

Title: Docker: Can't read class path resource from spring boot application

Post Body:

Reading a classpath resource as,

```
try {
    final ClassPathResource classPathResource = new ClassPathResource(format('location%sGeoLite2-City.mmdb', File.separator), this.getClass().getClassLoader());
```

I've packaged this with docker using following Dockerfile,

```
FROM java:8 ADD build/libs/*.jar App.jar CMD java -jar App.jar
```

But while running this application as **docker run -p 8080:8080 app-image** I can hit the application endpoint and from application logs I can see it fails to read this file (following is from logs),

```
Exception: java.io.FileNotFoundException: class path resource [location/GeoLite2-City.mmdb] cannot be resolved to absolute file path
```

Would appreciate any comment, Things to know before you comment,

**\*\***- Running on windows 10, intellij 2018.2, jdk 8

- Can run application successfully with intellij as well as command line - File exists in jar (I did extract jar and checked ) **\*\***

Accepted Answer:

It is not a good approach to use slashes.

Always use File Separators as they work irrespective of System OS.

Change

```
(location\\GeoLite2-City.mmdb)
```

to

```
('location'+ File.separator + 'GeoLite2-City.mmdb')
```

Refer this for more.

<https://www.journaldev.com/851/java-file-separator-separatorchar-pathseparator-pathseparatorchar>

[Difference between File.separator and slash in paths](#)

Highest Rated Answer:

Since you are using springboot you can try to use the following annotation for loading your classpath resource. Worked for me because I had the same exception. Be aware that the directory 'location' must be under the `src/main/resources` folder:

```
@Value('classpath:/location/GeoLite2-City.mmdb') private Resource geoLiteCity;
```

Without springboot you could try:

```
try (InputStream inputStream = getClass().getClassLoader().getResourceAsStream('/location/GeoLite2-City.mmdb')) { ... //convert to string }
```

Also the answers before were correct as the use of '/' is not good at all and `File.separator` would be the best choice.