

Checking for JavaScript Errors with Selenium

Posted At : June 7, 2010 5:54 PM | Posted By : Bob Silverberg

Related Categories: Selenium, TDD

I've recently been writing some end-to-end tests for [ValidateThis](#), my validation framework for ColdFusion objects, using [Selenium](#). One of the things I wanted to test were the client-side validations that the framework generates automatically. Selenium makes it very easy to test these, as I can use Selenium's *assertText* command to locate failure messages that have been generated by the [jQuery Validation plugin](#), using XPath, and check whether the message is what I expected.

I found, however, that I was often getting JavaScript errors when the page first loaded (because I was working on the JS and was introducing errors), so I was looking for a quick way to add an assert to my test to check whether any JS errors occurred. After some Googling I came to the conclusion that there is nothing built into Selenium to support this, but there are a number of hacks that can be used to accomplish it. I'm going to describe one of them here. Let me state again, for the record, that this is pretty hacky. I'd love to hear from others who may have better solutions.

I simply add a script to my page that will catch any JS errors by intercepting the *window.onerror* event:

```
<script type="text/javascript">

window.onerror=function(msg){

    $("body").attr("JSError",msg);

}

</script>
```

This will cause an attribute called *JSError* with a value corresponding to the JavaScript error message to be added to the *body* tag of my document if a JavaScript error occurs. Note that I'm using jQuery to do this, so this specific example won't work if jQuery fails to load. Then, in my Selenium test, I just use the command *assertElementNotPresent* with a target of *//body[@JSError]*. Now, if any JavaScript errors occur on the page my test will fail and I'll know I have to address them first. If, for some strange reason, I want to check for a particular JavaScript error, I could use the *assertElementPresent* command with a target of *//body[@JSError='the error message']*.

Note that I'm using this with a test fixture page, it's not an actual page that is part of an application. I'm not sure that this would be very useful in an automated testing environment, I'm just using it for some TDDing, where I want to write an end-to-end acceptance test first, and then write my unit tests. Getting the test to fail because of a JavaScript error is a nice way, imo, of satisfying the criteria that I must have a failing test before writing code to make it work. The usefulness of this technique in terms of pure testing is questionable, but I find it useful for my purposes, so I thought I'd put it out there for others.