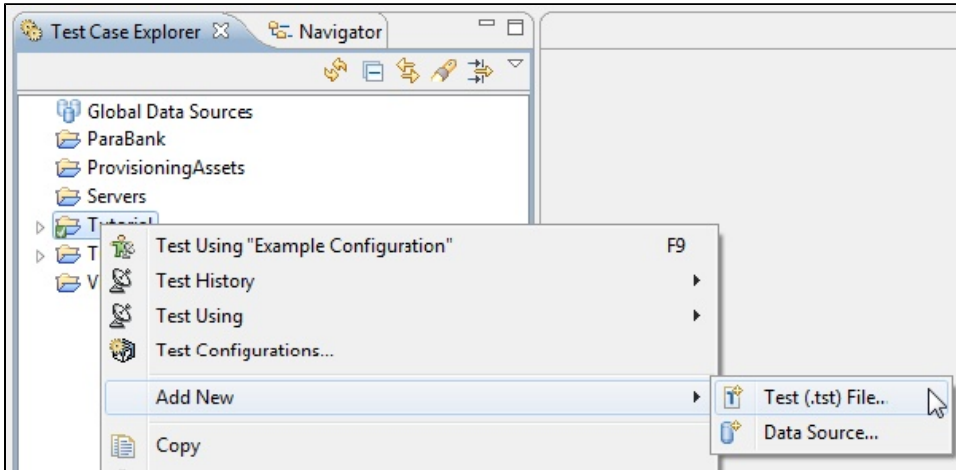


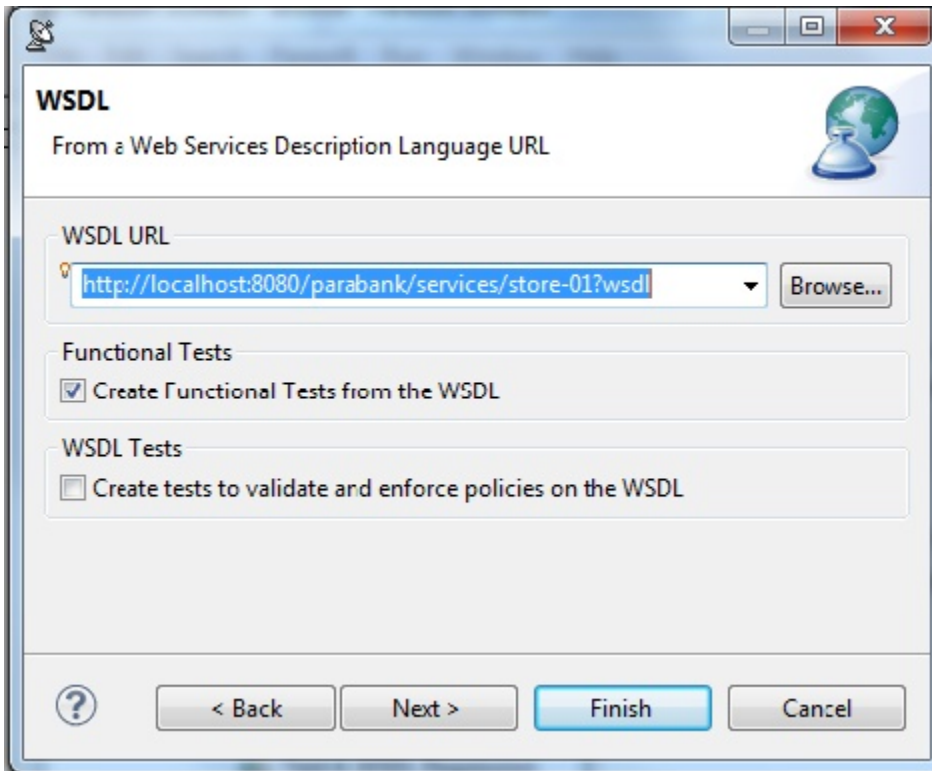
Creating Test Suites for Unit Tests

For this example, we add a new Test (.tst) file to the project created in the previous lesson.

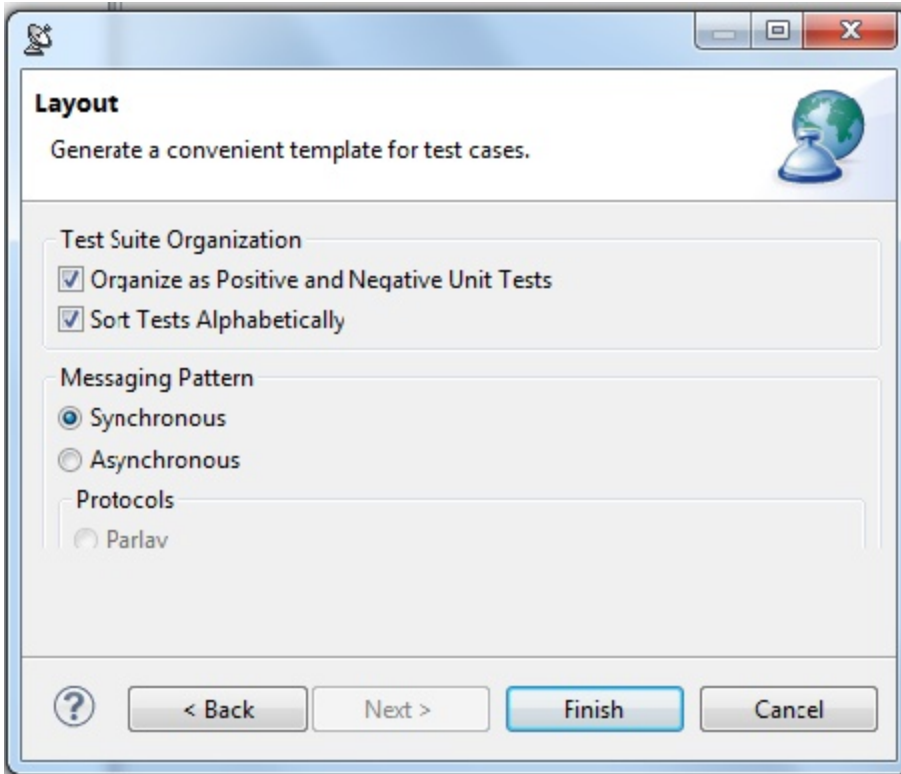
1. Right-click the project from the previous exercise, then choose **Add New> Test (.tst) File** from the shortcut menu.



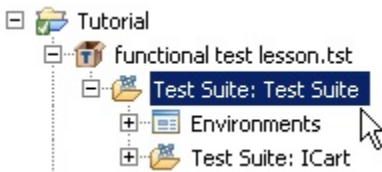
2. Enter a name for the file (e.g., functional test lesson), then click **Next**.
3. Select **SOA> WSDL**, then click **Next**.
4. Select <http://localhost:8080/parabank/services/store-01?wsdl> from the **WSDL URL** field.
5. Ensure that **Create Functional Tests from the WSDL** is enabled.



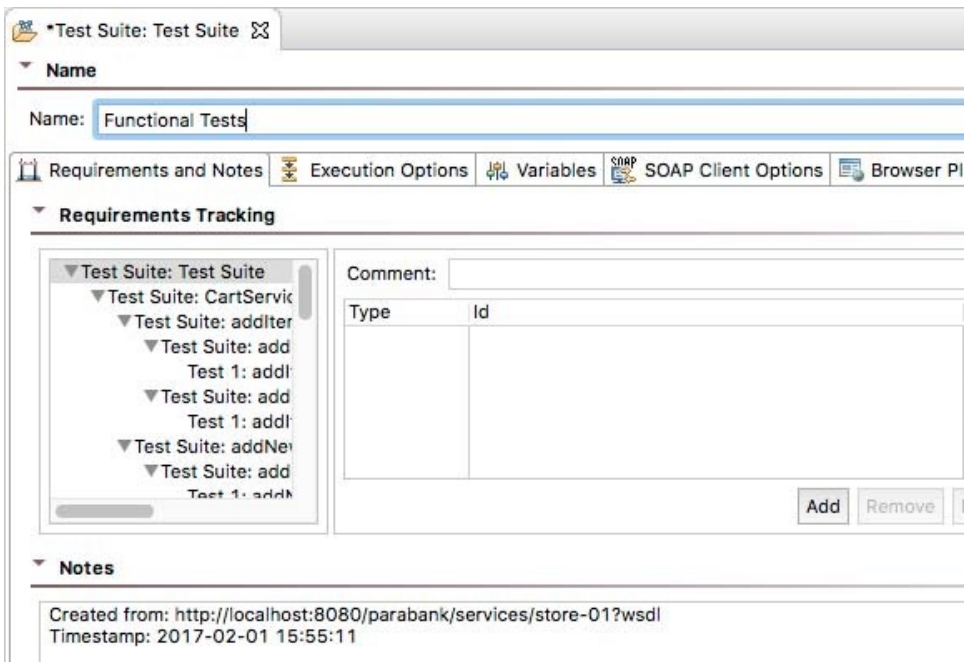
6. Click **Next** to accept the default settings until you reach the **Layout** dialog.
7. Enable **Organize as Positive and Negative Unit Tests**.



8. Click the **Finish** button. The newly created test suite displays in the **Test Case Explorer**.

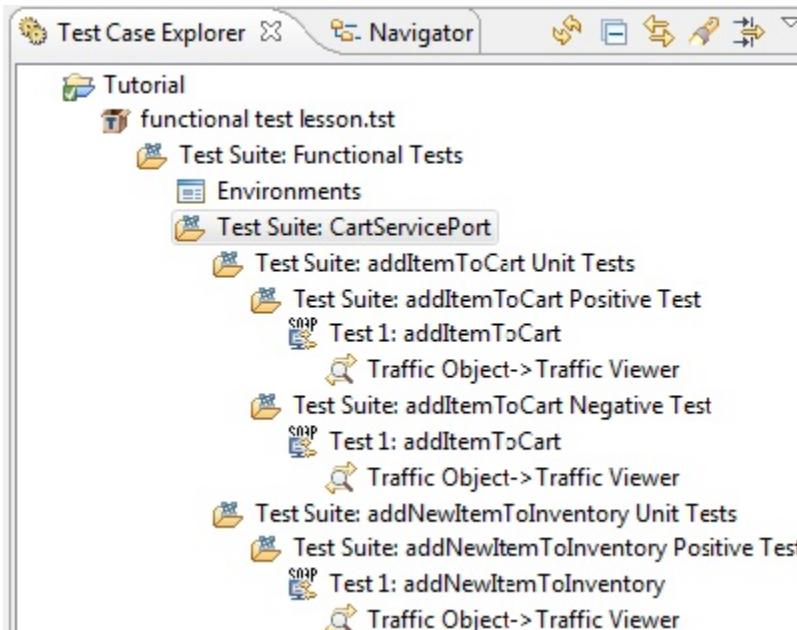


9. Double-click the new **Test Suite: Test Suite** node to open the test suite configuration panel and rename the test suite to `Functional Tests` in the **Name** field.



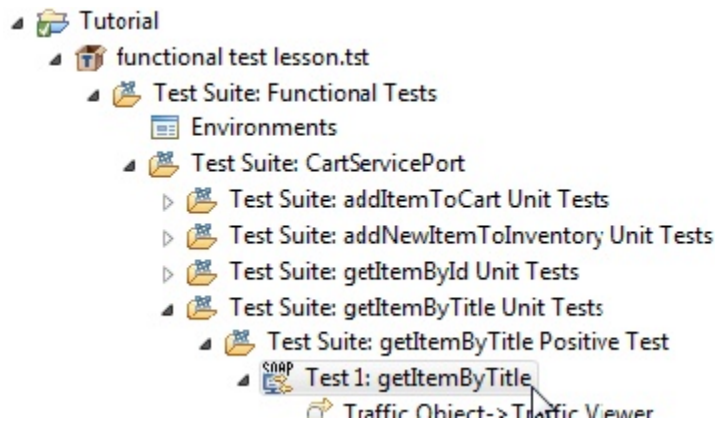
10. Save the file.

- Expand the **Functional Tests** test suite to show the **Test Suite: CartServicePort** node, which includes seven other test suites that test each operation of the WSDL.
- Right-click the **Test Suite: CartServicePort** node, then choose **Expand All**. This will display all seven test suites, as well as every test within each test suite.

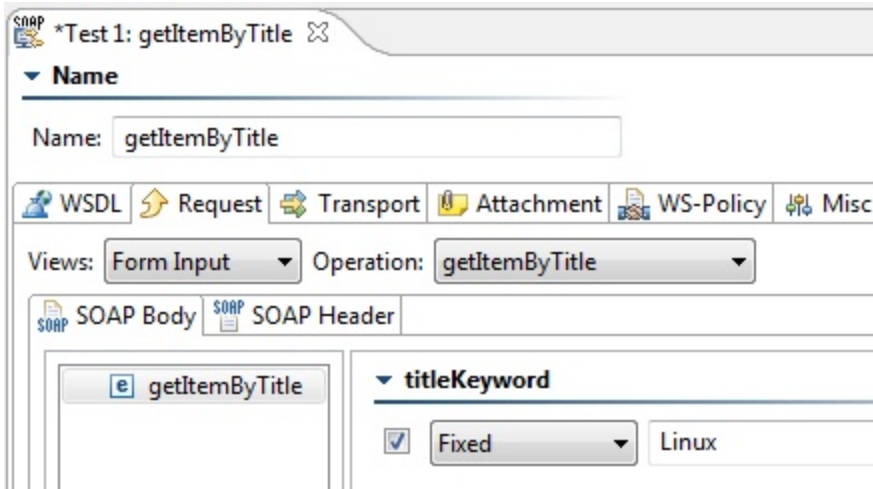


Each of the seven test suites contain both a positive and negative test for each operation. Why create both positive and negative tests? Because it is important to test situations where we send expected data as well as unexpected data to the server.

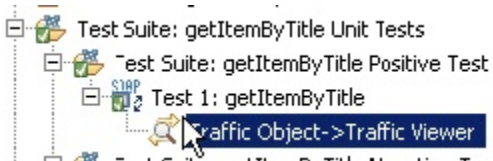
- Double-click the **Test Suite: getItemByTitle Positive Test> Test 1: getItemByTitle** node to open the test configuration panel.



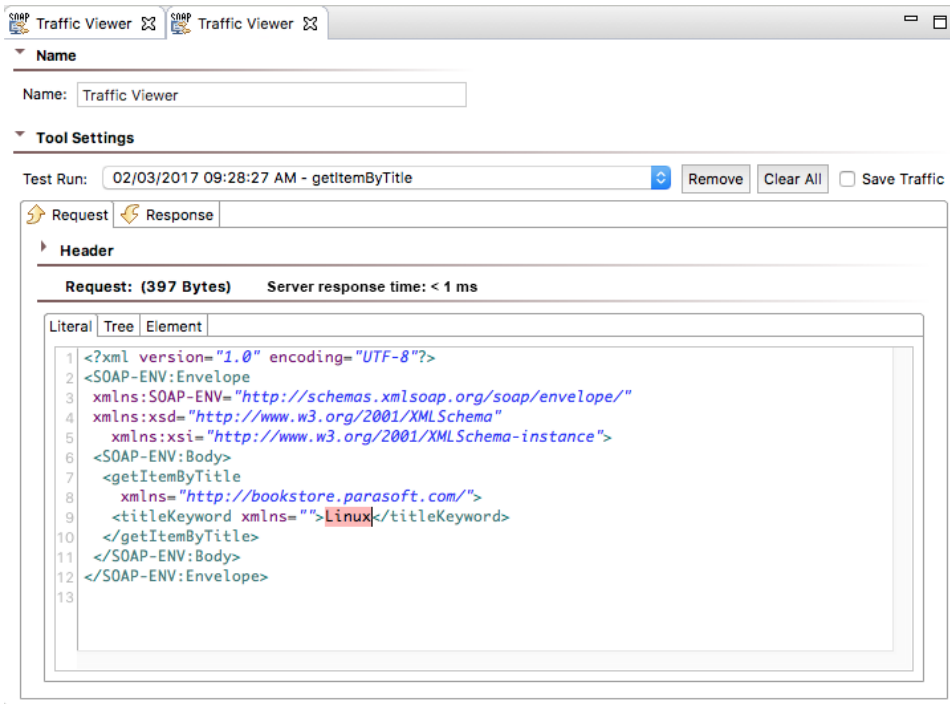
- Choose the **Request> SOAP Body** tabs in the test configuration panel.
- Enable the option in the **titleKeyword** entry field and choose **Fixed** from the drop-down menu.
- Enter `Linux` in the empty field and save the file. We will be searching for books with keyword **Linux**.



17. Select the **Test 1: getItemByTitle** node and click the **Test** tool bar button. The getItemByTitle operation is invoked with the parameter **Linux**.
18. Expand the **Test 1: getItemByTitle** node and double-click the **Traffic Object-> Traffic Viewer** node underneath.



The Traffic View opens and displays the traffic that was logged from the test run.



We now have a functional test that tests the getItemByTitle operation of our web service with a single input value. The same sequence of actions can be done to create functional tests for the other operations defined within the WSDL.