

Web Functional Testing

The following exercises demonstrate how to use SOAtest to perform functional testing on the Web interface. It covers:

- [Introduction](#)
- [Cross-browser Testing and Validation Video Overview](#)
- [Recording in a Browser](#)
- [Adding a Data Source](#)
- [Parameterizing a Form Input](#)
- [Configuring Validation on a Page Element](#)
- [Configuring a Browser Stub](#)
- [Playing a Recorded Scenario in a Browser](#)
- [Performing Static Analysis During Web Scenario Execution](#)

Introduction

Web interface testing is difficult to automate. Teams often abandon automated testing in favor of manual testing due to too many false positives or too much effort required to maintain the test suites.

SOAtest facilitates the creation of automated test suites that are reliable and dependable. Its ability to isolate testing to specific elements of the Web interface eliminates noise and provides accurate results.

SOAtest isolates and tests individual application components for correct functionality across multiple browsers without requiring scripts. Dynamic data can be stubbed out with constant data to reduce test case noise. Validations can be performed at the page object level as well as the HTTP message level. SOAtest also verifies the client-side JavaScript engine under expected and unexpected conditions through asynchronous HTTP message stubbing.

Cross-browser Testing and Validation Video Overview

Recording in a Browser

Before performing the following exercises, ensure that you have started the ParaBank Server (as described in [Setting Up ParaBank](#)) and that you can navigate successfully to the default address (`localhost:8080/parabank`). Close the browser window after you verify this. To record in a browser: Right-click the project from the previous exercises, then choose **Add New Test (.tst) File** from the shortcut menu.

Enter a name for the file, then click **Next**. Select **Web**. Record web scenario and click **Next**.

In the first Record Web Scenario wizard page, ensure that **Record new web scenario** is selected, then click **Next**.

Complete the next Record Web Scenario wizard page as follows:

- Enter `Web Functional Testing` in the **Test Suite Name** field.
- Enter `localhost:8080/parabank` in the **Start Recording From** field.

Ensure that the following options are checked, and the others are not:

- **Generate Functional Test**
- **Generate Asynchronous Request Tests**

/ResponseBody.png?version=1&modificationDate=1524517672729&api=v2" data-unresolved-comment-count="0" data-linked-resource-id="33857314" data-linked-resource-version="1" data-linked-resource-type="attachment" data-linked-resource-default-alias="ResponseBody.png" data-base-url="https://docs.parasoft.com" data-linked-resource-content-type="image/png" data-linked-resource-container-id="33857293" data-linked-resource-container-version="5">

Click Save to save the changes.<h2 id="WebFunctionalTesting-PlayingaRecordedScenarioinaBrowser">Playing a Recorded Scenario in a Browser</h2><p>To playback the recorded scenario in a browser:</p>In the Test Case Explorer, select the Scenario: Web Functional Testing node.Click the Test toolbar button.<p>The recorded scenario will now be played back in your browser, once for each search value that we parameterized in a previous section of this example. Please wait for each action to be played out in your browser.</p><p>Note if you have been following the complete tutorial, errors will be reported due to the Browser Stub tool that was previously added. This is expected.</p><h2 id="WebFunctionalTesting-PerformingStaticAnalysisDuringWebScenarioExecution">Performing Static Analysis During Web Scenario Execution</h2><p>To configure SOAtest to perform static analysis on the Web pages that the browser downloads as web scenarios execute:</p>In the Test Case Explorer, select the Scenario: Web Functional Testing node.Open the Test toolbar button's pull-down menu, then choose one of the available static analysis configurations.

<p>When SOAtest is finished performing static analysis, static analysis violations are shown in the Quality Tasks view. To facilitate review of these results:</p>Open the pull-down menu on the top right of the Quality Tasks view.

Choose Show SOAtest Static Analysis for Functional Tests Layout.</div></div></body></html>