A powerful aspect of XMLHttpRequest Debugging is that it allows you to create a new Ajax request based on a previous request. Click the “edit&replay” link to open the window, as shown in Figure 7-3. Here you’ll see the data listed from the request from which you started, but now all the fields are editable. You can change the request method, update the URL, add some request headers, and even add a string of data or XML that is sent along as part of the request body. Click the “send” link to send the newly created request. This particular tool is powerful when you want to experiment with different request types and data without having to manually update the script, deploy it to the Web server, and repeat the test.

The Greasemonkey extension for the Firefox browser, combined with the XMLHttpRequest Debugging user script, is a powerful tool for Ajax debugging and development. It can help trace exactly what is happening in the browser, and if something goes wrong, you’ll have the information you need to quickly debug the problem. It can help isolate whether a problem exists in the browser-side client script or in the server-side code that services the request.

**Figure 7-3.** A request from the previous example was changed to use the POST method, and the query string was moved from the URL to the request body.

## Debugging JavaScript

Sooner or later during your Ajax development you’re going to have some bugs in JavaScript code. Some of the bugs will be easy to find because they’re syntactic in nature. Others will be more difficult to find because they’re subtle errors in business logic. That’s when a debugger tool comes in handy so you can step through the code, line by line, and verify that the scripts are flowing as expected and that the variables have the correct values.

You can use three tools to help you debug JavaScript:

*Firefox JavaScript Console.* Firefox JavaScript Console records all errors and warnings that occur within JavaScript. It’s easy to use, and you can quickly diagnose most errors using the JavaScript Console, without resorting to a full-fledged debugging environment.

*Microsoft Script Debugger.* This tool integrates with Internet Explorer and provides basic debugging facilities such as the ability to set breakpoints and the ability to inspect and modify the values of variables during runtime. It’s a relatively primitive tool, although it may fit nicely in situations where many Internet Explorer–specific behaviors are used or where the debugging errors occur only in Internet Explorer.