

Framework Spring JPA Parte 2

Índice

2. Docker, Render y BBDD.

2.1. Despliegue de proyecto con Docker sin BBDD.

2.2. Tarea: Investiga cómo crear un proyecto con Docker usando BBDD.

2.3. Tarea: Añade tu proyecto a Render.

2. Docker. Render y BBDD.

2.1. Despliegue de proyecto con Docker sin BBDD.

Docker es un proyecto de código abierto que permite crear un paquetes de software, o contenedor, que se ejecuta en todos los tipos de dispositivos y sistemas operativos.

Los pasos para crear un proyecto con Docker son:

Paso1: Archivo de configuración Dockerfile. Para proyecto creado con Java Version 23 y packaging jar puedes usar las siguientes líneas de código dentro de un archivo llamado Dockerfile sin extensión:

```
# Stage 1: Build the application
FROM eclipse-temurin:23-jdk AS builder
# Set the working directory
WORKDIR /app
# Copy the application code
COPY . .
# Given permissions to mvnw
RUN chmod +x mvnw
# Build the application (requires Maven or Gradle)
RUN ./mvnw clean package -DskipTests

# Stage 2: Run the application
FROM eclipse-temurin:23-jre
# Set the working directory
WORKDIR /app
# Copy the JAR file from the builder stage
```

```
COPY --from=builder /app/target/*.jar app.jar
# Expose the port the app will run on
EXPOSE 8080
# Command to run the application
ENTRYPOINT ["java", "-jar", "app.jar"]
```

Paso2: En la terminal del IDE debes usar el siguiente comando para crear una imagen en Docker, accediendo a la ruta de tu proyecto, es decir, donde puedas ver el archivo DockerFile: “\$ dir Dockerfile HELP.md mvnw mvnw.cmd pom.xml src target”. A continuación, lanza el comando:

```
# docker build -t nombreimagen .
docker build -t spring-boot-docker .
```

Espera a que se genere la imagen. A continuación abre la aplicación Docker Desktop para revisar que ya está creada dicha imagen en la pestaña images.

Una imagen de Docker, o una imagen de contenedor, es un archivo ejecutable e independiente que se utiliza para crear un contenedor. Esta imagen de contenedor contiene las bibliotecas, las dependencias y los archivos que el contenedor necesita para ejecutarse. Una imagen de Docker se puede compartir y transportar; por lo tanto, se puede implementar la misma imagen en varias ubicaciones a la vez.

Paso3: En la terminal del IDE debes usar el siguiente comando para crear un contenedor de la imagen anterior (De esta forma puedes lanzar la aplicación / algoritmo).

```
# docker run -d -p puertousado nombreimagen
docker run -d -p 8080:8080 spring-boot-docker
```

En la aplicación Docker Desktop debe aparecer dentro de la pestaña containers. Pincha en run y prueba a lanzar desde el navegador tu aplicación (Ya no sería necesario un IDE para mostrar tu aplicación final/proyecto)

Un contenedor de Docker es un entorno de ejecución que tiene todos los componentes necesarios (como el código, las dependencias y las bibliotecas) para ejecutar el código de la aplicación sin utilizar las dependencias de la máquina host.

2.2. Tarea: Investiga cómo crear un proyecto con Docker usando BBDD.

En Internet hay manuales para usar Docker con proyectos que accedan a bases de datos reales. Entre las opciones disponibles, existe el fichero docker-compose.yml.

Ejercicio: Debes investigar cómo usar bases de datos en proyectos Docker. Para ello, crea un ejercicio sencillo que acceda a una tabla de una BD MySQL. Documenta todo el proceso en un PDF que sirva de manual con el paso a paso. *Sube el código y la explicación en PDF a la entrega llamada “Entrega Ejercicios Tema 4 Parte 2”*

2.3. Tarea: Añade tu proyecto a Render.

Una mejora interesante que ví en los proyectos finales de 2ºDAW (Junio 2024) podría ser usar Render para tu proyecto.

Para ello mira el siguiente vídeo explicativo donde, paso a paso, nos muestran qué es Render y cómo se usa: <https://www.youtube.com/watch?v=2rkoryDkf6o>

Ejercicio: Debes investigar cómo usar proyectos a partir de Render. Documenta todo el proceso en un PDF que sirva de manual con el paso a paso. *Sube el código y la explicación en PDF a la entrega llamada “Entrega Ejercicios Tema 4 Parte 3”*