

REST API For Employee Management

Training Report Submitted in Partial Fulfilment of the Requirements
for the Degree of

Bachelor of Engineering *in* **Computer Science and Engineering**

Submitted by

Manish Singh: (Roll No. 18UCSE4016)

Under the Supervision of

Miss Deepika Kamboj

Assistant Professor

MBM Engineering College, Jodhpur



Department of Computer Science and Engineering

M.B.M. Engineering College

Faculty of Engineering & Architecture

Jai Narain Vyas University, Jodhpur

May, 2020



Department of Computer Science & Engineering

M.B.M. Engineering College, Jai Narain Vyas University
Ratanada, Jodhpur, Rajasthan, India –342011

CERTIFICATE

This is to certify that the work entitled “**REST API For Employee Management**” was carried out by Mr. Manish Singh (Roll. No: 18UCSE4016) during his training from May 2020 to June 2020 under the supervision of Miss Deepika Kamboj, Assistant Professor at MBM Engineering College, Jodhpur.

Miss Deepika Kamboj
Assistant Professor
MBM Engineering College, Jodhpur

DECLARATION

I, ***Manish Singh*** hereby declare that this work titled “***REST API For Employee Management***” is a record of original work done by me under the supervision and guidance of ***Miss Deepika Kamboj, Assistant Professor at MBM Engineering College*** from May, 2020 to July 2020.

I, further certify that this work has not formed the basis for the award of the Degree/Diploma/Associateship/Fellowship or similar recognition to any candidate of any university and no part of this report is reproduced as it is from any other source without appropriate reference and permission.

(**Manish Singh**)
viith Semester, CSE
Enroll. – 17R/06986
Roll No. – 18UCSE4016

ACKNOWLEDGEMENT

I would like to express my kindest appreciation to all those who provided me the possibility to complete this report. Special gratitude to my guide “Miss Deepika Kamboj” for introducing me to the field of “REST API Development” and her contribution in simulating suggestions, constant supervision and encouragement helped me to coordinate my work. I would also like to acknowledge with much appreciation to the Head of CSE department, Dr. NC Barwar and also to all faculty members and staff of the department in carrying out my work and have willingly helped me out with their abilities. Last but not the least, I would like to express my gratitude towards my parents and friends for their kind cooperation and encouragement which helped me in the completion of my work.

ABSTRACT

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.

An API is a set of definitions and protocols for building and integrating application software. It's sometimes referred to as a contract between an information provider and an information user—establishing the content required from the consumer (the call) and the content required by the producer (the response). For example, the API design for a weather service could specify that the user supply a zip code and that the producer reply with a 2-part answer, the first being the high temperature, and the second being the low.

In other words, if one wants to interact with a computer or system to retrieve information or perform a function, an API helps communicate what you want to that system so it can understand and fulfil the request.

Contents

CERTIFICATE	2
DECLARATION	3
ACKNOWLEDGEMENT	4
ABSTRACT.....	5
List of Figures	7
Chapter 1.....	8
INTRODUCTION.....	8
Features of REST API-	8
Chapter 2_ Technology Used.....	10
Spring.....	10
Maven	10
Dependencies -	14
Hibernate -	15
Jackson-.....	16
MySql-	16
Chapter 3_ Work Completed	17
API Requirements –	17
REST API -	17
Development Process Step-By-Step -	18
Application Architecture:.....	18
Chapter 4_ Results	19
Response for http requests:	19
Chapter 5.....	22
Conclusion & Future Work.....	22
References.....	23

List of Figures

Figure 1:Working of Maven	11
Figure 2: Local, Remote & Central Repository	13
Figure 3: ORM Tool	15
Figure 4: API Requirements	17
Figure 5: Application Architecture	18
Figure 6:Response GET request	19
Figure 7:Response GET Request with parameter	20
Figure 8:POST Request add employee	20
Figure 9:PUT Request update employee	21
Figure 10: DELETE Request with parameter	21

Chapter 1

INTRODUCTION

REST is a set of architectural constraints, not a protocol or a standard. API developers can implement REST in a variety of ways.

When a client request is made via a RESTful API, it transfers a representation of the state of the resource to the requester or endpoint. This information, or representation, is delivered in one of several formats via HTTP: JSON (Javascript Object Notation), HTML, XML, Python, PHP, or plain text. JSON is the most generally popular programming language to use because, despite its name, it's language-agnostic, as well as readable by both humans and machines.

Something else to keep in mind: Headers and parameters are also important in the HTTP methods of a RESTful API HTTP request, as they contain important identifier information as to the request's metadata, authorization, uniform resource identifier (URI), caching, cookies, and more. There are request headers and response headers, each with their own HTTP connection information and status codes.

Features of REST API-

- A client-server architecture made up of clients, servers, and resources, with requests managed through HTTP.
- Stateless client-server communication, meaning no client information is stored between get requests and each request is separate and unconnected.
- Cacheable data that streamlines client-server interactions.
- A uniform interface between components so that information is transferred in a standard form. This requires that:
 - resources requested are identifiable and separate from the representations sent to the client.
 - resources can be manipulated by the client via the representation they receive because the representation contains enough information to do so.
 - self-descriptive messages returned to the client have enough information to describe how the client should process it.
 - hypertext/hypermedia is available, meaning that after accessing a resource the client should be able to use hyperlinks to find all other currently available actions they can take.

- A layered system that organizes each type of server (those responsible for security, load-balancing, etc.) involved the retrieval of requested information into hierarchies, invisible to the client.
- Code-on-demand (optional): the ability to send executable code from the server to the client when requested, extending client functionality.

Chapter 2

Technology Used

It is a REST API which is build using Spring framework.

Spring

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

The physical level structure of scheme is based upon Maven.

Maven

What is Maven?

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

What maven does?

Maven does a lot of helpful task like

1. We can easily build a project using maven.
2. We can add jars and other dependencies of the project easily using the help of maven.
3. Maven provides project information (log document, dependency list, unit test reports etc.)
4. Maven is very helpful for a project while updating central repository of JARs and other dependencies.
5. With the help of Maven we can build any number of projects into output types like the JAR, WAR etc without doing any scripting.
6. Using maven we can easily integrate our project with source control system (such as Subversion or Git).

How maven works?

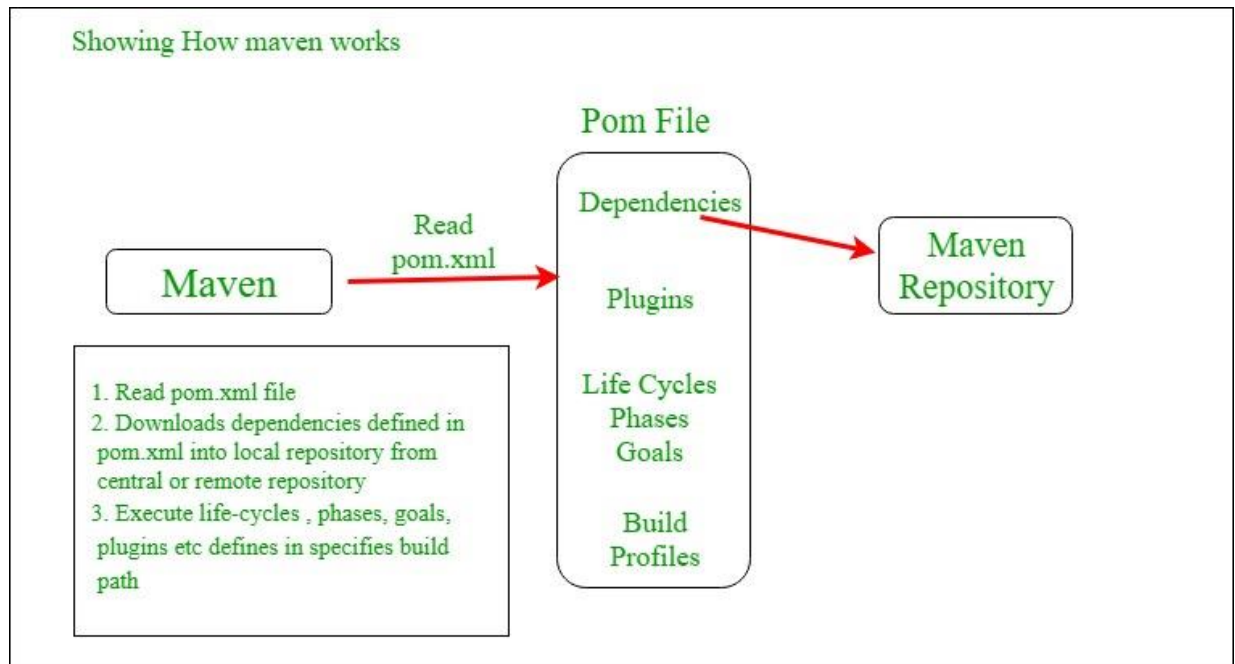


Figure 1:Working of Maven

Core Concepts of Maven:

1. **POM Files:** Project Object Model(POM) Files are XML file that contains information related to the project and configuration information such as dependencies, source directory, plugin, goals etc. used by Maven to build the project. When you should execute a maven command you give maven a POM file to execute the commands. Maven reads pom.xml file to accomplish its configuration and operations.
2. **Dependencies and Repositories:** Dependencies are external Java libraries required for Project and repositories are directories of packaged JAR files. The local repository is just a directory on your machine hard drive. If the dependencies are not found in the local Maven repository, Maven downloads them from a central Maven repository and puts them in your local repository.
3. **Build Life Cycles, Phases and Goals:** A build life cycle consists of a sequence of build phases, and each build phase consists of a sequence of goals. Maven command is the name of a build lifecycle, phase or goal. If a lifecycle is requested executed by giving maven command, all build phases in that life cycle are executed also. If a build phase is requested executed, all build phases before it in the defined sequence are executed too.
4. **Build Profiles:** Build profiles a set of configuration values which allows you to build your project using different configurations. For example, you may need to build your project for your local computer, for development and test. To enable different builds you can add different build profiles to your POM files using its profiles elements and are triggered in the variety of ways.

5. **Build Plugins:** Build plugins are used to perform specific goal. you can add a plugin to the POM file. Maven has some standard plugins you can use, and you can also implement your own in Java.

Maven pom.xml file

POM means Project Object Model is key to operate Maven. Maven reads pom.xml file to accomplish its configuration and operations. It is an XML file that contains information related to the project and configuration information such as **dependencies, source directory, plugin, goals etc.** used by Maven to build the project.

Elements used for Creating pom.xml file

1. **project-** It is the root element of the pom.xml file.
2. **modelVersion-** modelversion means what version of the POM model you are using. Use version 4.0.0 for maven 2 and maven 3.
3. **groupId-** groupId means the id for the project group. It is unique and Most often you will use a group ID which is similar to the root Java package name of the project like we used the groupId com.project.loggerapi.
4. **artifactId-** artifactId used to give name of the project you are building.in our example name of our project is LoggerApi.
5. **version-** version element contains the version number of the project. If your project has been released in different versions then it is useful to give version of your project.

Other Elements of Pom.xml file

1. **dependencies-** dependencies element is used to defines a list of dependency of project.
2. **dependency-** dependency defines a dependency and used inside dependencies tag. Each dependency is described by its groupId, artifactId and version.
3. **name-** this element is used to give name to our maven project.
4. **scope-** this element used to define scope for this maven project that can be compile, runtime, test, provided system etc.
5. **packaging-** packaging element is used to packaging our project to output types like JAR, WAR etc.

Maven Repository

Maven repositories are directories of packaged JAR files with some metadata. The metadata are POM files related to the projects each packaged JAR file belongs to, including what external dependencies each packaged JAR has. This metadata enables Maven to download dependencies of your dependencies recursively until all dependencies are download and put into your local machine.

Maven has three types of repository :

1. Local repository
2. Central repository
3. Remote repository

Maven searches for dependencies in this repositories. First maven searches in Local repository then Central repository then Remote repository if Remote repository specified in the POM.

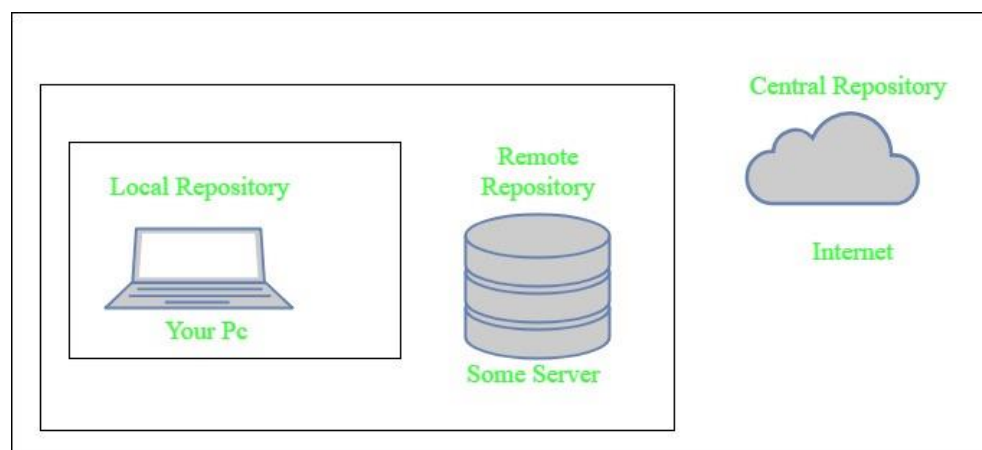


Figure 2: Local, Remote & Central Repository

1. Local repository- A local repository is a directory on the machine of developer. This repository contains all the dependencies Maven downloads. Maven only needs to download the dependencies once, even if multiple projects depends on them (e.g. ODBC). By default, maven local repository is user_home/m2 directory. example – C:\Users\manish_singh\.m2
2. Central repository- The central Maven repository is created Maven community. Maven looks in this central repository for any dependencies needed but not found in your local repository. Maven then downloads these dependencies into your local repository. You can view central repository by [this link](#).

3. Remote repository- remote repository is a repository on a web server from which Maven can download dependencies. It is often used for hosting projects internal to organization. Maven then downloads these dependencies into your local repository.

Practical Application of Maven

When working on a java project and that project contains a lot of dependencies, builds, requirement, then handling all those things manually is very difficult and tiresome. Thus, using some tool which these works can do is very helpful. Maven is such a *build management tool* which all the things like can adding dependencies, managing the classpath to project, generating war and jar file automatically and many other things.

Pros and Cons of using Maven

Pros:

1. Maven can add all the dependencies required for the project automatically by reading pom file.
2. One can easily build their project to jar, war etc. as per their requirements using Maven.
3. Maven makes easy to start project in different environments and one doesn't need to handle the dependencies injection, builds, processing, etc.
4. Adding a new dependency is very easy. One has to just write the dependency code in pom file.

Cons:

1. Maven needs the maven installation in the system for working and maven plugin for the ide.
2. If the maven code for an existing dependency is not available, then one cannot add that dependency using maven.

Dependencies -

- Hibernate
- Jackson
- MySql

Hibernate -

Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

ORM Tool

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.

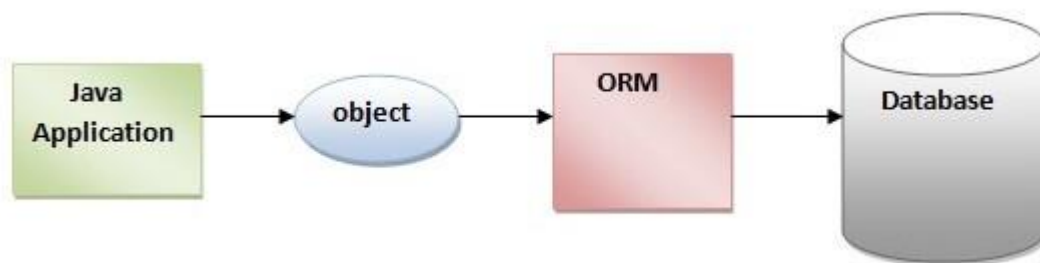


Figure 3: ORM Tool

The ORM tool internally uses the JDBC API to interact with the database.

Advantages of Hibernate Framework

Following are the advantages of hibernate framework:

1) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.

3) Database Independent Query

HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.

4) Automatic Table Creation

Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

5) Simplifies Complex Join

Fetching data from multiple tables is easy in hibernate framework.

6) Provides Query Statistics and Database Status

7) Hibernate supports Query cache and provide statistics about query and database status

Jackson-

In computing, **Jackson** is a high-performance JSON processor for Java. Its developers extol the combination of fast, correct, lightweight, and ergonomic attributes of the library.

MySql-

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

Chapter 3

Work Completed

API Requirements –

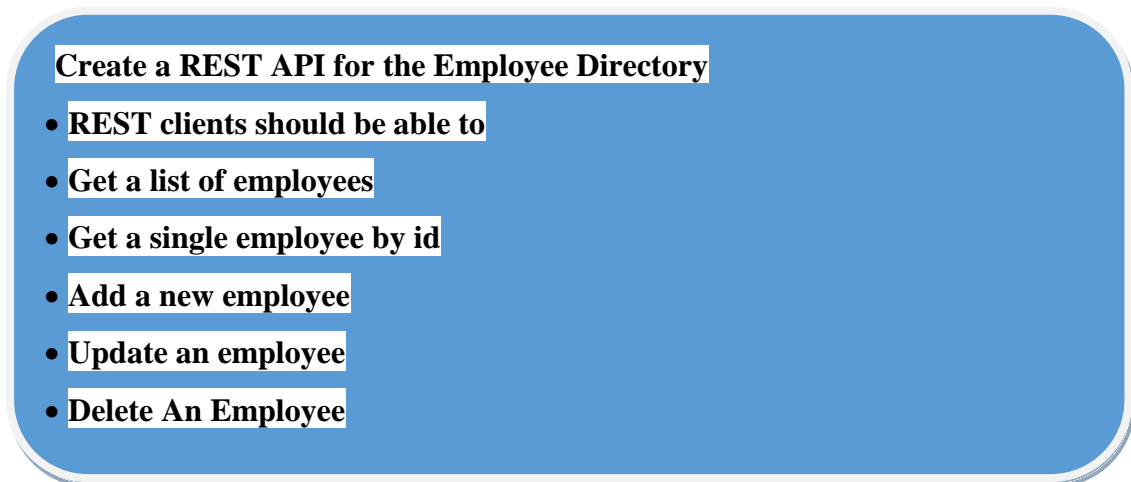


Figure 4: API Requirements

REST API -

HTTP Method		CRUD Action
POST	/api/employees	Create a new employee
GET	/api/employees	Read a list of employees
GET	/api/employees/{employeeId}	Read a single employee
PUT	/api/employees	Update an existing employee
Delete	/api/employees/{employeeId}	Delete an existing employee

Development Process Step-By-Step -

1. Set up Database Dev Environment
2. Create Spring Boot project using Spring Initializr
3. Get list of employees
4. Get single employee by ID
5. Add a new employee
6. Update an existing employee
7. Delete an existing employee

Application Architecture:

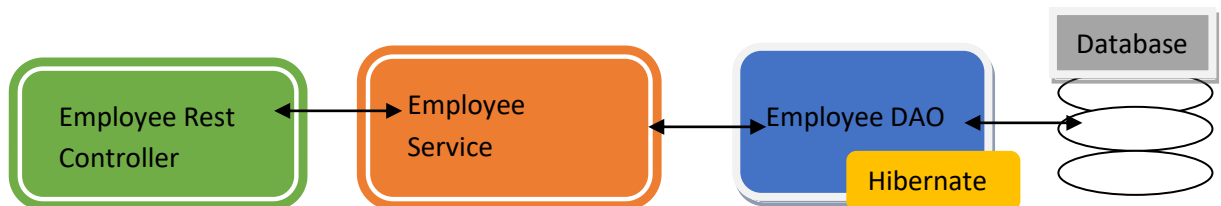


Figure 5: Application Architecture

Chapter 4

Results

Tested using postman.

Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so you can create better APIs—faster. Post request requires user to put data in a JSON format. Specific constrained query requests are generated using request parameters.

This particular REST API can be integrated with any frontend client such as Angular/React.

Jackson is a crucial part as it processes all the incoming & outgoing http requests.

Response for http requests:

GET request - "<http://localhost:8080/spring-crm-rest/api/employees>"

Response- The response is JSON formatted list of all employees present.

JSON	Raw Data	Headers
Save	Copy	Collapse All Expand All Filter JSON
▼ 0: id: 1 firstName: "David" lastName: "Adams" email: "david@luv2code.com"		
▼ 1: id: 5 firstName: "Maxwell" lastName: "Dixon" email: "max@luv2code.com"		
▼ 2: id: 2 firstName: "John" lastName: "Doe" email: "john@luv2code.com"		
▼ 3: id: 6 firstName: "Marta" lastName: "La Croft" email: "marta.Lacroft@gmail.com"		
▼ 4: id: 4 firstName: "Mary" lastName: "Public" email: "mary@luv2code.com"		
▼ 5: id: 8 firstName: "Tony" lastName: "Stark"		

Figure 6:Response GET request

GET Request – “<http://localhost:8080/spring-crm-rest/api/employees/2>”

Description – Retrieve the employee with id = 2.

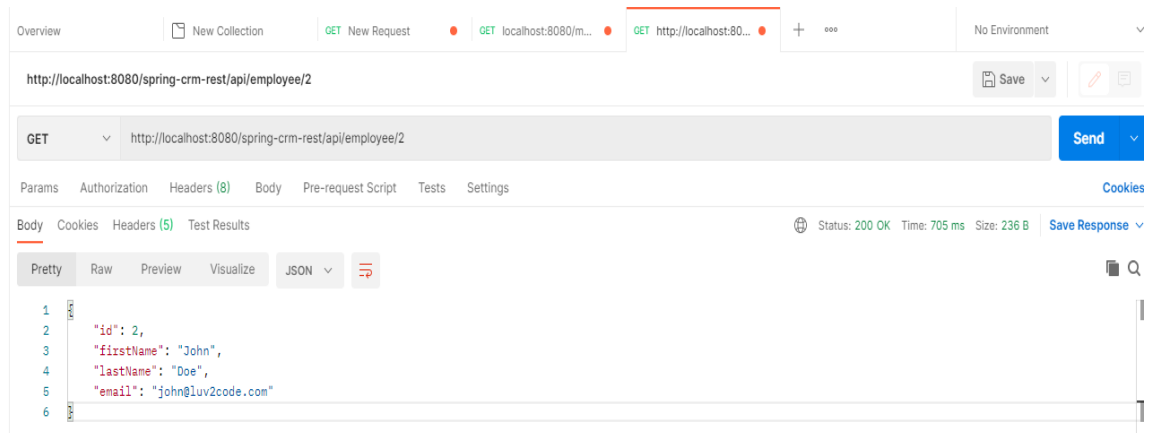


Figure 7:Response GET Request with parameter

POST Request – “<http://localhost:8080/spring-crm-rest/api/employees>”

Description – Create a new Employee.

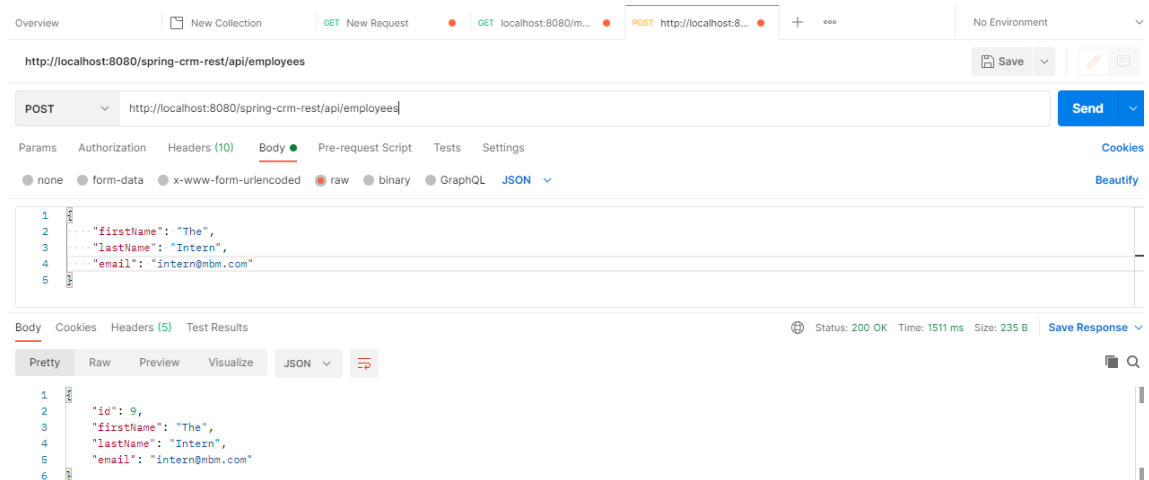


Figure 8:POST Request add employee

On successful request, the api will return the new employee data with the id of the same generated in database.

PUT Request – “<http://localhost:8080/spring-crm-rest/api/employees>”

Description – Update an employee with a particular id.

Response – returns the updated data if the entry exists with the particular id in the database.

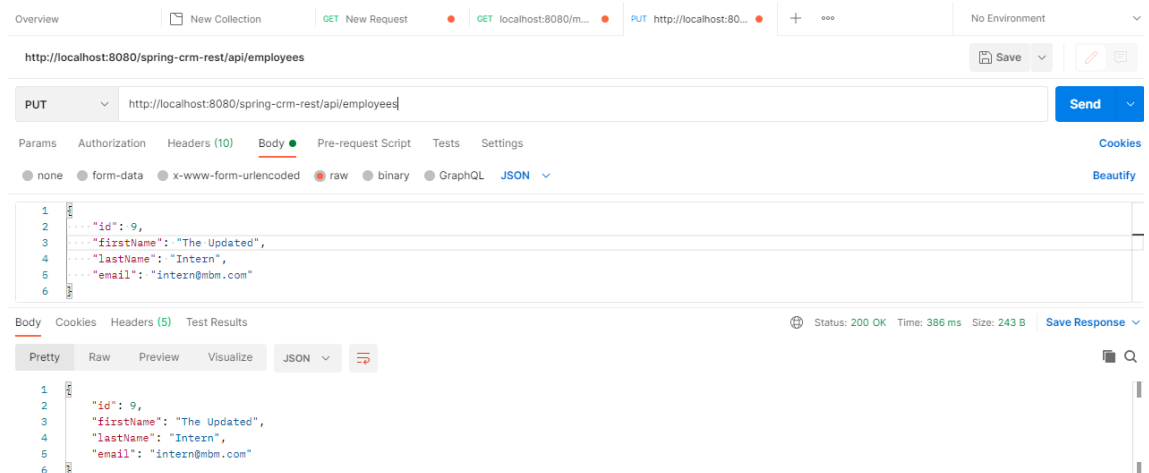


Figure 9: PUT Request update employee

DELETE Request – “localhost:8080/spring-crm-rest/employees/9”

Description - Returns the acknowledgement if the employee is deleted.

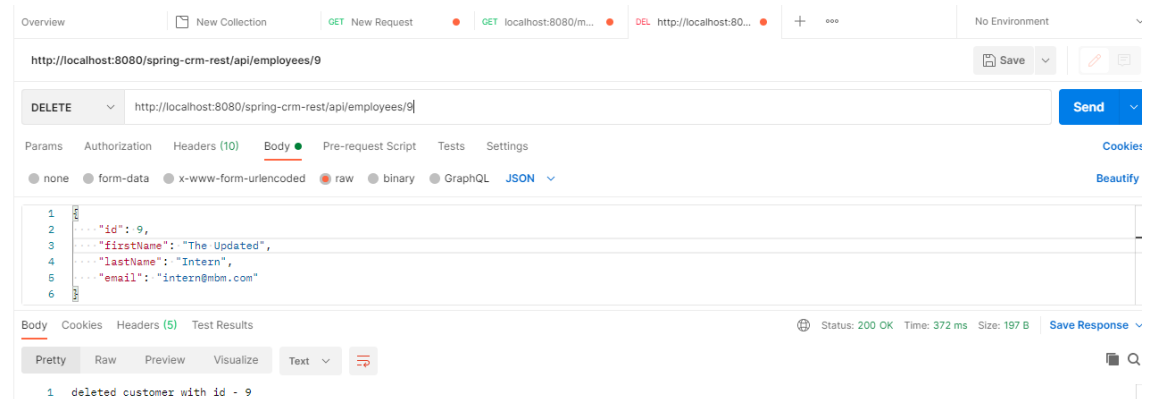


Figure 10: DELETE Request with parameter

Chapter 5

Conclusion & Future Work

API works well until under good concurrent transaction are applied. The reason being Hibernate can be slow for concurrent requests. For that reason, Spring Data JPA should be used. It lets developer decide with which ORM framework will the application communicate with database.

Spring security can also be applied to secure and implement authorization on endpoints of the API.

References

- [1] <https://spring.io/projects/spring-framework>
- [2] <https://hibernate.org>
- [3] POSTMAN - <https://www.postman.com/>