## Testing with E2E SOAP Test Tool Lesson 3.1 MD18

$\left.\sum-\right\rangle-1=1-1-1-1->-1$
Now, you can test the xUML service with the E2E SOAP Test Tool.

## Testing the File Reading




Second Iteration

## On this Page:

- Testing the File Reading
- Testing the Filtering

The log view shows the service having been deployed to the internal server of the development environment and now running in the E2E Embedded Runtime.


Open the E2E SOAP Test Tool by clicking Open Test Case. The SOAP Test Tool window opens and shows a test case for lesson3.EducationLesson3.EducationLesson3.


Have a look at the default test case queryProducts. The gearwheel in gray color indicates that the test case has not yet been run. On the Request tab in the right panel, the input parameter userlnput is displayed.

Remember, that parameters of complex type have to be created by a create statement. Create the parameter userInput by selecting Create from the context menu.


Expanding userlnput, the title field and the keywords array are displayed. Arrays are labeled with the icon an have to be created, too.

Create the keywords array.


The keywords array has been created as an array with no elements. This is indicated by the phrase Str ing [0] in the parameter's tree node. You could append array items now, but in this first step you will not test the filtering functionality.

Run the test case without entering keywords. The Analyzer switches to the Response tab and shows the result.


If the test case reports no error, the test case's gearwheel will turn green and eight products from the file catalog.xml should be displayed.
If an error occurred, the gearwheel turns red. Go on reading this page, even if no error occurred.
In case of an error, open the detail node on the Response tab. Expand the first message node to read the description of the error.

The message might read: Failed stating file "[...]/catalog.xml".
This means, that the data source file catalog.xml could not be found in the location you defined in the backend interface artifact XMLFileArtifact in the component diagram.

To solve the problem, copy the file catalog.xml to the location you have defined in the component diagram. As an alternative, you can also check the current location path of the file and adjust the path in the backend interface artifact XMLFileArtifact in the component diagram. In this case you must compile and deploy the xUML service again. This applies also, if there are typing errors in the file's path name.

After your corrections, run the test case again.
The attribute of the output object productList (product array, labeled with is displayed. The array elements, which are of type Product, contain all product records retrieved from the XML file. The name of the array product is being followed by the number of array elements (8) in square brackets: product: Product[8].

## Testing the Filtering

Now, you will create a second test case to test the filtering functionality of your xUML service.
Create a second test case.



Normally, you would have to create the array in the SOAP Test Tool first, which results in an empty array. As, in this case, you want to add an array element, you can directly append an element without creating the array before. The SOAP Test Tool will create the array automatically.


Run the test case.
On the Response tab, expand the tree. The records shown in the picture below will be selected if you used the keywords Nemo, Lord, and Pirates on the Request tab.


