Title: Spring boot 1.5.9, 404 error while accessing resources images within Docker container

Post Body:

In a Spring Boot 1.5.9 application I want to access static resources through HTTP using Google Chrome.

In Docker I have **`404` error**, it worked fine in embedded tomcat.

Configuration

This is the `Dockerfile`:

```
FROM openjdk:8-jre-alpine VOLUME /tmp  RUN addgroup -S spring --gid 13371 && adduser -S spring -G spring --uid 13371 RUN mkdir
```

This is how I configure the static resources:

```
public class MyWebMvcConfigurerAdapter extends WebMvcConfigurerAdapter {      @Override     public void addResourceHandlers(Re
```

Result

This is the result log when I access the URL from chrome **in docker**
`http://localhost:8080/resources/promotions/37/lg/31723d95-eec3-4ff9-aa42-e3477559e9df.png`

```
Counting up[http-nio-8080-Acceptor-0] latch=1 Socket: [org.apache.tomcat.util.net.NioEndpoint$NioSocketWrapper@45deea1:org.apa
```

Expected

This is what I expect (in the development environment) while accessing
`http://localhost:8080/resources/promotions/39/lg/843291ef-2309-4809-9f37-38257e087a97.jpeg`

```
Counting up[http-nio-8080-Acceptor-0] latch=1 Socket: [org.apache.tomcat.util.net.NioEndpoint$NioSocketWrapper@4d64a855:org.ap
```

Investigation

I have replicated the production environment locally and I used the same absolute path `file:/srv/upload/images` in development, I can confirm that status code are different within docker and without it:

- In embedded tomcat: 200
- In docker: 404

The problem is not related to the path being different but only because of docker.

Question

- Is there a step I have forgotten?

Any hint would be much appreciated.

Accepted Answer:

I don't have enough reputation points to make a comment, so I write my comment as an answer :(

I had a similar issue with docker and resources, so I would like to help you.

Could you provide logs with `TRACE` level? This level helps us to see what is going on inside `PathResourceResolver` and according to this implementation trace is used in most cases.

In my case, it was useful and helped me.

According to this

> You can enable a “trace” mode by starting your application with a --trace flag (or trace=true in your application.properties)

Highest Rated Answer: None