

# Scenario Testing

After unit tests have been created, they can be leveraged into scenario-based tests without any additional work. Scenario tests allow you to emulate business logic or transactions that may occur during normal usage of the web service. This also allows you to find bugs that may surface only after a certain sequence of events.

Parasoft SOAtest allows you to create multi-step scenarios, with each step exercising the same service or completely different application interfaces. This allows you to automate the modeling of business use case scenarios (as described in [End-to-End Business Process Testing](#)). This also allows you to discover unexpected behaviors that may surface only after a certain sequence of events.

The scenario test example given in the test suite "Scenario Test - Search, Add to Cart, and Remove from Cart" represents a typical sequence of operations that a customer may invoke when using a book store web service. In this case, it represents a situation where a customer searches for a book, places an order for that book, and then removes the previously placed order.

This scenario test introduces a tool called the [XML Data Bank](#). This tool allows you to extract XML element values and store these values in memory so that they can be used in later tests. In this example, you will be storing the book ID returned by the service after searching for a book—then, in the subsequent test, use that ID to purchase the book. You will also store the order number returned after placing an order for the book—then, in the subsequent test, use that order number to remove the order from the system.

When you complete this section of the tutorial, your test suite should resemble the "Scenario Test - Search, Add to Cart, and Remove from Cart" test suite in the `SOAtestTutorial.tst` file.

Note that this lesson assumes that you have completed the previous lessons. To follow along with this lesson, please ensure that you have completed the previous lessons.

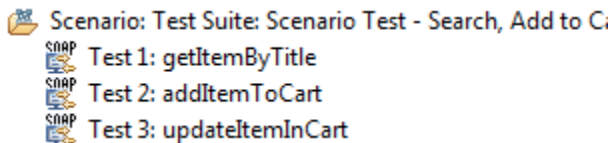
## Creating a Scenario Test Suite

To create this scenario:

1. Select the **Test Suite: Functional Tests** node from the previous exercise and click the **Add Test Suite** button.



2. Select **Empty** from the **Add Test Suite** wizard and click the **Finish** button.
3. Double-click the new **Test Suite: Test Suite** node that was added to the test suite tree.
4. In the test suite configuration panel, enter `Scenario Test - Search, Add to Cart, and Remove from Cart` into the **Name** field, and then click the **Save** toolbar button.
5. Copy the positive `getItemByTitle`, `addItemToCart`, and `updateItemInCart` SOAP Client test nodes from the previously-created `CartServicePort` test suite and paste them into the **Scenario Test - Search, Add to Cart, and Remove from Cart** test suite. If needed, you can drag and drop to reorder them.

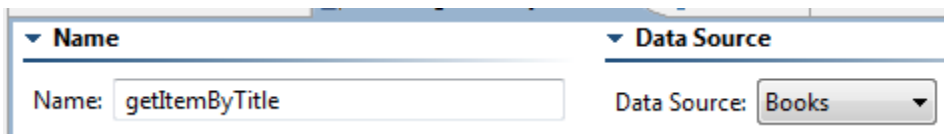


These three tests represent a typical business transaction a customer may invoke. This will be the basis for our scenario test.

## Storing Results to Be Used in Subsequent Tests

You can store results by configuring an XML Data Bank as follows:

1. Double-click the **Test 1: getItemByTitle** node in the **Scenario: Scenario Test - Search, Add to Cart, and Remove from Cart** test suite.
2. In the test configuration panel, select **Books** from the **Data Source** drop-down menu at the top right.



3. Check `titleKeyword`, then select **Parameterized** and **Keywords** from the `itemId` element drop-down menus.

▼ **titleKeyword**

Parameterized

- Double-click the **Test 2: addItemToCart** node in the **Scenario Test - Search, Add to Cart, and Remove from Cart** test suite.
- For **itemId**, select **Parameterized** and **Use Data Source Wizard** from the drop-down menus.

▼ **Name**      ▼ **Data Source**

Name:       Data Source:

WSDL   Request   Transport   Attachment   WS-Policy   Misc

Views:       Operation:

SOAP Body   SOAP Header

addItemToCart

- itemId
- quantity

▼ **cartId**

Fixed      0

▼ **itemId**

▼ **quantity**

- Complete the Parameterize with Value From Existing Message dialog as follows so that when this test is run, the value stored from **Test 1** will be automatically inserted as the value for the **itemId** element:
  - Select **Test 1: getItemByTitle** from the **Existing message** menu at the top of the dialog.
  - Select the **id** element from the **Expected Message** tree and click the **Add** button. The **id** element displays in the **Selected Element** list with a **Data Source Column Name** corresponding to the selected test.

Existing message:

Source:

Expected Message:

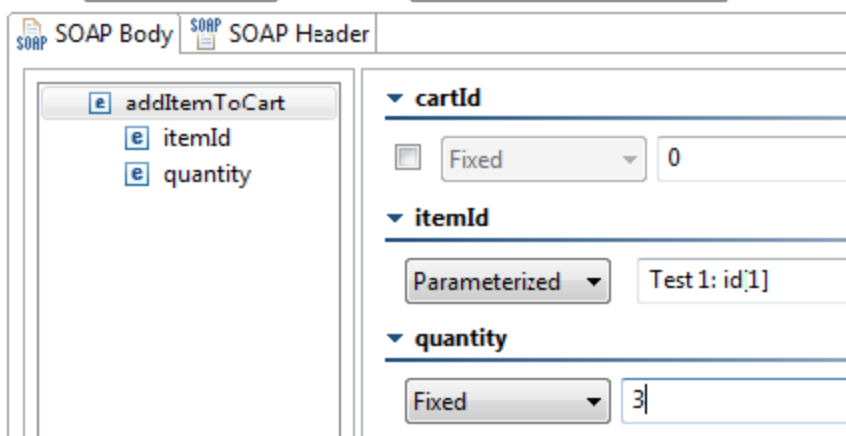
- Envelope
  - Body
    - getItemByTitleResponse
      - book
        - id**
        - name
        - price
        - stockQuantity
        - authors
        - description
        - ISBN
        - publicationDate
        - publisher

Selected Element	Data Source Column
../id[1]	Test 1: id[1]

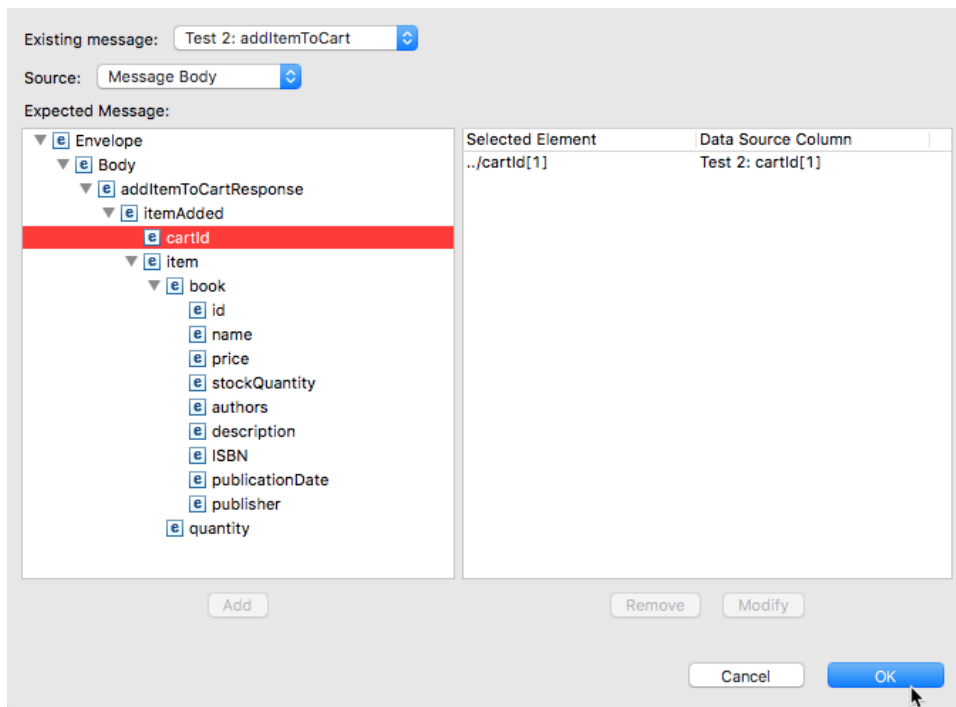
Add      Remove      Modify

Cancel      **OK**

- Click the **OK** button. **Test 1:id[1]** now displays in the right GUI panel as a parameterized value for **itemId**. You will also notice that a **Response SOAP Envelope> XML Data Bank** node now appears underneath the **Test 1: getItemByTitle** node in the **Scenario Test - Search, Add to Cart, and Remove from Cart** test suite.
- In the test configuration panel, enter a Fixed value of 3 for the **quantity** element, then click the **Save** toolbar button.

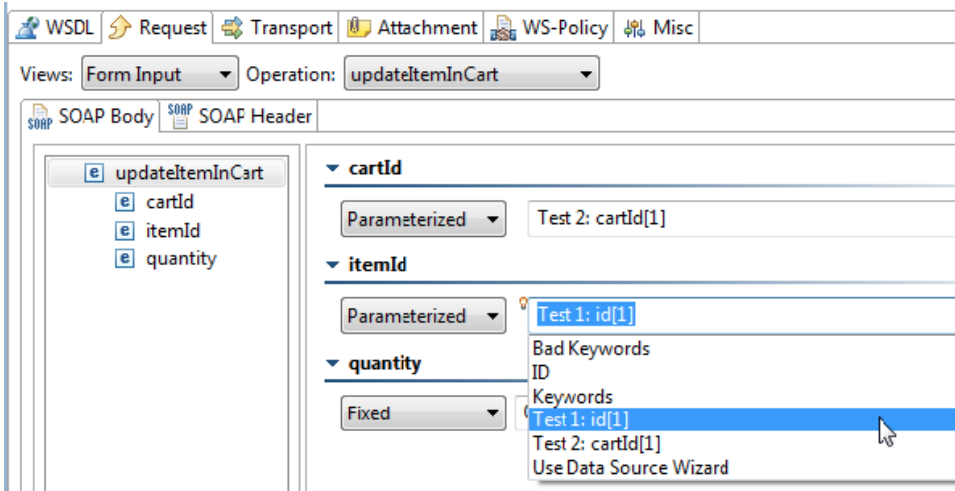


8. Double-click the **Test 3: updateItemInCart** node.
9. For **cartId**, select **Parameterized** and **Use Data Source Wizard** from the drop-down menus.
10. Complete the Parameterize with Value From Existing Message dialog as follows so that when this test is run, the **cartId** element value stored from **Test 2** will be automatically inserted as the value for the **cartId** element here:
  - a. Select **Test 2: addItemToCart** from the **Existing message** menu at the top of the dialog.
  - b. Select the **cartId** element from the **Expected Message** tree and click the **Add** button. The **cartId** element displays in the **Selected Element** list with a **Data Source column name** corresponding to the selected test.



- c. Click the **OK** button.

**Test 2:cartId[1]** now displays in the test configuration panel as a parameterized value for **cartId**. You will also notice that a **Response SOAP Envelope > XML Data Bank** node now appears underneath the **Test 2: addItemToCart** node in the **Scenario Test - Search, Add to Cart, and Remove from Cart** test suite.
11. Configure **itemId** to use the ID from Test 1.



12. Click the **Save** toolbar button.
13. Select the **Scenario: Scenario Test - Search, Add to Cart, and Remove from Cart** node and click the **Test** toolbar button. When this test is run, the **id** element value stored from **Test 1** will be automatically inserted as the value for the **itemId** element in **Test2** and **Test3**. Additionally, the **cartId** from **Test2** will be used in **Test3**.
14. Explore the traffic by expanding **Scenario: Scenario Test - Search, Add to Cart, and Remove from Cart** and double-clicking each test's **Traffic Object> Traffic Viewer** nodes.
15. Notice that the **id** of the book returned from **Test1** is used as the input for **Test2**. Also, the **cartId** of the order placed in **Test2** is used as the input for **Test3**.
16. Right-click the **Scenario: Scenario Test - Search, Add to Cart, and Remove from Cart** node, select **Create/Update Regression Controls**.
17. In the Response Validation Wizard, select **Create Regression Controls**, click **Next**, select **Create Multiple Controls**, and then click the **Finish** button. The tests are run and a Regression Control is added to each SOAP Client test.
18. Select the **Scenario: Scenario Test - Search, Add to Cart, and Remove from Cart** node and click the **Test** toolbar button. Notice the test failures reported.
19. Examine the error messages that appear in the Quality Tasks view. These regression failures are due to dynamic content that appears within the response messages. In the following steps we will ignore elements with this type of dynamic data.
20. In the Quality Tasks view, right-click the first error reported under each **TestSuite** node and select **IgnoreXPath** from the shortcut menu. In the **IgnoreXPathSettings** dialog that displays, click the **OK** button. You should ignore two XPath's in this step.
21. Select the **Test Suite: Scenario Test - Search, Add to Cart, and Remove from Cart** node and click the **Test** toolbar button. All the tests should now succeed.

You have now created a fully functional scenario test that tests one possible business transaction which may occur during normal usage of the book store service. For extra practice, you can try to create other scenarios that may possibly occur. Negative test cases could also be created for expanded test coverage.