

Stubbing Test Requests and Responses

This topic explains how to stub test requests/responses for web scenarios.

Sections include:

- [Understanding Stubs for Web Scenarios](#)
- [Creating Stubs](#)
- [Configuring the Browser Stub Tool](#)

Understanding Stubs for Web Scenarios

Testing applications with dynamic data can cause many false positives, creating extra overhead for the developers and QA who have to determine which failures are real, and which are “noise.” To solve this, SOAtest has the ability to “stub” data that is sent back to the client. Stubbing helps to ensure that any changes to the client-side code do not affect the final resultant html page.

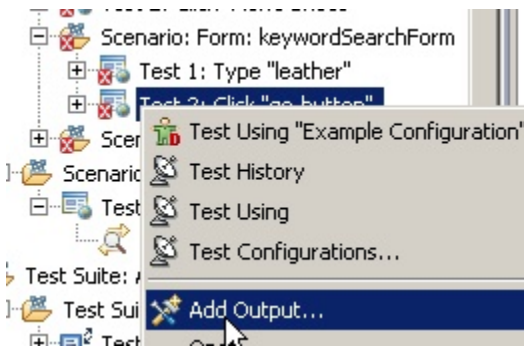
A stub is static data that SOAtest saves when recording a functional test scenario through a web site. Since the data that will be fed to the client is unchanging, any new errors that occur while processing the data can be attributed to changes in the client-side JavaScript that processes the data.

Creating Stubs

While recording a web scenario, SOAtest keeps track of each request made by the client, as well as the response.

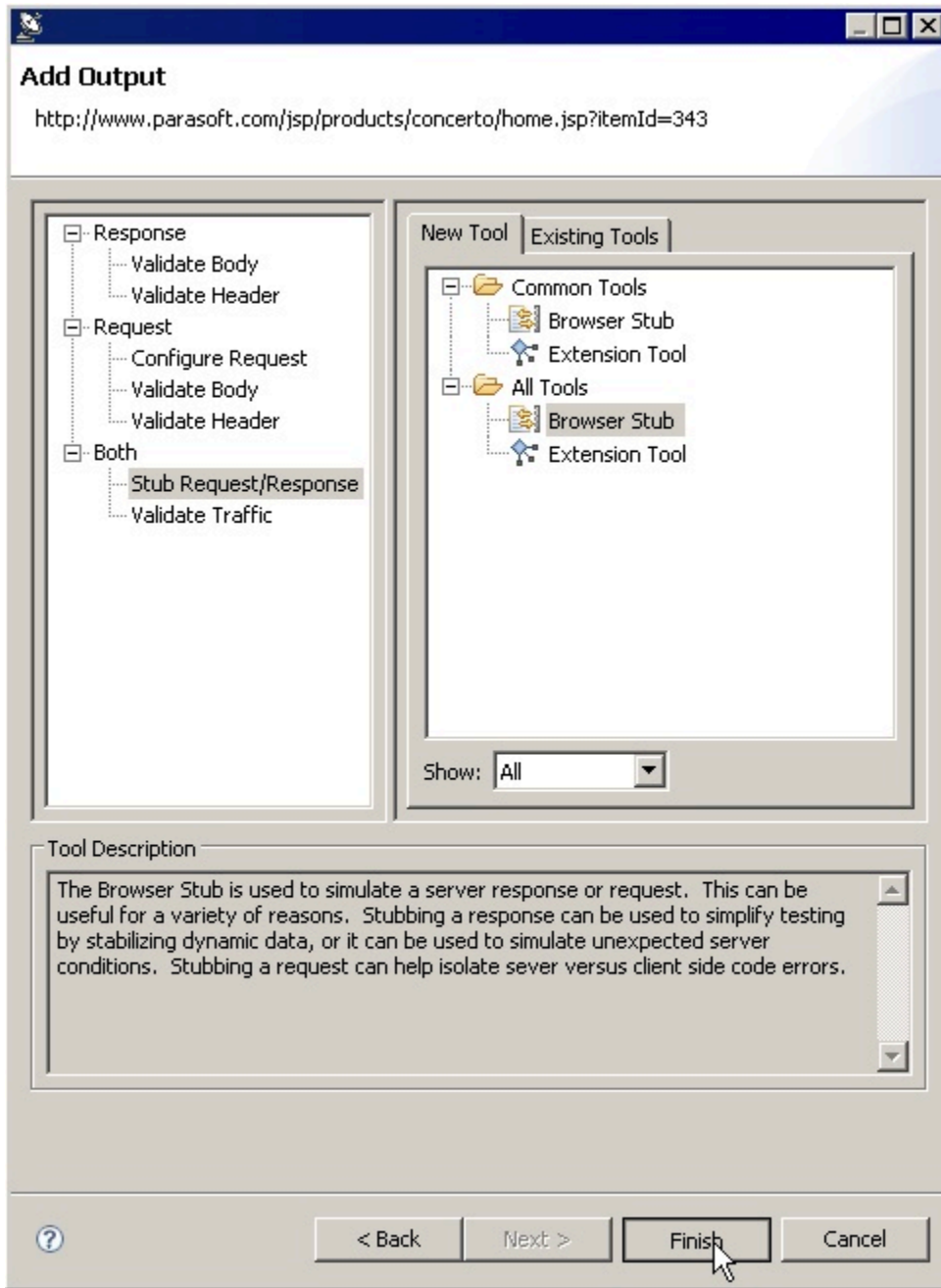
To create a stub, complete the following:

1. Right-click the test from which you would like to return the static data and select **Add Output** from the shortcut menu.



2. In the wizard that opens, choose **HTTP traffic**, then click **Next**.
3. In the next page, choose the browser request that you want to stub.
4. In the left panel, select **Both - Stub Request/Response**.
5. In the right panel, select any **Browser Stub**.

6. Click **Finish**.



Configuring the Browser Stub Tool

To configure a Browser Stub tool that has been added to an functional test:

1. Double-click the **Stub Request/Response for <URL>** node that was added to the test.
2. In the Browser Stub test configuration panel, you can modify the following options:
 - **Name:** Specifies the name for the Browser Stub.
 - **Response Header View:** Customizes the returned request/response headers. Select the following from the drop-down menu:
 - **Literal:** This view allows you to modify the raw text returned from the request.
 - **Form:** This view provides the following sub-options:
 - **General:** Selecting this option from the list will allow you to modify the version of HTTP and response code that is returned.
 - **Request/ Response Headers:** This panel lets you add/remove/modify the list of headers that are sent to the server, or received by the client. To add a new header, click the **Add** button. This will add a new entry to the list that can be modified by clicking the **Modify** button. Clicking **Modify** will open up a dialog with two text fields.

The top field is for the header name, and the bottom option is for the header value. Click OK to make the changes permanent. To remove headers, select the header you wish to remove and click the **Remove** button.

- **URL Parameters:** This option, only available if configuring a request, allows you to add/remove/modify any arguments to be placed into the URL when the browser makes its request to the server.
- **Parameterized:** This option will allow you to return values stored in data sources created in the project. For more information on setting up data sources see [Parameterizing Tests with Data Sources, Variables, or Values from Other Tests](#).
- **Scripted:** This option allows you to use a custom script to return the correct values. This option is identical to using an Extension tool in SOAtest. For more information see [Extension Tool for Custom Scripting](#).
- **Response Body View:** Customizes the returned request/response body. Select the following from the drop-down menu:
 - **Literal:** This view allows you to modify the raw text returned from the request.
 - **Parameterized:** This option will allow you to return values stored in data sources created in the project. For more information on setting up data sources see [Parameterizing Tests with Data Sources, Variables, or Values from Other Tests](#).
 - **Scripted:** This option allows you to use a custom script to return the correct values. This option is identical to using an Extension tool in SOAtest. For more information see [Extension Tool for Custom Scripting](#).