

Re: Automated testing of cgi / perl

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2007-02/msg00870.html>

- *From:* Mark Clements <mark.clementsREMOVETHIS@xxxxxxxxxx>
 - *Date:* Fri, 16 Feb 2007 19:09:36 +0100
-

Michael Vilain wrote:

In article <1171565491.430048.105110@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>, spam@xxxxxxxxxx wrote:

Hello, I have the following situation:

- Existing site with about 18,000 cgi pages.
- Most pages are data driven (fill out form, submit, get results). Exact results may change from day to day as data get updated.
- Some static html mixed in
- Many pages require authentication. It is a custom in-house authentication mechanism, but ultimately it stores a temporary session ID in a session cookie.

I would like to implement automated testing that will:

- Tell me if anything breaks.
- Tell me if a new installation of the same overall site is working the same as the original (for an upgrade project).
- Tell me if I break anything during maintenance (regression testing)

Can anyone give me some pointers on this? Tool? Overall approach? Pitfalls? Any clues appreciated.

Larry

I don't know of any tools that can test in a web application environment. The regression testing tool in Unix is expect which is command-line only.

I came across Selenium today, which looks rather useful, and has the advantage of being able to run tests under various browsers. Also worthy of note is jsunit.

With so many pages, there must be an automated test plan. Otherwise, how did the site get so large without detecting and fixing bugs? Or was that what your customers are for?

Re: Automated testing of cgi / perl

You can write perl code that uses HTML CPAN libraries to access pages, pretending to be a client. Said connection can input information into fields, even store cookies for session ID between pages.

As an addendum to this:

Test::WWW::Mechanize

would be my first port of call.

If you don't already have an automated testing environment for your web application, it's rather late to start developing one.

He's probably inherited a large and rather crufty application and wants to make sure nothing breaks as he tries to knock it into shape. Best of luck :)

Mark

.