrencyConvertorSoap
  - Simple Types
  - Anonymous Complex Types
  - Global Elements
  - Schemas
  - Messages
  - PortTypes
  - Bindings
  - Services
    - CurrencyConvertor

http://www.webservicex.com/currencyconvertor.asmx?WSDL

```
1  <wsdl:definitions targetNamespace="http://www.webservic
2     <wsdl:types>
3        <s:schema elementFormDefault="qualified" targetNa
4           <s:element name="ConversionRate">
5              <s:complexType>
6                 <s:sequence>
7                    <s:element minOccurs="1" maxOccurs="
8                    <s:element minOccurs="1" maxOccurs="
9                 </s:sequence>
10             </s:complexType>
11          </s:element>
12          <s:simpleType name="Currency">
```

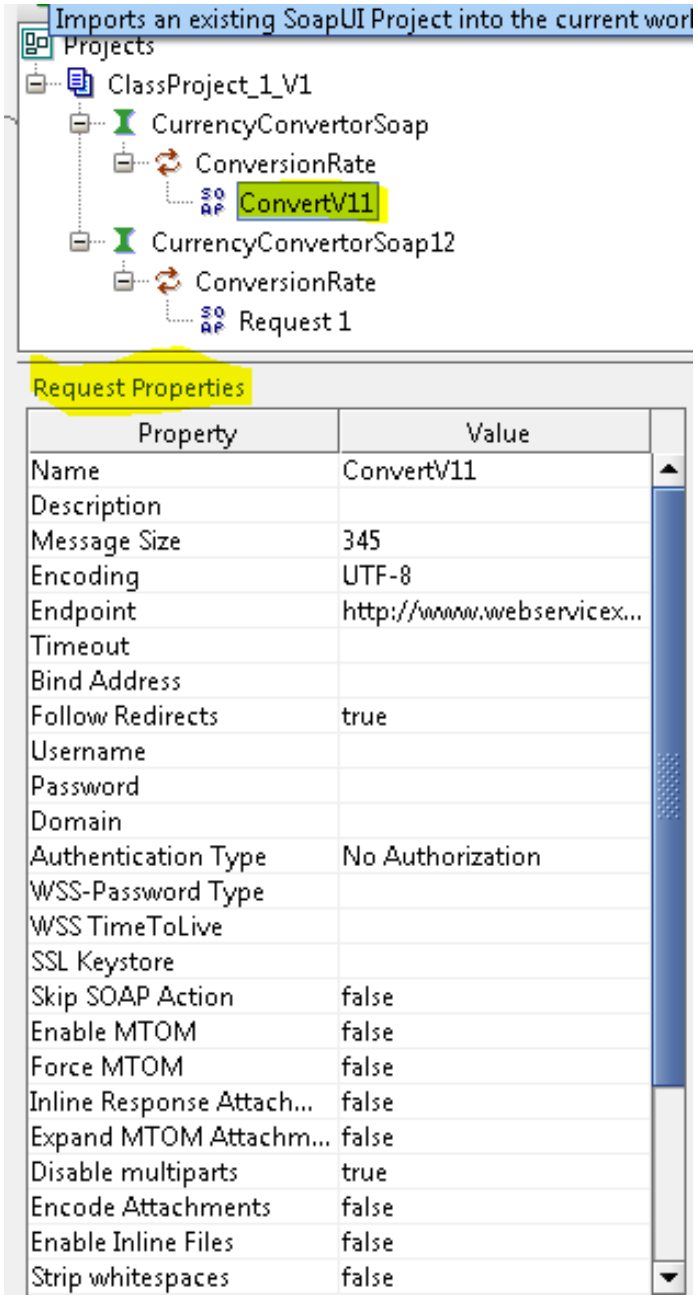
# Operation Properties






QUALITEST software testing experts

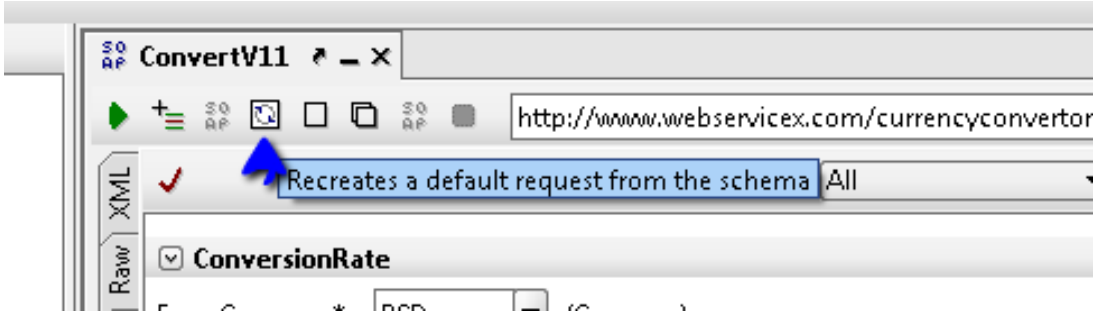
# Request Properties

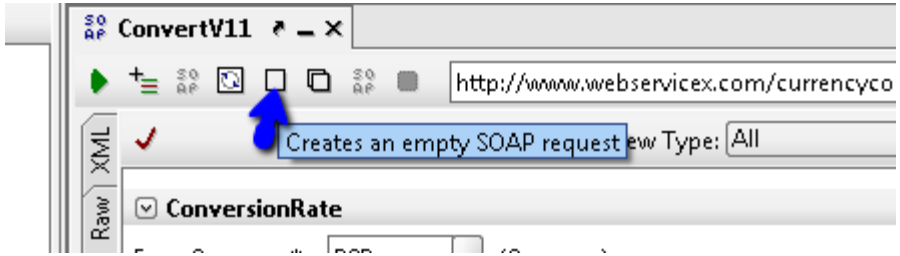


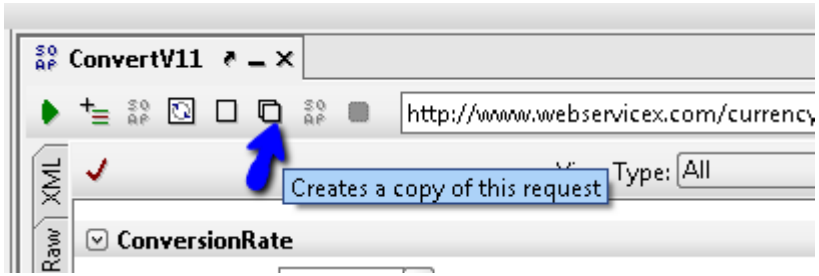


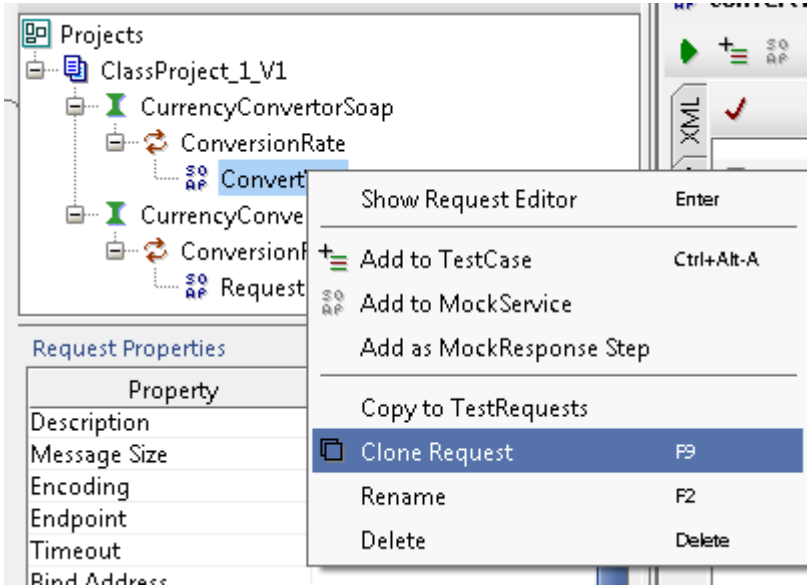

QUALITEST software testing experts

# Create Multiple Requests



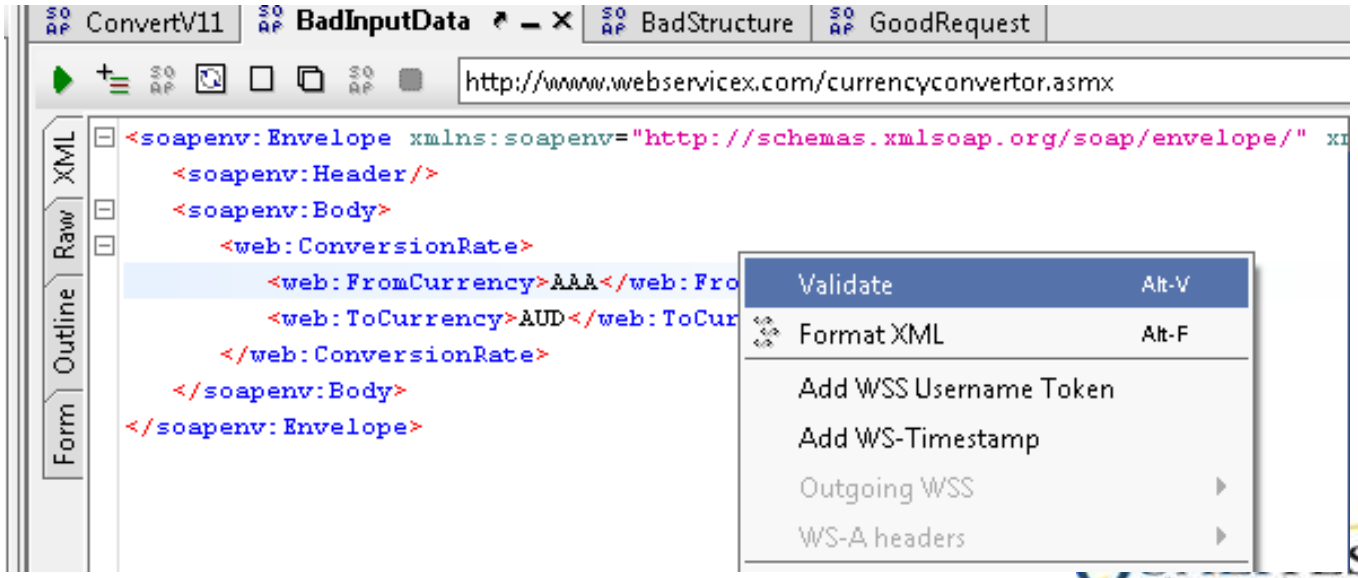






QUALITEST software testing experts

# Validate Requests

# Validate Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
    <soapenv:Header/>
    <soapenv:Body>
        <web:ConversionRate>
            <web:FromCurrency>AAA</web:FromCurrency>
            <web:ToCurrency>AUD</web:ToCurrency>
        </web:ConversionRate>
    </soapenv:Body>
</soapenv:Envelope>
```

line 5: string value 'AAA' is not a valid enumeration value for Currency in namespace http://www.webservice

Double click on error to directly navigate

# Structural Validation

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
   <soapenv:Header/>
   <soapenv:Body>
      <web:ConversionRate>
         <web:FromCurrency>CAD</web:FromCurrency
         <web:ToCurrency>AUD</web:ToCurrency>
      </web:ConversionRate>
   </soapenv:Body>
</soapenv:Envelope>
```

line -1: error: Unexpected character encountered while in a closing tag: '<'
line 5: Unexpected character encountered while in a closing tag: '<'

# Exercise

- Create new project using wsdl - **http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL**
- Look at the structure of the wsdl and create multiple requests for
  - Valid request
  - Invalid requests (bad data/bad structure)
- Validate the requests before submitting them
- Save your work regularly
  - soap saves internally immediately but no auto save unless you set it up