Title: Spring boot application won't run when trying to run from the jar file

Post Body:

I have a spring boot application which works fine when run through intellij. But when I run it from the jar I am getting the below exception.

```
        2017-02-13 05:18:28.596  WARN 8581 --- [           main] ationConfigEmbeddedWebApplicationContext : Exception encounte
```

Below is my pom file.

```
<?xml version='1.0' encoding='UTF-8'?> <project xmlns='http://maven.apache.org/POM/4.0.0' xmlns:xsi='http://www.w3.org/2001/XM
```

I tried to create the jar by both

```
mvn clean package
```

and

```
mvn clean install spring-boot:repackage
```

Then I run the jar by java -jar -DParamName='someParam'

What am I doing wrong here? I am new to Spring. Any help would be much appreciated.

Accepted Answer:

You have `spring-boot-starter-jersey` in your pom.xml. The error you are getting is `/BOOT-INF/classes (No such file or directory)`. This is the issue with jersey which doesn't work well with spring-boot fat jar. Please refer this and this for more details. For work around you have to package the classes that should be scanned by jersey in a jar and extract the jar as given at Spring-Boot documentation Extract specific libraries when an executable jar runs

Highest Rated Answer:

@abaghel answer has the correct explanation, and arguably the best solution as well.

Since Jersey's scanning step is what's failing, another way to work around it is to explicitly register each resource. I replaced this:

```
@Component public JerseyConfig() extends ResourceConfig {     packages('com.mydomain.resource'); }
```

with

```
@Component public JerseyConfig() extends ResourceConfig {     register(TimeResource.class);     register(SpaceResource.class);
```

(Tested with Spring Boot 2.1.3)

My code has about 25 Jersey of these resource classes to register. This highlights the downside to this approach, namely that the explicit registration has to be maintained. New resource classes won't be found automatically, and it can be a confusing error to run into.

On the other hand, you don't need to split your project up. So explicit registration might occasionally be good enough.