PhantomJS SOURCE CODE DOCUMENTATION API

Please take a moment to improve this document with anything that could be useful to other develope

Documentation

Get Started

- D Download
- D Build
- D Releases
- D Release Names
- D REPL

Learn

- D Quick Start
- D Headless Testing
- D Screen Capture
- D Network Monitoring
- D Page Automation
- D Inter Process Communicatio
- D Command Line Interface

Get Help

- D Troubleshooting
- D FAQ

Explore

- D Examples
- D Best Practices
- D Tips and Tricks
- D Supported Web Standards
- D Buzz
- D Who's using PhantomJS?
- D Related Projects

Contribute

- D Contributing
- D Source Code
- D Test Suite
- D Release Preparation
- D Crash Reporting

Screen Capture

Since PhantomJS is using WebKit, a real layout and rendering engine, i as a screenshot. Because PhantomJS can render anything on the web p convert contents not only in HTML and CSS, but also SVG and Canvas.

The following script demonstrates the simplest use of page capture. It loa and then saves it as an image, `github.png`.

```
var page = require('webpage').create();
page.open('http://github.com/', function() {
  page.render('github.png');
  phantom.exit();
});
```

To run this example create a new file called `github.js`. Copy and paste t `github.js` file. In the commandline, run this newly created script with Pha

```
phantomjs github.js
```

Beside PNG format, PhantomJS supports JPEG, GIF, and PDF.

In the `examples` subdirectory, there is a script rasterize.js (30 lines) whic complete rendering feature of PhantomJS. An example to produce the re Tiger (from SVG):

```
phantomjs rasterize.js http://ariya.github.io/svg/tiger.svg tiger.png
```

which gives the following `tiger.png`:

Another example is to show polar clock (from RaphaelJS):

```
phantomjs rasterize.js http://raphaeljs.com/polar-clock.html clock.png
```

Producing PDF output is also easy, e.g. from a Wikipedia article:

```
phantomjs rasterize.js 'http://en.wikipedia.org/w/index.php?title=Jakar
```

You can change the size of the screenshot and the webpage using the a

```
var page = require('webpage').create();
//viewportSize being the actual size of the headless browser
page.viewportSize = { width: 1024, height: 768 };
//the clipRect is the portion of the page you are taking a screenshot of
page.clipRect = { top: 0, left: 0, width: 1024, height: 768 };
//the rest of the code is the same as the previous example
page.open('http://example.com/', function() {
  page.render('github.png');
  phantom.exit();
});
```

Canvas can be easily constructed and converted to an image. The inclu
(50 lines) produces the following color wheel:

It is possible to build a web screenshot service using PhantomJS. Some
easy to create such a service.

© Copyright 2010-2015 Ariya Hidayat — Homepage design by Maurice Svay — Documentation design b