

Spring Boot CRUD Application - Project Documentation

1. Project Overview

This is a Java Spring Boot application for CRUD operations with a MySQL database. It is containerized with Docker and managed using Docker Compose.

2. Tech Stack

Java 17
Spring Boot
MySQL (Docker)
Maven
Docker & Docker Compose

3. Features

- REST API for CRUD
- Dockerized Deployment
- MySQL Integration
- Health Checks

4. Installation & Build

1. Clone repo: `git clone <repo-url>`
2. Build: `mvn clean package`
3. Docker build & run: `docker-compose up --build`

5. Dockerfile

```
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY ./target/*.jar app.jar
ENV SPRING_DATASOURCE_URL=jdbc:mysql://mysql-container:3306/myapplication?createDatabaseIfNotExist=true
ENV SPRING_DATASOURCE_USERNAME=root
ENV SPRING_DATASOURCE_PASSWORD=1234
EXPOSE 8081
ENTRYPOINT ["java", "-jar", "app.jar"]
```

6. docker-compose.yml

```
version: '3.8'
```

```
services:
```

```
  mysql:
```

```
    image: mysql:latest
```

```
    container_name: mysql-container
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: 1234
```

Spring Boot CRUD Application - Project Documentation

ports:

- "3306:3306"

networks:

- my-network

healthcheck:

test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]

interval: 10s

timeout: 5s

retries: 5

sakgroup:

build:

context: .

dockerfile: Dockerfile

container_name: springapp

image: sakgroup

ports:

- "8081:8081"

networks:

- my-network

depends_on:

mysql:

condition: service_healthy

networks:

my-network:

name: my-network

driver: bridge

7. Accessing the App

Spring Boot App: <http://localhost:8081>

MySQL: localhost:3306

MySQL user: root

Password: 1234

8. Push to Docker Hub

docker tag sakgroup your-dockerhub-username/sakgroup

docker login

docker push your-dockerhub-username/sakgroup