

# jboss-docker-e2e-solution (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-09-10 11:52:15 UTC

# Table of Contents

1. Introduction .....	1
2. Implementation .....	2
2.1. docker (Oracle, JBoss, Selenium Hub + Chrome).....	2
2.2. project under test (legacy JavaEE app running in JBoss) .....	2
2.3. e2e-tests .....	2
2.4. documentation .....	4
3. Links .....	5

# Chapter 1. Introduction

This repository contains end-to-end testing of legacy system which is basically absolutely doesn't has any tests at all.

Testing is going base on selenium-hub with chrome node

Test could be implemented with JUnit 5 and written using modern languages, like latest fancy java 8/10/11 or Kotlin

Also we going to bootstrap fresh new oracle instance before tests

Read [project reference documentation](#)

Initially generated by using [generator-jvm](#) yeoman generator (java-parent-multi-project)

# Chapter 2. Implementation

## 2.1. docker (Oracle, JBoss, Selenium Hub + Chrome)

*docker-compose*

```
./mvnw -pl legacy-java-ee-app

./mvnw -pl docker docker-compose:up
./mvnw -pl docker -P up
./mvnw -pl docker -P tail

./mvnw -pl docker docker-compose:down
./mvnw -pl docker -P down
```

## 2.2. project under test (legacy JavaEE app running in JBoss)

*docker-compose*

```
./mvnw -pl legacy-java-ee-app clean package docker-compose:up

./mvnw -pl legacy-java-ee-app docker-compose:down
```

## 2.3. e2e-tests

1) *pull needed images*

```
docker pull selenium/hub:3.14.0-beryllium
docker pull selenium/node-chrome-debug:3.14.0-beryllium
docker pull selenium/node-firefox-debug:3.14.0-beryllium
```

2) *run hub and link browser nodes (use `--shm-size=2g` if needed)*

```
docker run -d -p 4444:4444 --rm --name selenium-hub selenium/hub:3.14.0-beryllium
```

3) *verify if grid console available*

```
open :4444/grid/console
# or
docker logs -f selenium-hub &
```

4) link browser nodes (use `--shm-size=2g` if needed)

```
docker run -d -P --link selenium-hub:hub --rm --name node-chrome-debug selenium/node-chrome-debug:3.14.0-beryllium
docker logs -f node-chrome-debug &
```

```
docker run -d -P --link selenium-hub:hub --rm --name node-firefox-debug selenium/node-firefox-debug:3.14.0-beryllium
docker logs -f node-firefox-debug &
```

5) finally, run e2e tests

```
./mvnw -DargLine="-Dwebdriver.browsers=chrome -Dremote=http://127.0.0.1:4444/wd/hub" test
./mvnw -DargLine="-Dbrowsers=firefox -Dremote=http://127.0.0.1:4444/wd/hub" test
```



see `docker/docker-compose-selenium-hub.yaml` file for detail, how to setup selenium grid

```
version: '2.1'
```

```
services:
```

```
  selenium-hub:
```

```
    image: selenium/hub:3.14.0-beryllium
```

```
    ports: ['4444:4444']
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      backing-services:
```

```
        aliases:
```

- hub
- selenium-hub
- hub.daggerok.github.com
- selenium-hub.daggerok.github.com

```
  node-chrome-debug:
```

```
    image: selenium/node-chrome-debug:3.14.0-beryllium
```

```
    depends_on: [selenium-hub]
```

```
    environment:
```

```
      HUB_HOST: selenium-hub
```

```
    ports: ['5900']
```

```
    restart: unless-stopped
```

```
    networks:
```

```
      backing-services:
```

```
        aliases:
```

- chrome
- node-chrome-debug
- chrome.daggerok.github.com
- node-chrome-debug.daggerok.github.com

```
#shm_size: 2g
```

```
node-firefox-debug:
```

```
  image: selenium/node-firefox-debug:3.14.0-beryllium
```

```
  depends_on: [selenium-hub]
```

```
  environment:
```

```
    HUB_HOST: selenium-hub
```

```
  ports: ['5900']
```

```
  restart: unless-stopped
```

```
  networks:
```

```
    backing-services:
```

```
      aliases:
```

```
        - firefox
```

```
        - node-firefox-debug
```

```
        - firefox.daggerok.github.com
```

```
        - node-firefox-debug.daggerok.github.com
```

```
#shm_size: 2g
```

```
networks:
```

```
  backing-services:
```

```
    driver: bridge
```

*selenium*

```
docker ps
```

CONTAINER ID	IMAGE	STATUS	PORTS	COMMAND
c702811463c3	selenium/node-chrome-debug:3.14.0-beryllium	17 seconds ago	Up 16 seconds	"/opt/bin/entry_poin..."
			0.0.0.0:32768->5900/tcp	

```
# to connect by using vnc client use:
```

```
# addr:port
```

```
127.0.0.1:32768
```

```
# password:
```

```
secret
```

## 2.4. documentation

```
./mvnw -pl docs
```

# Chapter 3. Links

- [GitHub repo](#)
- [GitHub pages](#)