Spring Boot
Lab Book

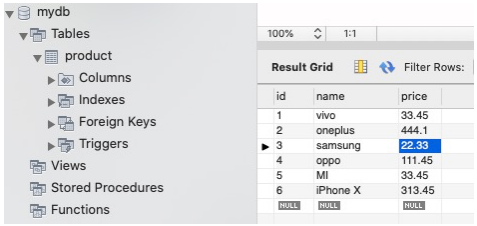
Every Project implemented in Spring Boot must have the below flow:

POSTMAN->Spring REST Controller->Service Layer->Spring Data JPA Repository Layer->Database

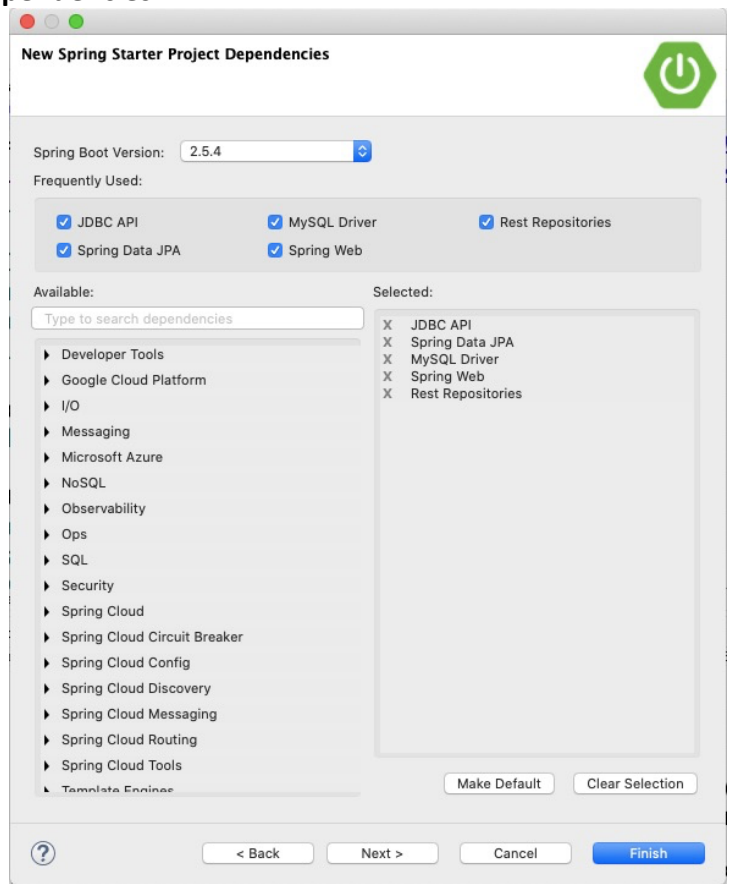
Module Name: Product

- ID in int
- Name in String
- Price in float

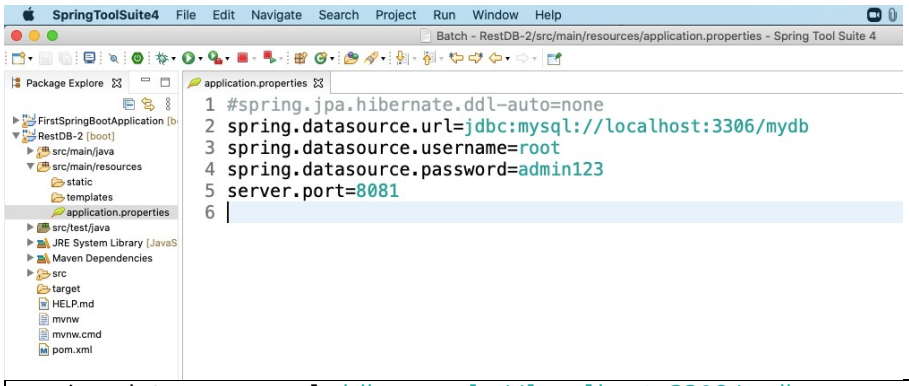
Setup 1: Database Implementation at MySQL Server Workbench



Setup 2: Create Spring Starter Project in **Spring Tool Suite(STS)**
Add the below dependencies :



- Add the following script in **application.properties**




```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=admin123
server.port=8081
```

Setup 3: In Spring Tool Suite(STS)

Step 1: Create a class for **Product** Entity

- **Entity - *Model***

Step 2: Create a class for **ProductController**

- **RESTController - Controller**

Step 3: Create a class for **ProductService**

- **Autowired**

Step 4: Create a interface for **ProductRepository**

- **Data JPA Repository**

Step 1: Create a class for **Product** Entity

```
package net.codejava;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class Product
{
    private Integer id;
    private String name;
    private float price;

    public Product()
    {
    }

    public Product(Integer id, String name, float price)
    {
        this.id = id;
        this.name = name;
        this.price = price;
    }

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    public Integer getId()
    {
        return id;
    }

       public void setId(Integer id)
       {
              this.id = id;
       }

       public String getName()
       {
              return name;
       }

       public void setName(String name)
       {
              this.name = name;
       }

       public float getPrice()
       {
              return price;
       }

       public void setPrice(float price)
       {
              this.price = price;
       }

       @Override
       public String toString()
       {
       return "Product [id=" + id + ", name=" + name + ", price=" + price + "]";
       }


}
```

Step 2: Create a class for **ProductController**

```
package net.codejava;
import java.util.*;
import org.springframework.beans.factory.annotation.*;
import org.springframework.http.*;
import org.springframework.web.bind.annotation.*;

@RestController
public class ProductController
{
    @Autowired
    private ProductService service;

    // RESTful API methods for Retrieval operations
    @GetMapping("/products")
    public List<Product> list()
    {
        return service.listAll();
    }
    @GetMapping("/products/{id}")
    public ResponseEntity<Product> get(@PathVariable Integer id)
    {
        try
        {
            Product product = service.get(id);
            return new ResponseEntity<Product>(product, HttpStatus.OK);
        }
        catch (NoSuchElementException e)
        {
            return new ResponseEntity<Product>(HttpStatus.NOT_FOUND);
        }
    }
    // RESTful API method for Create operation
    @PostMapping("/products")
    public void add(@RequestBody Product product)
    {
        service.save(product);
    }

    // RESTful API method for Update operation
    @PutMapping("/products/{id}")
    public ResponseEntity<?> update(@RequestBody Product product, @PathVariable Integer id)
    {
        try
        {
            Product existProduct = service.get(id);
            service.save(product);
            return new ResponseEntity<>(HttpStatus.OK);
        }
        catch (NoSuchElementException e)
        {
            return new ResponseEntity<>(HttpStatus.NOT_FOUND);
        }
    }
    // RESTful API method for Delete operation
    @DeleteMapping("/products/{id}")
    public void delete(@PathVariable Integer id)
    {
        service.delete(id);
    }
}
```

Step 3: Create a class for **ProductService**

```
package net.codejava;
import java.util.List;
import javax.transaction.Transactional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
@Transactional
public class ProductService
{

    @Autowired
    private ProductRepository repo;

    public List<Product> listAll()
    {
        return repo.findAll();
    }

    public void save(Product product)
    {
        repo.save(product);
    }

    public Product get(Integer id)
    {
        return repo.findById(id).get();
    }

    public void delete(Integer id)
    {
        repo.deleteById(id);
    }
}
```

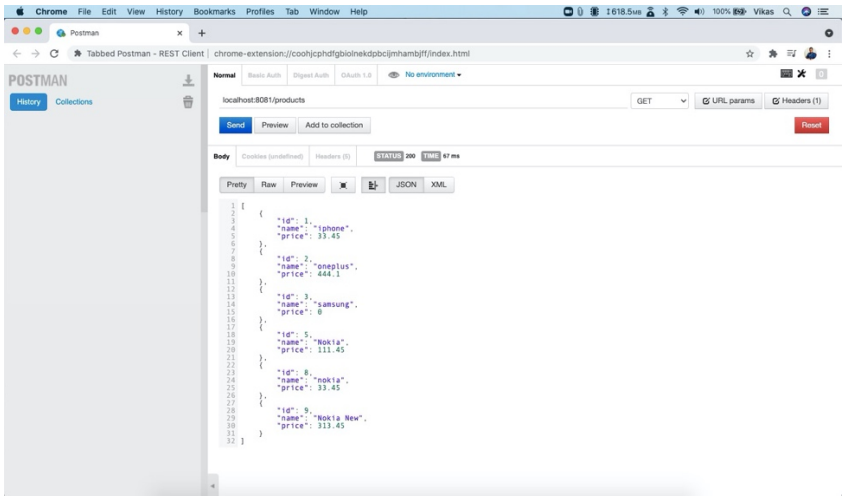
Step 4: Create a interface for **ProductRepository**

```
package net.codejava;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ProductRepository extends JpaRepository<Product, Integer>
{

}
```

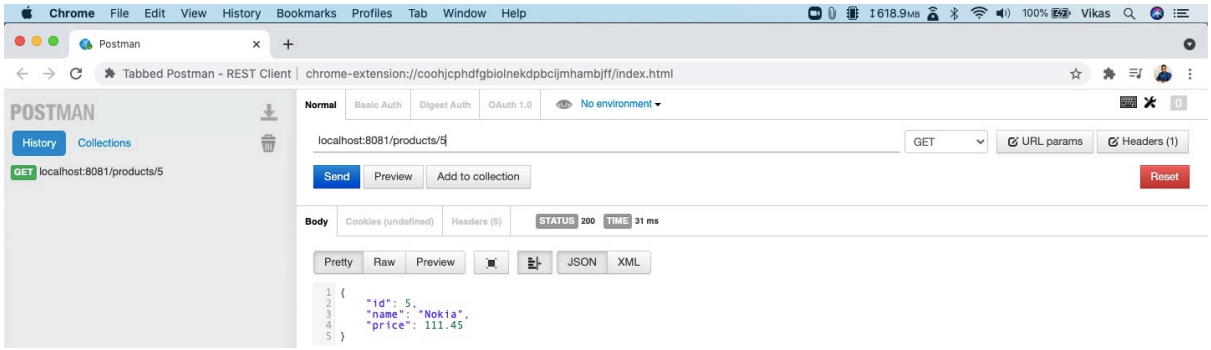
Step 5: Run Application.Java in src/main/java

Step 6: Install & Open POSTMAN Rest in Browser. Perform CRUD operations

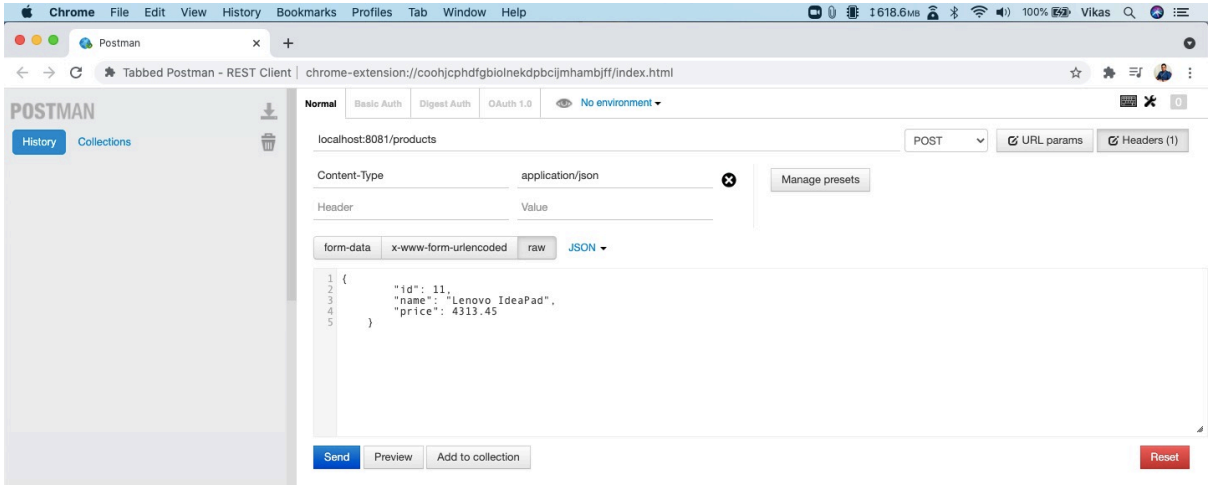
Output 1: GET method – Display all Records



Output 2: GET method – Display Specific Records



Output 3: POST method – Insert row into the database through POSTMAN



Output 4: DELETE method – delete a specific row