

HTTP Component Testing

with WebSphere Studio Application Developer

Paul Glezen
pglezen@us.ibm.com
January 27, 2003

Abstract

WSAD provides several mechanisms to drive unit tests of server-side Java code. The HTTP unit testing tool can be useful for automating HTTP invocations. The current online documentation for HTTP unit testing is somewhat dated. This document was written to demonstrate its use as it exists today.

Prerequisites

In addition to WSAD V5 GM, you must also have the WebSphere **Remote Agent Controller** installed on the machine from which the HTTP requests will be invoked. Most likely, this will be a developer workstation. The **Component Testing Tool** plugin is included with WSAD.

Project Setup

You will create test metadata that defines the structure and behaviour of the tests using the WSAD GUI. From this metadata, Java code is generated and compiled to actually carry out the tests. The execution of these tests will generate result records. It's a good idea to setup a WSAD project for all these artifacts separate from your servlet project.

Make the new project a Java project to hold the generated Java code. In the Java project creation wizard, specify "src" as the source subdirectory and bin as the build output folder.

Host Creation

The following steps guide you through defining a host on which to run the test cases. It is assumed that all tests will be run on the local host.

1. Change to the Component Test perspective. From the **Window** dropdown menu choose **Open Perspective --> Component Test**. If "Component Test" does not appear among the selections, choose **Other** and select **Component Test** from the presented list.
2. In the upper-left view, make sure the **Definition** tab is selected. There should be two folders, **Hosts** and **Testcases**.
3. Right click on **Hosts** and select **new --> hosts**. The host referred to here is the host that will be invoking the HTTP requests defined by the test cases. Select the Java project you created for storing test data. Any name will suffice. Since the tests are to be run on the local machine, enter "localhost" for the host name field. Click **Next**.
4. Enter "A host defined for the local machine" for the description and click **Finish**.

5. You should be presented with an editor for the new host you created. In the IP Address field enter **127.0.0.1** and then save the change. Close the host editor.

The host just defined can be reused for all the test cases you created for your application. It is sensible to have all test cases defined in this project to refer to this host.

Test Case Creation

It is recommended that the application you intend to test be running in the test environment during this phase. That's because each step has a quick-test option that permits validation of a URL and its parameters if the server is already running.

Make sure you are still in the Component Test perspective and that the **Definition** tab is selected in the upper left view.

1. Right click on the **Testcases** folder and select **New --> testcase**. In the first panel of the wizard, choose the test folder you created during the setup step. Give the test case a name. Click **Next**.
2. Under Schedulers choose **HTTP**. Enter a short description for the testcase. Click **Finish**.

At this point, you should see the new test case reflected in the editor window. In the outline view (bottom left) you should see that your new test case contains only a "Main Block." Testing tasks are organized into blocks. You can add a task or another block to a block. Blocks are useful for controlling the number of iterations for a set of tasks.

3. Right click on **Main Block** and select **New --> Task --> HTTP**. A new task is created with the default name of T1. The editor panel is loaded with this new task. Change the name to something that reflects the nature of your first HTTP request.
4. Click on the **Design** node on the left side of the task editor. This displays all the properties to be defined for the request. The **Body** section exists only for POST requests. Leave it blank for GET requests. The hostname is the name of the host on which the servlet is running. Since you will be running the servlet in your WSAD test environment, enter **localhost**. The absolute path starts with the first slash after the port. So the URL is `http://localhost:8080/loans/application?age=34`, the absolute path is `/loans/application?age=34`.
5. (Optional) If the server is currently running, you can click **Execute request** to ensure that a response is returned. The sign that the response returned is a "Response: 200" message at the bottom of the panel.
6. Save the request (Ctrl-S).
7. Right click the **Main Block** in the outline view and select **New --> Task --> Delay**. Rename the delay task. Enter a delay amount in milliseconds. Save the task.
8. Add a third task like the first one.
9. Close the test case editor.

Test Case Instance Creation

Now that the test case metadata is defined, a test case instance can be generated. A test case instance is generated testing code associated with a agent host.

1. Make sure the **Definition** tab is still selected in the Component Test prospective. Right click on the test case and select **Prepare**. In the location field, select the source directory in which to place the generated test code.
2. Specify an instance name.
3. Choose **Save instance in the definition's location**. Click **Next**.
4. Choose the **localhost** host. Click **Next**.
5. Click **Finish**.

A test case instance is created. The Component Test prospective activates the **Execution** tab of the upper-left view. The new instance name appears in the Execution view and instance details appear in the right sidee. You can switch to the Java prospective to view the Java code that was generated for the test.

Running the Test Case Instance

Close all the test case editors. Make sure the **Execute** tab is active in the Component Test prospective.

1. Start the test server if it isn't running already.
2. Start the WebSphere Agent Controller if it isn't running already.
3. Right click the test case instance in the Execute view. Select **Run**.
4. Choose a name for this run. **execution1** might be a suitable default. Click **Next**.
5. Choose the localhost host defined earlier. Click **Next**.
6. Enter a description if you have one. Otherwise click **Finish**.

This begins the execution of your test case. In the outline view, the man-running icon by the "execution1" run indicates the test is still running. When the tests finishes there will be either a check mark or an error symbol to denote the success or failure of the test.

To inspect test results, click on the test instance name in the outline. In the editor panel, the status node presents the overall results. The Event Log node displays messages logged by the generated test code.

More detailed information can be obtained by drilling down in the outline view to the individual tasks.