

Εργαστηριακή άσκηση 7: Containers

1 Open project from lab 05

Open previous project or Checkout from git the source code from lab 5

2 Change pom.xml

Pom xml must be changed to build a jar that contains all the libraries

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.4.0</version>
    <relativePath /> <!-- lookup parent from repository -->
  </parent>
  <groupId>gr.upatras</groupId>
  <artifactId>rest.example</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>rest.example</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>11</java.version>
    <swagger.version>3.0.0</swagger.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>

    <!-- swagger -->
    <dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-boot-starter</artifactId>
      <version>${swagger.version}</version>
    </dependency>
    <dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-swagger2</artifactId>
      <version>${swagger.version}</version>
    </dependency>

    <dependency>
      <groupId>io.springfox</groupId>
      <artifactId>springfox-swagger-ui</artifactId>
      <version>${swagger.version}</version>
    </dependency>

  </dependencies>

  <repositories>
    <repository>
      <id>spring-milestones</id>
```

```

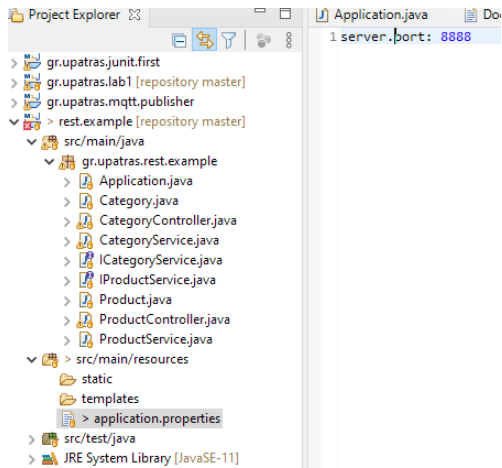
        <name>Spring Milestones</name>
        <url>https://repo.spring.io/milestone</url>
        <snapshots>
            <enabled>false</enabled>
        </snapshots>
    </repository>
    <repository>
        <id>spring-snapshots</id>
        <name>Spring Snapshots</name>
        <url>https://repo.spring.io/snapshot</url>
        <releases>
            <enabled>false</enabled>
        </releases>
    </repository>
</repositories>
<pluginRepositories>
    <pluginRepository>
        <id>spring-milestones</id>
        <name>Spring Milestones</name>
        <url>https://repo.spring.io/milestone</url>
        <snapshots>
            <enabled>false</enabled>
        </snapshots>
    </pluginRepository>
    <pluginRepository>
        <id>spring-snapshots</id>
        <name>Spring Snapshots</name>
        <url>https://repo.spring.io/snapshot</url>
        <releases>
            <enabled>false</enabled>
        </releases>
    </pluginRepository>
</pluginRepositories>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <version>2.4.0</version>
            <configuration>
                <classifier>exec</classifier>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>

```

3 Change application.properties

application.properties file is a special file for java Springboot framework. Make it like:



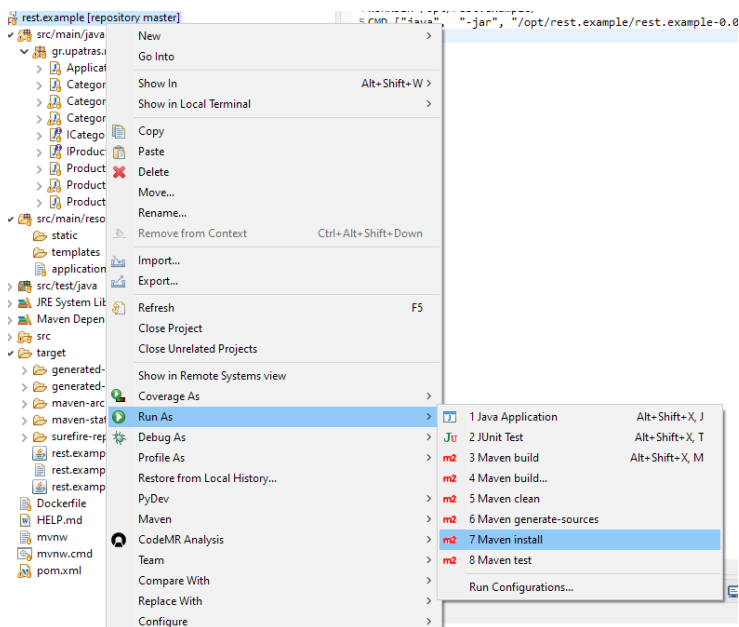
`server.port: 8888`

The application will now listen on port 8888

Run it and make a GET on

`curl -X GET "http://localhost:8080/product/" -H "accept: application/json;charset=utf-8"`

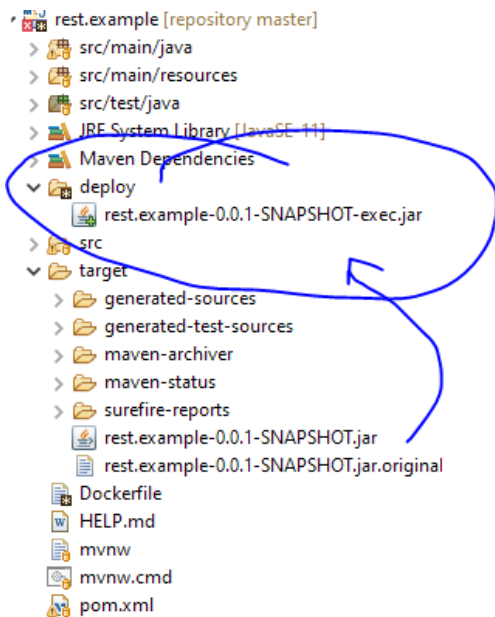
4 Build the project



Select Run AS-> Maven Install

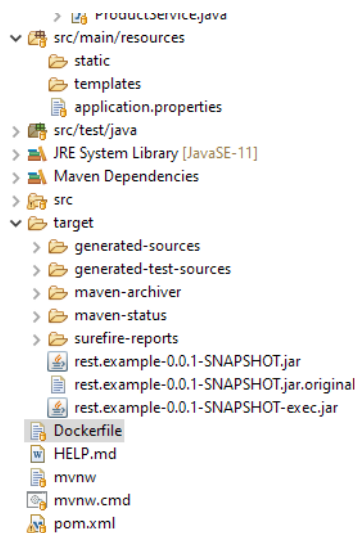
Check the target folder for the files

Create a new folder called deploy and move there the file rest.example-0.0.1-SNAPSHOT-exec.jar:



5 Create the Dockerfile

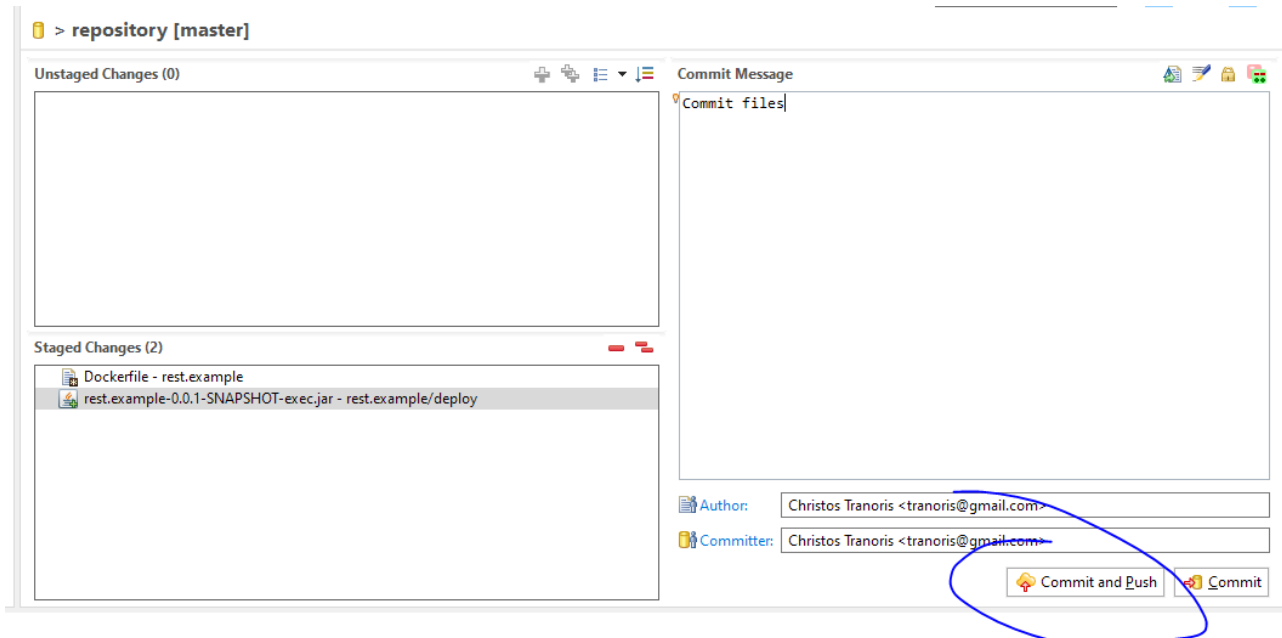
Create a new file called Dockerfile in the root folder of the project as following:



Type the following:

```
FROM adoptopenjdk/openjdk13-openj9:latest
MAINTAINER ece.upatras.gr
COPY deploy/rest.example-0.0.1-SNAPSHOT-exec.jar /opt/rest.example/
WORKDIR /opt/rest.example/
CMD ["java", "-jar", "/opt/rest.example/rest.example-0.0.1-SNAPSHOT-exec.jar"]
EXPOSE 8888
```

6 Push changes to Git



7 Build the docker container

You need to do the following on a machine that has docker installed

```
~/labs/lab1_test/rest.example > master $ sudo docker build -t examplereast .
Sending build context to Docker daemon 25.36MB
Step 1/6 : FROM adoptopenjdk/openjdk13-openj9:latest
--> 9a988b64a140
Step 2/6 : MAINTAINER ece.upatras.gr
--> Using cache
--> 167dda4fc04d6
Step 3/6 : COPY deploy/rest.example-0.0.1-SNAPSHOT-exec.jar /opt/rest.example/
--> Using cache
--> 8b8df9f05a51
Step 4/6 : WORKDIR /opt/rest.example/
--> Using cache
--> 20fd9a516b03
Step 5/6 : CMD ["java", "-jar", "/opt/rest.example/rest.example-0.0.1-SNAPSHOT-exec.jar"]
--> Using cache
--> d1d7f4e17d65
Step 6/6 : EXPOSE 8888
--> Using cache
--> e747f3b6ae1f
Successfully built e747f3b6ae1f
Successfully tagged examplereast:latest
~/labs/lab1_test/rest.example > master $
```

See all images:

```
Successfully tagged examplereast:latest
~/labs/lab1_test/rest.example > master $ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
examplereast        latest             e747f3b6ae1f       About a minute ago 445MB
ctranoris/hello-lab v1                 962472a4c78a       2 months ago      912MB
```

Execute your container

```
sudo docker run -d examplere
```

```
~/labs/lab1_test/rest.example > master sudo docker run -d examplere
c5bd4687a023b9a15f9156d2ebf25d29e1e73483862fd4e2dcbc9f930cd00e70
~/labs/lab1_test/rest.example > master
```

The container now run as a daemon

```
sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c5bd4687a023	examplere	"java -jar /opt/rest..."	About a minute ago	Up About a minute	8888/tcp	vibrant_robinson

However, we did not exposed a port from the container! The docker file exposed port 8888.

So if you make a GET localhost:8888/product/ it **will NOT return** you the data.

Kill the container

```
sudo docker kill <CONTAINER_ID>
```

e.g. `sudo docker kill c5bd4687a023`

Rerun the container with the following command:

```
sudo docker run -d -p 9090:8888 examplere
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c7cea942451c	examplere	"java -jar /opt/rest..."	4 seconds ago	Up 2 seconds	0.0.0.0:9090->8888/tcp, :::9090->8888/tcp	sad_chatelet

make a GET localhost:9090/product/

NOTICE: If you are running from Virtualbox you need to expose the port 9090 to your host machine!