PhantomJS and CasperJS

See Browser Automation.

PhantomJS is a headless WebKit with JavaScript API. It has fast and native support for various web standards: DOM handling, CSS selector, JSON, Canvas, and SVG. PhantomJS is an optimal solution for fast headless testing, site scraping, pages capture, SVG renderer, network monitoring and many other use cases.

CasperJS is a navigation scripting & testing utility for PhantomJS and SlimerJS (still experimental). It eases the process of defining a full navigation scenario and provides useful high-level functions, methods & syntactic sugar for doing common tasks such as:
- defining & ordering navigation steps
- filling forms
- clicking links
- capturing screenshots of a page (or an area)
- making assertions on remote DOM logging & events
- downloading resources, even binary ones
- catching errors and react accordingly
- writing functional test suites, exporting results as JUnit XML (xUnit)

How to install CasperJS and PhantomJS for Tiki

Starting in Tiki18, PhantomJS and CasperJS are available via Packages

1.1.1. Test the installation

The simplest is to use: PluginCasperJS. Otherwise, use the following script to test (in the root of tiki, adjust path if other folder):

Create a file (ex.: scrape.php) with the following content

```php
<?php
// Require composer autoload
require_once __DIR__ . '/vendor/autoload.php';

use Browser\Casper;

$casper = new Casper(__DIR__ . '/bin/');
$casper->setOptions(array(
    'ignore-ssl-errors' => 'yes'
));

// screen shot of homepage
$casper
    ->start('http://tiki.org')
    ->setViewPort(1024,800)
    ->waitForText('About Tiki')
    ->wait(200)
    ->capture(
        array(
            'top' => 0,
            'left' => 0,
            'width' => 1024,
            'height' => 800
        )
    );
```
Once you visit the newly created file, it will fetch the info at http://tiki.org and generate custom-capture.png.

1.1.2. Expose them system-wide (optional)

In case you need to run some custom programs on a server which require casperjs and phantomjs, you may expose those programs to the whole server so that they can get accessed system wide from the usual /usr/bin folder where all programs are located on symlinked.

Imagine that your root tiki folder is /var/www/html/. Then you can create a symbolic link for each of the programs:

```
ln -s /var/www/html/bin/casperjs /usr/bin/casperjs
ln -s /var/www/html/bin/phantomjs /usr/bin/phantomjs
```

This way you will be able to call casperjs or phantomjs from anywhere in the server, if you need it for your other custom scripts doing scraping for you server-side.

Troubleshooting

If you get the following error, just stop the script and try again later:

```
> Tiki\Composer\BootstrapCompiler::build
> Tiki\Composer\CleanVendors::clean
> PhantomInstaller\Installer::installPhantomJS
   - Installing phantomjs (1.9.8)
      Downloading: Connecting...
Could not fetch
https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-1.9.8-linux-x86_64.tar.bz2, please create a bitbucket OAuth token to go over the API rate limit
Follow the instructions on
https://confluence.atlassian.com/bitbucket/oauth-on-bitbucket-cloud-238027431.html
to create a consumer. It will be stored in "/root/.composer/auth.json" for future use by Composer.
Ensure you enter a "Callback URL" or it will not be possible to create an Access Token (this callback url will not be used by composer)
Consumer Key (hidden):

To test, and check version
[root@ -]# phantomjs -v
1.9.8
```
Related links

- PhantomJS
- Browser Automation
- http://wikisuite.org/How-to-install-PhantomJS-on-ClearOS

Open Hub

alias

- PhantomJS | CasperJS