Tech Tip 3 - Synthetic transactions with Casperjs

Tech Tip 3 (02/2016) - Monitoring synthetic transactions with Casperjs in GroundWork

> ⚠ This Tech Tip applies to versions of GroundWork prior to 7.1.1.

While the native Nagios plugins are powerful, they are limited when it comes to testing the end-user experience. There are ways to use open source tools like Selenium to measure and test web site performance, but these can be complex to set up, and may be too powerful for some users needs. There are also commercial solutions which are good, but can get expensive if leveraged in a regular monitoring context. Fortunately, however, advances in "headless" or "phantom" browsers have made it pretty easy to record simple tests in a browser and play them back using plugins that call the headless browsers.

In this tech tip will take a look at using Casperjs and Phantomjs, which can play back actions you record using Resurrectio, a plugin for the Chrome browser.

Requirements:

- Any reasonable Linux system or VM. This would probably work on a Raspberry Pi running Raspbian Linux, with some modifications.
- A way to execute plugins in perl (we will use a GroundWork server, but GDMA works just fine).
- A Chrome browser (for running Resurrectio and recording the actions to test).

How do you do it?

1. Install Phantomjs
2. Install Casperjs
3. Install check_casperjs.pl
4. Install Resurrectio
5. Record your test(s) in Resurrectio
6. Copy the saved test script to the GroundWork Server
7. Run the tests as services in GroundWork

First, we will assume you are in a clean directory with a root shell on the GroundWork server where you want to install it. We will ignore the fact that most of the steps will require "sudo", and we will give the example commands for a 64-bit Centos/Red hat system, though other systems like Ubuntu or SuSe should be quite similar. Here's how you do each step:

1. Install Phantomjs

   As of this writing, Casperjs needs to see a version 1.x of Phantomjs, which is currently on version 2.1.1. So, you need to install an earlier version:

   ```
   wget https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-1.9.8-linux-x86_64.tar.bz2
   tar xvf phantomjs-1.9.8-linux-x86_64.tar.bz2
   ```

   That will get you a directory called phantomjs-1.9.8-linux-x86_64. Copy the self-contained phantomjs executable to a place in the path, like this:

   ```
   cp phantomjs-1.9.8-linux-x86_64/bin/phantomjs /usr/local/bin/
   ```

   You might prefer to link it rather than adding files to /usr/local/bin. It's up to you.

   There is a dependency on fontconfig. If it is not installed you can do so with yum:

   ```
   yum install fontconfig
   ```

2. Install Casperjs

   Download the tarball:

   ```
   wget -O casperjs-1.1-beta5.tgz https://github.com/n1k0/casperjs/tarball/1.1-beta5
   ```

Untar it:

```
tar xvf casperjs-1.1-beta5.tgz
```

That will give you a directory called n1k0-casperjs-4f105a9 (this will be different if you use a different version).
Copy (or link) the binary into the path like this:

```
cp n1k0-casperjs-54a7a05/bin/casperjs /usr/local/bin/
```

3. Install check_casperjs.pl

   Download the tarball:

   ```
   wget -O check_casperjs-master.tgz
   https://git.netways.org/plugins/check_casperjs/archive-tarball/master
   ```

   Then untar the file:

   ```
   tar xvf check_casperjs-master.tgz
   ```

   That will give you a directory called plugins-check_casperjs. Copy this directory to the nagios plugins directory:

   ```
   mv plugins-check_casperjs /usr/local/groundwork/nagios/libexec/check_casperjs
   ```

   Change the directory permissions:

   ```
   chown -R nagios.nagios /usr/local/groundwork/nagios/libexec/check_casperjs
   ```

   Next we update the shebang to use the perl provided by GroundWork:

   ```
   sed -e 's#/usr/bin/perl#/usr/local/groundwork/perl/bin/perl#' \
       -i /usr/local/groundwork/nagios/libexec/check_casperjs/check_casperjs.pl
   ```

   You should now be able to test the plugin as user nagios:

   ```
   su - nagios
   /usr/local/groundwork/nagios/libexec/check_casperjs.pl  -w 1000 -c 100000 -t
   tests/simple_example.js
   ```

4. Install Resurrectio

   This is a Chrome plugin, and is straightforward to install. Just type "resurrectio" into a working chrome browser address line and follow the link (or click here in chrome), and click "Add to Chrome".

5. Record your test(s) in Resurrectio

   Use the Resurrectio plugin to record your actions for monitoring a web application. Click the Resurrectio icon and hit "Stop recording" when done. Then click "Export Casper JS". You will get a browser window with some javascript in it.

6. Copy the saved test script to the GroundWork Server

   Save this javascript to a file in the `/usr/local/groundwork/nagios/libexec/check_casperjs/tests` directory, for example like this:

   ```
   vi /usr/local/groundwork/nagios/libexec/check_casperjs/tests/mytest.js
   ```

   Then insert the javascript you saved, and save the file.
   Now you are ready to run the plugin and have it test the actions you recorded. For example, as user nagios:

```
cd /usr/local/groundwork/nagios/libexec/check_casperjs/
./check_casperjs.pl -w 5000 -c 10000 -t tests/mytest.js
```

will run the steps recorded, and will go to warning if they take over 5 seconds, and critical if over 10 seconds.
Try --help for more options, including capturing screenshots of each step.

7. Run the tests as services in GroundWork

   You can now set up these recorded tests as services in GroundWork. As a best practice, keep the tests simple, and try not to make any one script too long - 3-5 steps at most.
   Instead of one long script, break up the tests into discrete functional areas, so failures can be more precisely detected with separate service checks.

   Implementing the tests as services in GroundWork is really the same as for any other plugin. Here's a profile you can upload to **/usr/local/groundwork/core/profiles**, to get you started with a simple service that you can copy and modify as needed.

Name	Size	Creator	Creation Date	Comment
service-profile-web_transactions_ca...	3 kB	Thomas Stocking	Apr 02, 2018 17:43	

References

- Resurrectio: https://chrome.google.com/webstore/detail/resurrectio/kicncbplfjgjlliddogifpohdhkbjogm?hl=en
- Plugin development and documentation: https://git.netways.org/plugins/check_casperjs
- Phantomjs development: http://phantomjs.org
- Casperjs development: http://casperjs.org