

Testing the APIs using SoapUI

[Go to full article](#)

Overview

This article contains instructions on how to test the Emersion APIs using SOAP UI.

SoapUI is a [freeware client-based app](#) that gives developers and IT staff the ability to test APIs. In this article, we demonstrate how to use SoapUI to test a given API end point.

This article in no way warrants or endorses SoapUI as an application and Emersion do not provide support for it should you decide to use it for your own testing and troubleshooting purposes. However if you want to test an API function or call, this tool can assist you to do that. SoapUI also allows you to view the methods contained within the API and learn the structure of a request and a response without having to write to a system.

For general advice on testing with Emersion and the creation of test accounts and services, please [read this article](#).

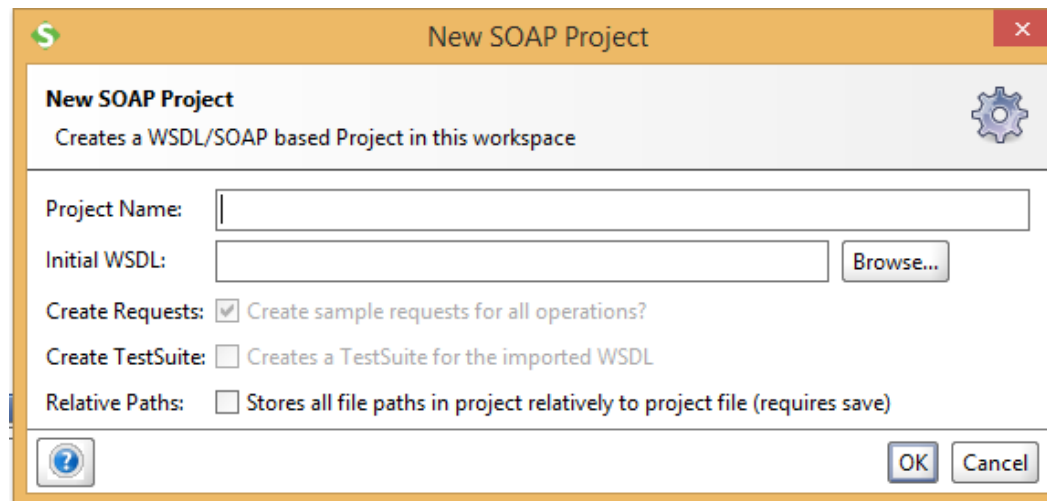
To begin, open SoapUI.

Create New Project

The first step is to set up a new project. Click the SOAP button in the ribbon menu to begin a new Soap project.



Give the project a name.



In the **Initial WSDL** field, enter the file path of the wsdl file you are wanting to test.

 This URL should be able to be plugged into your browser and load a definitions file. If not, the URL is incorrect.

Click the **OK** button to save.

Once the definitions have been loaded, your screen should look like the one below. The Project and API calls will be listed on the left-hand side, and the input parameters on the right.



Now that you have finished setting up the project, you can move onto configuration the authentication.

Configure the Authentication

We secure the API layer by incorporating the access credentials from the external user account into the SOAP envelope header. If the user is authenticated and authorised (meaning your configuration is correct), the API function will be served. If not, an appropriate error message will be provided.

Follow the steps below to configure the authentication for a SOAP project.

1. Right-click the SOAP project and select **Show Project View**.
2. Click on the **WS-Security Configurations** tab.
3. Select the sub tab **Outgoing WS-Security Configurations**.
4. Add a new configuration with any name. Select this configuration and add an entry by pressing the + (plus) button beneath it.
5. Select the type of entry as *Username* and provide the External User user name and password.
6. Tick the **Add Nonce**
7. Tick **Add Created**
8. select the password type as **Password Digest**.
9. Close the **Project View** window.

Next:

1. Click the **Auth** button at the bottom of the request window to open the authentication/authorisation window.
2. Choose the **Basic** authorisation type from the drop down
3. select the **Outgoing WSS** which you created in the *Outgoing WS-Security Configuration* section of the Project View.

You can show/hide the authorisation dialogue by clicking the **Auth (Basic)** button at any time.

With the authentication now configured, you are ready to submit a request.

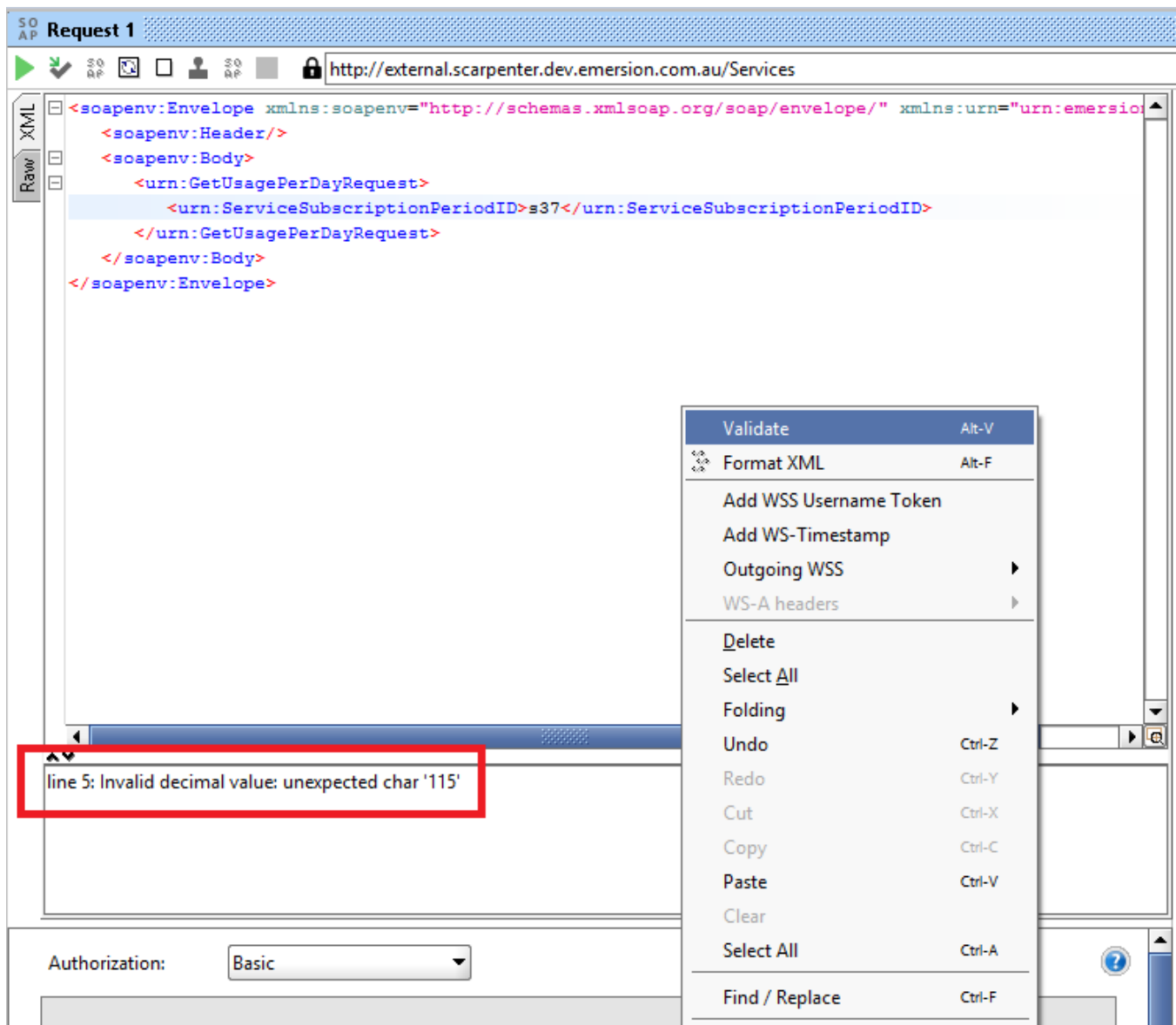
API Call Execution

To submit the API request, start at the *Request* window as shown below.

Enter the appropriate values in the fields.

Remove any optional values and comments.

Right click anywhere in the white space and choose **Validate** (Alt-V).



If you have entered any values that won't pass validation, they will be listed as per the screenshot above. In this case, fix your inputs and try again. When you have a request that will pass validation, you will see **message Validation OK**.

Once your request is validated, submit the API request by clicking the green triangle icon in the top-left of the *Request* window.

If your configuration is correct, you should see your expected results. If not, review your configuration.

Working Example

This is an example of a successful request to the GetUsagePerDay API call in the Services end point. The red numbers/sections show:

- 1 - The Project configured with the **Services.wsdl** end point.
- 2 - The API Call GetUsagePerDay with *Request* number 1.
- 3 - Request 1 input with a valid service subscription period ID (which is what this call requires to return results).
- 4 - The results of the call when submitted.
- 5 - Submit Request button.

The screenshot displays the SoapUI application interface with the following components:

- Project Explorer (Left):** Shows a project named 'Web Dev New Auth - Services' containing a 'ServiceInterface'. The 'GetUsagePerDay' request is highlighted.
- Request Properties (Left):** A table showing details for 'Request 1':

Property	Value
Name	Request 1
Description	
Message Size	344
Encoding	UTF-8
Endpoint	http://external.scarp...
Timeout	
Bind Address	
Follow Redirects	true
Username	
- XML Editor (Center):** Displays the SOAP request XML:


```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/' xmlns:urn='urn:emeralds...'>
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetUsagePerDayRequest>
      <urn:ServiceSubscriptionPeriodID>1625237</urn:ServiceSubscriptionPeriodID>
    </urn:GetUsagePerDayRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
- XML Viewer (Right):** Displays the SOAP response XML:


```
<?xml version='1.0' encoding='UTF-8'>
<nsl:TelephonyUsageSummary>
  <nsl:Bo1On>
    <nsl:Bo1OnPeriodID>5748921</nsl:Bo1OnPeriodID>
    <nsl:Bo1OnPeriodStart>2016-11-17T12:16:08</nsl:Bo1OnPeriodStart>
    <nsl:Bo1OnPeriodEnd>2016-12-01T00:00:00</nsl:Bo1OnPeriodEnd>
    <nsl:PercentageUsed>0.52</nsl:PercentageUsed>
    <nsl:IncludedValueUsed>12061</nsl:IncludedValueUsed>
    <nsl:IncludedValueActual>233333.33</nsl:IncludedValueActual>
    <nsl:IncludedValueBase>5000000</nsl:IncludedValueBase>
    <nsl:IncludedValueTypeID>3</nsl:IncludedValueTypeID>
    <nsl:IncludedValueTypeID_Kilobytes><nsl:IncludedValueTypeID_Kilobytes>
  </nsl:Bo1On>
  <nsl:Bo1Ons>
  </nsl:Bo1Ons>
</nsl:TelephonyUsageSummary>
<nsl:Day>
  <nsl:Day>
    <nsl:Data>2016-11-29</nsl:Data>
    <nsl:TelephonyUsageSummary>
      <nsl:Bo1On>
        <nsl:Bo1OnPeriodID>5748921</nsl:Bo1OnPeriodID>
        <nsl:Bo1OnPeriodStart>2016-11-17T12:16:08</nsl:Bo1OnPeriodStart>
        <nsl:Bo1OnPeriodEnd>2016-12-01T00:00:00</nsl:Bo1OnPeriodEnd>
        <nsl:PercentageUsed>0.46</nsl:PercentageUsed>
        <nsl:IncludedValueUsed>10700.3333</nsl:IncludedValueUsed>
        <nsl:IncludedValueActual>233333.33</nsl:IncludedValueActual>
        <nsl:IncludedValueBase>5000000</nsl:IncludedValueBase>
        <nsl:IncludedValueTypeID>3</nsl:IncludedValueTypeID>
        <nsl:IncludedValueTypeID_Kilobytes><nsl:IncludedValueTypeID_Kilobytes>
      </nsl:Bo1On>
      <nsl:Bo1Ons>
      </nsl:Bo1Ons>
    </nsl:Bo1On>
    <nsl:Bo1OnAddOnPeriodID>328082</nsl:Bo1OnAddOnPeriodID>
    <nsl:Bo1OnAddOnPeriodStart>2016-11-29T19:38:29</nsl:Bo1OnAddOnPeriodStart>
    <nsl:Bo1OnAddOnPeriodEnd>2016-12-01T00:00:00</nsl:Bo1OnAddOnPeriodEnd>
    <nsl:PercentageUsed>0.85</nsl:PercentageUsed>
    <nsl:IncludedValueUsed>948.6667</nsl:IncludedValueUsed>
    <nsl:IncludedValueActual>1000000</nsl:IncludedValueActual>
    <nsl:IncludedValueBase>1000000</nsl:IncludedValueBase>
    <nsl:IncludedValueTypeID>3</nsl:IncludedValueTypeID>
    <nsl:IncludedValueTypeID_Kilobytes><nsl:IncludedValueTypeID_Kilobytes>
  </nsl:Bo1OnAddOn>
  <nsl:Bo1OnAddOns>
  </nsl:Bo1OnAddOns>
</nsl:Day>
</nsl:Bo1Ons>
</nsl:TelephonyUsageSummary>
<nsl:GetUsagePerDayResponse>
</nsl:GetUsagePerDayResponse>
</SOAP-ENV:Envelope>
```