

Title: Spring Boot REST app returns 400 when requested from other docker-compose service by service name

Post Body:

I'm trying to introduce a Spring Boot REST service in our development setup. The development setup is using docker-compose and an API gateway to expose the individual services on the same domain (ie. localhost).

When I try to make a HTTP request to my service from inside another container via the service name in the shared docker-compose file, the service returns a 400.

The setup

I've edited our docker-compose file, so it looks like the below to introduce the Spring Boot Java service. The service is based on spring-boot-starter-parent (2.0.3.RELEASE) and spring-boot-starter-web. I haven't configured anything related to the web server (except adding the server.server-header property to ensure myself that the request is hitting my service).

```
version: '3' services: ... hello_java: build: context: ../hello-java/ dockerfile: Dockerfile depends_
```

The behaviour

If I request the hello service from outside the containers (e.g. in my browser on localhost:8301) it replies back correctly. If I'm inside a container, but obtain the IP that the container with my new service gets in the docker network and use that the new service also responds correctly back.

Below I have shown a request from inside the API gateway container to the Java service, first by using the service name and then afterwards with the IP that was resolved. It only replies with a correct response in the IP case.

```
# curl -v http://hello_java:8080/hello-java/greet?username=Java * Hostname was NOT found in DNS cache * Trying 172.19.0.6...
```

The questions

Is there something in the standard spring-boot-starter-web setup that prevents the web server from servicing the request, when the client adds the 'Host: hello_java:8080' header? Or why is the web server behaving differently in the two scenarios? And what can I do about it?

Accepted Answer: None

Highest Rated Answer:

After some experimentation it turned out that the it was the underscore in the service name that caused the issue. Changing the service name to not have an underscore solved the problem.