

Title: Unable to run Springboot app using Selenium RemoteWebDriver with docker-compose

Post Body:

i am containerising my spring boot app which uses selenium/standalone-firefox-debug i have created docker compose file,but when i up it it gives me error as

Could not start a new session. Possible causes are invalid address of the remote server or browser start-up failure.

But if i run the spring-boot app directly and selenium/standalone-firefox-debug seperatly it works.I want to run it with docker-compose

Dockerfile:

```
FROM openjdk:8-jdk-alpine VOLUME /tmp ARG JAR_FILE ADD ${JAR_FILE} app.jar ENTRYPOINT exec java -jar /app.jar
```

Dockercompose:

```
version: '2.2' services:  employer-url:      image: 'adib/employer-url'      ports:      - '8080:8080'      depends_on:      - f
```

this is how i create driver in spring app

```
RemoteWebDriver driver = new RemoteWebDriver(new URL('http://localhost:4444/wd/hub'), DesiredCapabilities.firefox());
```

Accepted Answer:

Problem

This `http://localhost:4444/wd/hub` URL refers to the localhost which belongs to the container runtime. The springboot container does not have port 4444 running that's why it's complaining.

Solution

You should access the selenium service from its hostname (not localhost). In springboot application, you can use, `http://firefox:4444/wd/hub` URL and you'd be good to go.

Rationale

You are missing the core of the networking concept in containers here. Both of these images (viz springboot and selenium) are running inside containers and hence they have their separate environments. If you refer to localhost inside any container, it means the localhost of that container. You are expecting localhost to refer to the localhost of the docker host machine. You exposed port 4444 on the docker host machine. So if you try to run your jar from the docker host (while selenium is containerized) `localhost:4444` would work but if you access it from inside a container, it's not going to work. Following Diagram shows the concept:

Highest Rated Answer: None