

# README FILE

## (PROJECT 2)

---

- **INTRODUCTION**

Implementation of the spring-boot project for Employee/Department Management module, which has basic CRUD APIs using Spring JPA.

"Project 2: Extend the project 1 :

- 1) Use @transactional in the project ( implement transactions)
- 2) Use hibernate relationship mapping annotation ( eg. one to many)
- 3) Redis as a cache ( implement for nonfrequent changing data - eg. Department table)
- 4) Implement JUNITs using Mockito

"

- **REQUIREMENTS**

Pre-requisite: Install Mysql, JAVA, IntelliJ.

- **SETUP.**

- 1) Download the zip & extract it.
- 2) Open the project in IntelliJ
- 3) Reload maven dependencies
- 4) Running the application, run the demoApplication.java
- 5) For unit testing, open test folder and run.

- **CURL COMMANDS**

- 1) **GET REQUEST**

curl <http://localhost:8080/api/v1/employee>

- 2) **POST REQUEST**

curl -d  
'{"name":"William","email":"william@gmail.com","phnumber":"987515627","age":"22","salary":"32000","department":"Backend Dev"}' -H "Content-Type: application/json" -X POST <http://localhost:8080/api/v1/employee>

- 3) **PUT REQUEST**

curl -X PUT -H "Content-Type: application/json"  
<http://localhost:8080/api/v1/employee/2?name=Shriki>

#### 4) DELETE REQUEST

curl -X DELETE <http://localhost:8080/api/v1/employee/3>

---

To access individual data in API change from employee to department or vice versa.