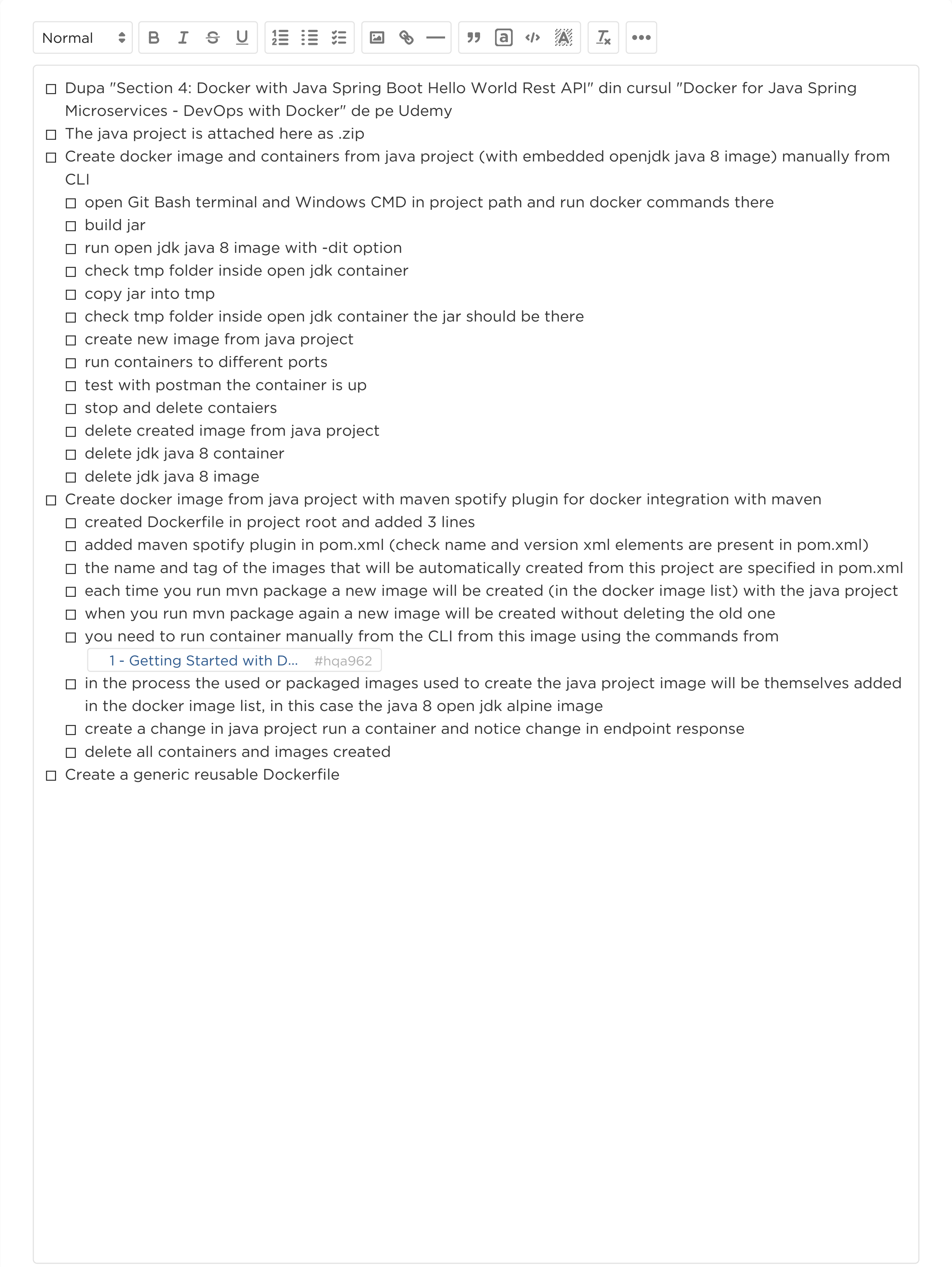
Docker Udemy



- □ Dupa "Section 4: Docker with Java Spring Boot Hello World Rest API" din cursul "Docker for Java Spring Microservices - DevOps with Docker" de pe Udemy
- □ The java project is attached here as .zip
- □ Create docker image and containers from java project (with embedded openjdk java 8 image) manually from CLI
  - □ open Git Bash terminal and Windows CMD in project path and run docker commands there
  - □ build jar
  - □ run open jdk java 8 image with -dit option
  - □ check tmp folder inside open jdk container
  - □ copy jar into tmp
  - □ check tmp folder inside open jdk container the jar should be there
  - □ create new image from java project
  - □ run containers to different ports
  - □ test with postman the container is up
  - □ stop and delete contaiers
  - □ delete created image from java project
  - □ delete jdk java 8 container
  - □ delete jdk java 8 image
- □ Create docker image from java project with maven spotify plugin for docker integration with maven
  - □ created Dockerfile in project root and added 3 lines
  - □ added maven spotify plugin in pom.xml (check name and version xml elements are present in pom.xml)
  - □ the name and tag of the images that will be automatically created from this project are specified in pom.xml
  - □ each time you run mvn package a new image will be created (in the docker image list) with the java project
  - □ when you run mvn package again a new image will be created without deleting the old one
  - □ you need to run container manually from the CLI from this image using the commands from

    1 - Getting Started with D... #hqa962
  - □ in the process the used or packaged images used to create the java project image will be themselves added in the docker image list, in this case the java 8 open jdk alpine image
  - □ create a change in java project run a container and notice change in endpoint response
  - □ delete all containers and images created
- □ Create a generic reusable Dockerfile