

# Build RESTful with Spring Boot

RestApiTestApplication:

```
package com.test.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class RestApiTestApplication {

    public static void main(String[] args) {
        SpringApplication.run(RestApiTestApplication.class, args);
    }

}
```

ProductEntity:

```
package com.test.example;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;

@Entity
public class ProductEntity {

    @jakarta.persistence.Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id", updatable = false, nullable = false)
    private int id;
    @Column
    private String name;
    @Column
    private String description;

    public ProductEntity() {
        super();
    }
    public ProductEntity(int id, String name, String description) {
        super();
        this.id = id;
        this.name = name;
        this.description = description;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getDescription() {
```

```

        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}

```

ProductRepository:

```

package com.test.example;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ProductRepository extends JpaRepository<ProductEntity,
Integer>{

}

```

ProductService:

```

package com.test.example;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class ProductService {

    @Autowired
    ProductRepository productRepository;

    public List<ProductEntity> getAllProduct() {
        return productRepository.findAll();
    }

    public ProductEntity getProduct(int id) {
        return productRepository.findById(id).get();
    }

    public void addProduct(ProductEntity pe) {
        productRepository.save(pe);
    }

    public void updateProduct(int id, ProductEntity pe) {
        if(productRepository.findById(id).isPresent()) {
            ProductEntity
oldProductEntity=productRepository.findById(id).get();
            oldProductEntity.setName(pe.getName());
            oldProductEntity.setDescription(pe.getDescription());
            productRepository.save(oldProductEntity);
        }
    }

    public void deleteProduct(int id) {
        productRepository.deleteById(id);
    }
}

```

ProductController:

```
package com.test.example;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(path = "/webapi")
public class ProductController {

    @Autowired
    ProductService productService;

    @RequestMapping("/allproduct")
    public List<ProductEntity> getAllProduct() {
        return productService.getAllProduct();
    }

    @RequestMapping("/product/{id}")
    public ProductEntity getProduct(@PathVariable int id) {
        return productService.getProduct(id);
    }

    @RequestMapping(method=RequestMethod.POST, value="/product")
    public void addProduct(@RequestBody ProductEntity pe) {
        productService.addProduct(pe);
    }

    @RequestMapping(method=RequestMethod.PUT, value="/product/{id}")
    public void updateProduct(@PathVariable int id, @RequestBody
ProductEntity pe) {
        productService.updateProduct(id, pe);
    }

    @RequestMapping(method=RequestMethod.DELETE, value="/product/{id}")
    public void deleteProduct(@PathVariable int id) {
        productService.deleteProduct(id);
    }
}
```

```
eclipse4 - RestApiTest/src/main/java/com/test/example/RestApiTestApplication.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Markers Properties Servers Data Source Explorer Snippets Terminal Console X Run
RestApiTest - RestApiTestApplication [Spring Boot App] C:\Users\murug\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.x20230425-1502\jre\bin\javaw.exe (15-Jun-2023, 3:22:49 pm) [pid: 20344]

:: Spring Boot ::
(v3.1.0)

2023-06-15T15:22:50.454+05:30 INFO 20344 --- [main] c.test.example.RestApiTestApplication : Starting RestApiTestApplication using Java 17.0.7 with PID 20344 (C:\Users\murug\OneDrive\Desktop\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.x20230425-1502\jre\bin\javaw.exe)
2023-06-15T15:22:50.457+05:30 INFO 20344 --- [main] c.test.example.RestApiTestApplication : No active profile set, falling back to 1 default profile: "default"
2023-06-15T15:22:50.879+05:30 INFO 20344 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-06-15T15:22:50.913+05:30 INFO 20344 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 29 ms. Found 1 JPA repository interfaces
2023-06-15T15:22:51.242+05:30 INFO 20344 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-06-15T15:22:51.250+05:30 INFO 20344 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-06-15T15:22:51.250+05:30 INFO 20344 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-06-15T15:22:51.330+05:30 INFO 20344 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-06-15T15:22:51.330+05:30 INFO 20344 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 830 ms
2023-06-15T15:22:51.411+05:30 INFO 20344 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-06-15T15:22:51.537+05:30 INFO 20344 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:07fe7032-b85a-4774-ala8
2023-06-15T15:22:51.538+05:30 INFO 20344 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-06-15T15:22:51.572+05:30 INFO 20344 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHN000204: Processing PersistenceUnitInfo [name: default]
2023-06-15T15:22:51.605+05:30 INFO 20344 --- [main] org.hibernate.Version : HHN000412: Hibernate ORM core version 6.2.2.Final
2023-06-15T15:22:51.607+05:30 INFO 20344 --- [main] org.hibernate.cfg.Environment : HHN000406: Using bytecode reflection optimizer
2023-06-15T15:22:51.701+05:30 INFO 20344 --- [main] o.h.b.i.BytecodeProviderInitiator : HHN000021: Bytecode provider name : bytebuddy
2023-06-15T15:22:51.781+05:30 INFO 20344 --- [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer
2023-06-15T15:22:51.814+05:30 INFO 20344 --- [main] org.hibernate.com.dialect : HHN035001: Using dialect: org.hibernate.dialect.H2Dialect, version: 2.1.214
2023-06-15T15:22:51.939+05:30 INFO 20344 --- [main] o.h.b.i.BytecodeProviderInitiator : HHN000021: Bytecode provider name : bytebuddy
2023-06-15T15:22:52.331+05:30 INFO 20344 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHN000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction
2023-06-15T15:22:52.351+05:30 INFO 20344 --- [main] o.h.t.s.i.e.GenerationTargetToDatabase : HHN000476: Executing script '[injected ScriptSourceInputNonExistentImpl script
2023-06-15T15:22:52.353+05:30 INFO 20344 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-06-15T15:22:52.505+05:30 WARN 20344 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may
2023-06-15T15:22:52.788+05:30 INFO 20344 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-06-15T15:22:52.795+05:30 INFO 20344 --- [main] c.test.example.RestApiTestApplication : Started RestApiTestApplication in 2.598 seconds (process running for 3.263)
```