



# CasperJS + PhantomJS

## Headless automation testing

Anup Menon



## What is it that we are trying to achieve ?

- **We are trying to achieve headless browser automation using Casperjs and Phantomjs.**
- **We chose to work with the CasperJS framework because it works headless.**
- **Easier to integrate it with Jenkins.**
- **We don't need any extra libraries to carry on our automation and there is no dependency on external library bindings.**
- **All you need is casperjs (and python) installed in your machine and an editor where you can start coding.**

## What is CasperJS and PhantomJS ?

- **CasperJS acts as a wrapper for PhantomJS that helps extending the automation capability and functionalities of PhantomJS.**
- **Has features like taking screenshots, applying assertions etc.**
- **PhantomJS is more of a scripted headless browser which is also used for automating web page interaction.**
- **PhantomJS in its background uses ‘GhostDriver’ to carry on the headless functionalities.**
- **It is not necessary to use PhantomJS as a browser. User can also define other user agents also.**

# Installation

## For Mac

**--Installing from Homebrew (OSX)**

Installation of both PhantomJS and CasperJS can be achieved using Homebrew, which is a popular package manager for Mac OSX.

```
$ brew update
```

```
$ brew install casperjs --devel (Recommended to install the development version)
```

```
$ brew install casperjs (For stable version). If you want to upgrade to the latest version then just do
```

```
$ brew upgrade casperjs
```

## For Linux

First we have to install phantomJS and then casperJS, make sure you install phantomJS 1.9.2 or above.

```
$ git clone git://github.com/n1k0/casperjs.git
```

```
$ cd casperjs
```

```
$ ln -sf `pwd`/bin/casperjs /usr/local/bin/casperjs
```

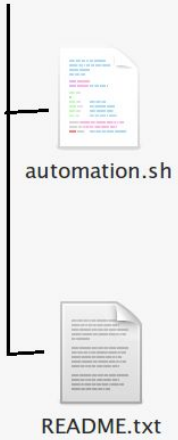
## Some Basic Methods

- To begin a test suite
  - `casper.test.begin(test_name, number of test cases, function suite () { });`
- To open a URL in a browser and perform some action
  - `this.open(url).then(function() { });`
- To take a screenshot
  - `this.capture(relative path + name of the image)`
- To wait for until a resource is loaded in the page
  - `this.waitForResource(Resource, function() { });`
- To start a test
  - `casper.start(url, function() { });`
- ★ To handle an iframe
  - ★ `casper.withFrame(iframe name, function() { });`

# The Structure we are following.



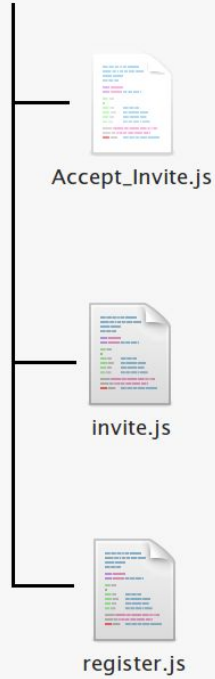
README  
2 items



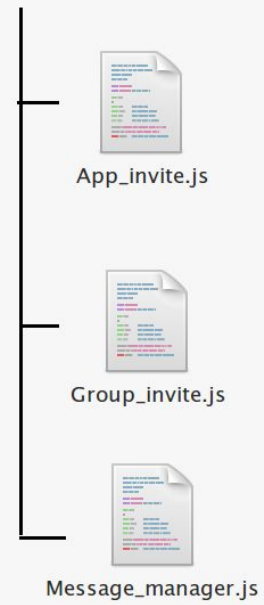
Reports  
2 items



Scenarios  
3 items



Testcases  
1 item



config.js

**DEMO**

**Questions ???**



# Thank you

[anup.menon@srijan.in](mailto:anup.menon@srijan.in)