Assignment 3 Sesi 17 – Rest API

Nama: Mikail Gibran Fernanda Lubis

Kode Peserta: JVSB001ONL020

Membuat operasi POST, GET & DELETE dengan REST API menggunakan Post man

# Struktur Folder

* com.belajar.spring -> BelajarApiLatihanApplication.java
* com.belajar.spring.controller -> ProdukController.java, UiController.java
* com.belajar.spring.model -> Produk.java
* com.belajar.spring.repository -> ProdukRepository.java
* com.belajar.spring.service -> ProdukService.java

# Depedencies

|  |
| --- |
| <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.7.0</version>  <relativePath/> <!-- lookup parent from repository -->  </parent>  <groupId>com.belajar.spring</groupId>  <artifactId>BelajarAPILatihan</artifactId>  <version>0.0.1-SNAPSHOT</version>  <name>BelajarAPILatihan</name>  <description>Demo project for Spring Boot</description>  <properties>  <java.version>17</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-jpa</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-thymeleaf</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-test</artifactId>  <scope>test</scope>  </dependency>  <dependency>  <groupId>io.github.jpenren</groupId>  <artifactId>thymeleaf-spring-data-dialect</artifactId>  <version>3.4.0</version>  </dependency>  </dependencies>  <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build> |

# Project Application Properties

|  |
| --- |
| server.port=8282  spring.datasource.url=jdbc:mysql://localhost:3306/db\_produk?useSSL=false&useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC  spring.datasource.username=root  spring.datasource.password=root  # The SQL dialect makes Hibernate generate better SQL for the chosen database  spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQL5InnoDBDialect  # To beautify or pretty print the SQL  spring.jpa.properties.hibernate.format\_sql=true  # Hibernate ddl auto (create, create-drop, validate, update)  spring.jpa.hibernate.ddl-auto=update  # The simplest way to dump the queries  spring.jpa.show-sql=true  # he first line logs the SQL queries, and the second statement logs  logging.level.org.hibernate.SQL=DEBUG  logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE  # Logging JdbcTemplate Queries  logging.level.org.springframework.jdbc.core.JdbcTemplate=DEBUG  logging.level.org.springframework.jdbc.core.StatementCreatorUtils=TRACE |

# Codes

## BelajarApiLatihanApplication.java

|  |
| --- |
| package com.belajar.spring;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class BelajarApiLatihanApplication {  public static void main(String[] args) {  SpringApplication.run(BelajarApiLatihanApplication.class, args);  }  } |

## ProdukController.java

|  |
| --- |
| package com.belajar.spring.controller;  import java.util.List;  import java.util.NoSuchElementException;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.http.HttpStatus;  import org.springframework.http.ResponseEntity;  import org.springframework.web.bind.annotation.CrossOrigin;  import org.springframework.web.bind.annotation.DeleteMapping;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.PutMapping;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RestController;  import com.belajar.spring.model.Produk;  import com.belajar.spring.service.ProdukService;  @RestController  @RequestMapping("/api")  public class ProdukController {  @Autowired  ProdukService produkService;    @CrossOrigin  @GetMapping("/produks")  public List<Produk> list() {  return produkService.listAllProduk();  }    @CrossOrigin  @GetMapping("/produks/{id}")  public ResponseEntity<Produk> produks(@PathVariable Integer id){  try {  Produk produk = produkService.getProduk(id);  return new ResponseEntity<Produk>(produk, HttpStatus.OK);  } catch (NoSuchElementException e) {  return new ResponseEntity<Produk>(HttpStatus.NOT\_FOUND);  }  }    @CrossOrigin  @PostMapping("/produks")  public ResponseEntity<?> produks(@RequestBody Produk produk) {  produkService.saveProduk(produk);  return new ResponseEntity<>("Berhasil Simpan Data Produk", HttpStatus.OK);  }    @CrossOrigin  @PutMapping("/produks/{id}")  public ResponseEntity<?> update(@RequestBody Produk produk, @PathVariable Integer id) {  try {  Produk existProduk = produkService.getProduk(id);  if (existProduk.getId() != id) {  return new ResponseEntity<>("ID Tidak Ada !!",HttpStatus.OK);  }  produk.setId(id);  produkService.saveProduk(produk);  return new ResponseEntity<>("Berhasil Edit Data Supirs Dengan Nama "+produk.getNama\_produk(), HttpStatus.OK);  } catch (NoSuchElementException e) {  return new ResponseEntity<>("TIDAK ADAAA",HttpStatus.NOT\_FOUND);  }  }    @CrossOrigin  @DeleteMapping("/produk/{id}")  public ResponseEntity<?> delete(@PathVariable Integer id) {  produkService.deleteProduk(id);  return new ResponseEntity<>("Berhasil Hapus Data Produk", HttpStatus.OK);  }    } |

## UiController.java

|  |
| --- |
| package com.belajar.spring.controller;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Controller;  import org.springframework.ui.ModelMap;  import org.springframework.web.bind.annotation.GetMapping;  import com.belajar.spring.repository.ProdukRepository;  import com.belajar.spring.service.ProdukService;  import org.springframework.data.domain.Pageable;  @Controller  public class UiController {    @Autowired  private ProdukService produkService;    @GetMapping("/index")  public ModelMap getAll(Pageable pageable) {  return new ModelMap().addAttribute("produks", produkService.listAllProduk());  }  } |

## Produk.java

|  |
| --- |
| package com.belajar.spring.model;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.GenerationType;  import javax.persistence.Id;  import javax.persistence.Table;  @Entity  @Table(name = "produk")  public class Produk {  private int id;  private String nama\_produk;  private double harga\_jual;  private double harga\_beli;    public Produk() {    }    public Produk(String nama\_produk, double harga\_jual, double harga\_beli) {  this.nama\_produk = nama\_produk;  this.harga\_beli = harga\_beli;  this.harga\_jual = harga\_jual;  }    @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  public int getId() {  return id;  }  public String getNama\_produk() {  return nama\_produk;  }  public void setNama\_produk(String nama\_produk) {  this.nama\_produk = nama\_produk;  }  public double getHarga\_jual() {  return harga\_jual;  }  public void setHarga\_jual(double harga\_jual) {  this.harga\_jual = harga\_jual;  }  public double getHarga\_beli() {  return harga\_beli;  }  public void setHarga\_beli(double harga\_beli) {  this.harga\_beli = harga\_beli;  }  public void setId(int id) {  this.id = id;  }      } |

## ProdukRepository.java

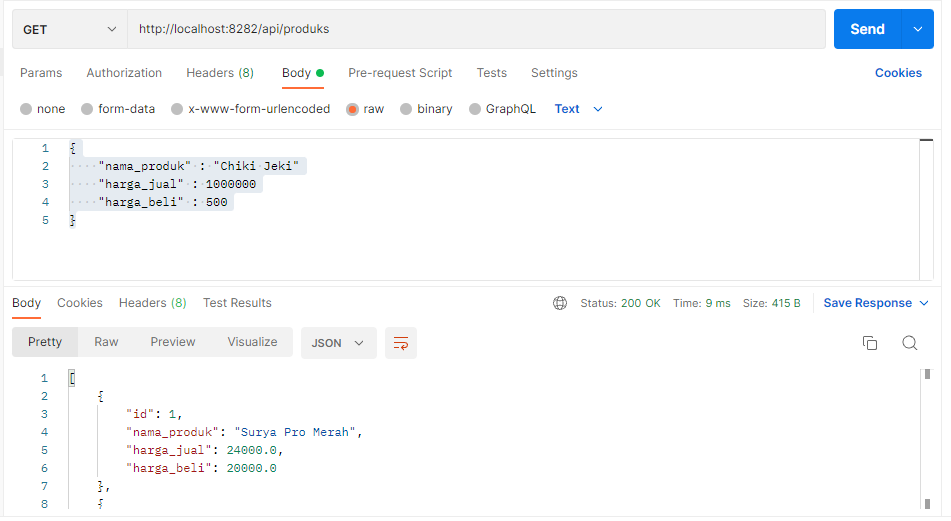
|  |
| --- |
| package com.belajar.spring.repository;  import org.springframework.data.jpa.repository.JpaRepository;  import com.belajar.spring.model.Produk;  public interface ProdukRepository extends JpaRepository<Produk, Integer>{  } |

## ProdukService.java

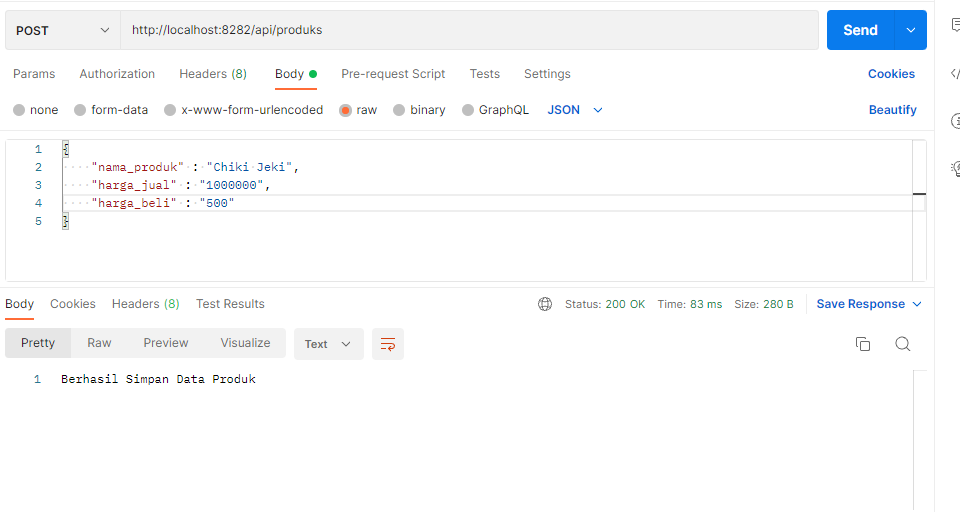
|  |
| --- |
| package com.belajar.spring.service;  import java.util.List;  import javax.transaction.Transactional;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import com.belajar.spring.model.Produk;  import com.belajar.spring.repository.ProdukRepository;  @Service  @Transactional  public class ProdukService {    @Autowired  private ProdukRepository produkRepository;    public List<Produk> listAllProduk() {  return produkRepository.findAll();  }    public void saveProduk(Produk produk) {  produkRepository.save(produk);  }    public Produk getProduk(Integer id) {  return produkRepository.findById(id).get();  }    public void deleteProduk(Integer id) {  produkRepository.deleteById(id);  }    } |

# Operasi Postman

## Operasi Get



## Operasi POST



## Operasi DELETE by ID

