The Godfather of Talent | Futurense

Futurense

Democratizing Tech Talent to deliver impact at scale

The Godfather of Talent | Futurense

Project:

Electricity Bill Generator

Presented by:

Sri Sai Naga Prasad

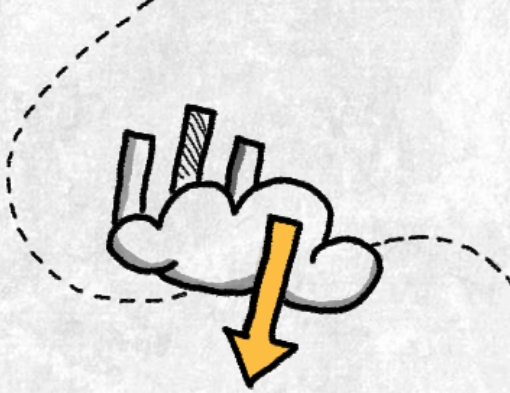
FT571

The Godfather of Talent | Futurense

Index:

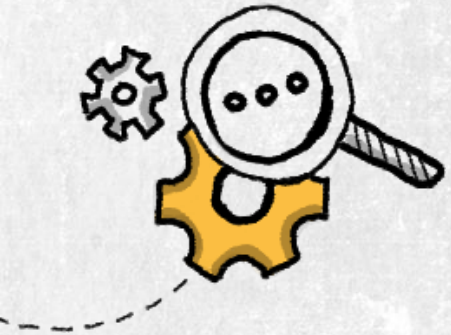
- Overview
- Tools and Technologies used
- Statistics
- Code Overview
- Code Snippets
- Future Scope

The Godfather of Talent | Futurense



Overview:

- This is a basic project that deals with maintaining the customer records of their electricity.
- The application takes the basic info about the customer along with the units consumed by the customer and calculates the bill itself.



Tools and Technologies Used:

Java:

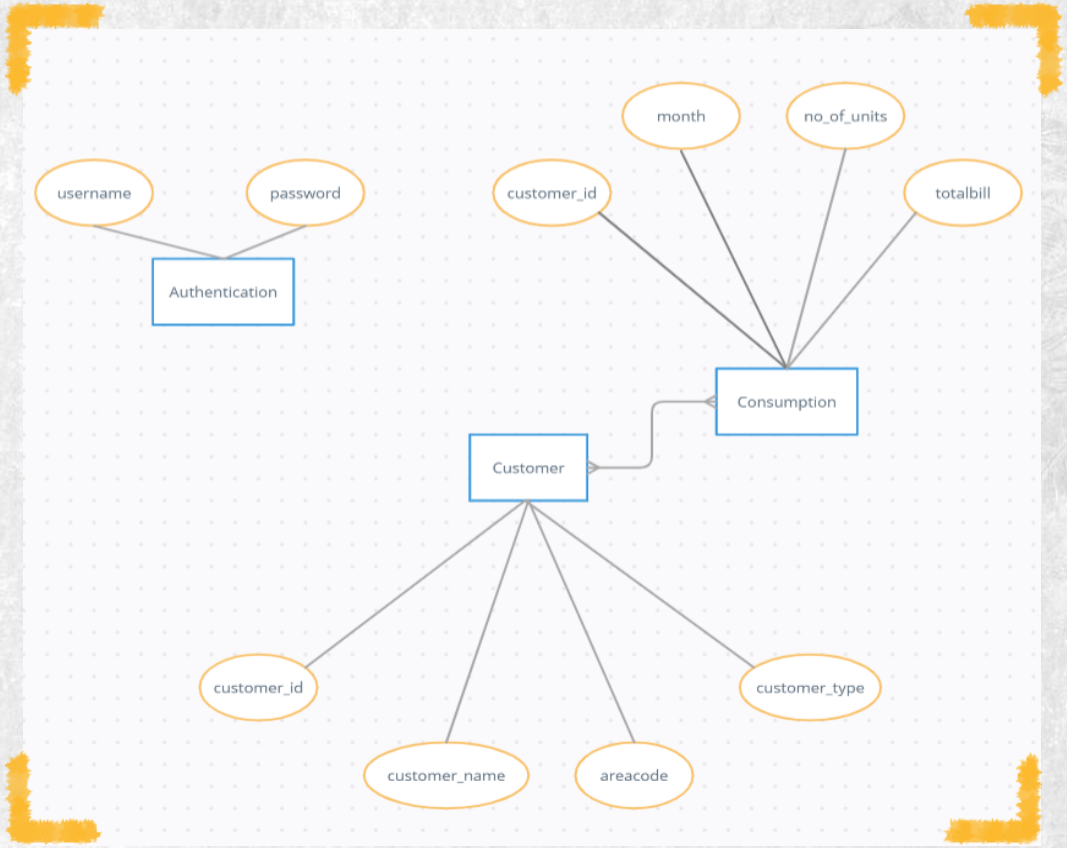
- Spring Boot
- Rest API
- JPA Repository

MySQL

Postman

Eclipse

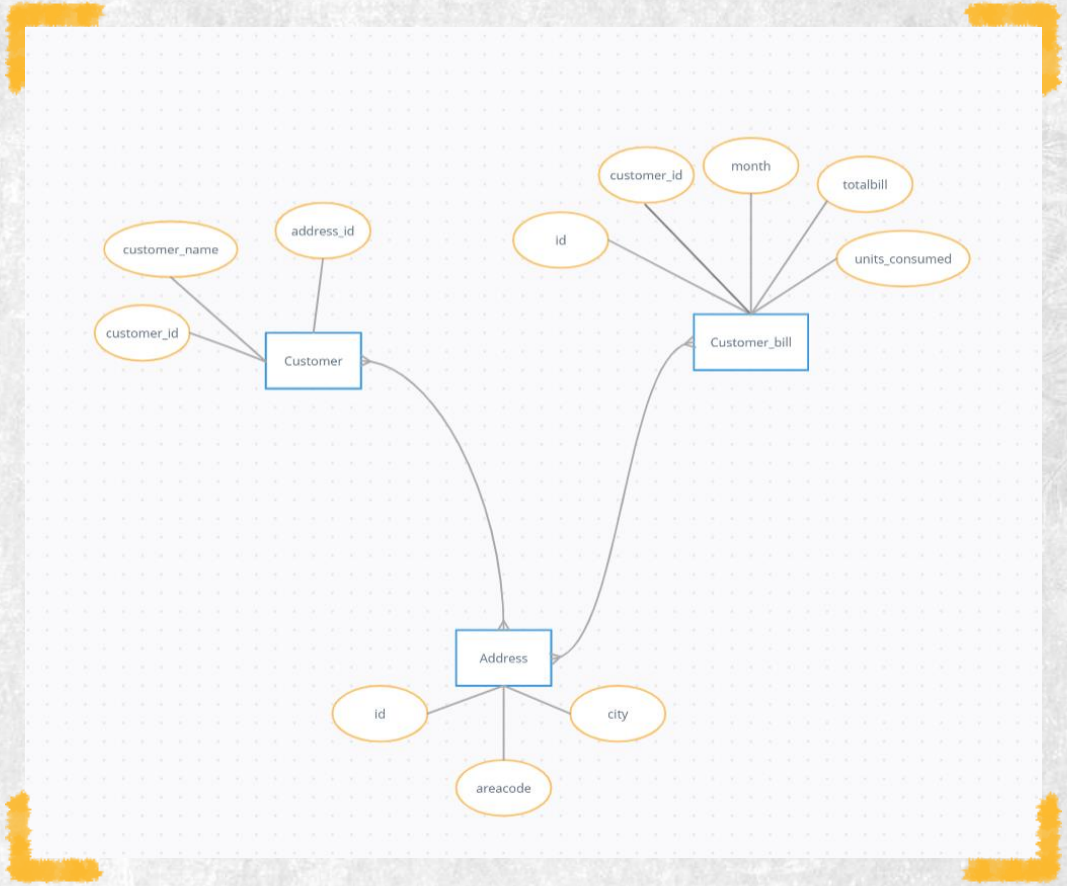
Basic Java Project:



- Tables: 3
- Indexes: 4
- Procedures: 3
- Functions: 3

The Godfather of Talent | Futurense

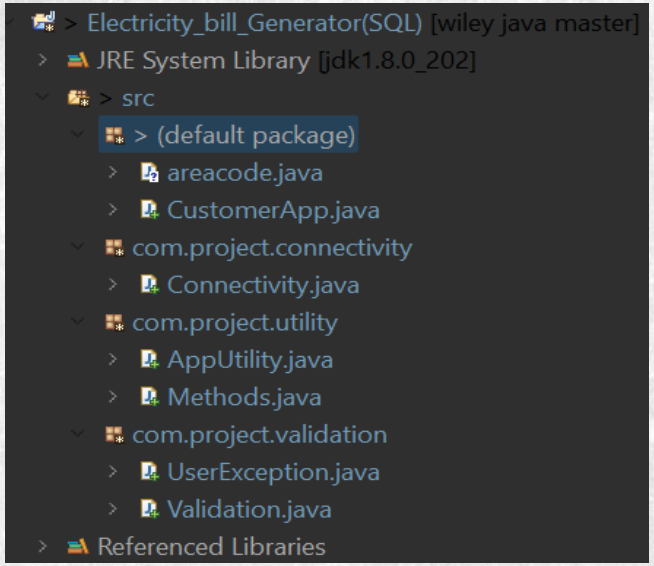
Spring boot Java Application:



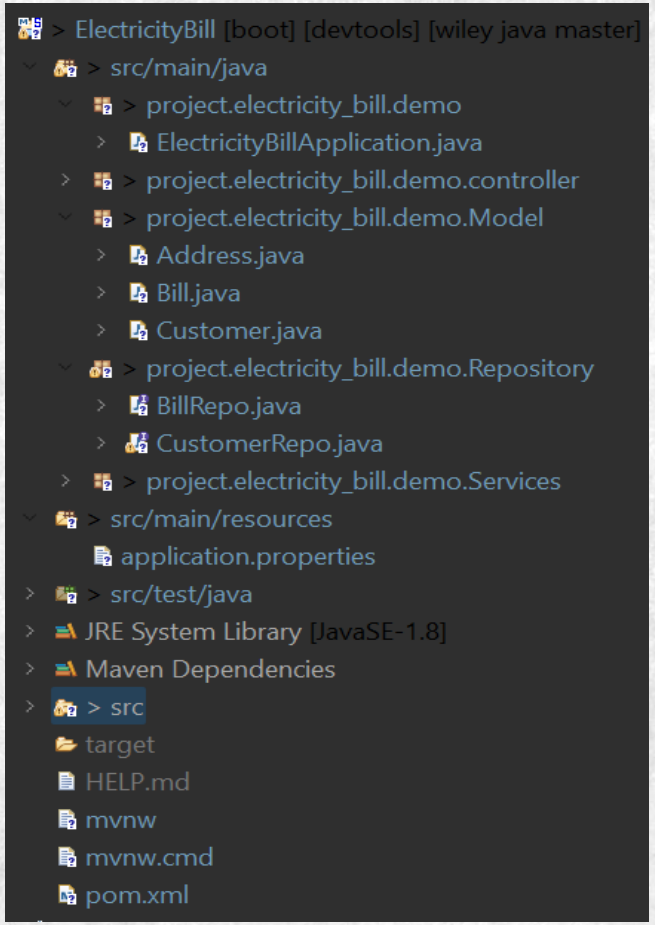
- Tables: 3
- Fields: 11
- Relationships: 2
- Indexes: 4

The Godfather of Talent | Futurense

Programs:



Java SQL



Java Spring boot SQL

Connectivity:

To connect our Java program with the SQL Database.

```java
package com.project.connectivity;

import java.sql.Connection;

public class Connectivity {
    public static Connection createconnection() {
        try {

            String url = "jdbc:mysql://localhost:3306/electricity_bill_generator";
            String root = "root";
            String pass = "Sai@1234";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection c = DriverManager.getConnection(url, root, pass);
            return c;
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Procedure:

To get customers details who have their bills in one particular month.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getCustomersByBill`(
    IN month_no VARCHAR(50)
)
BEGIN
    SELECT c.*,con.total_bill FROM consumption con
    INNER JOIN customer c ON con.customer_id = c.customer_id
    WHERE month = month_no ORDER BY total_bill DESC;
END
```

Function:

To calculate bill when values

are inserted into the table.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `totalbill`(
units int
) RETURNS decimal(10,2)
    DETERMINISTIC
BEGIN
    DECLARE bill decimal(10,2);
    if(units<101) then
        set bill=units*0.5;
    elseif(units<151) then
        set bill=(100*0.5)+(units-100)*0.75;
    else
        set bill=100*0.5+50*0.75+(units-150);
    end if;
RETURN bill;
END
```

Calling Procedure & Function:

Procedure

```
public static void displayCustomer(Connection conn) throws SQLException {
    CallableStatement cs = conn.prepareCall("{call getCustomer()}");
    ResultSet rs = cs.executeQuery();
    System.out.println("-----------------------------------------------------------------");
    System.out.printf("| %-5s|  %-20s|  %-20s|  %-10s|\n", "Id", "Name", "Customer Type", "Area Code");
    System.out.println("-----------------------------------------------------------------");
    while (rs.next()) {
        System.out.printf("| %-5s|  %-20s|  %-20s|  %-10s|", rs.getInt(1), rs.getString(2), rs.getString(3),
                rs.getString(4));
        System.out.println();
    }
    System.out.println("-----------------------------------------------------------------");
    System.out.println("\n");
}
```

Function

```
public static void displayBillOfParticularCustomer(String name, String Month, Connection conn) throws SQLException {
    CallableStatement cs = conn.prepareCall("select getBillForParticularCustomer(?,?) ");
    cs.setString(1, name);
    cs.setString(2, Month);
    ResultSet rs = cs.executeQuery();
    System.out.println("---------------");
    System.out.printf("|  %-11s |\n", "Total Bill");
    System.out.println("---------------");
    while (rs.next()) {
        System.out.printf("|  %-10s  |\n", rs.getDouble(1));

    }
    System.out.println("---------------");
    System.out.println("\n");
}
```

Query in repository:

With the use of JPA repository, we can just write the methods using naming conventions of JPA repository and the method will be written directly by the compiler itself using JPA repository.

```java
import java.util.List;
import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import project.electricity_bill.demo.Model.Bill;

@Repository
public interface BillRepo extends JpaRepository<Bill, Long> {
    Optional<Bill> findBillByCustomerIdAndMonth(long customerId,String month);
    List<Bill> findBillByMonth(String month);
    @Query(value="select sum(total_bill) from customer_bill where month=:month",nativeQuery=true)
    public double findtotal_bill(@Param("month") String month);
    public void deleteBillByCustomerIdAndMonth(long id,String month);
}
```

The Godfather of Talent | Futurense

Controller Class:

A Rest API which contains all the services provided by the application to the user.

```java
@RestController
public class BillController {
    @Autowired
    private BillServiceImplementation bsi;

    @GetMapping("/bill")
    public List<Bill> getAllName(){
        return bsi.getAllBill();
    }
    @GetMapping("/bill/{id}/{month}")
    public Bill getcustomerbillbymonth(@PathVariable Long id,@PathVariable String month) {
        return bsi.getCustomerBillByMonth(id,month);
    }
    @GetMapping("/bill/month/{month}")
    public List<Bill> getbillbymonth(@PathVariable String month){
        return bsi.getBillByMonth(month);
    }
    @PostMapping("/bill")
    public String saveBill(@RequestBody Bill p) {
        return bsi.savebill(p);
    }

    @GetMapping("/bill/{id}")
    public double gettotalbillbymonth(@PathVariable String id) {
        return bsi.getTotalBillByMonth(id);
    }

    @DeleteMapping("/bill/{id}/{month}")
    public String deleteName(@PathVariable Long id,@PathVariable String month) {
        return bsi.deleteBill(id, month);
    }
```

The Godfather of Talent | Futurense

Output:

```
Are you the admin
1.Yes
2.No
1
Enter user name
root
Enter user password
root
Enter any of the following operation
1:Display list of all customers
2:Display bill of all customers in a purticular month
3:Display bill of a customer for a purticular month
4:Total bill of all customers in purticular month
5:Displayed the customer list with their bills for a month, in descending order of the bill
6:Add or modify customer details
Any any other number to exit excluding 0
```

```
1
-------------------------------------------------------------------
| Id   |  Name                |  Customer Type       |  Area Code |
-------------------------------------------------------------------
| 1    |  SAGAR               |  REGULAR             |  123456    |
| 2    |  sai                 |  Commercial          |  236591    |
| 3    |  JAID                |  INDUSTRIAL          |  123457    |
| 4    |  Tejaswini           |  Regular             |  530076    |
| 5    |  SAI                 |  COMMERCIAL          |  123456    |
| 6    |  DEEPAK              |  INDUSTRIAL          |  123457    |
| 13   |  SaiNagaPrasad       |  Industrial          |  234123    |
| 23   |  sai                 |  Commercial          |  324213    |
| 100  |  Nagaprasad          |  Regular             |  251725    |
| 121  |  sai                 |  Regular             |  415161    |
| 199  |  Shivam              |  Regular             |  251735    |
| 200  |  sai                 |  Commercial          |  514261    |
| 251  |  BTSShivam           |  Commercial          |  163571    |
-------------------------------------------------------------------
```

```
Enter any of the following operation
1:Display list of all customers
2:Display bill of all customers in a purticular month
3:Display bill of a customer for a purticular month
4:Total bill of all customers in purticular month
5:Displayed the customer list with their bills for a month, in descending order of the bill
6:Add or modify customer details
Any any other number to exit excluding 0
2
Enter the month to be searched
1:January
2:February
3:March
4:April
5:May
6:June
7:July
8:August
9:September
10:October
11:November
12:December

5
5
-------------------------------------------------------------------------------------
| Id    |  Name                 |  Customer Type        |  Area Code       |    Bill      |
-------------------------------------------------------------------------------------
| 1     |  SAGAR                |  REGULAR              |  123456          |    320.34    |
| 199   |  Shivam               |  Regular              |  251735          |    82.25     |
| 200   |  sai                  |  Commercial           |  514261          |    87.5      |
-------------------------------------------------------------------------------------
Enter any of the following operation
```

CustomerApp.java | Console

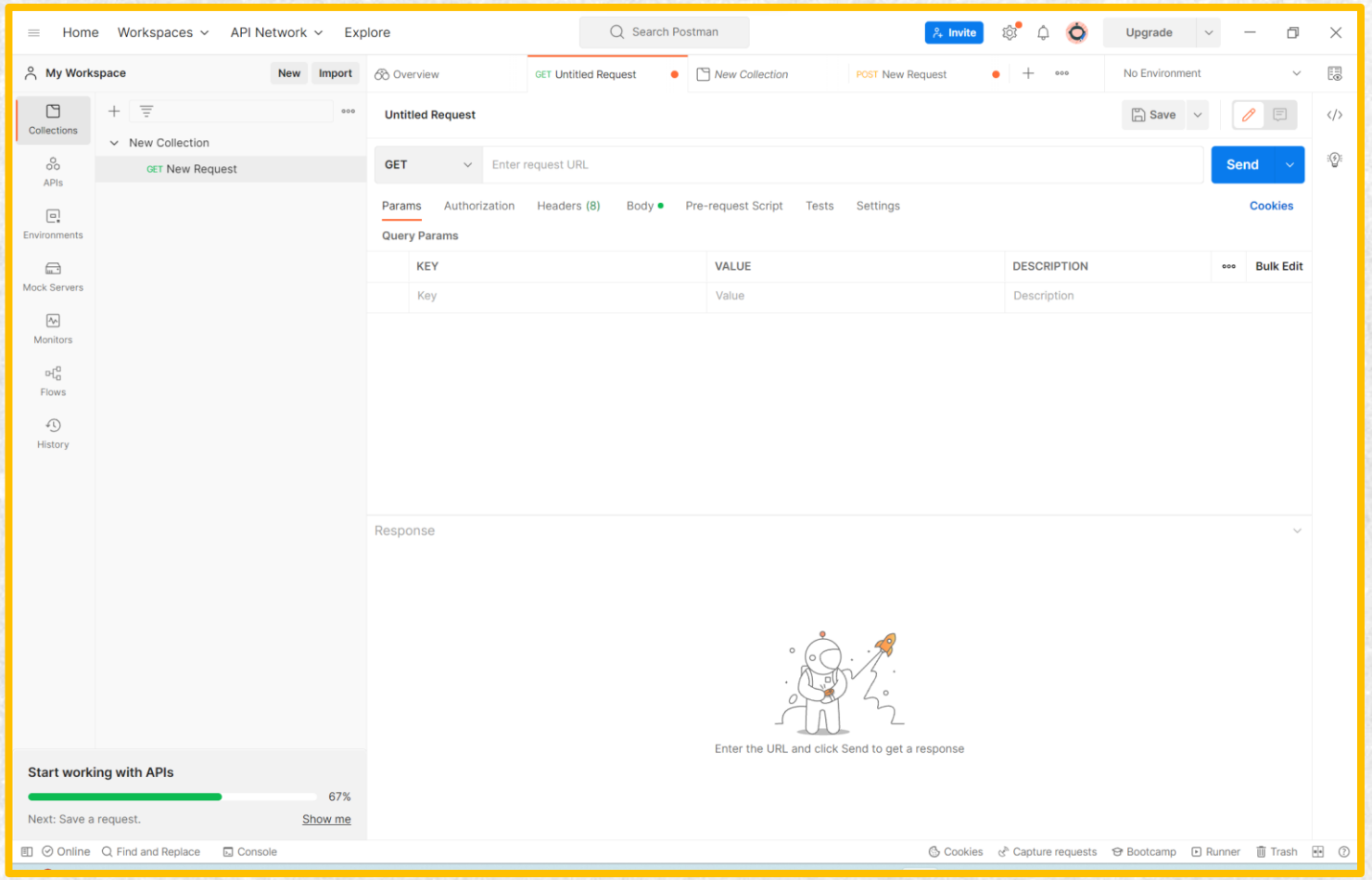
```
CustomerApp (1) [Java Application] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (10-Dec-2022, 4:02:09 PM) [pid: 9384]
Are you the admin
1.Yes
2.No
2
Want to add your details or modify your details press any key else press n
y
1:Add Customer
2:Add Customer electricity comsumption details
3:Modify customer data
enter other to exit
1
Id : 13
Name : NagaPrasad
Area Code : 518001
Enter customer type
1:Industrial
2:Commercial
3:Regular
2
Added Successfully
1:Add Customer
2:Add Customer electricity comsumption details
3:Modify customer data
enter other to exit
2
Id : 13
13
Enter month of the bill you are adding in the numerical format
1:January
2:February
3:March
4:April
5:May
6:June
7:July
8:August
9:September
10:October
11:November
12:December

4
4
Number of units consumed : 189
Added Successfully
1:Add Customer
2:Add Customer electricity comsumption details
3:Modify customer data
enter other to exit
```

```
1:Add Customer
2:Add Customer electricity comsumption details
3:Modify customer data
enter other to exit
3
Id : 13
Name : SaiNagaPrasad
Area Code : 234123
Enter customer type
1:Industrial
2:Commercial
3:Regular
1
Added Successfully
```

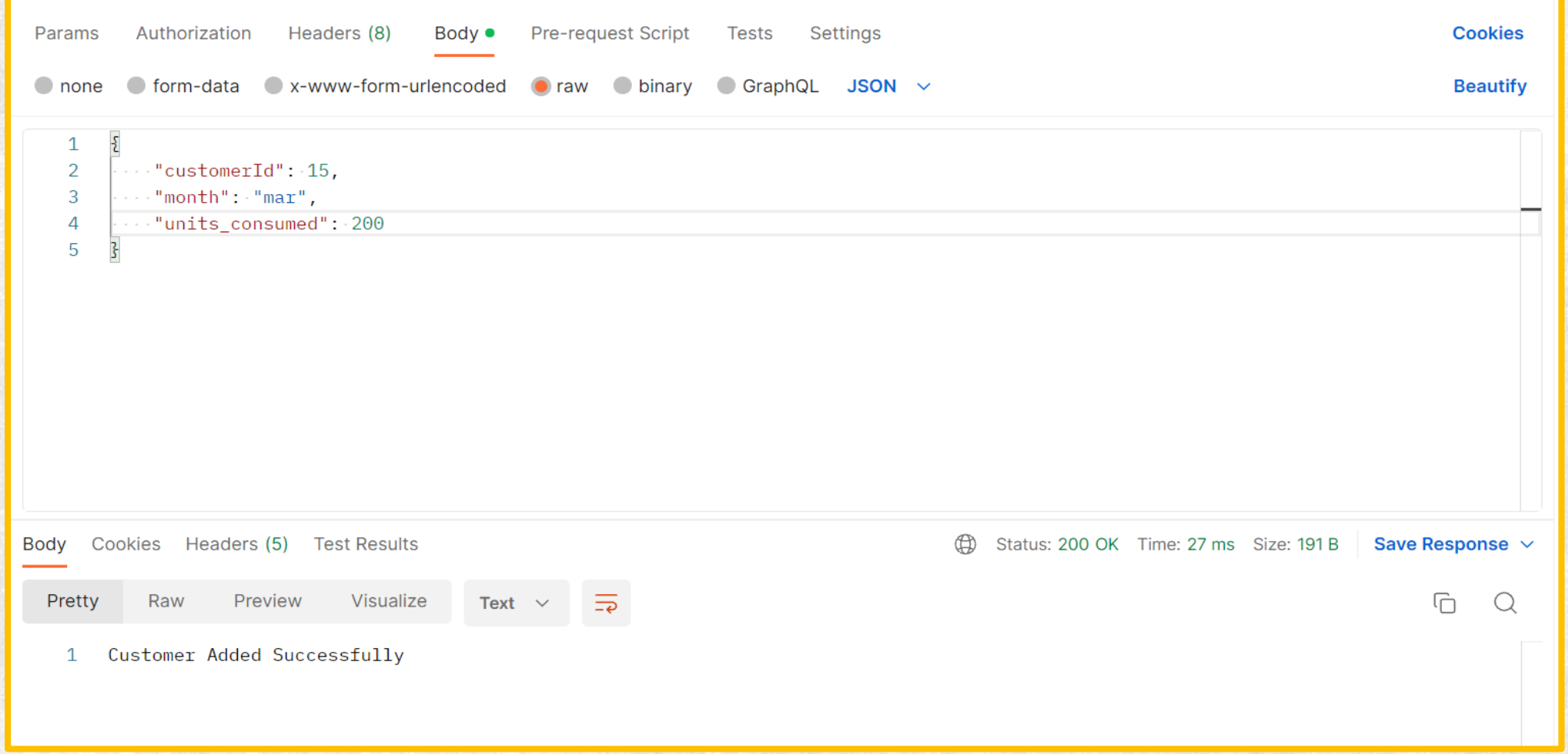
The Godfather of Talent | Futurense

Spring boot Application:

```
Hibernate:

    create table address (
       id integer not null auto_increment,
        area_code varchar(255),
        city varchar(255),
        primary key (id)
    ) engine=InnoDB
Hibernate:

    create table customer (
       customer_id bigint not null,
        customer_name varchar(255),
        address_id integer,
        primary key (customer_id)
    ) engine=InnoDB
Hibernate:

    create table customer_bill (
       id bigint not null auto_increment,
        customer_id bigint,
        month varchar(255),
        total_bill double precision,
        units_consumed integer,
        primary key (id)
    ) engine=InnoDB
```



The Godfather of Talent | Futurense



The Godfather of Talent | Futurense

GET http://localhost:8080/bill Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

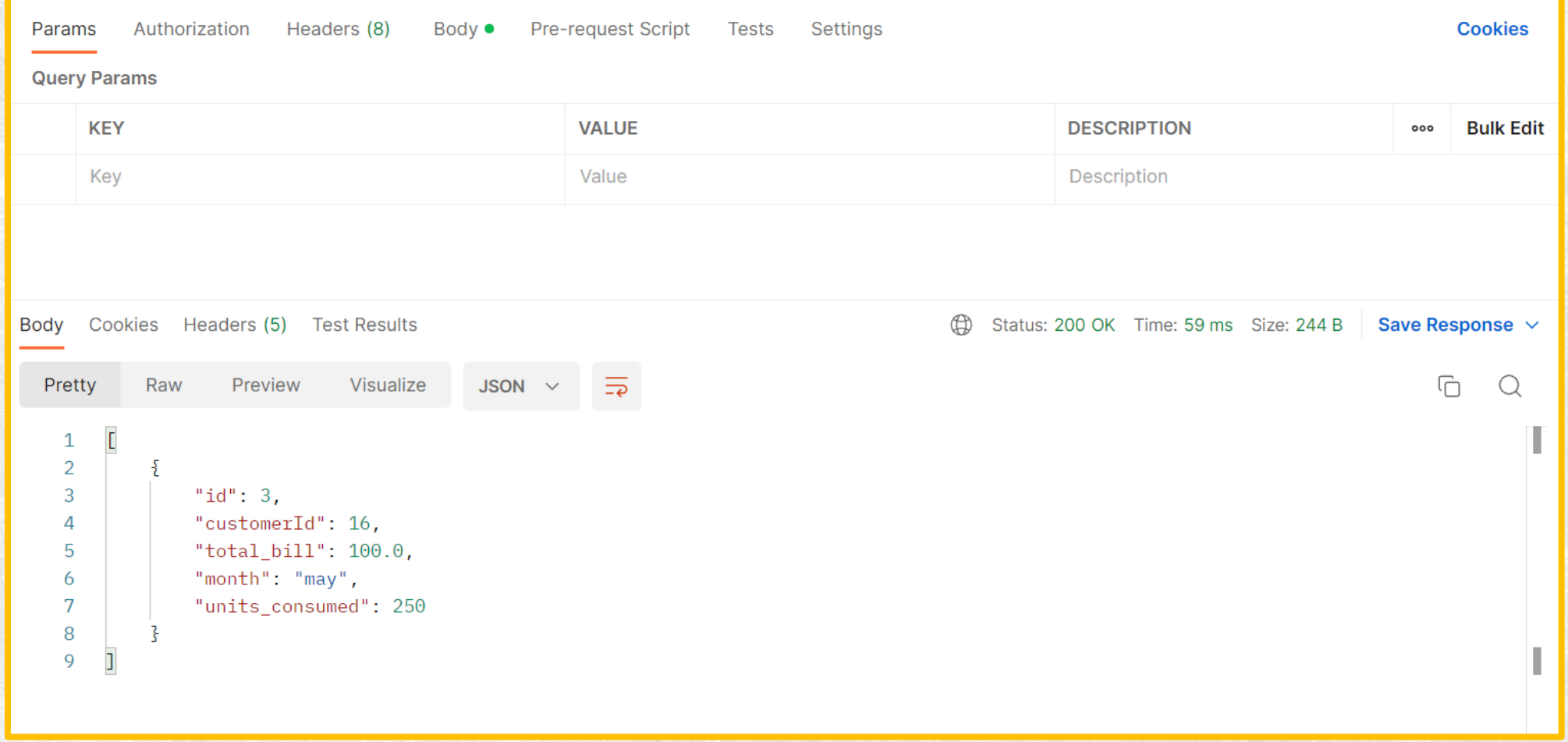
KEY	VALUE	DESCRIPTION	ooo	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 9 ms Size: 320 B Save Response

Pretty Raw Preview Visualize JSON


```
[
    {
        "id": 1,
        "customerId": 13,
        "total_bill": 27.5,
        "month": "feb",
        "units_consumed": 55
    },
    {
        "id": 2,
        "customerId": 15,
        "total_bill": 50.0,
        "month": "mar",
        "units_consumed": 200
    }
]
```

The Godfather of Talent | Futurense



Future Scope:

- The application can be further developed by adding security.
- It can be extended using front end technology and made as a full fledged application.
- The developed version can be used as a real time application used for the elections which are being done.

The Godfather of Talent | Futurense

Thank You