1. Solution and implementation for Problem Statement 1:

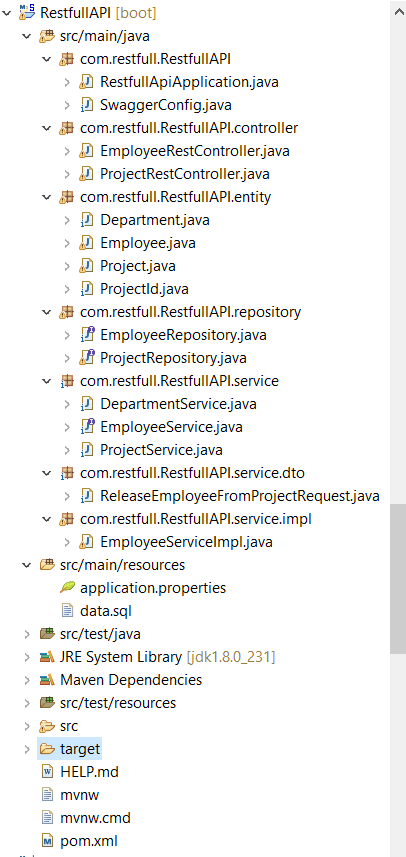
High level Design and Configuration of the Applications:

1. Solution is implemented using Spring boot, Maven, Java, Packaging jar and Java 8

2. Configure with Rest, Spring data jpa and H2

3. plugged- in with swagger and junit

Project structure:



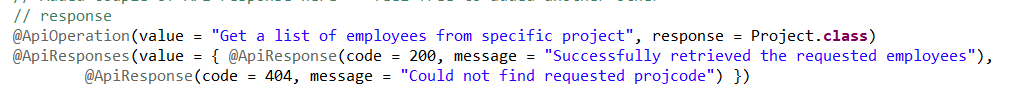
Main class – RestfullApplication

API implémentation 🡪 com.restfull.RestfullAPI.Controller package

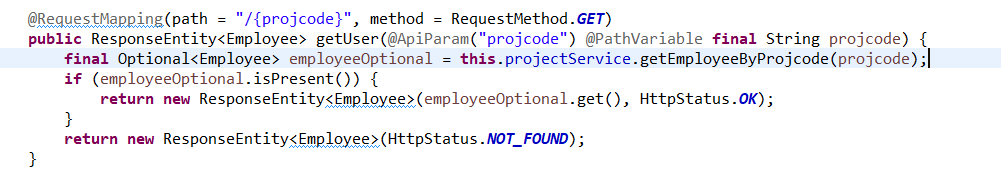
Hibernate pojo implémentation 🡪 com.restfull.RestfullAPI.entity

API parameters handling -> ProjectRestController.java: with first API request, I have handled the parameter using url and can be retrieve the employee details based on below query: http://localhost:8080/<project code>

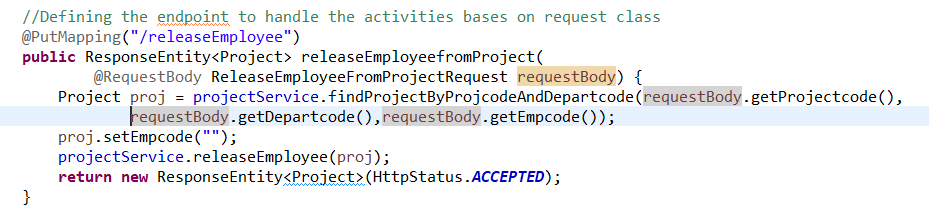
Exception handling of API request: It was implemented along with custom generated exception and provided with 2 response code however can be added any number of responses based on need.



But code can handle any types of responses.



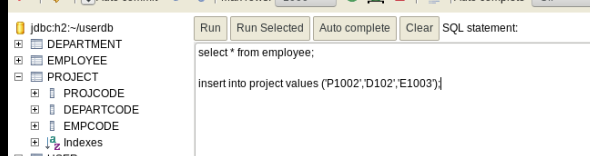
For handling multiple parameters rather than using pathvariable I have handled the input based on Request Body by creating a separate class and we can use end point to access the details.



Swagger is plugged in with the implementation hence can be test the flows using Swagger using URL: http://localhost:8080/swagger-ui.html

Junit – As of now, I have commented the code however created the framework to Mockito the API calls using that.

H2 DB can be accessible using url: http://localhost:8080/h2/ : User – sa and no password



DDL Commands to be used to create the table structures, although I have already placed data.sql file on resource folder so that by triggering the project it will create the DB.

I have designed based on basic requirement and there is not much columns added on it so please consider this based on constraints provided:

DB Structure and SQL -- for table structure the data

Note: Please use the exactly same structure else it would be problematic to access the data

Table Creation of Employee using below SQL statement:

Create table Employee (empcode varchar2(6) primary key, empname varchar2 (30));

DML – Inserts for Employee table:

insert into employee values ('E10001','Test1');

insert into employee values ('E10002','Test2');

insert into employee values ('E10003','Test3');

insert into employee values ('E10004','Test4');

insert into employee values ('E10005','Test5');

insert into employee values ('E10006','Test6');

insert into employee values ('E10007','Test7');

Table Creation of Department using below SQL statement:

Create table Department (departcode varchar2(4) primary key, departname varchar2(30));

DML – Inserts for Department table:

insert into department values ('D101','Department1');

insert into department values ('D102','Department2');

insert into department values ('D103','Department3');

insert into department values ('D104','Department4');

Table Creation of Project using below SQL statement:

Create table Project (projcode varchar2(5), departcode varchar2(4),empcode varchar2(6), primary key(projcode,departcode),foreign key (departcode) references Department(departcode), foreign key (empcode) references Employee (empcode));

DML – Inserts for Department table:

Note: Ensure data is getting exactly same as per empcode and departcode else it will not allow to insert as references are defined in db level

insert into project ('P1001','D101','E10001');

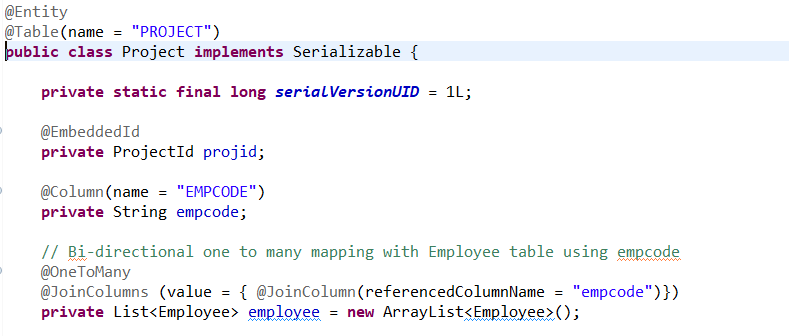
insert into project ('P1001','D102','E10001');

insert into project ('P1002','D101','E10003');

insert into project ('P1002','D103','E10004');

insert into project ('P1002','D104','E10005');

Although, it was not required to handle composite key on db level but ensure smooth functionality I have create the project class by ensure all the configuration are defined based on db structure



Also implemented the composite key using ProjectId class :

