Nasruddin A'aidil Shari

[Email address]

Abstract

Short read me on how to build test and deploy the application

EMPLoyee api

NPHC take home challenge

# Pre-requisites

The following are pre-requisites for the application to work

|  |  |
| --- | --- |
| Program | Version |
| Postman | v9.20.3 |
| JDK | 17 |
| Postgres | 14 |

# Cloning

Clone the repository by running the following command:

git clone <https://github.com/DinDragon/nphcChallenge.git>

# Setting up DB

Install Postgress from https://www.enterprisedb.com/downloads/postgres-postgresql-downloads and choose the appropriate version

Setup postgres db with the following properties:

port 8000

username postgres

password pswd1234

Run the following:

Create new database

-- Database: NPHC

-- DROP DATABASE IF EXISTS "NPHC";

CREATE DATABASE "NPHC"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_Singapore.1252'

LC\_CTYPE = 'English\_Singapore.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

If you wish to use your own configuration and update the {project folder}/src/main/resources/application.properties with the following variables:

spring.datasource.url=jdbc:postgresql://localhost:{your db port}/{your db name}

spring.datasource.username={your username}

spring.datasource.password={your password}

# Setting up DB table

Spring boot will automatically create the table for you once you run the web application.

# To build the file

open command line and navigate to project folder /Employee/

run command "mvn clean install"

# To Run Unit test

open command line and navigate to project folder /Employee/

run command "mvn test"

# To Run web application

open command line and navigate to project folder /Employee/target/

run command "java -jar Employee-0.0.1-SNAPSHOT.jar"

# To test the application with postman

Install postman

The requests are as specified in the specification given. However for the fetch API, there are additional parameters for sorting and filtering. These params are:

sortBy: The data you want it sorted by. Will accept only id,login,name,salary and startDate

order: To specify the order of how the data appears. Will accept only “asc”, “desc”

name: To filter out the data list using the employee name.

For the update API, the request body or params was not specified in the specs. However, in the request parameters you can specify the following for update:

1. login,
2. name
3. salary
4. startDate

Optional: I have exported json file for Postman which comes preloaded with the requests required for testing. Please feel free to use the file.

# Future improvement on solutions:

For future improvements, I would like to implement a proper converter for the request params. Currently it is using the default converter which is unable to respond to exception handling in an elegant way. By implementing the Convert for class Employee, parsing exception can be integrated and return a proper responise entity.

Due to lack of time, was unable to implement mock mvc and Mockito for testing which can smooth out testing greatly. An improvement if given the time would like to implement these features.