

LAND REGISTRY

---

# SoapUI Basics: Automating Web Services

# Table of Contents

|  |    |
|--|----|
| Web Service Basics .....                                     | 1  |
| 1.1 What is a Web Service? .....                             | 1  |
| 1.2 What does the Land Registry use? .....                   | 1  |
| 1.3 What do we currently use to test our Web Services? ..... | 1  |
| 1.4 What is SoapUI? .....                                    | 2  |
| 1.5 Why are we using at SoapUI? .....                        | 2  |
| Installing SoapUI .....                                      | 3  |
| 2.1 Downloading SoapUI.....                                  | 3  |
| Creating a Project with a WSDL .....                         | 10 |
| Creating a Test Suite and Test Case.....                     | 13 |
| 4.1 Creating a Test Suite.....                               | 13 |
| 4.2 Creating a Test Case .....                               | 16 |
| 4.3 Creating Test Steps – Initial Request. ....              | 19 |
| 4.4 Creating Test Steps – Checking the Response .....        | 27 |
| Running Tests .....  | 33 |
| 5.1 Running Test Steps Individually .....                    | 33 |
| 5.2 Running Tests as a Test Case .....                       | 33 |
| 5.3 Running Tests as a Test Suite .....                      | 37 |
| Conclusion .....   | 40 |
| Appendix.....  | 41 |
| A.1 Table of Environment Placeholders for WSDLs .....        | 41 |
| A.2 WSDL Addresses .....                                     | 42 |

---

# Web Service Basics

## 1.1 What is a Web Service?

A web service is a software system that allows two systems to interact with each other over a network or most, commonly the internet. There are many variances of web service and the main two that you will hear about are 'SOAP web services' and restful web services.

## 1.2 What does the Land Registry use?

The Land Registry uses SOAP based web services for all of its business gateway services. SOAP is a communication protocol that uses XML. This permits applications belonging to another company to communicate to our business gateway using HTTP; something that which all internet browsers and servers use. We use WSDL, which are descriptions of the web services used in combination with SOAP and XML to tell the customer what we expect to receive from them.

## 1.3 What do we currently use to test our Web Services?

We currently use the business gateway test harness, a piece of software developed in house which allows us to create and send SOAP requests to the required service and test region.

The screenshot displays the Business Gateway Test Harness web application. The top navigation bar includes links for Enquiries, Land Charges, Official Copies, Others, Register Extract, Searches, Verification, VIPR, eDRS, Settings, and Extras. The main interface is divided into two sections: InputView and OutputView.

**InputView:** This section contains several input fields and buttons. The 'User Id' field is populated with 'GJohnson3000' and the 'Password' field with 'green123'. The 'Message Id' field contains 'DfItMsgId-20140424-135052'. Below these fields is a dropdown menu for 'Previously used Message Ids'. The 'Locale' is set to 'en'. There is a checkbox labeled 'Generate new message id on submit or convert'. At the bottom of the InputView, there are buttons for 'Submit', 'Convert', 'Reset', and 'Submit Poll Request'.

**OutputView:** This section displays the XML response generated by the SOAP request. The XML is as follows:

```
<ns3:eDocumentRegistration xmlns:ns2="http://www.oscre.org/ns/eReg-Final/2012/RequestApplicationToChangeRegisterV1_0" xmlns:ns3="http://drsv1_0.ws.bg.lf.gov"/>
  <arg0>
    <ns2:MessageId>DfItMsgId-20140424-135052</ns2:MessageId>
    <ns2:ExternalReference></ns2:ExternalReference>
    <ns2:Product>
      <ns2:Reference></ns2:Reference>
      <ns2:TotalFeeInPence>0</ns2:TotalFeeInPence>
      <ns2:Email></ns2:Email>
      <ns2:TelephoneNumber></ns2:TelephoneNumber>
      <ns2:AP1WarningUnderstood>false</ns2:AP1WarningUnderstood>
      <ns2:DisclosableOverridingInterests>false</ns2:DisclosableOverridingInterests>
    </ns2:Product>
    <ns2:Titles>
      <ns2:TitleNumber></ns2:TitleNumber>
    </ns2:Titles>
    <ns2:Applications>

```

Below the XML response, there is a section for 'Application to Change Register' with fields for 'External Ref', 'Postcode', 'Reference', 'Local Authority', 'Application date', 'Telephone', 'Fee (in pence)', and 'Email'.

## **1.4 What is SoapUI?**

SoapUI is an open source system which allows you to script and run automated tests for various types of web services. You can directly use the WSDLs that we provide for the web service you are testing to create a project, with SoapUI creating example requests.

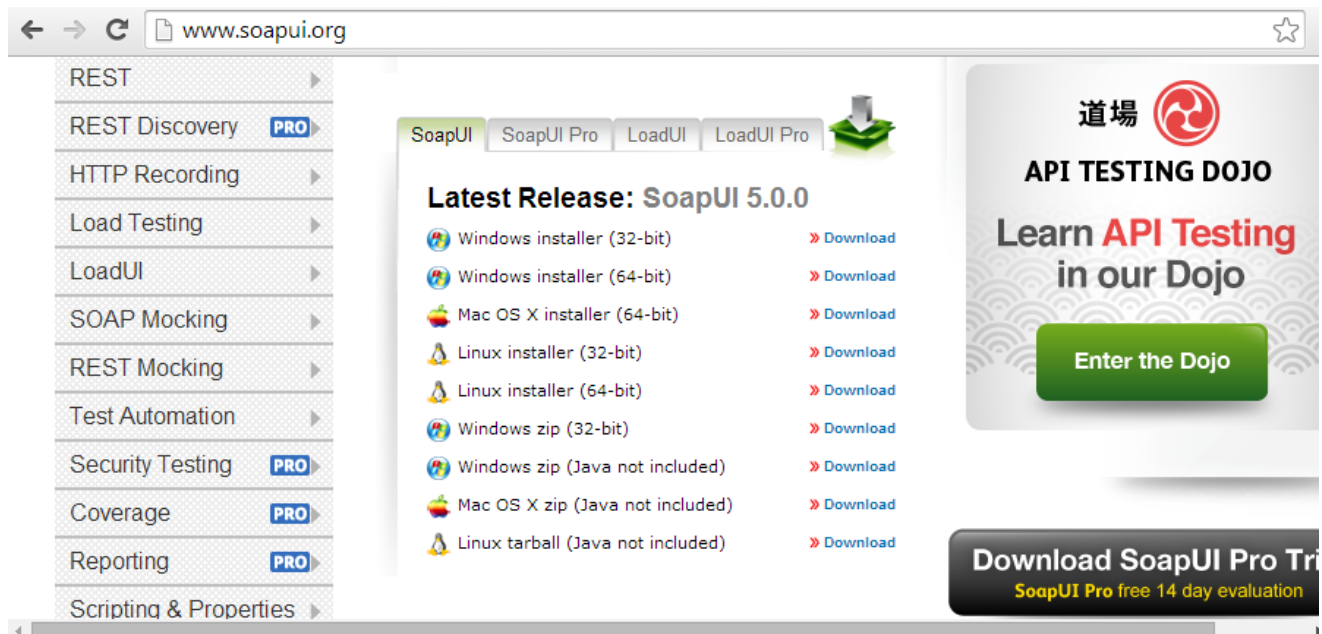
## **1.5 Why are we using at SoapUI?**

By expanding our toolset for automation we have the opportunity to get closer to our test automation targets.

# Installing SoapUI

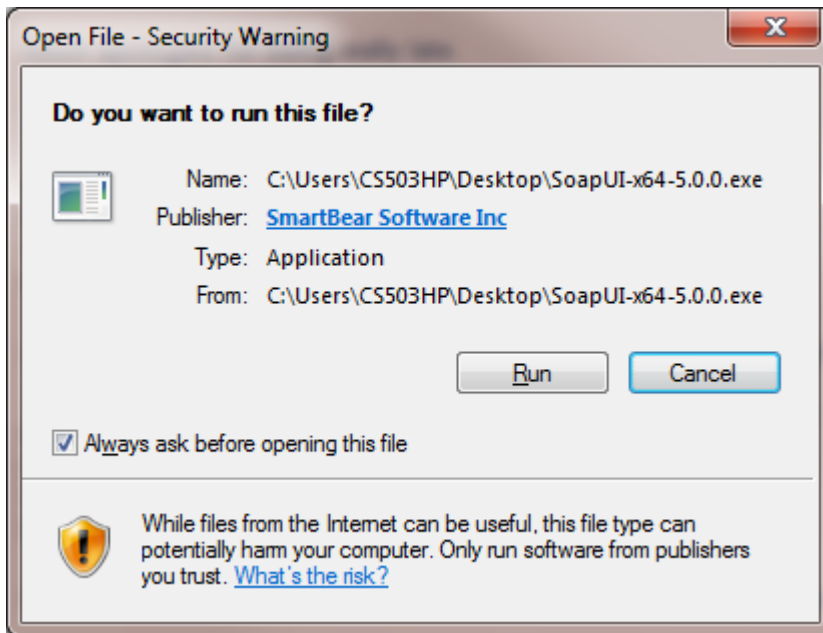
## 2.1 Downloading SoapUI

1. First step is to visit [www.soapui.org](http://www.soapui.org) which is the main website for SoapUI.



2. On the page you will see a box which contains the download links for the latest release of SoapUI.
3. As are currently running on 64 bit Windows 7 you will need to click the download link 'Windows installer (64-bit)'.
4. When the download starts, select to save somewhere you can access easily (personal folder – preferably not your desktop).

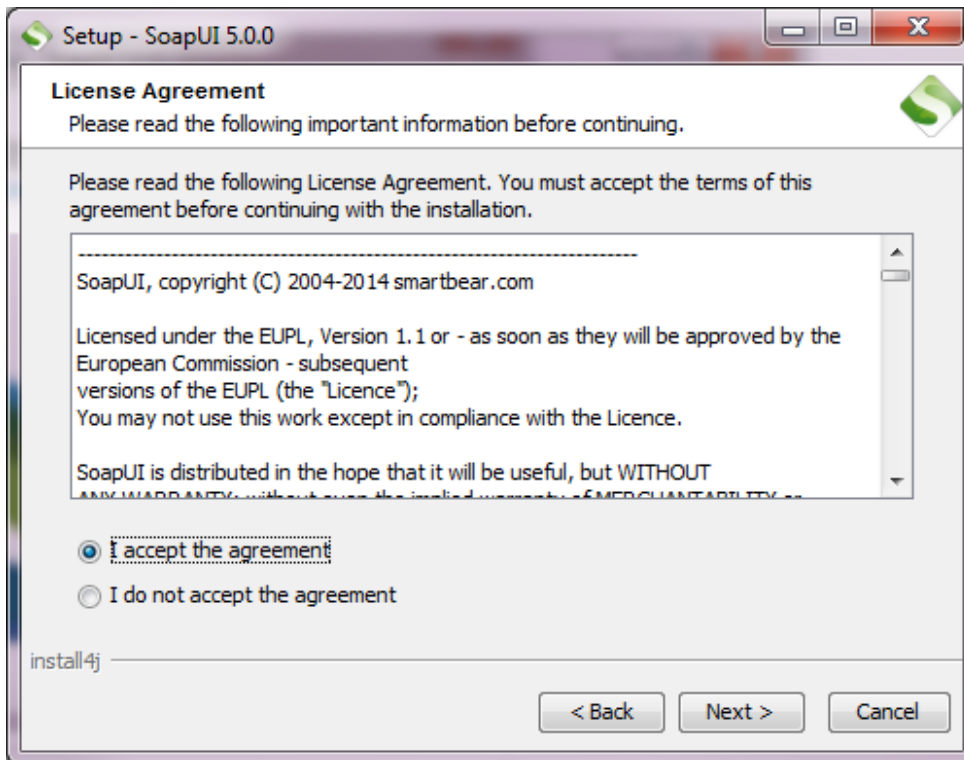
5. When the download has completed open the file and click 'Run' on this dialogue box:



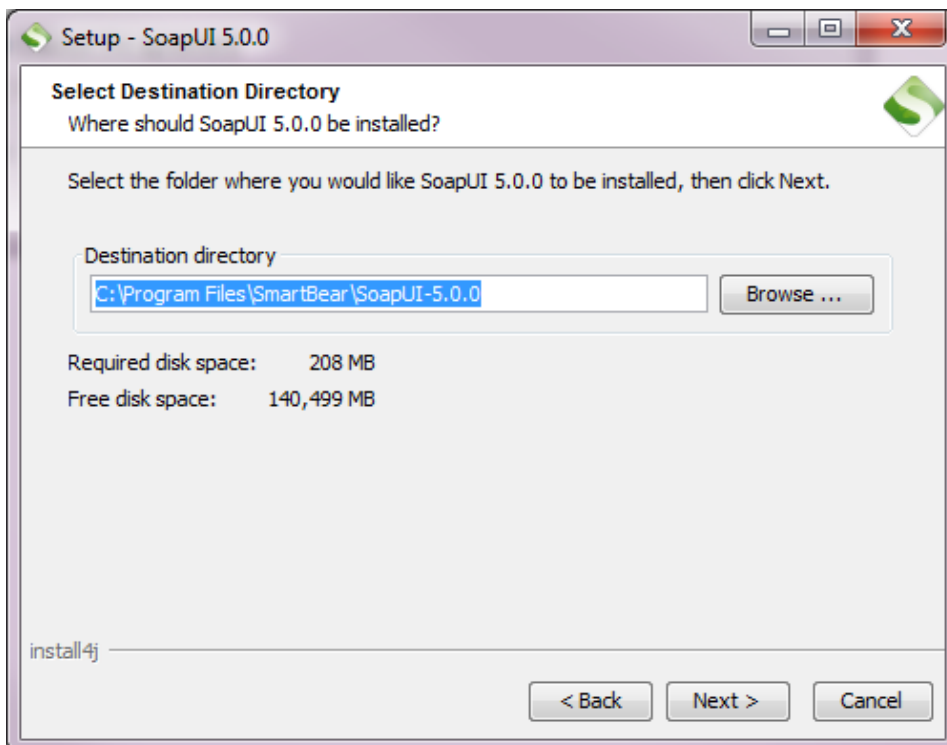
6. If this doesn't work for some reason (Smartbear's downloads occasionally corrupt at time of writing) visit this link to our 'O' drive: O:\ESG\Resource\SoapUI-Exe and install the most up to date version we have there.
7. Click 'Next' to go through the install wizard.



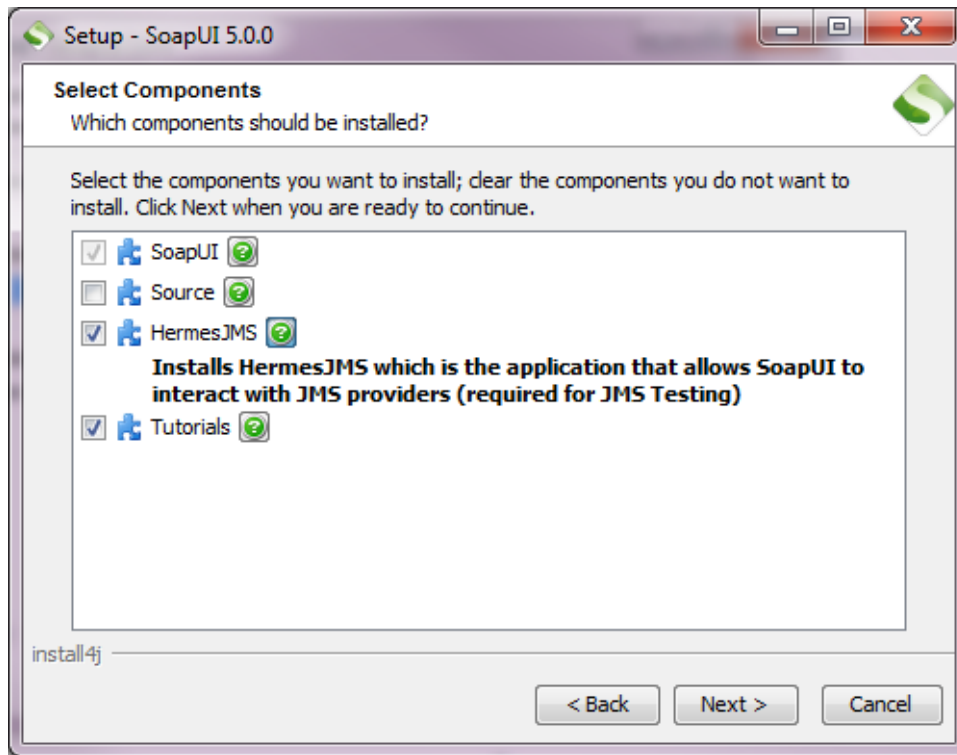
## 8. Select 'I accept the agreement' and click 'Next'



## 9. Click 'Next' again.



10. Click 'Next' again.

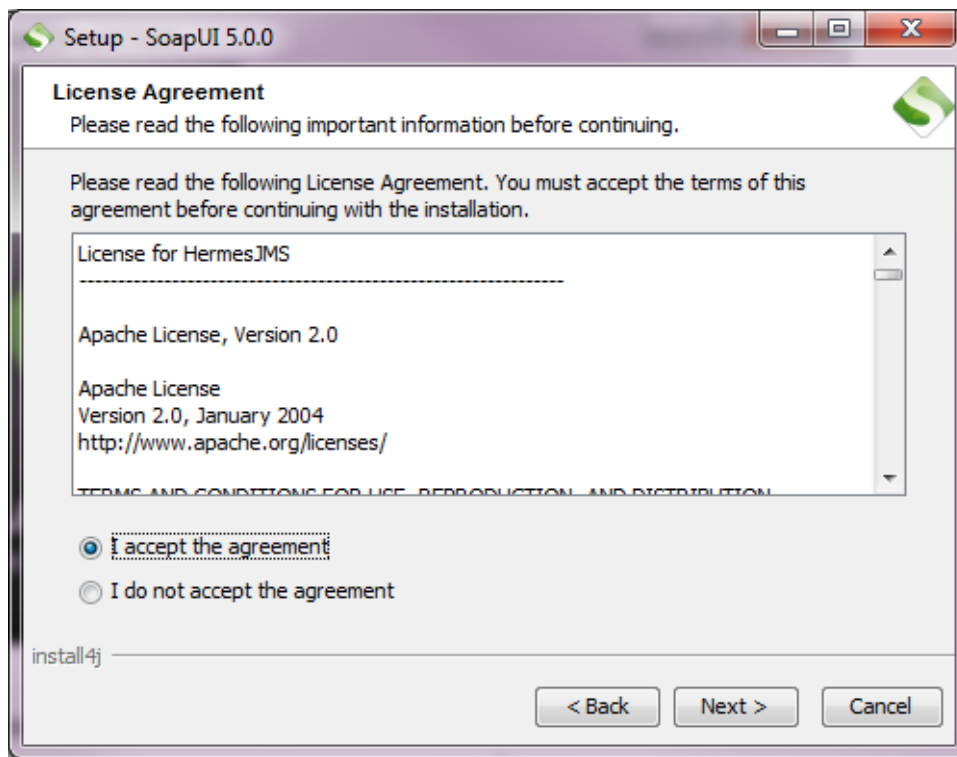


11. Without ticking the tick box at the top of the window. Click 'Next'.

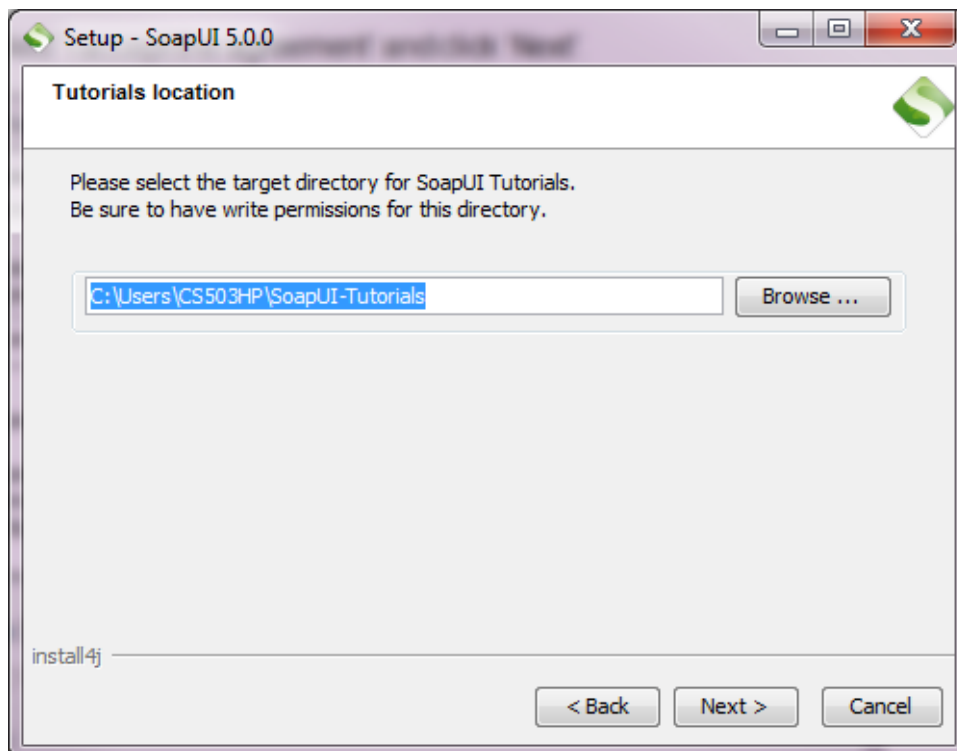




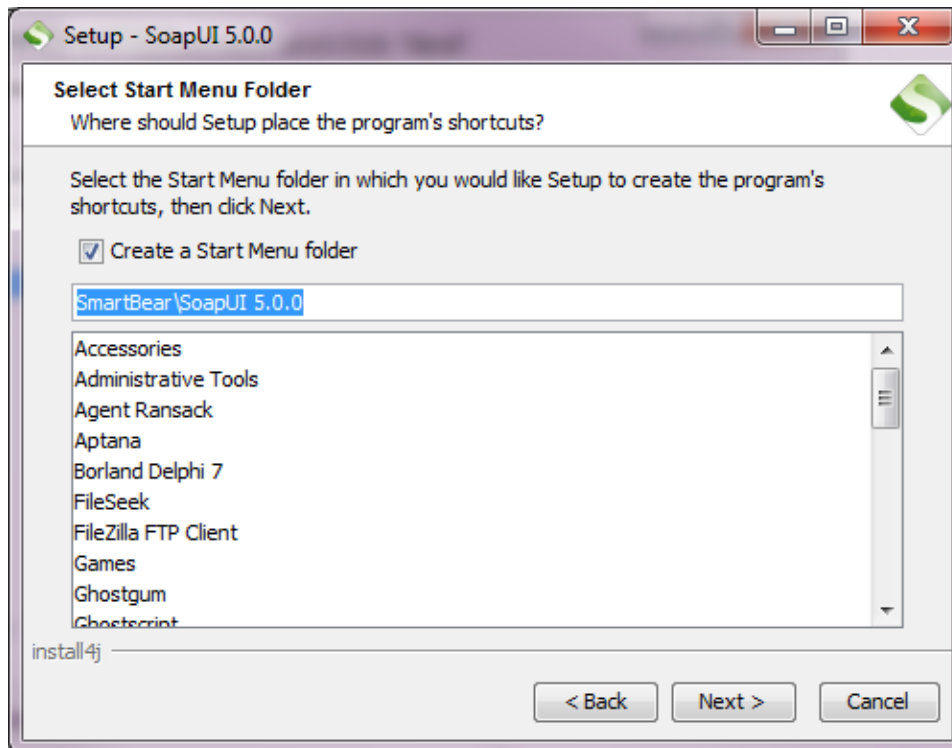
12. Select 'I accept the agreement' and click 'Next'.



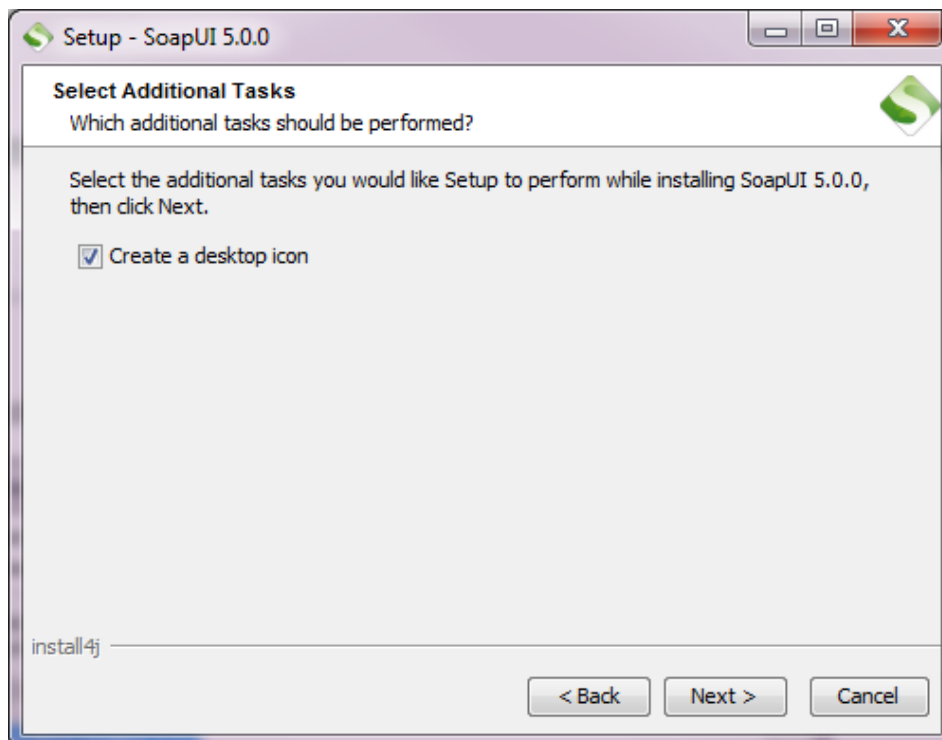
13. Click 'Next'.



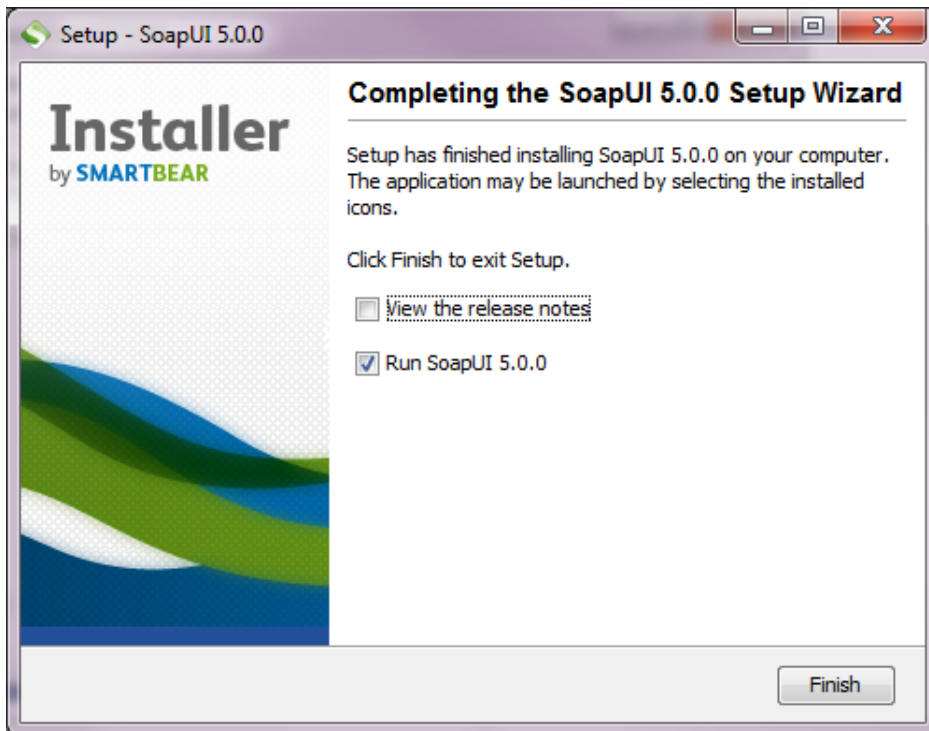
14. Click 'Next'.



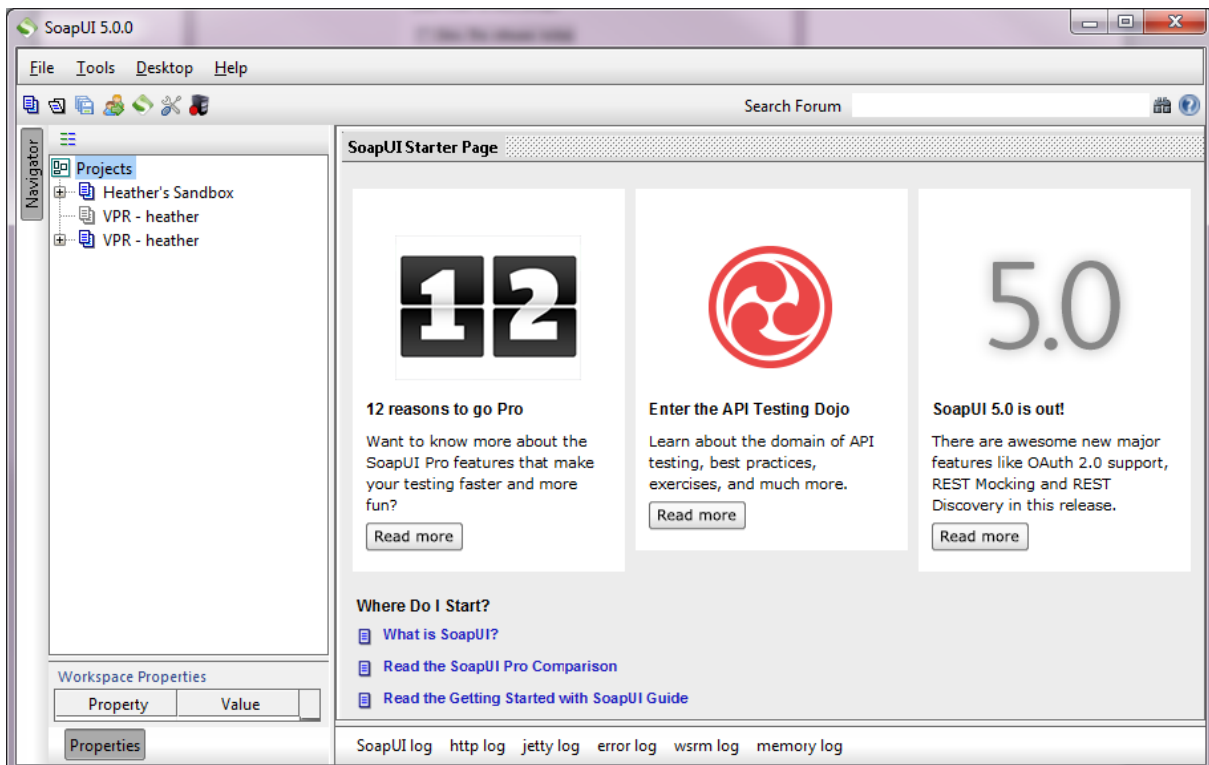
15. Click 'Next' (un tick the box if you don't want a desktop icon)



16. Un tick the box for viewing the release notes and click 'Finish'.



17. SoapUI should now be open.



## Creating a Project with a WSDL

In this chapter we will create a project using a WSDL for the web service we plan to test.

1. Firstly we need the WSDL for which we are using for our testing. We have a large amount of web services within the land registry and these can be used in each of our test environments. Each test environment has four addresses, which either go through Webseal or not and an address that is used for test stubs.

In the appendix in **sub chapter A.1** the environment addresses are supplied in the '**Table of Environment Placeholders for WSDLs**'.

When typically testing we have used the environment placeholders that have 'via\_webseal' set to 'N' and stub set to 'N'

In this example we will be using the SYT2 environment placeholder of:

2. **`https://X6.b2b.http.server.landregistry.gov:19009/`**
3. The next part is dependent on what service we are testing. In the appendix in **sub chapter A.2** there is a table labelled '**WSDL Addresses**'.

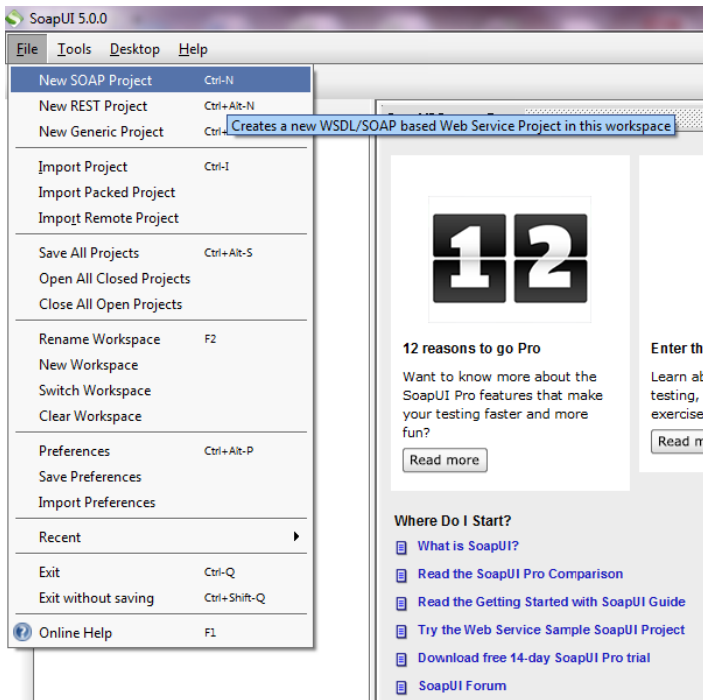
In this guide we shall be looking at the Daylist Enquiry business gateway web service and therefore our WSDL address part shall be:

**`BGSoapEngine/DaylistEnquiryV2_0WebService/WEB-INF/wsd/DaylistEnquiryV2_0Service.wsdl`**

4. Combining these two will produce the WSDL that we will use when creating our new project:

**`https://X6.b2b.http.server.landregistry.gov:19009/BGSoapEngine/DaylistEnquiryV2_0WebService/WEB-INF/wsd/DaylistEnquiryV2_0Service.wsdl`**

5. As this will be testing a SOAP based web service, we will select to create a new SOAP project from File > New SOAP Project as shown below.

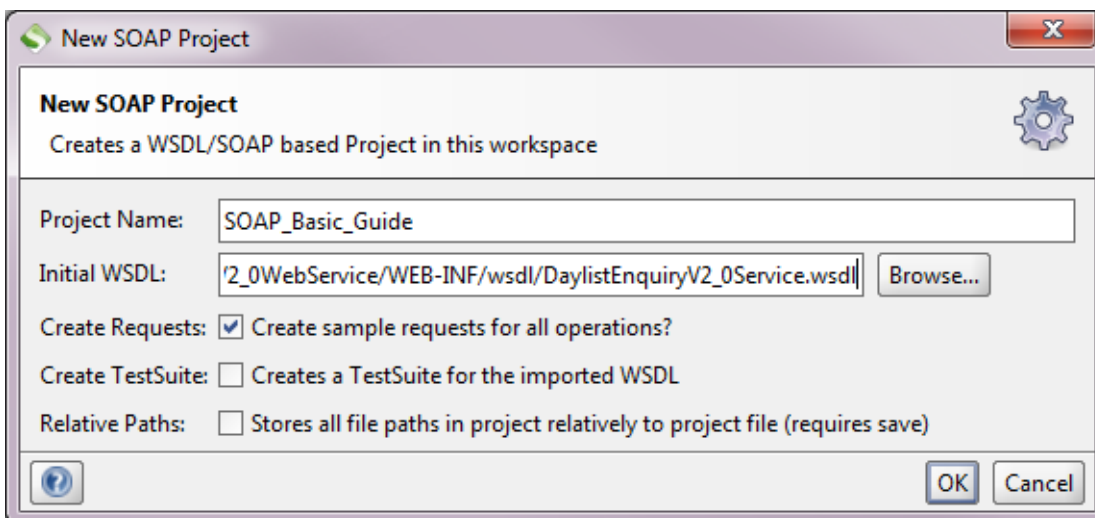


6. This will bring up the below window.

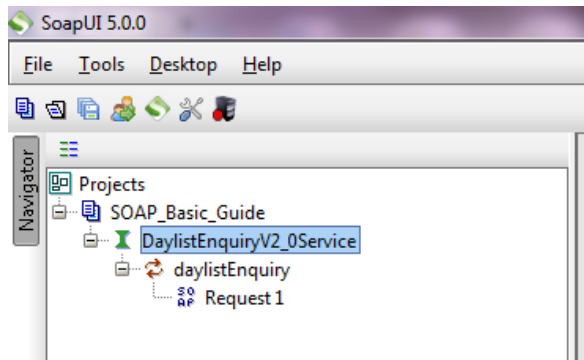
In Project Name enter the name of your project.

In Initial WSDL enter the WSDL address that we put together in step 4.

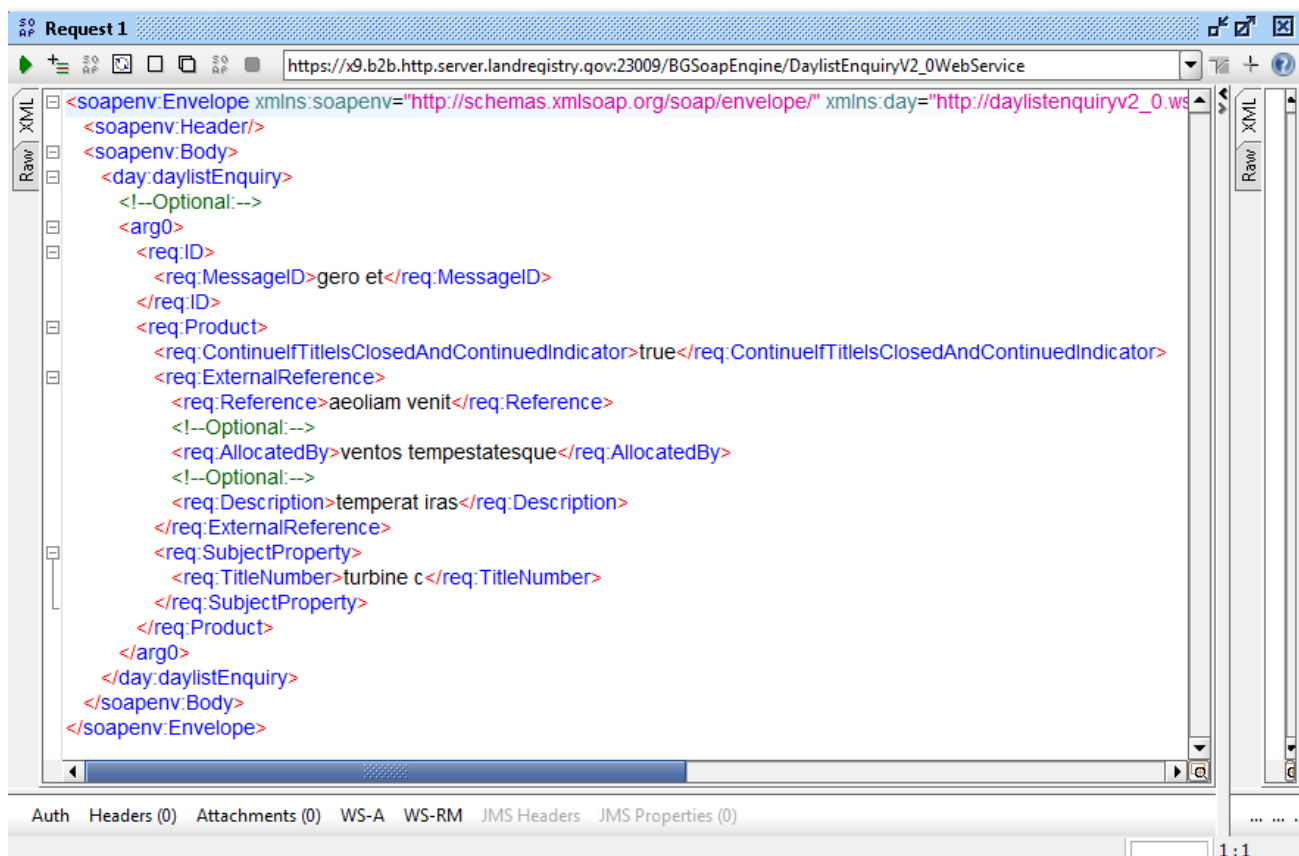
Click 'Ok'



7. Clicking 'Ok' will load the definitions from the WSDL, and as we had the option ticked in the previous window it will have produced a sample request:



And should look like this (if using a SoapUI version lower than 5.0 the text will be replaced with '?') :



We now have a project that we can start working in.

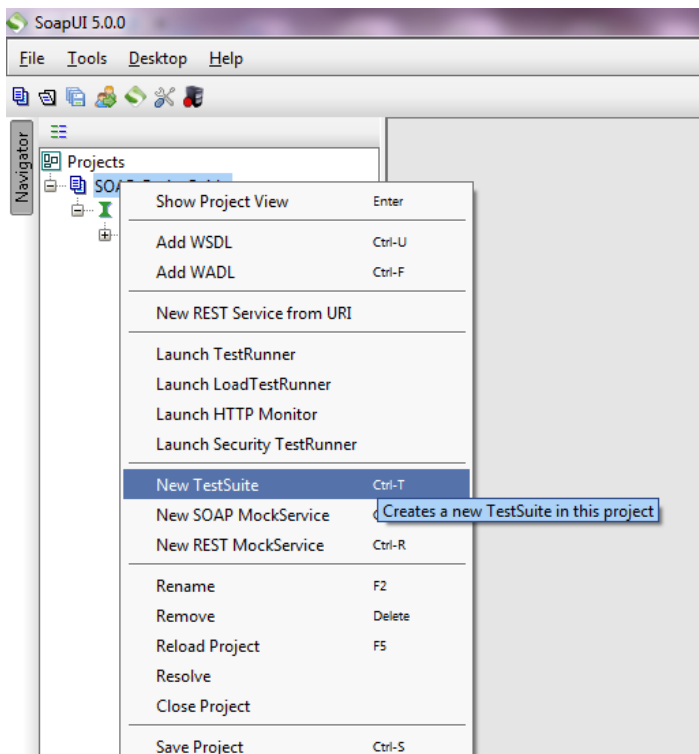
# Creating a Test Suite and Test Case

## 4.1 Creating a Test Suite

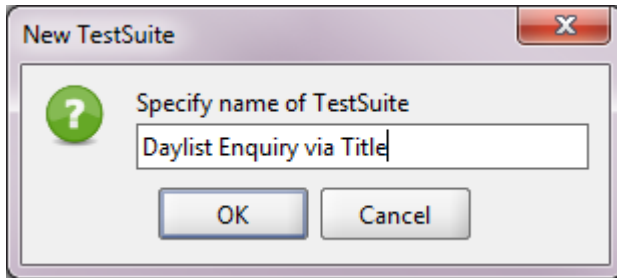
The method in which tests are organised in SoapUI are similar to Quality Centre. You have Test Suites, Test Cases and Test Steps. These can vary in structure to how you want to organise your testing and what we will go through in this guide will be a simple and basic structure.

1. A test suite is a collection of test cases, it can be labelled, given a description and you can also choose to run all tests within a test suite collectively.

You can create a test suite by right clicking on your project and selecting 'New Test Suite' from the menu displayed.

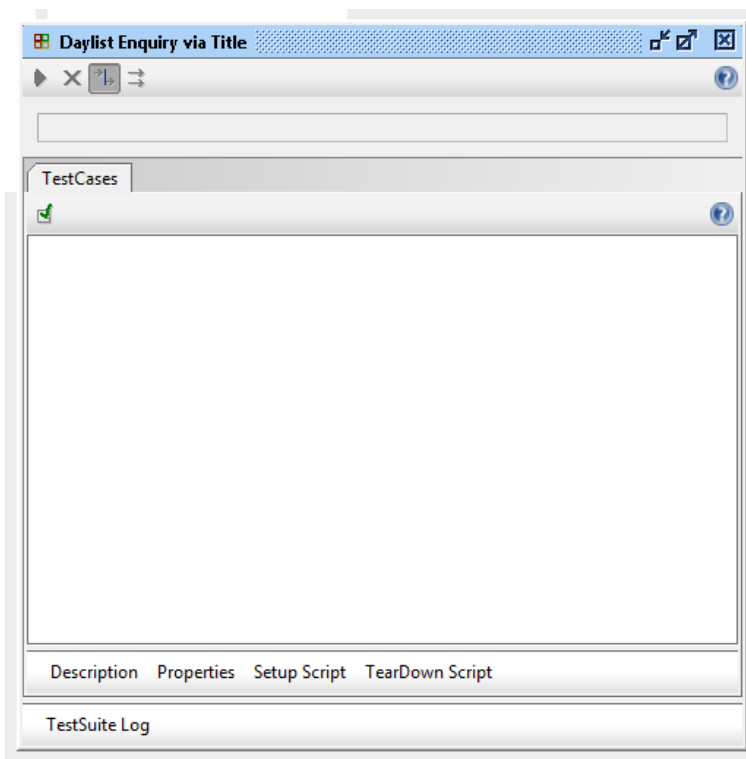


2. Give the test suite a suitable name:



And click 'OK'.

3. When created it will produce this:



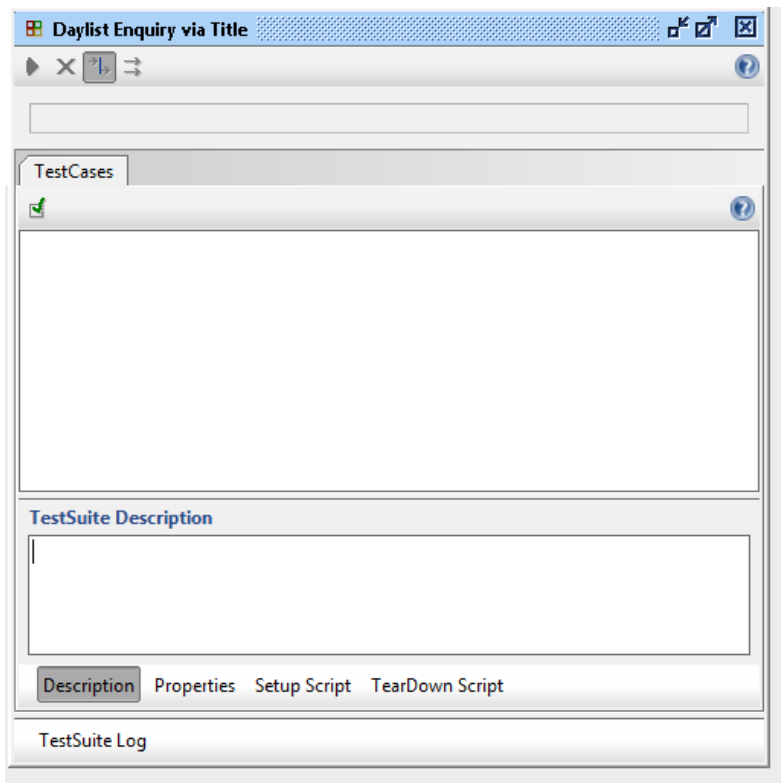
This is the window opened when you open the test suite. Currently this is empty but when we have created test cases, they will be listed in this window.

There are four tabs at the bottom listed as Description, Properties, Setup Script and Teardown script.

As we are just learning the basics we will just add a description to allow us to identify the test suite at a later date.

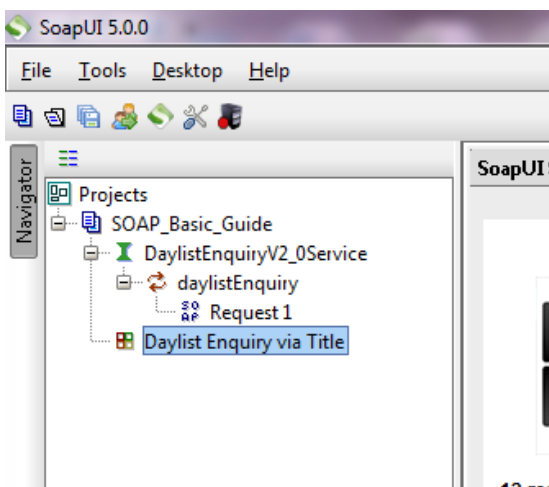


4. Selecting the 'Description' tab at the bottom will expand it into a text field.



Enter a test suite description and close the window using the cross button in the top right corner (this is used to close all windows within SoapUI).

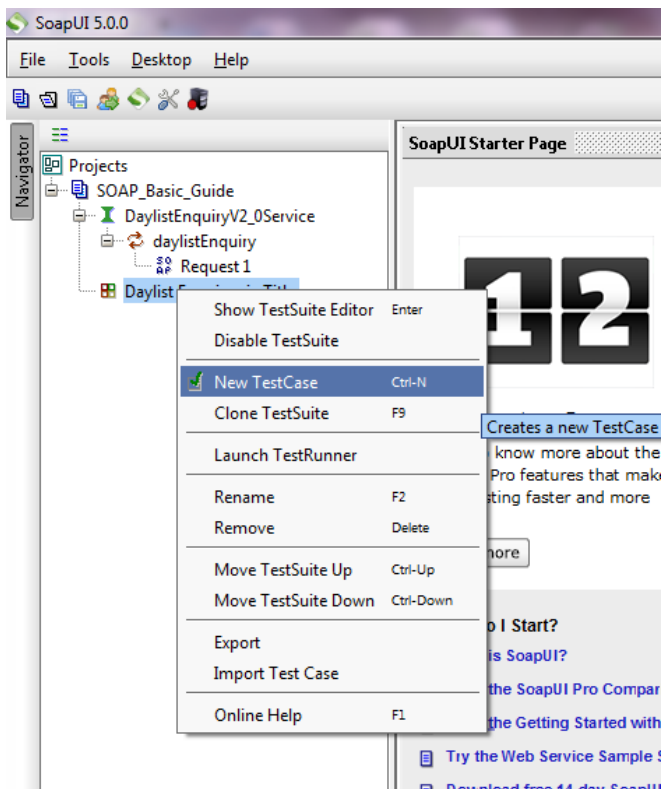
5. You will now have a test suit displayed in the left side menu.



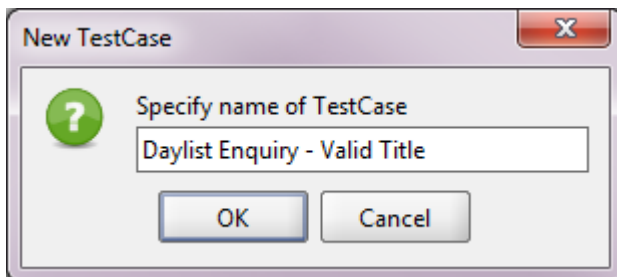
## 4.2 Creating a Test Case

A test case that is created in SoapUI acts like a test in quality centre and will contain test steps to recreate what is being tested.

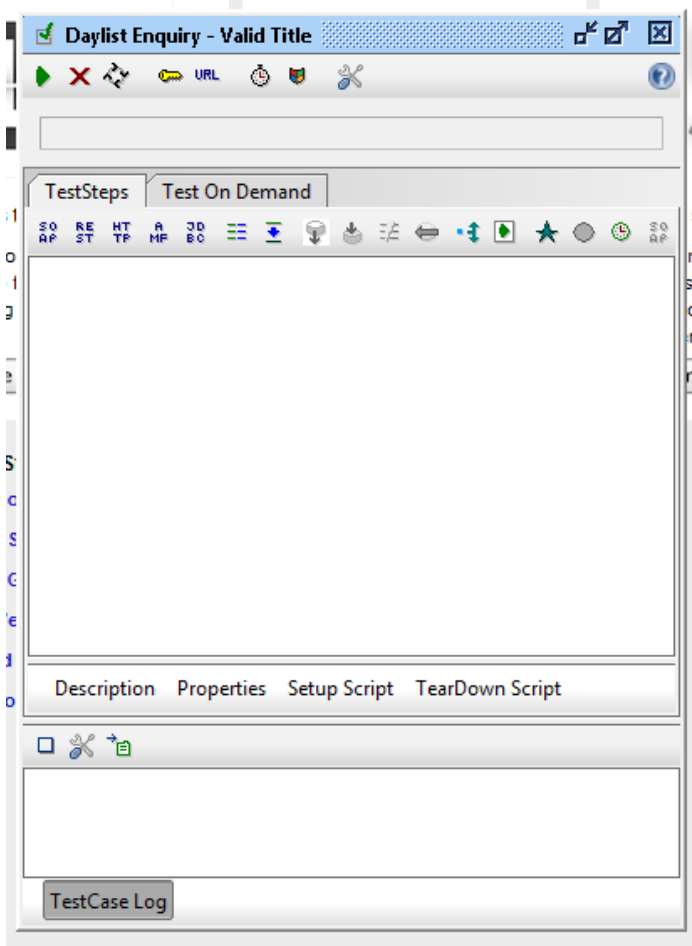
1. To create a test case in the required test suite, you right click on the test suite and select 'New TestCase'.



2. It will bring up a window that will allow you to name the test. This labelling should be clear and concise and allow you to identify what area the test is in.



3. Clicking 'OK' will create the test case and you are provided a similar window to the test suite.



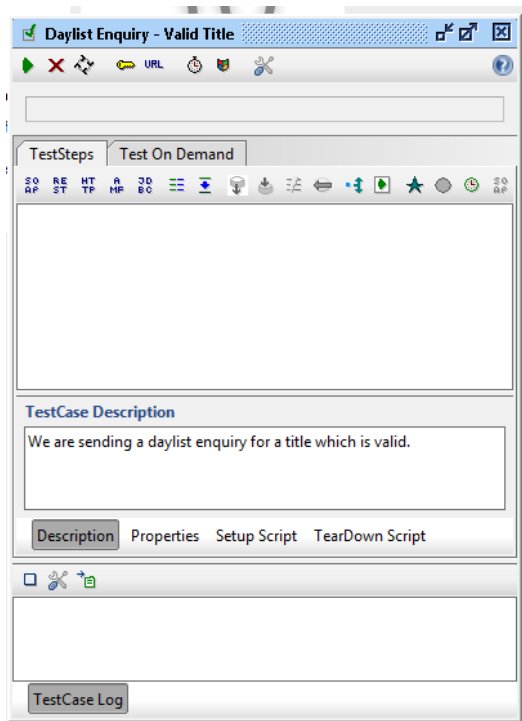
There are two tabs at the top, TestSteps and Test on Demand. During this guide we will only be interested in the TestSteps tab which displays all the test steps for the test case.

As with the test suite the lower tabs are Description, Properties, Setup Script and TearDown Script.

4. In the description we should enter what we need to identify the test case and what it does.

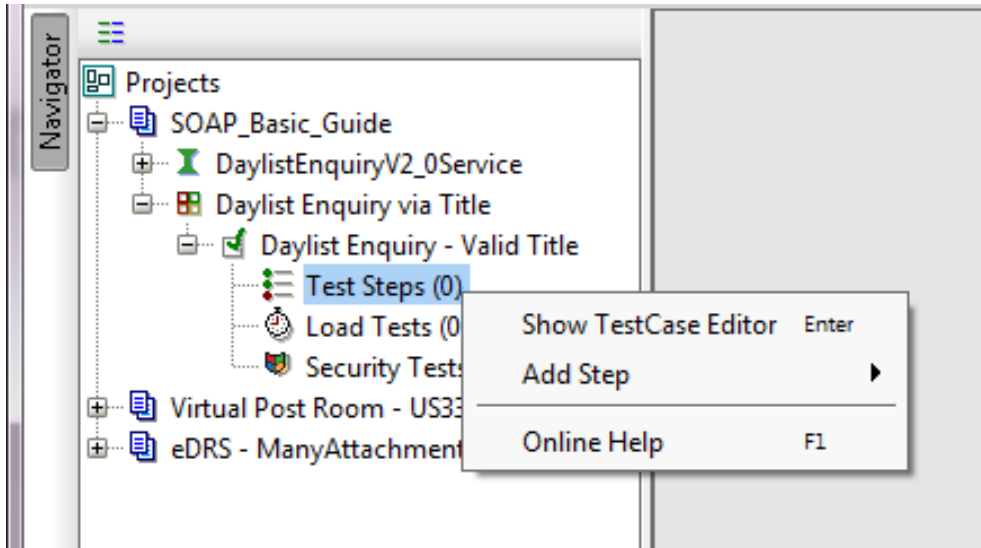
In this instance we are writing a test script that will submit a daylist enquiry with a valid title, so therefore we will put something to that effect in the description field that is displayed when you click on 'Description'.

After the description has been completed just close the test case window.

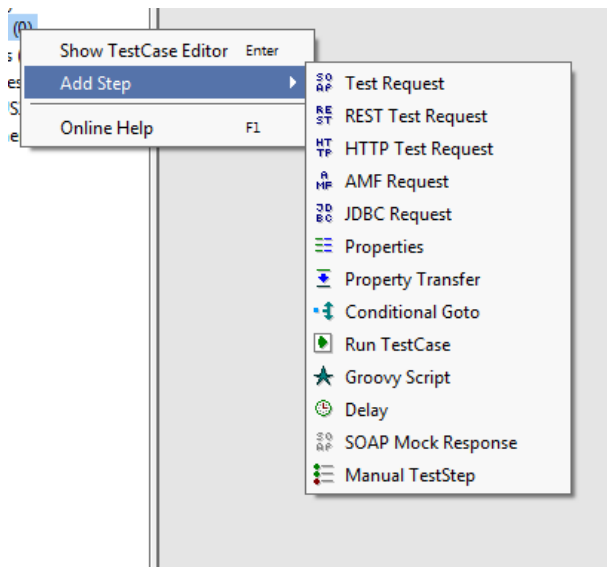


### 4.3 Creating Test Steps – The Initial Request.

1. To create a new test step within our test case, expand the test case and right click on 'Test Steps'



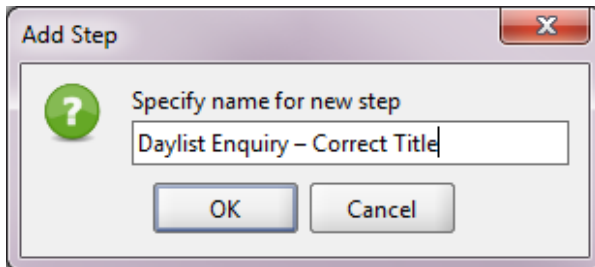
Go to 'Add Step' to display this selection of test steps which you can create.



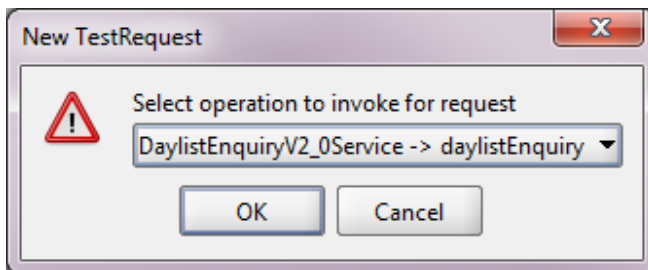
As you can see there is a large selection of test step types, SoapUI is quite accommodating so if you have any web service related testing its worth looking at when considering automation.

For this guide we are going to use a 'Test Request' which handles the SOAP request that we want to do.

2. Clicking on 'Test Request' will display a window for you to name your test step. Once again how you name your test steps are up to how you are organising your project, 'step one' could be sufficient or you might want more of a descriptive title such as 'Daylist Enquiry – Correct Title'.

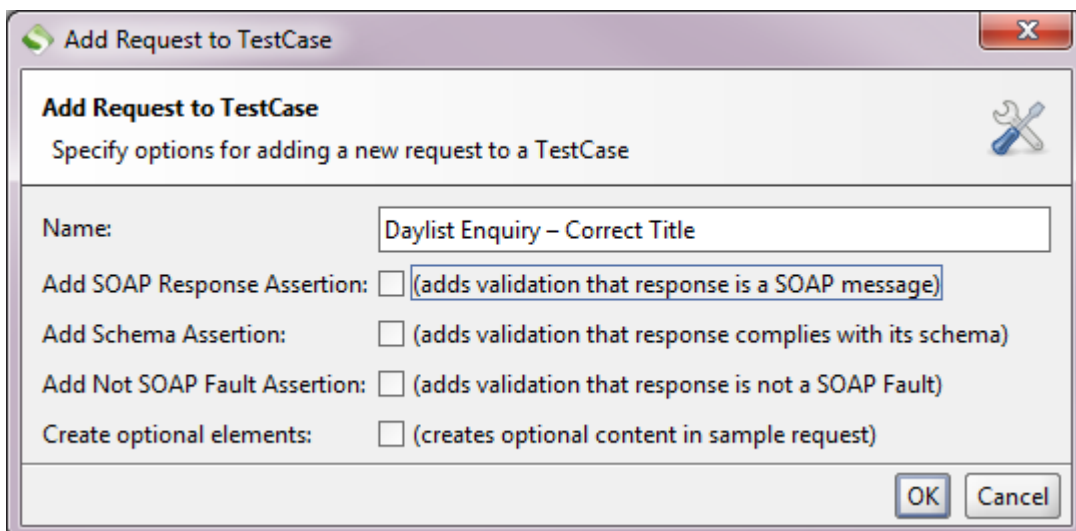


3. Clicking 'OK' will display the next window below. As we have correctly added our service via a WSDL earlier, we can select the type of request we want to use below. At the moment only one will be listed, but as a project can have several its worth checking you have selected the correct one.



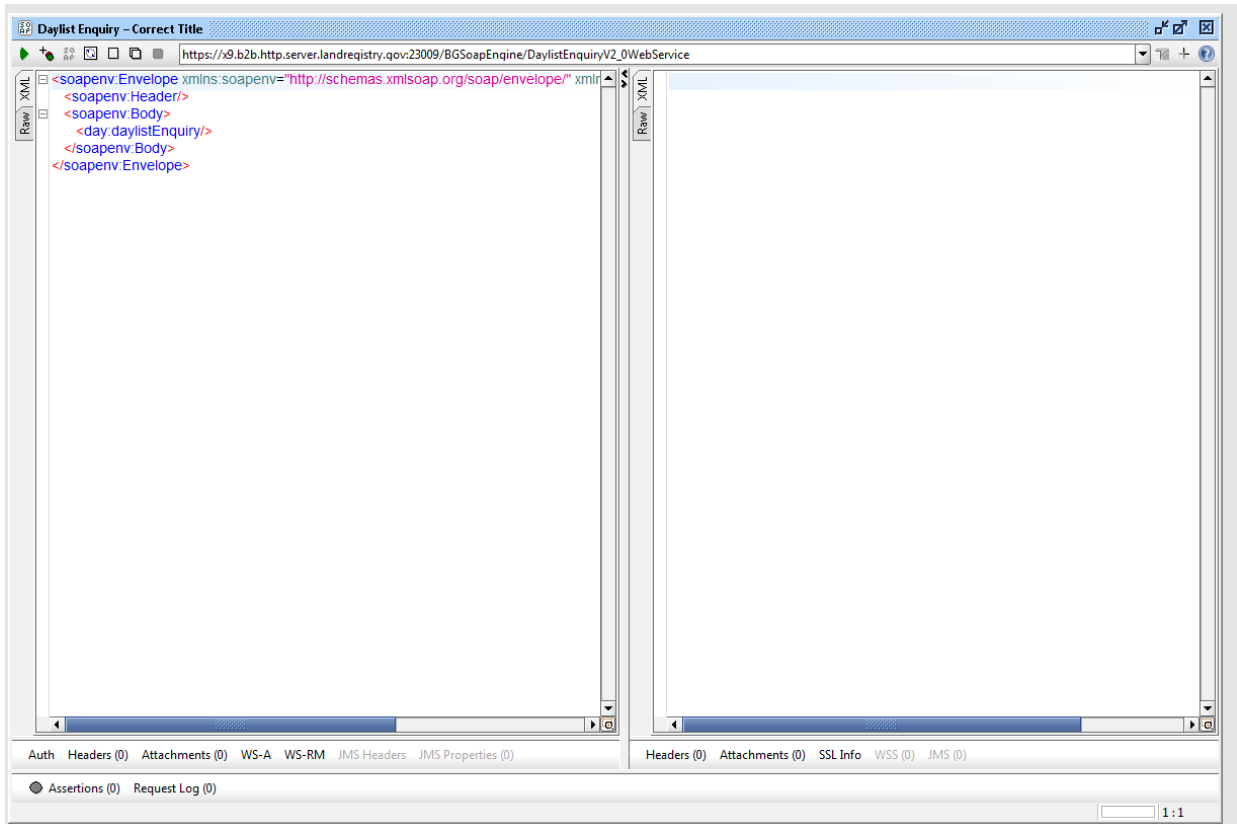
After you have selected the correct operation you need for the request, click 'Ok'.

4. The next window is confirmation of adding the request as a test step. It is also where we add generic assertions that will check that we receive the correct response.



When you get used to creating tests, it will sometimes just be easier to tick the box that you need, but as we will be looking at assertions later just un tick all boxes on the screen and click 'Ok'.

5. We now get presented the request screen where we build the xml request for the test. How the schema has been written and how many optional fields there are will determine what you are presented with. As the main chunk of xml fields in this service are optional we are left with a rather blank request.



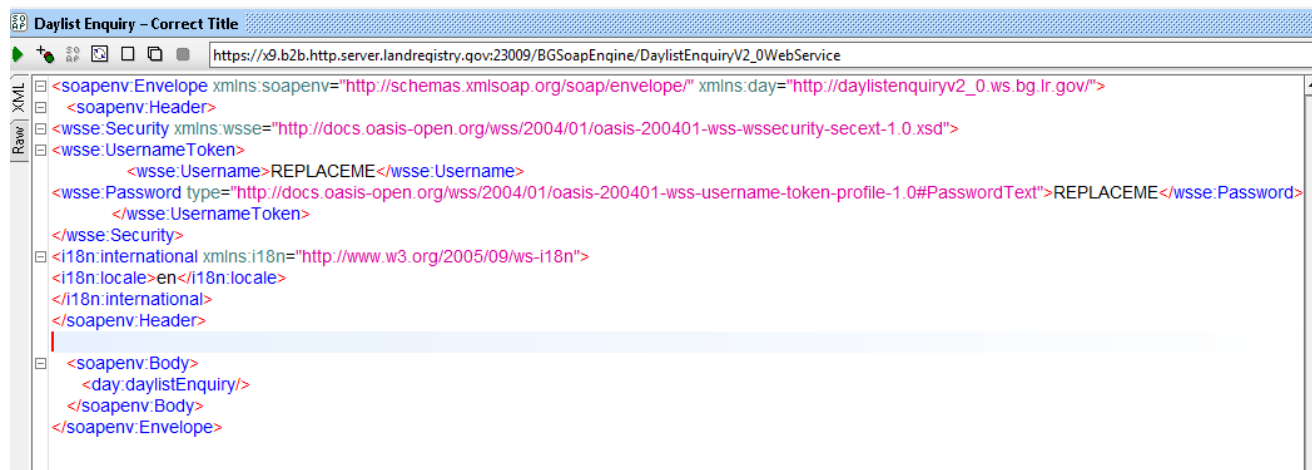
6. Before we go any further with amending the request we have to make sure that it has the log in information. This is displayed like this in the business gateway harness:

In SoapUI you can either set elsewhere in the project, however as we are just going through the basics we will put the required log in XML on the request itself.

This is the XML which is contained in the 'header' and used to gain access to the service:

```
<soapenv:Header>
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  <wsse:UsernameToken>
    <wsse:Username>REPLACEME</wsse:Username>
    <wsse:Password type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">REPLACEME</wsse:Password>
  </wsse:UsernameToken>
</wsse:Security>
<i18n:international xmlns:i18n="http://www.w3.org/2005/09/ws-i18n">
  <i18n:locale>en</i18n:locale>
</i18n:international>
</soapenv:Header>
```

Copy and paste this into the request replacing the <soapenv:Header/> tag that is in the XML:

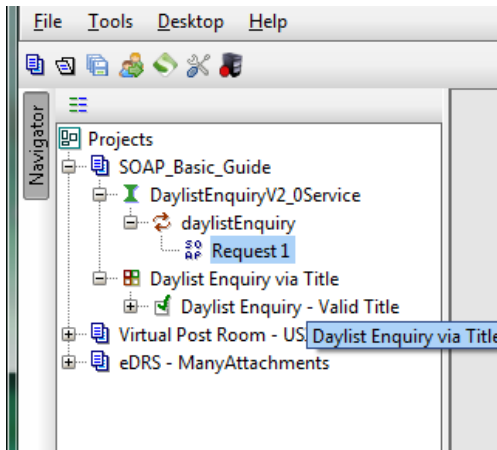


7. Replace the 'REPLACEME' text in the 'Username' and 'Password' XML tags with appropriate login details and you will have the required log in part of the XML request.

NOTE: You will have to change the username according to what environment you are in.

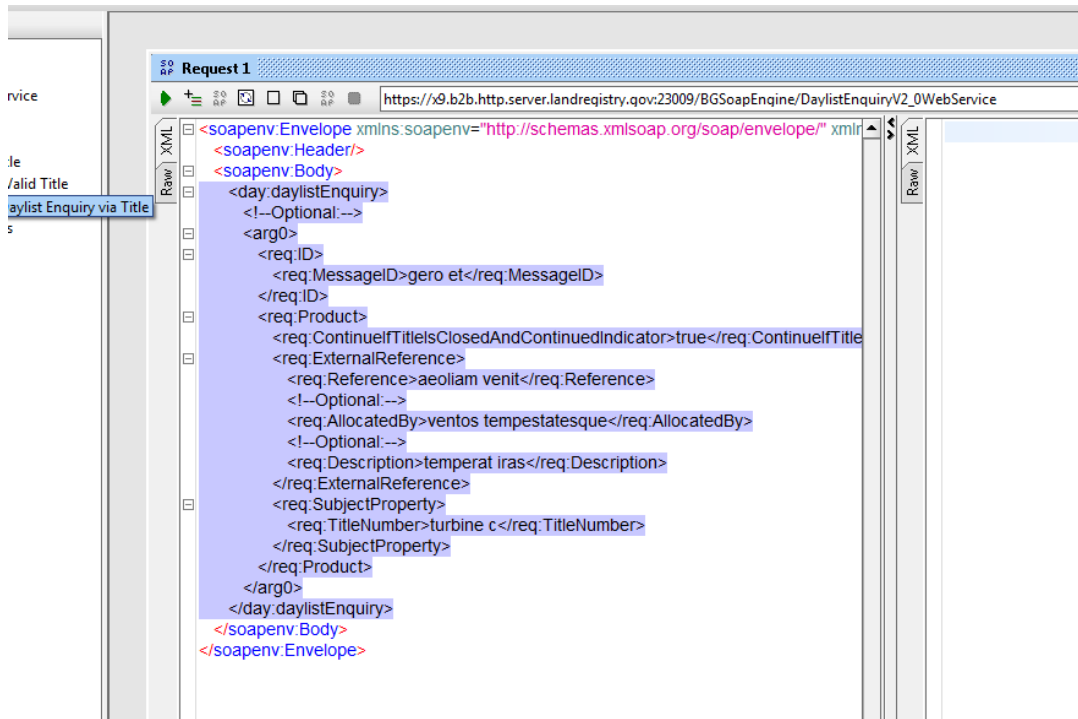


8. Going back to the main part which is still rather empty, thankfully SoapUI have an example of the full xml that can be included in the request. This can be accessed via the 'DaylistEnquiryV2\_0Service' which has a green icon. Expand daylistEnquiry, and double click Request 1.



9. As you can see this request has a bit more XML in it! There are 'Optional' tags which show which bits are not needed to send a request through – as you can see for some reason most of its optional.

Sometimes the request that is generated in your test step will have all the needed XML however in cases like this, we can select the XML we want from the example, copy it (ctrl + c) and then paste it into our test step.



10. At this point it's worth mentioning the importance of XML tags, as you might have noticed looking at the request that an XML tag is used to contain information. As it is used to contain information, it will have a tag at the beginning and at the end.

Such as:

```

<req:Product>
</req:Product>

```

The tags at the end with have a "/" at the beginning to say it's the closing tag.

You can have tags within tags, as such:

```

<req:Product>
  <req:TitleNumber>
  </req:TitleNumber>
</req:Product>

```

If the tags are contained, they have to begin and end within the other set of tags.

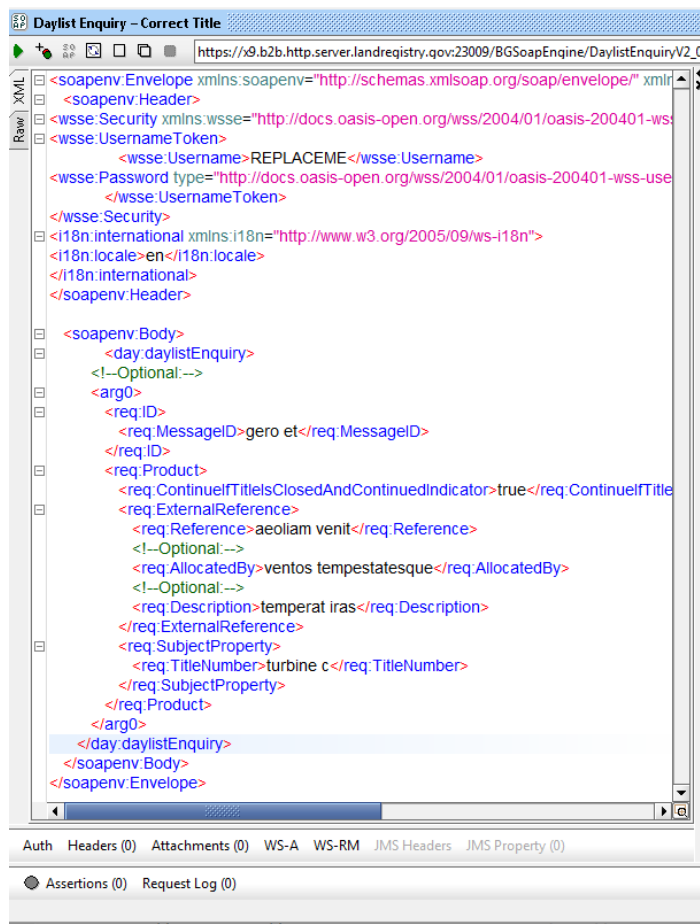
If we get the XML wrong and we were to submit an error response would be returned.

With this in mind, it's important that we get our copied XML inbetween the correct existing XML tags of 'Body'. With this particular request we can just select the single XML tag below

```
<soapenv:Body>
  <day:daylistEnquiry/>
</soapenv:Body>
</soapenv:Envelope>
```

And copy in its place the XML that we copied in step 10 as shown below:

As we are doing a request that only requires us to submit an application with a correct title number we can remove some of the XML which is optional and not needed for this test.



After removing the following XML tags:

- AllocatedBy
- Description

To make sure that we submit XML that our service understands we have to amend the 'namespace' of the XML tags that we are using. Currently SoapUI have set this to a generic 'req' however the Land Registry use the ns1, ns2 and ns3 prefix to help identify their XML tags that may be replicated through other services.

Replacing 'req' with 'ns2' (without out changing any of the XML that surrounds the 'req' part) should be the last changes.

The last bit to change to the overall tags which contain the request itself  
Change:

<day:daylistEnquiry>

to

```
<ns3:daylistEnquiry xmlns:ns2="http://www.oscre.org/ns/eReg-
Final/2011/RequestDaylistEnquiryV2_0"
xmlns:ns3="http://daylistenquiryv2_0.ws.bg.lr.gov/">
```

And the end tag </day:daylist>

to

</ns3:daylistEnquiry>

You should be left with this remaining XML (it's worth checking at this point)

**Note:** As the messageID needs to be unique (If the same one is used it will respond with a previous response) we can instead of putting a static value, enter this code in between the messageID XML tags:

```
${(new Date()).format("yyyyMMdd-HH:mm:ss-SSS")}
```

This will generate a new message ID based on the current date and time that the request is sent, so will always be different. Code can be used throughout SoapUI but this will be the only time we touch upon its use in the basic guide.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:day="http://daylistenquiryv2_0.ws.bg.lr.gov/">
  <soapenv:Header>
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">
<wsse:UsernameToken>
      <wsse:Username>ibell3000</wsse:Username>
<wsse:Password type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-
1.0#PasswordText">green123</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
<i18n:international xmlns:i18n="http://www.w3.org/2005/09/ws-i18n">
<i18n:locale>en</i18n:locale>
</i18n:international>
</soapenv:Header>
  <soapenv:Body>
    <ns3:daylistEnquiry>
      <!--Optional:-->
      <arg0>
        <ns2:ID>
          <ns2:MessageID> ${=(new Date()).format("yyyyMMdd-HH:mm:ss-SSS")}</ns2:MessageID>
        </ns2:ID>
        <ns2:Product>
<ns2:ContinuelfTitlesClosedAndContinuedIndicator>true</ns2:ContinuelfTitlesClosedAndContinuedIndicator>
          <ns2:ExternalReference>
            <ns2:Reference>basicguide</ns2:Reference>
          </ns2:ExternalReference>
          <ns2:SubjectProperty>
            <ns2:TitleNumber>GR519040</ns2:TitleNumber>
          </ns2:SubjectProperty>
        </ns2:Product>
      </arg0>
    </ns3:daylistEnquiry>
  </soapenv:Body>
</soapenv:Envelope>

```

11. The last step we have before we have XML that can be submitted successfully is actually putting the data in. SoapUI has put random text which needs to be replaced with data that will work in our test environment.

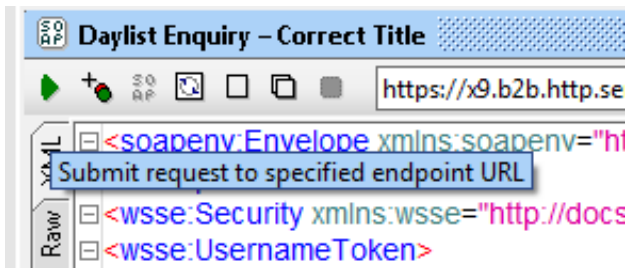
As this guide is working in SYT2 we will need a SYT2 title number. The other two bits of data we need to enter is a message id for 'MessageID' (this needs to be unique and not used previously) and a reference (which can be any text).

'ContinuelfTitlesClosedAndContinuedIndicator' can be left as 'true'.

## 4.4 Creating Test Steps – Checking the Response

1. Now that we have put together the XML request we can check that we will get a response.

Pressing the green play button in the top left hand corner will submit the application.



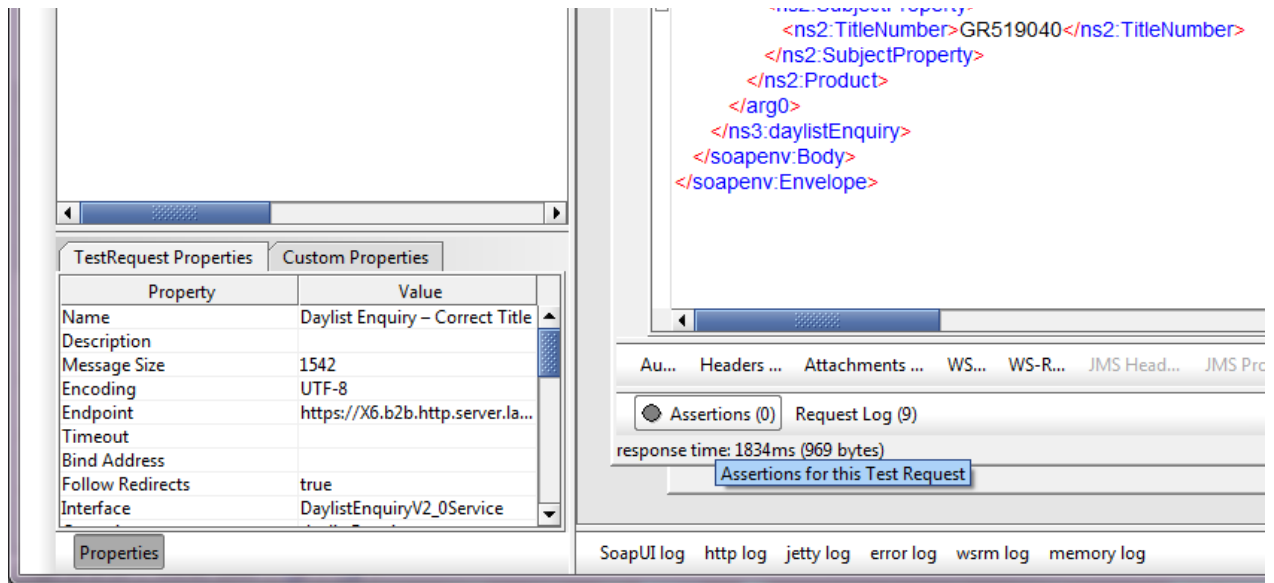
And we will get a response similar to this:



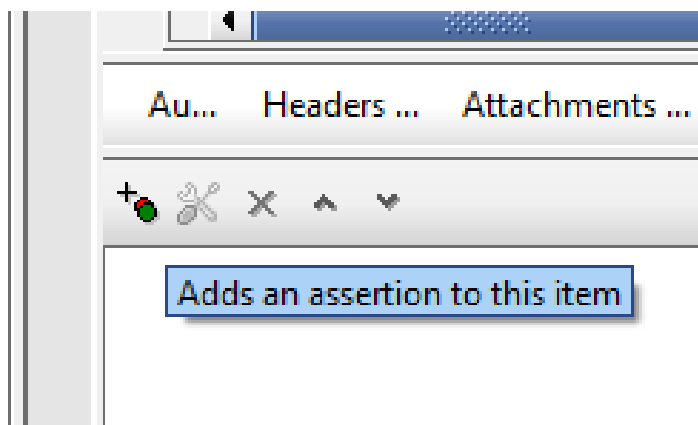
- How you would validate a response in SoapUI will depend on what you are testing. Our test here is to make sure that when using a valid title in a Daylist Enquiries Request, that we get a valid response and not a failure response.

So we will be putting two assertions against our response. One to check that it's a SOAP response (and not a SOAP failure which is an error message response) and the second checks that we have the 'TypeCode' of '30' which indicates the application request was successful.

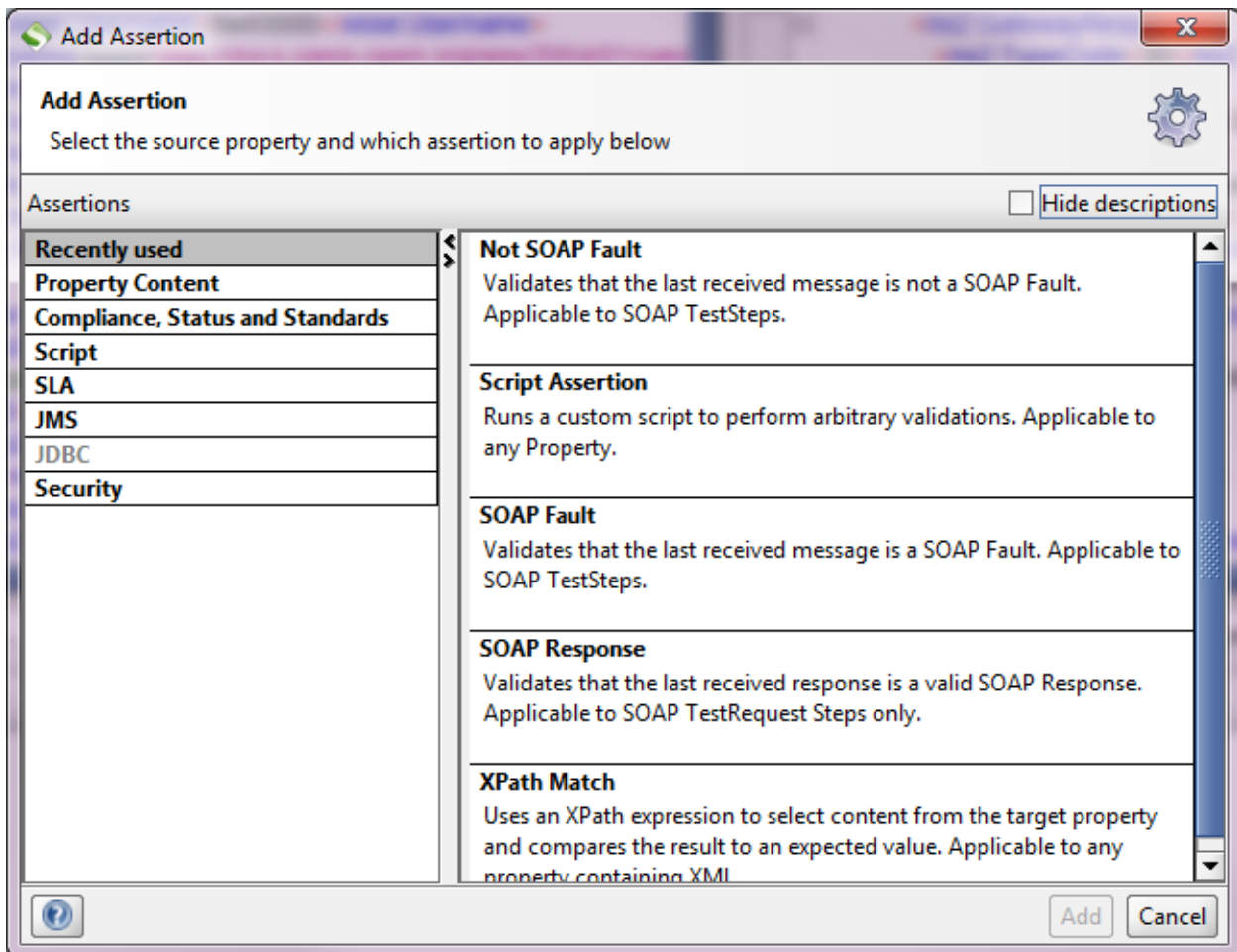
To add an assertion to the test, click 'assertions' on the bottom left hand corner of your test step window:



- It will expand the Assertions part of the window, which will be empty until we add some. Click the button which is two circles and a plus symbol:



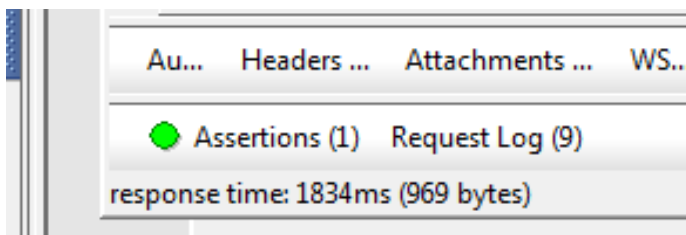
And you will be displayed the 'Add Assertion' window as shown below:



Unlike here as you are working on a fresh install of SoapUI the 'Recently Used' tab will actually be empty.

4. Our first assertion that we will create is in the 'Compliance, Status and Standards' tab, after selecting that tab select 'SOAP Response' and click 'Add'.

Assertions will then minimize and look like this on the bottom menu:



As we already have a response which fulfills the assertion then its currently showing as passed, expanding the 'Assertions' tab once again will display it.

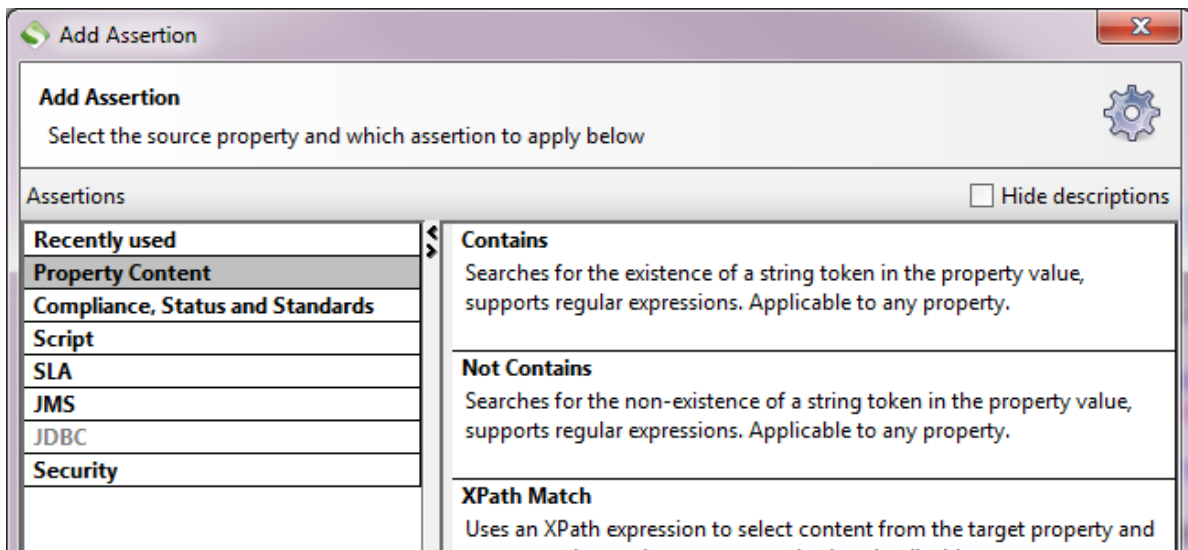




## SOAP Response - VALID

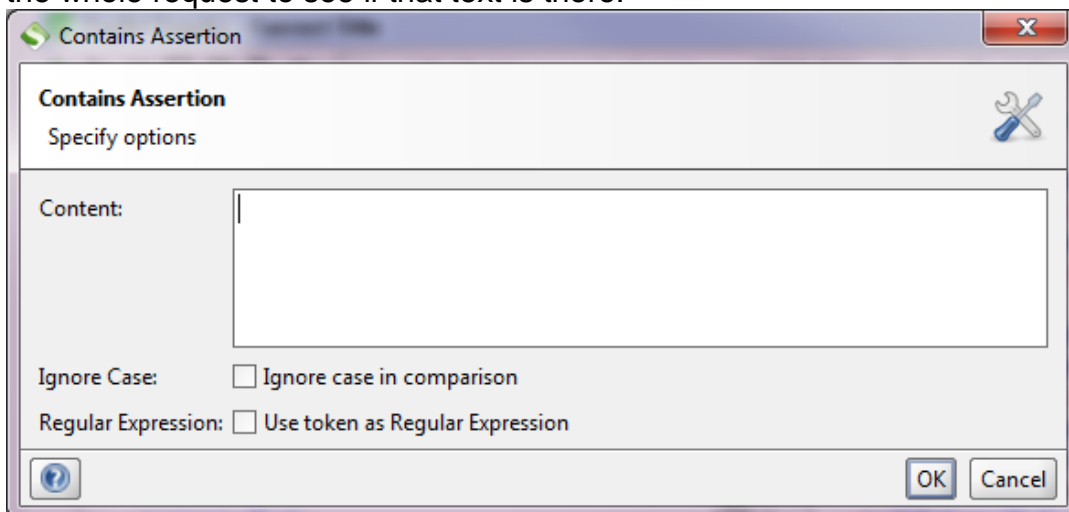
- The second assertion will be checking the content of the response, and whether it has been returned with the Type Code of 30 indicating that the application has been accepted.

Select to add an assertion, and then select 'Property Content' from the left hand menu.



As you can see there are numerous matches you can do (including a handy not contains), however as part of the guide we will be using the more straight forward 'Contains'.

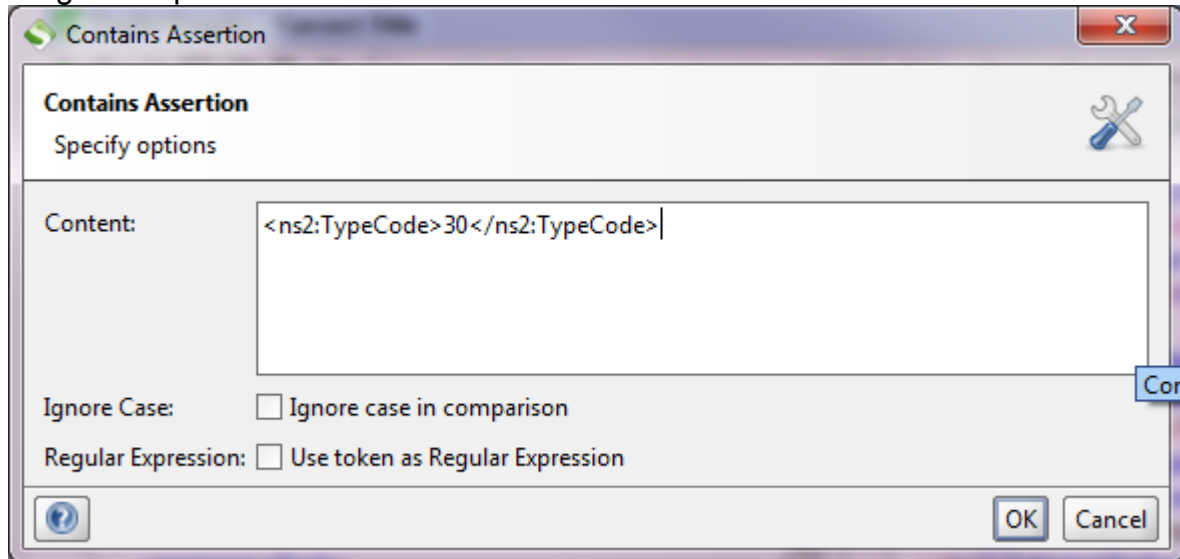
- All we need to do is put the text that we want to confirm is there and it will check against the whole request to see if that text is there.



In this guide we are going to check that the XML tags and the data they contain are correct, so we will enter in the 'Content' field:

```
<ns2:TypeCode>30</ns2:TypeCode>
```

As we are entering the text in the correct case we will not tick the 'Ignore Case' or the 'Regular Expression' tick boxes.



7. Clicking 'Ok' will add this assertion to your test and will show as passed as it has found this content in your current response.
8. What we have now is a working test step which makes up our test. We are testing that when we send a Daylist Enquiry with a valid title number, we receive the expected acceptance response.

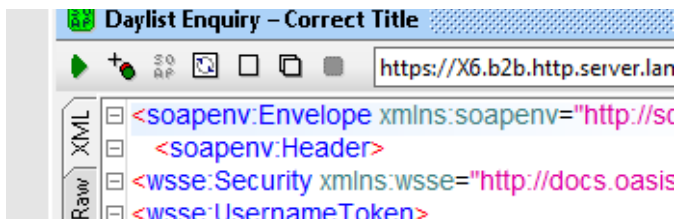
## Running Tests

The test that has been created in this guide can now be run in numerous ways. While they are all valid ways of running the test it will later depend on what tests you want to run, or how many you have.

You can run these tests at different levels according to what results you need.

### 5.1 Running Test Steps Individually

1. We have already looked at running a test step individually, by pressing the green play button the test step itself:

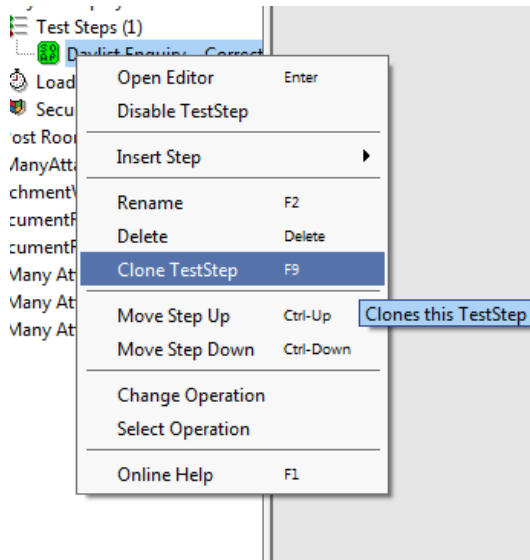


This is best used when either we are checking that the test connects to the web service or when we just want to check that the response of that single test. You will see its passed by seeing that the individual assertions are green and that the response XML message you can see in the right side of the window.

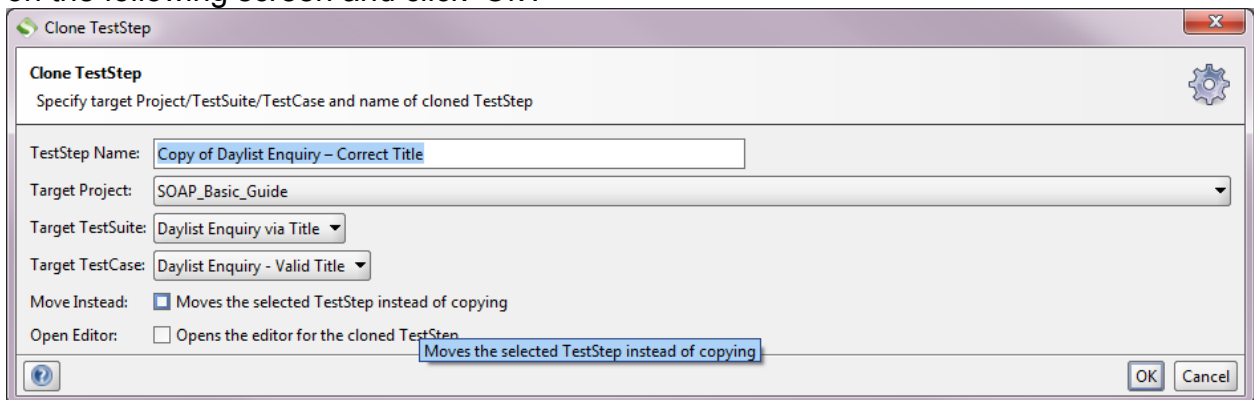
### 5.2 Running Tests as a Test Case

1. While we haven't touched on a test case with multiple steps within this guide, they do exist. These are constructed when a test needs several submissions to complete the submission of an application, and eDRS or ViPR application which has several image attachments is an application which would create a test case with several test steps.
2. To allow us to get a look at this we are going to copy and paste the test step that we have created into the same test case.

Right Click on the test step and select 'Clone TestStep'

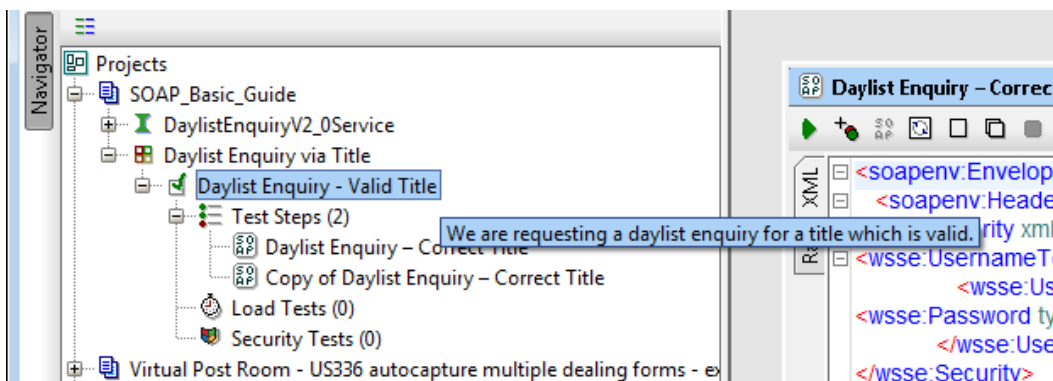


- As we are just copying it into the same test case and project do not change any settings on the following screen and click 'Ok'.

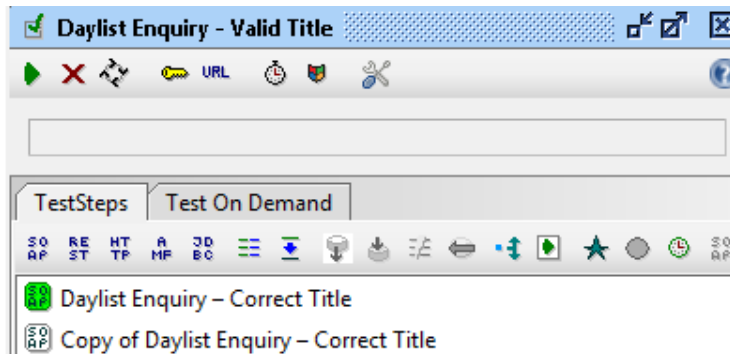


- Your testcase will now have two listed test steps.

Open up the testcase 'Daylist Enquiry - Valid Title' by double clicking on it.

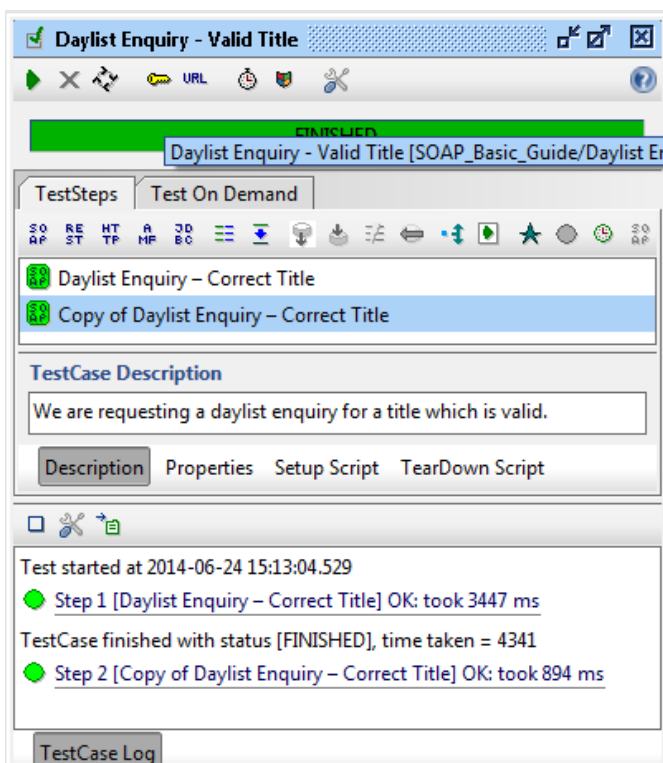


5. Last time when we were creating our test case this window was empty; we now have our two test steps displayed below.

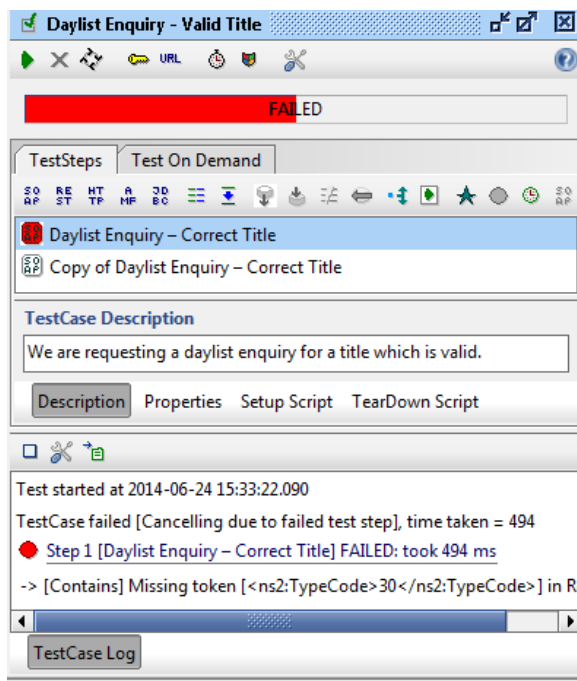


In the top left hand corner we have the green play button, this is consistently for running whatever is displayed, whether a test step, case or suite.

Clicking the play button will run the test steps sequentially, display the passing result as green as well as provide links to the actual request message and response as shown in the picture below:



To access the request and response XML, just click the step you want to view on the bottom part of the TestCase window.

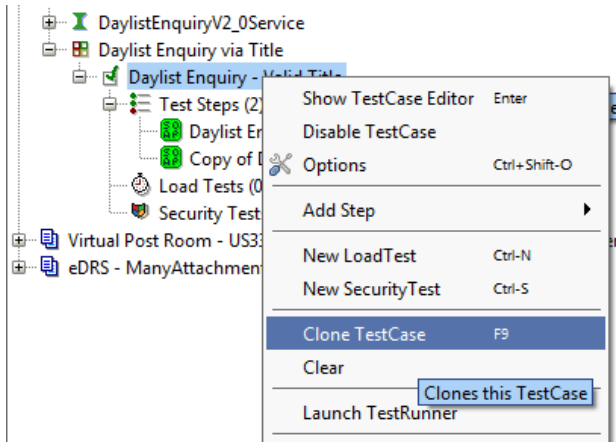


If a step were to fail it would display like to the left, with the test stopping at the first failed step unless instructed otherwise.

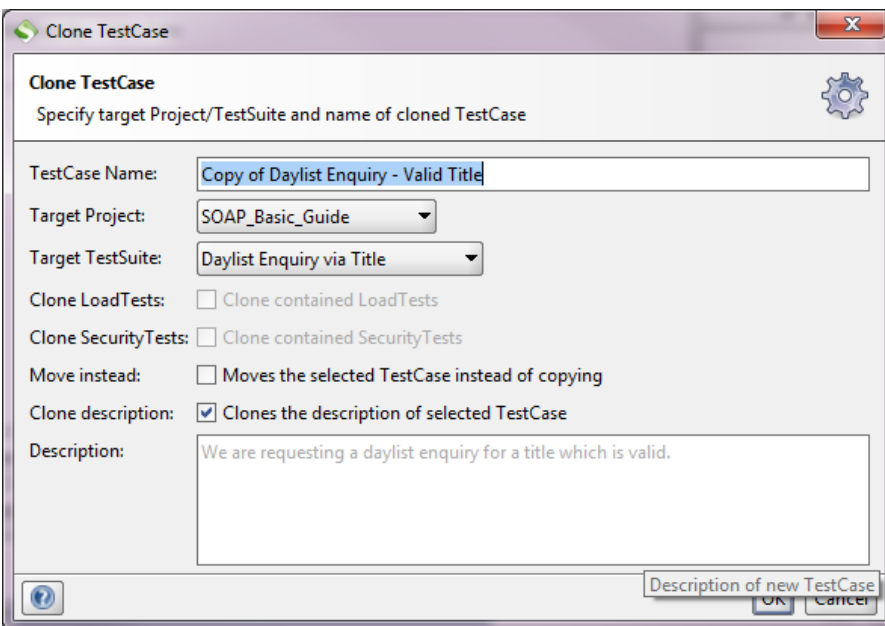
### 5.3 Running Tests as a Test Suite

The more tests you build, whether for a regression pack or created from a single user story the more likely you will be running tests as full test suites and projects.

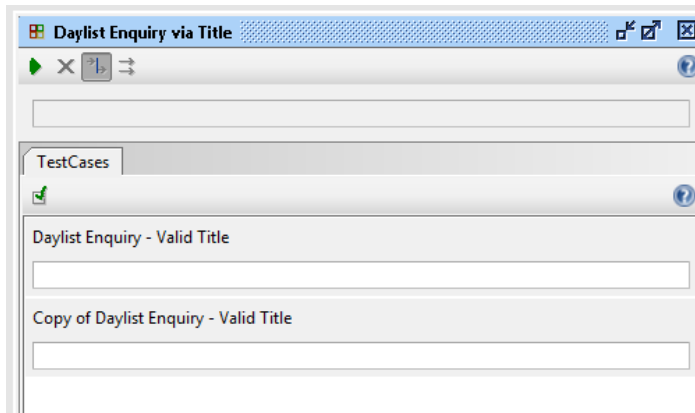
1. To show this functionality, we shall create a second testcase by right clicking on it and selecting to clone it:



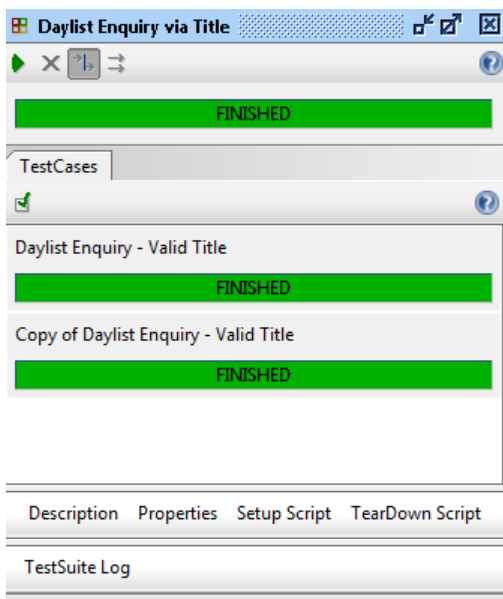
As previously, do not change any details on the clone window and just click 'Ok'.



2. Double click on the testsuite 'Daylist Enquiry via Title' which will open the test suite now containing two test cases.

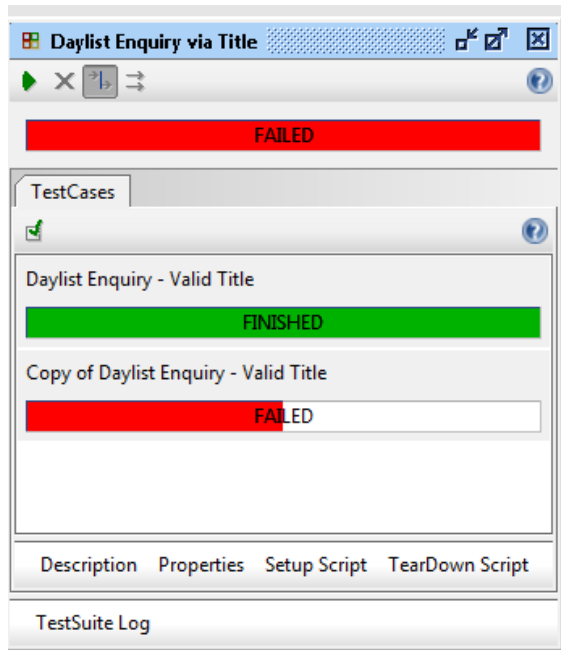


Pressing the play button will run all the test cases that the test suite contains and display the passing results like this:



Failing results will display what test case has failed like below:





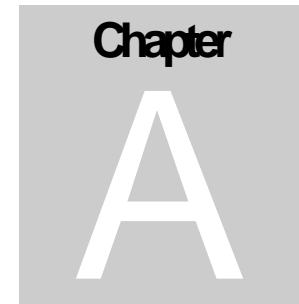
As well as test suite level, you can also run tests at project level, which will act and look like what we have done here.

## Conclusion

In this basic guide to SoapUI we have looked at web services and how the Land Registry uses them and how to create a test with static data that can test that they are working correctly.

We have run that test at different levels show what the passed and failed tests display as.

As SoapUI is a rather vast system and this guide was to introduce you to it and help you start utilising the basics that help you see benefits that it could bring to your work.



# Appendix

## A.1 Table of Environment Placeholders for WSDLs

| ID | REGION | VIA_WEBSEAL | STUB | PLACE HOLDER  |
|----|--------|-------------|------|---|
| 1  | SYT1   | N           | N    | <a href="https://X2.b2b.http.server.landregistry.gov:15009/">https://X2.b2b.http.server.landregistry.gov:15009/</a>       |
| 2  | SYT1   | N           | Y    | <a href="https://X2.node2.b2b.server.landregistry.gov:15069/">https://X2.node2.b2b.server.landregistry.gov:15069/</a>     |
| 3  | SYT1   | Y           | N    | <a href="https://192.168.74.1:444/x2b2b/">https://192.168.74.1:444/x2b2b/</a>   |
| 4  | SYT1   | Y           | Y    | NOT CONFIGURED  |
| 5  | SYT2   | N           | N    | <a href="https://X6.b2b.http.server.landregistry.gov:19009/">https://X6.b2b.http.server.landregistry.gov:19009/</a>       |
| 6  | SYT2   | N           | Y    | <a href="https://X6.node2.b2b.server.landregistry.gov:19069/">https://X6.node2.b2b.server.landregistry.gov:19069/</a>     |
| 7  | SYT2   | Y           | N    | <a href="https://192.168.74.1:444/x6b2b/">https://192.168.74.1:444/x6b2b/</a>   |
| 8  | SYT2   | Y           | Y    | NOT CONFIGURED  |
| 9  | SYT3   | N           | N    | <a href="https://X9.b2b.http.server.landregistry.gov:23009/">https://X9.b2b.http.server.landregistry.gov:23009/</a>       |
| 10 | SYT3   | N           | Y    | <a href="https://X9.node2.b2b.server.landregistry.gov:23069/">https://X9.node2.b2b.server.landregistry.gov:23069/</a>     |
| 11 | SYT3   | Y           | N    | <a href="https://192.168.74.1:444/x9b2b/">https://192.168.74.1:444/x9b2b/</a>   |
| 12 | SYT3   | Y           | Y    | NOT CONFIGURED  |
| 13 | UAT1   | N           | N    | <a href="https://Y0.NODE1.B2B.SERVER.LANDREGISTRY.GOV:13049/">https://Y0.NODE1.B2B.SERVER.LANDREGISTRY.GOV:13049/</a>     |
| 14 | UAT1   | N           | Y    | NOT CONFIGURED  |
| 15 | UAT1   | Y           | N    | <a href="https://uatbusinessgateway.landregistry.gov.uk/y0b2b/">https://uatbusinessgateway.landregistry.gov.uk/y0b2b/</a> |
| 16 | UAT1   | Y           | Y    | NOT CONFIGURED  |
| 17 | UAT2   | N           | N    | <a href="https://Y0.NODE1.B2B.SERVER.LANDREGISTRY.GOV:17049/">https://Y0.NODE1.B2B.SERVER.LANDREGISTRY.GOV:17049/</a>     |
| 18 | UAT2   | N           | Y    | <a href="https://Y0.NODE2.B2B.SERVER.LANDREGISTRY.GOV:17069/">https://Y0.NODE2.B2B.SERVER.LANDREGISTRY.GOV:17069/</a>     |
| 19 | UAT2   | Y           | N    | <a href="https://uatbusinessgateway.landregistry.gov.uk/y4b2b/">https://uatbusinessgateway.landregistry.gov.uk/y4b2b/</a> |
| 20 | UAT2   | Y           | Y    | NOT CONFIGURED  |

**A.2 WSDL Addresses**

| <b>Web Service</b>                               | <b>WSDL</b>   |
|--|---|
| Daylist Enquiry 2.0                              | BGSoapEngine/DaylistEnquiryV2_0WebService/WEB-INF/wsdl/DaylistEnquiryV2_0Service.wsdl   |
| Daylist Enquiry 2.0 Poll Request                 | BGSoapEngine/DaylistEnquiryV2_0PollRequestWebService/WEB-INF/wsdl/DaylistEnquiryV2_0PollRequestService.wsdl                     |
| Enquiry by Property Description 2.0              | BGSoapEngine/EnquiryByPropertyDescriptionV2_0WebService/WEB-INF/wsdl/SearchByPropertyDescriptionV2_0Service.wsdl                |
| Enquiry by Property Description 2.0 Poll Request | BGSoapEngine/EnquiryByPropertyDescriptionV2_0PollWebService/WEB-INF/wsdl/SearchByPropertyDescriptionV2_0PollRequestService.wsdl |
| Land Charges Bankruptcy Search 2.0               | BGSoapEngine/BankruptcySearchV2_0WebService/WEB-INF/wsdl/BankruptcySearchV2_0Service.wsdl                                       |
| Land Charges Bankruptcy Search 2.0 Poll Request  | BGSoapEngine/BankruptcySearchV2_0PollRequestWebService/WEB-INF/wsdl/BankruptcySearchV2_0PollRequestService.wsdl                 |
| Land Charges Bankruptcy Search 2.1               | BGSoapEngine/BankruptcySearchV2_1WebService/WEB-INF/wsdl/BankruptcySearchV2_1Service.wsdl                                       |
| Land Charges Bankruptcy Search 2.1 Poll Request  | BGSoapEngine/BankruptcySearchV2_1PollRequestWebService/WEB-INF/wsdl/BankruptcySearchV2_1PollRequestService.wsdl                 |
| Land Charges Full Search 2.0                     | BGSoapEngine/FullSearchV2_0WebService/WEB-INF/wsdl/FullSearchV2_0Service.wsdl   |
| Land Charges Full Search 2.0 Poll Request        | BGSoapEngine/FullSearchV2_0PollRequestWebService/WEB-INF/wsdl/FullSearchV2_0PollRequestService.wsdl                             |
| Land Charges Full Search 2.1                     | BGSoapEngine/FullSearchV2_1WebService/WEB-INF/wsdl/FullSearchV2_1Service.wsdl   |

|  |   |
|--|---|
| Land Charges Full Search<br>2.1 Poll Request               | BGSoapEngine/FullSearchV2_1PollRequestWebService/WEB-INF/wsdl/FullSearchV2_1PollRequestService.wsdl                     |
| Official Copy Title Known 2.0                              | BGSoapEngine/OfficialCopyTitleKnownV2_0WebService/WEB-INF/wsdl/OC1TitleKnownV2_0Service.wsdl                            |
| Official Copy Title Known 2.0<br>Poll Request              | BGSoapEngine/OC1TitleKnownV2_0PollRequestWebService/WEB-INF/wsdl/OC1TitleKnownV2_0PollRequestService.wsdl               |
| Official Copy Title Known 2.1                              | BGSoapEngine/OfficialCopyTitleKnownV2_1WebService/WEB-INF/wsdl/OC1TitleKnownV2_1Service.wsdl                            |
| Official Copy Title Known 2.1<br>Poll Request              | BGSoapEngine/OC1TitleKnownV2_1PollRequestWebService/WEB-INF/wsdl/OC1TitleKnownV2_1PollRequestService.wsdl               |
| Official Search of Part with<br>Priority 2.0               | BGSoapEngine/OfficialSearchOfPartV2_0WebService/WEB-INF/wsdl/OfficialSearchOfPartV2_0Service.wsdl                       |
| Official Search of Part with<br>Priority 2.0 Poll Request  | BGSoapEngine/OfficialSearchOfPartV2_0PollRequestWebService/WEB-INF/wsdl/OfficialSearchOfPartV2_0PollRequestService.wsdl |
| Official Search of Part with<br>Priority 2.1               | BGSoapEngine/OfficialSearchOfPartV2_1WebService/WEB-INF/wsdl/OfficialSearchOfPartV2_1Service.wsdl                       |
| Official Search of Part with<br>Priority 2.1 Poll Request  | BGSoapEngine/OfficialSearchOfPartV2_1PollRequestWebService/WEB-INF/wsdl/OfficialSearchOfPartV2_1PollRequestService.wsdl |
| Official Search of Whole with<br>Priority 2.0              | BGSoapEngine/OfficialSearchV2_0WebService/WEB-INF/wsdl/OfficialSearchV2_0Service.wsdl                                   |
| Official Search of Whole with<br>Priority 2.0 Poll Request | BGSoapEngine/OfficialSearchV2_0PollRequestWebService/WEB-INF/wsdl/OfficialSearchV2_0PollRequestService.wsdl             |
| Official Search of Whole with<br>Priority 2.1              | BGSoapEngine/OfficialSearchV2_1WebService/WEB-INF/wsdl/OfficialSearchV2_1Service.wsdl                                   |
| Official Search of Whole with                              | BGSoapEngine/OfficialSearchV2_1PollRequestWebService/WEB-   |

|  |   |
|--|---|
| Priority 2.1 Poll Request                              | INF/wsdl/OfficialSearchV2_1PollRequestService.wsdl  |
| Register Extract Service 2.0                           | BGSoapEngine/OfficialCopyWithSummaryV2_0WebService/WEB-INF/wsdl/OCWithSummaryV2_0Service.wsdl                                 |
| Register Extract Service 2.0<br>Poll Request           | BGSoapEngine/OfficialCopyWithSummaryV2_0PollRequestWebService/WEB-INF/wsdl/OCWithSummaryV2_0PollRequestService.wsdl           |
| Register Extract Service 2.1                           | BGSoapEngine/OfficialCopyWithSummaryV2_1WebService/WEB-INF/wsdl/OCWithSummaryV2_1Service.wsdl                                 |
| Register Extract Service 2.1<br>Poll Request           | BGSoapEngine/OfficialCopyWithSummaryV2_1PollRequestWebService/WEB-INF/wsdl/OCWithSummaryV2_1PollRequestService.wsdl           |
| Search of the Index Map 2.0                            | BGSoapEngine/SearchOfIndexMapV2_0WebService/WEB-INF/wsdl/SearchOfIndexMapV2_0Service.wsdl                                     |
| Search of the Index Map 2.0<br>Poll Request            | BGSoapEngine/SearchOfIndexMapV2_0PollRequestWebService/WEB-INF/wsdl/SearchOfIndexMapV2_0PollRequestService.wsdl               |
| Search of the Index Map 2.1                            | BGSoapEngine/SearchOfIndexMapV2_1WebService/WEB-INF/wsdl/SearchOfIndexMapV2_1Service.wsdl                                     |
| Search of the Index Map 2.1<br>Poll Request            | BGSoapEngine/SearchOfIndexMapV2_1PollRequestWebService/WEB-INF/wsdl/SearchOfIndexMapV2_1PollRequestService.wsdl               |
| <b>eDRS</b>  |   |
| Application to Change the<br>Register 1.0              | ECDRS_SoapEngine/EDocumentRegistrationV1_0WebService/WEB-INF/wsdl/EDocumentRegistrationV1_0Service.wsdl                       |
| Application to Change the<br>Register 1.0 Poll Request | ECDRS_SoapEngine/EDocumentRegistrationV1_0PollRequestWebService/WEB-INF/wsdl/EDocumentRegistrationV1_0PollRequestService.wsdl |
| Attachment 1.0   | ECDRS_SoapEngine/AttachmentV1_0WebService/WEB-INF/wsdl/AttachmentV1_0Service.wsdl   |
| Attachment 1.0 Poll Request                            | ECDRS_SoapEngine/AttachmentV1_0PollRequestWebService/WEB-INF/wsdl/AttachmentV1_0PollRequestService.wsdl                       |

|   |   |
|---|---|
| Correspondence 1.0                                  | ECDRS_SoapEngine/CorrespondenceV1_0PollRequestWebService/WEB-INF/wsdl/CorrespondenceV1_0PollRequestService.wsdl               |
| Early Completion 1.0                                | ECDRS_SoapEngine/EarlyCompletionV1_0PollRequestWebService/WEB-INF/wsdl/EarlyCompletionV1_0PollRequestService.wsdl             |
| Application to Change the Register 2.0              | ECDRS_SoapEngine/EDocumentRegistrationV2_0WebService/WEB-INF/wsdl/EDocumentRegistrationV2_0Service.wsdl                       |
| Application to Change the Register 2.0 Poll Request | ECDRS_SoapEngine/EDocumentRegistrationV2_0PollRequestWebService/WEB-INF/wsdl/EDocumentRegistrationV2_0PollRequestService.wsdl |
| Attachment 2.0                                      | ECDRS_SoapEngine/AttachmentV2_0WebService/WEB-INF/wsdl/AttachmentV2_0Service.wsdl   |
| Attachment 2.0 Poll Request                         | ECDRS_SoapEngine/AttachmentV2_0PollRequestWebService/WEB-INF/wsdl/AttachmentV2_0PollRequestService.wsdl                       |
| <b>Verification Services</b>                        |   |
| Online Ownership Verification 1.0                   | EOOV_SoapEngine/OnlineOwnershipVerificationV1_0WebService/WEB-INF/wsdl/OnlineOwnershipVerificationV1_0Service.wsdl            |