

Admin View (ambari-admin) Development

Frontend Development

Follow the instructions here to ease frontend development for Admin View (ambari-admin module):

1. Follow the [Quick Start Guide](#) to install and start Ambari Server (cluster need not be deployed).
2. Follow the "Frontend Development" section in [Quick Start Guide](#) to check out the Ambari source using git. This makes the entire Ambari source available via /vagrant/ambari from the Vagrant VM.
3. From the Ambari Server host:
 - a. `cd /var/lib/ambari-server/resources/views/work` <- if this directory does not exist, you have not started ambari-server; run "ambari-server start" to start it
 - b. `mv ADMIN_VIEW\{2.5.0.0\} /tmp`
 - c. `ln -s /vagrant/ambari/ambari-admin/src/main/resources/ui/admin-web/dist ADMIN_VIEW\{2.5.0.0\}`
 - d. `cp /tmp/ADMIN_VIEW\{2.5.0.0\}/view.xml ADMIN_VIEW\{2.5.0.0\}/`
 - e. `ambari-server restart`
4. Now you can change the source code for Admin View and run gulp locally, and the changes are automatically reflected on the server.

Functional Tests

To run end-to-end functional tests on the browser, execute:

```
npm run update-webdriver
npm start (This starts http server at 8000 port)
```

Open another terminal at same path and execute: `npm run protractor` (does e2e test in the browser. This library works on top of selenium jar).

Unit Tests

To run unit tests:

Go to path: `/ambari/ambari-admin/src/main/resources/ui/admin-web`
Execute `npm run test-single-run` (this uses PhantomJS headless browser; it's the same used by the ambari-web unit tests)

Note:

"npm test" command starts karma server at <http://localhost:9876/> and runs unit tests. This server remain up, autoreloads any changes in the test code and reruns the tests. This is useful while developing unit tests.