The information on [Protractor’s site](http://www.protractortest.org) is rather complete. The site not only displays how to setup, configure, to write and execute protractor tests it also provides a [style guide](http://www.protractortest.org/#/style-guide). Protractor is a [Node.js](https://nodejs.org/en/) program. To execute protractor, you will need to have Node.js installed. Protractor is built on top [WebDriverJS](https://github.com/SeleniumHQ/selenium/wiki/WebDriverJs). Java Development Kit (JDK) needs to be installed to run a local Selenium Server. Selenium WebDriver supports several browser implementations or drivers. The default behavior driven development (BDD) test framework in the Protractor install is [Jasmine](http://jasmine.github.io/). Below is the version of Protractor and supporting software installed on my personal computer.

|  |  |
| --- | --- |
| Description | Command |
| Version of Node | $ node --version  v4.4.5 |
| Version of npm | $ npm --version  2.15.5 |
| Version of Protractor (also the version of Jasmine and Selenium WebDriver) | $ protractor --version  Version 3.3.0  $ npm ls -depth=0  protractor@3.3.0 /usr/local/lib/node\_modules/protractor  ├── adm-zip@0.4.7  ├── chalk@1.1.3  ├── glob@6.0.4  ├── jasmine@2.4.1  ├── jasminewd2@0.0.9  ├── optimist@0.6.1  ├── q@1.4.1  ├── request@2.67.0  ├── saucelabs@1.0.1  ├── selenium-webdriver@2.52.0  └── source-map-support@0.4.0 |
| Version of Java | $ java -version  java version "1.8.0\_45"  Java(TM) SE Runtime Environment (build 1.8.0\_45-b14)  Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode) |

Protractor needs a test file and a configuration file to run. This [reference.js](https://github.com/angular/protractor/blob/master/docs/referenceConf.js) shows all of the configuration options that may be passed to Protractor. The configuration file tells Protractor where the test file(s) is located and where to talk to the Selenium Server. The test files use the syntax of your test framework, for example Jasmine, and the [Protractor API](http://www.protractortest.org/#/api). Below demonstrates how to run a Protractor test.

|  |  |
| --- | --- |
| Description | Command |
| 1. Start Selenium Server by either: |  |
| a. Start Selenium WebDriver | webdriver-manager start |
| b. or update the configuration file to reference the selenium standalone server jar | Reference to the selenium standalone server jar.  seleniumServerJar: './selenium\_server\_standalone/selenium-server-standalone-2.53.1.jar', |
| 2. Run Protractor Test | protractor conf.js |

Below is the Protractor configuration, test file, and page object files used in the [AngularJSWebsiteTestByProtractorJasmine](https://github.com/gdombchik/AngularJSWebsiteTestByProtractorJasmine) GitHub repository.

|  |  |  |
| --- | --- | --- |
| Functionality | Description | JavaScript Files |
| Protractor Configuration File | Protractor configuration options used:   * seleniumAddress: “To connect to a Selenium Server which is already running.” * seleniumServerJar: “Reference to the selenium standalone server jar. * capabilities:{‘browser’:}: “Protractor will launch specified browser. In our example we are launching a headless browser PhantomJS.” * framework: “Framework to use. Jasmine is the default test framework.” * specs: “The Protractor test files.”   jasmineNodeOpts: “Options to be passed to Jasmine.” | conf.js |
| Page Object Design Pattern | A Protractor test file to test the AngularJS home page. The angularjsSpec.js test file calls the page objects located in angularjsWebsite/pageObjects folder.  Use the [Page Objects](http://www.protractortest.org/#/style-guide) design pattern.  Reasons to use the Page Objects design pattern as specified by the Protractor style guide:   * Encapsulate information about the elements on the page under test * They can be reused across multiple tests   Decouple the test logic from implementation details | *Test File:*  angularjsSpec.js  *Page Objects:*  homePage.js  downloadAngularJSOnePage.js  theBasics.js  addSomeControl.js  createComponents.js  wireUpABackend.js |

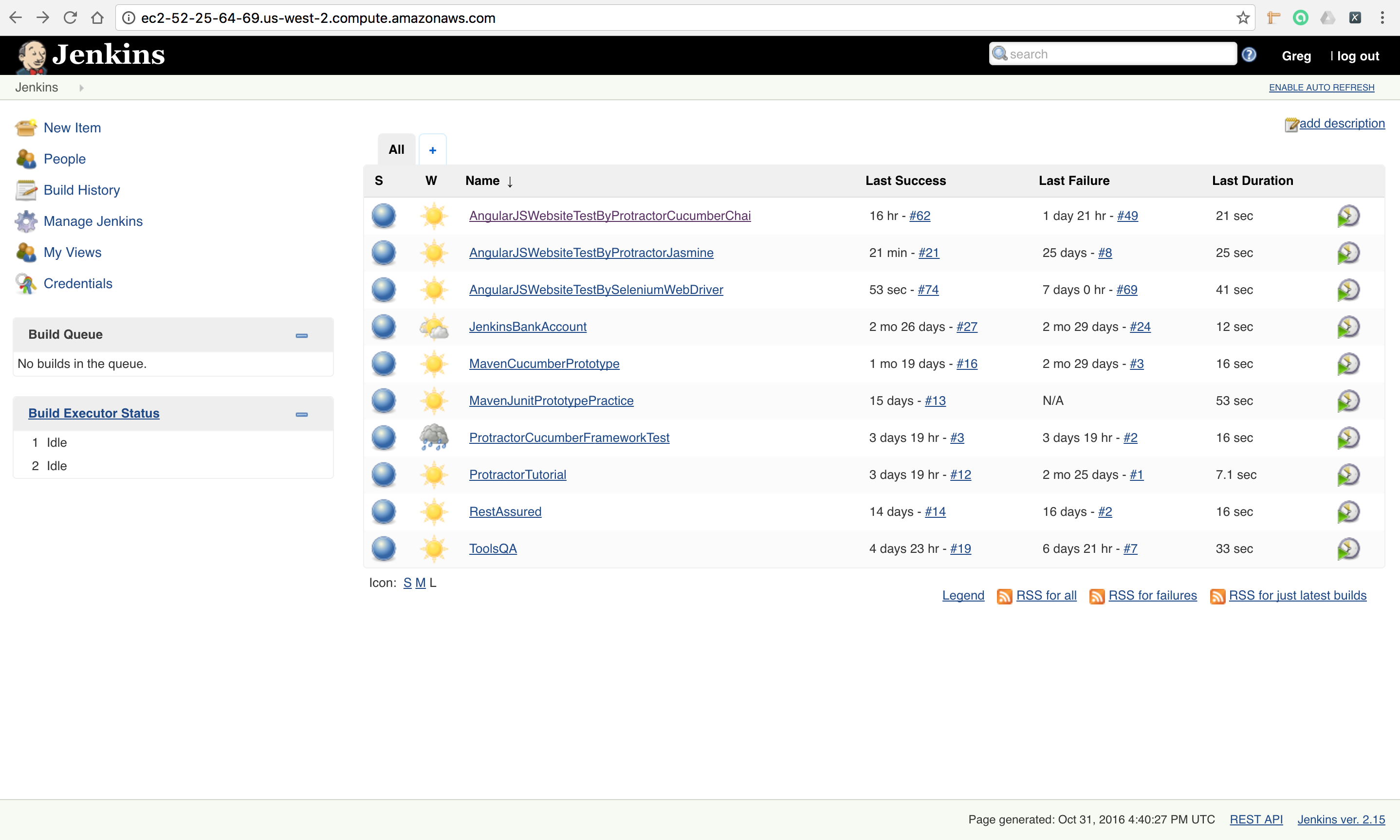
In reviewing how to run the Protractor tests using a continuous integration (ci) tool such as Jenkins, an option is to use the standalone Selenium Server jar start the selenium server, use Jenkins “execute shell” option to execute the Protractor configuration file, run the tests using a headless environment using tools such as PhantomJS, and capture test results.

The steps below are to be executed on the server where Jenkins is hosted. This example Jenkins has been installed on an AWS E2 Ubuntu server.

|  |  |
| --- | --- |
| Description | Command |
| Installation of nodejs | $ curl -sL https://deb.nodesource.com/setup\_4.x | sudo -E bash -  $ sudo apt-get install -y nodejs  $ npm –version  2.15.8 |
| Global Installation of Protractor and webdriver-manager | $ sudo npm install -g protractor  $ protractor --version  Version 4.0.2  $ sudo webdriver-manager update |
| Global Installation of PhantomJS | $ sudo npm install -g phantomjs-prebuilt  $ phantomjs --version  2.1.1 |
| Update the configuration file to reference PhantomJS | Update conf.js and change the browserName to phantomjs.  capabilities: {  'browserName': 'phantomjs' }, |
| Update the configuration file to reference the selenium standalone server | Reference selenium server standalone jar.  seleniumServerJar: './selenium\_server\_standalone/selenium-server-standalone-2.53.1.jar', |

The Jenkins setup of a Protractor project in GitHub. Does not include the installation and setup of Jenkins on AWS E2 Ubuntu server.

|  |  |
| --- | --- |
| Description | Command |
| *AWS E2 Ubuntu server* | |
| Switch to the Jenkins user. | $ sudo su – Jenkins |
| Create public private key pair. | $ cd .ssh  $ ssh-keygen -t dsa |
| Cat the public key and copy the results. | $ cat AngularJSWebsiteTestByProtractorJasmine.pub |
| *GitHub* | |
| Add and paste deploy key in GitHub project. | <https://github.com/gdombchik/AngularJSWebsiteTestByProtractorJasmine/settings/keys> |
| *Manage Jenkins – Configure System* | |
| Jenkins Location | Scroll to the “Jenkins Location” section.  Jenkins URL:  <http://ec2-52-25-64-69.us-west-2.compute.amazonaws.com/>  System Admin e-mail address:  [greg@gregorydombchik.com](mailto:greg@gregorydombchik.com) |
| E-mail Notification | Scroll to the “E-mail Notification” section.  SMTP server:  mail.gregorydombchik.com |
| *Jenkins Server – New Project* | |
| Add A Jenkins Project for Protractor project. | Select New Item.  Enter an item name.  Select Freestyle project.  Select Ok. |
| Specify the URL of the remote GitHub repository. | Scroll to the “Source Code Management” section.  Select the “Git” option.  Enter Repository URL:  <https://github.com/gdombchik/AngularJSWebsiteTestByProtractorJasmine.git> |
| Update the Poll SCM. | Scroll to the “Build Triggers” section.  Select the “Poll SCM” option.  Enter the following in the “Schedule” text box:  H \*/3 \* \* \* |
| Update the Build. | Scroll to the “Build” section.  Select from the “Add build step” and select the “Execute shell” option.  Enter the following in the “Command” text box:  ./angularjsWebsite.sh  I source angularjsWebsite.sh on the Jenkins server so I can move from the root of the AngularJSWebsiteTestByProtractorJasmine folder to the angularjsWebsite folder and execute the Protractor configuration file.  ubuntu@ip-172-31-18-39:/var/lib/jenkins/workspace/ AngularJSWebsiteTestByProtractorJasmine $ source angularjsWebsite.sh |
| Update E-mail Notification. | Scroll to the Post-build Actions.  Select from the “Add post-build action” and select the “E-mail Notification” option.  Enter email address in the “Recipients” text box:  [greg@gregorydombchik.com](mailto:greg@gregorydombchik.com) |
| *GitHub* | |
| Add Jenkins GitHub Plugin you can automatically trigger build jobs when  pushes are made to GitHub. | <https://github.com/gdombchik/AngularJSWebsiteTestByProtractorJasmine/settings/installations>  Select from the “Add service” and select the “Jenkin’s (Git plugin)” option. (NOT GITHUB PLUGIN. This service does not appear to work).  Enter the following in the “Jenkins url” text box:  <http://ec2-52-25-64-69.us-west-2.compute.amazonaws.com/> |



In summation, a front-end test framework can be built by installing Protractor and supporting software, generate Page Objects to represent the web application, build test or specs referencing the generated Page Objects in the Protractor and Jasmine test framework, and then porting the solution into a continuous integration environment.