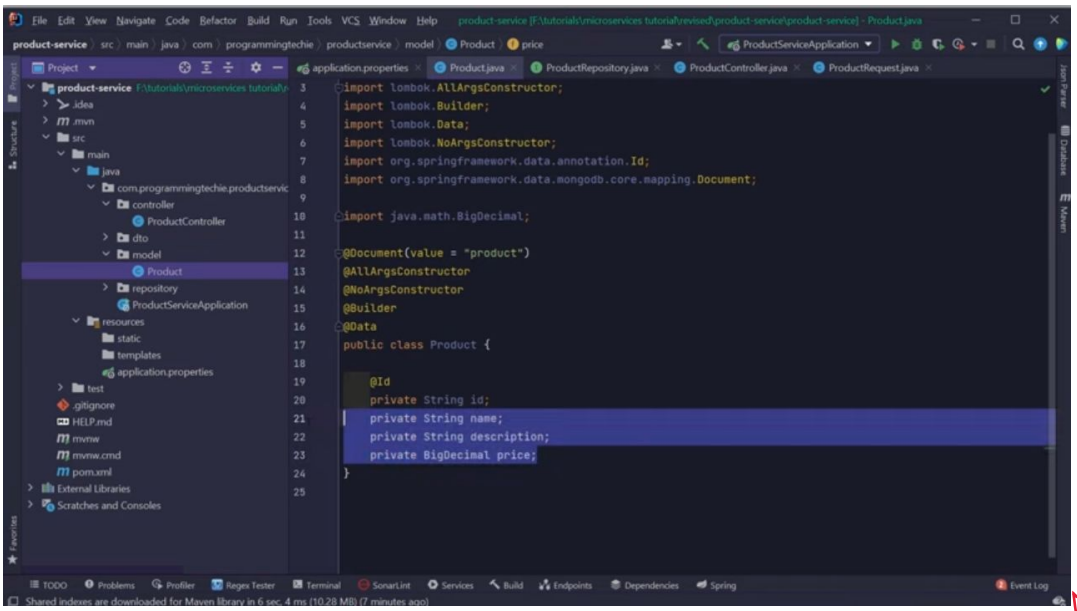
View, add and edit your notes in the app

Spring Boot Microservices Project Example - Part 1 | Building Services

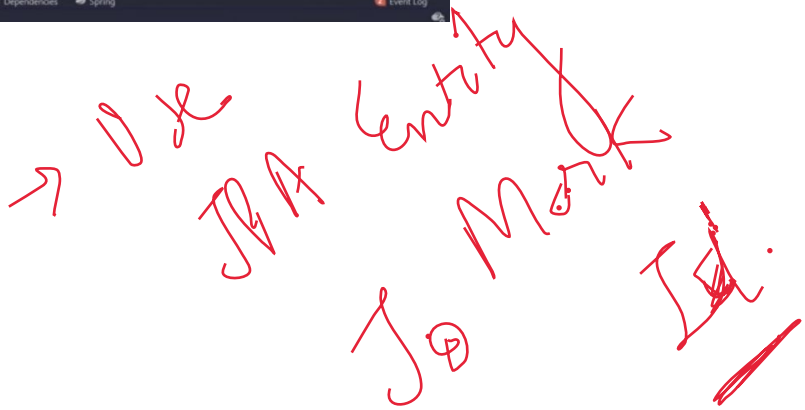
Generated on July 24, 2023

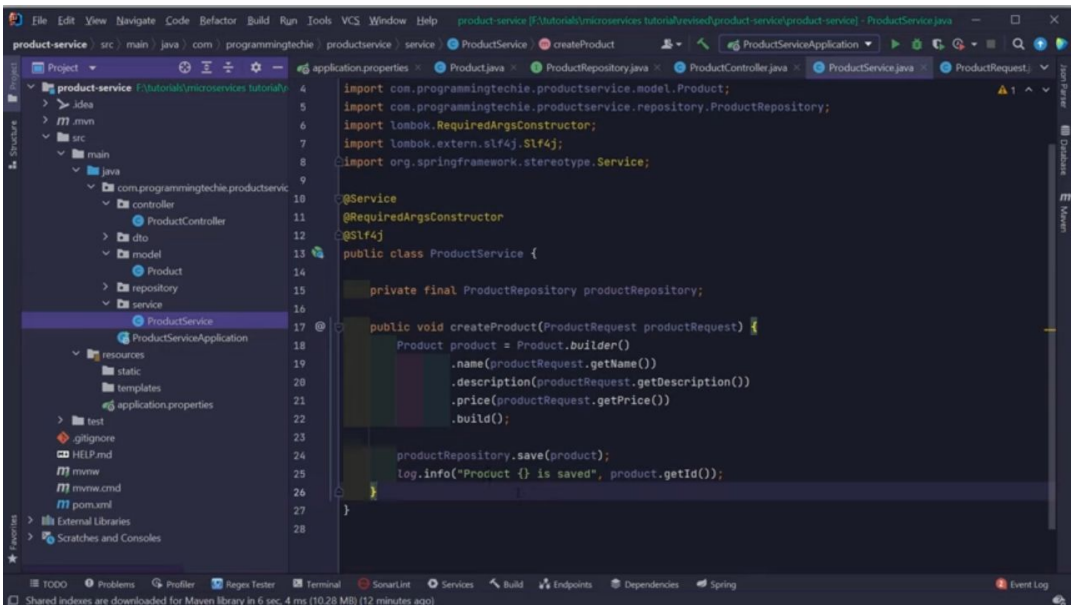
Summary

Notes	Screenshots
1	106

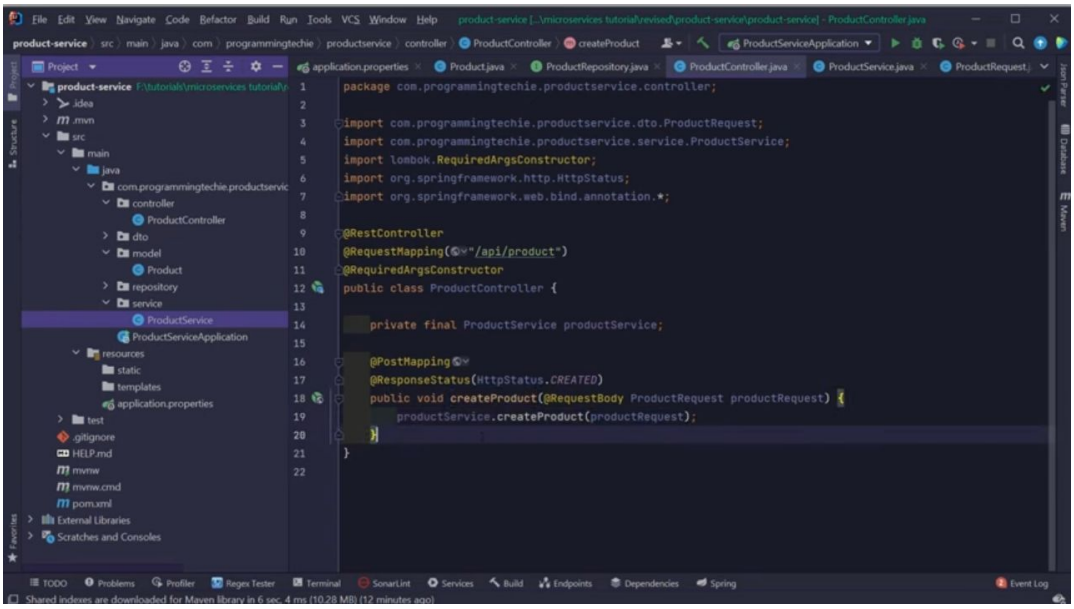


11:00

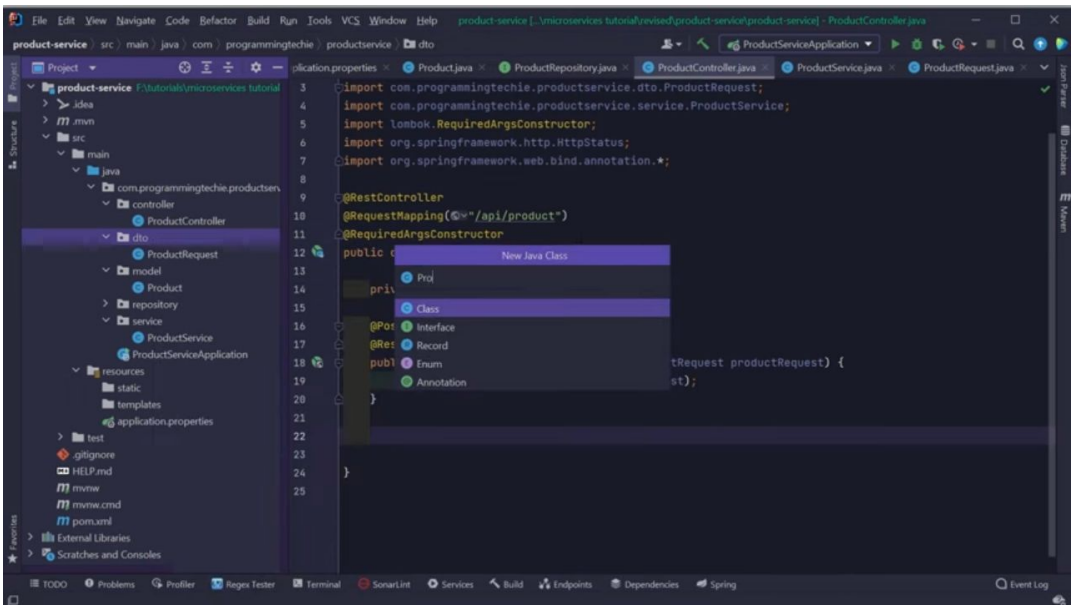



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService createProduct
ProductServiceApplication
Project
application.properties
Product.java
ProductRepository.java
ProductController.java
ProductService.java
ProductRequest.j
product-service F:\tutorials\microservices tutorial\r
.idea
.mvn
src
main
java
com.programmingtechie.productservic
controller
ProductController
dto
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
import com.programmingtechie.productservice.model.Product;
import com.programmingtechie.productservice.repository.ProductRepository;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
@Service
@RequiredArgsConstructor
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Procuct {} is saved", product.getId());
}
}
TODO Problems Profiler Regex Tester Terminal SonarLint Services Build Endpoints Dependencies Spring
Event Log
Shared indexes are downloaded for Maven library in 6 sec, 4 ms (10.28 MB) (12 minutes ago)

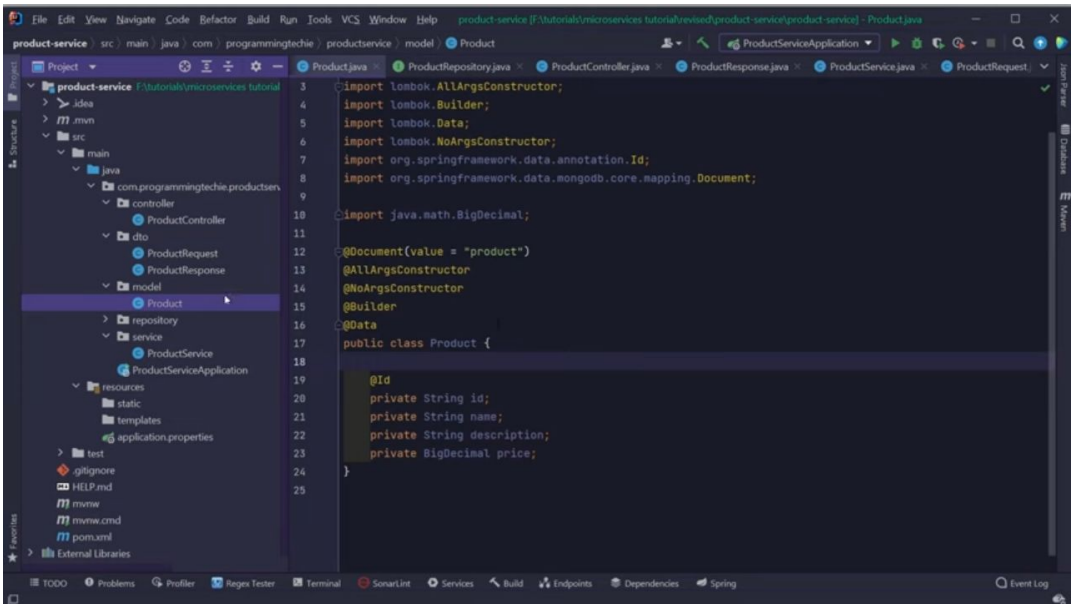
15:18


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [...\microservices tutorial\revised\product-service\product-service] - ProductController.java
product-service src main java com programmingtechie productservice controller ProductController createProduct
ProductServiceApplication
Project
application.properties
Product.java
ProductRepository.java
ProductController.java
ProductService.java
ProductRequest.j
product-service F:\tutorials\microservices tutorial\r
.idea
.mvn
src
main
java
com.programmingtechie.productservic
controller
ProductController
dto
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.programmingtechie.productservice.controller;
import com.programmingtechie.productservice.dto.ProductRequest;
import com.programmingtechie.productservice.service.ProductService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.*;
@RestController
@RequestMapping("/api/product")
@RequiredArgsConstructor
public class ProductController {
private final ProductService productService;
@PostMapping
@ResponseStatus(HttpStatus.CREATED)
public void createProduct(@RequestBody ProductRequest productRequest) {
productService.createProduct(productRequest);
}
}
TODO Problems Profiler Regex Tester Terminal SonarLint Services Build Endpoints Dependencies Spring
Event Log
Shared indexes are downloaded for Maven library in 6 sec, 4 ms (10.28 MB) (12 minutes ago)

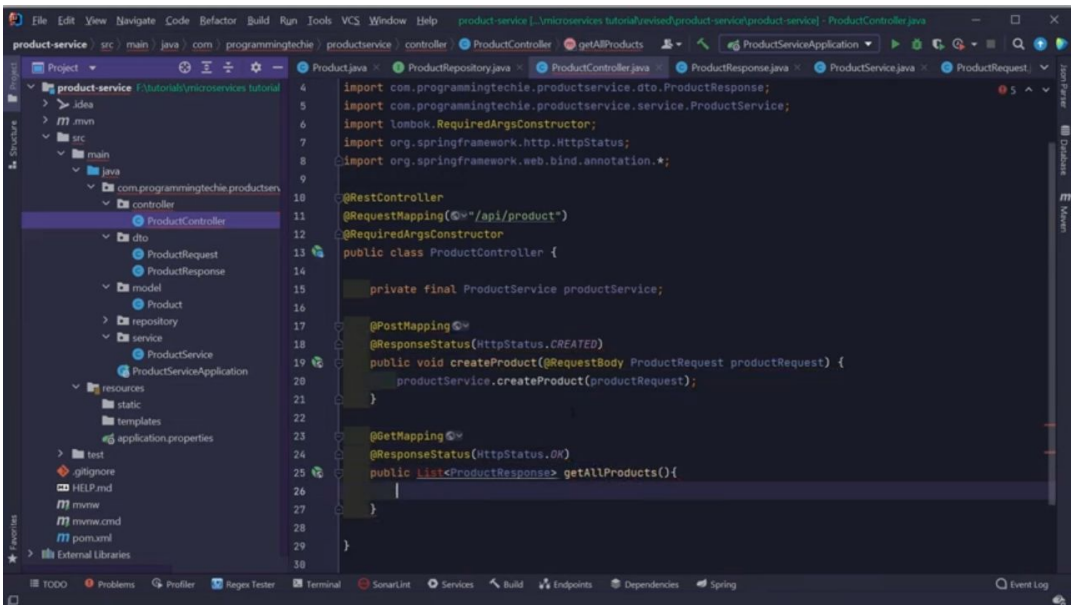
16:08


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [...\microservices tutorial\revised\product-service\product-service] - ProductController.java
product-service src main java com programmingtechie productservice dto
ProductServiceApplication
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
application.properties
Product.java
ProductRepository.java
ProductController.java
ProductService.java
ProductRequest.java
import com.programmingtechie.productservice.dto.ProductRequest;
import com.programmingtechie.productservice.service.ProductService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.*;
@RestController
@RequestMapping("/api/product")
@RequiredArgsConstructor
public
New Java Class
Pro
Class
Interface
Record
Enum
Annotation
priv
@Po
@Re
publ
tRequest productRequest) {
st);
}
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

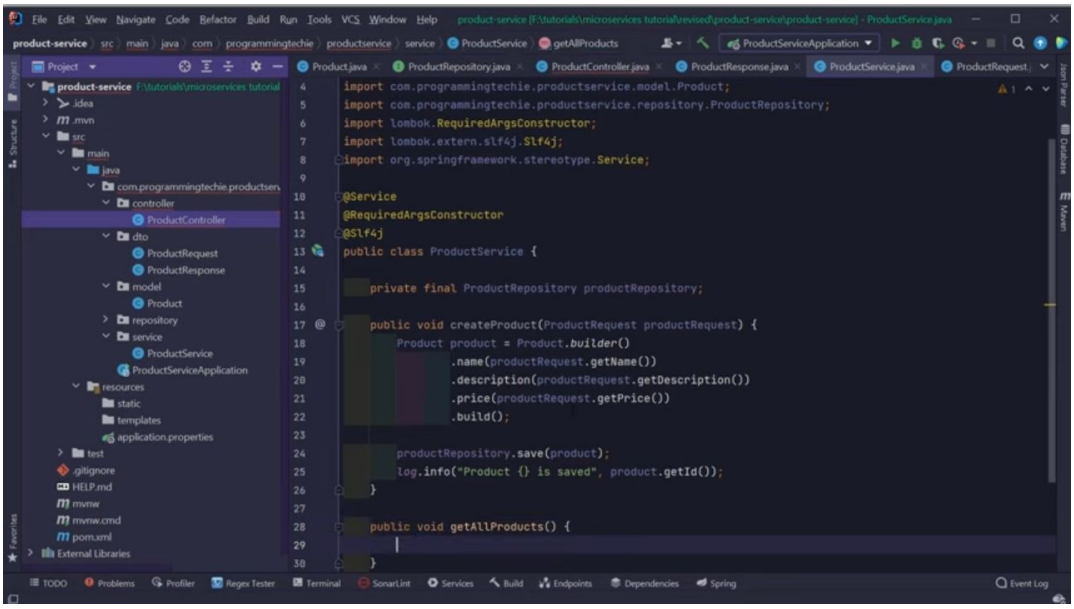
16:19


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - Product.java
product-service src main java com programmingtechie productservice model Product
ProductServiceApplication
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.mapping.Document;
import java.math.BigDecimal;
@Document(value = "product")
@AllArgsConstructor
@NoArgsConstructor
@Builder
@Data
public class Product {
@Id
private String id;
private String name;
private String description;
private BigDecimal price;
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

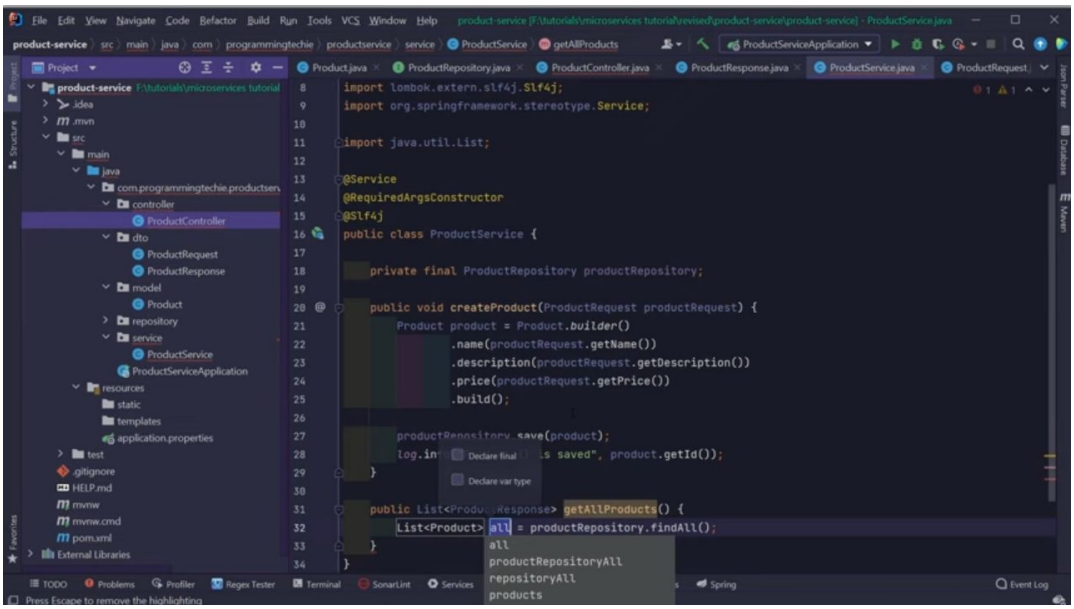
17:07


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [...\microservices tutorial\revised\product-service\product-service] - ProductController.java
product-service src main java com programmingtechie productservice controller ProductController getAllProducts
ProductServiceApplication
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
import com.programmingtechie.productservice.dto.ProductResponse;
import com.programmingtechie.productservice.service.ProductService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.*;
@RestController
@RequestMapping("/api/product")
@RequiredArgsConstructor
public class ProductController {
private final ProductService productService;
@PostMapping
@ResponseStatus(HttpStatus.CREATED)
public void createProduct(@RequestBody ProductRequest productRequest) {
productService.createProduct(productRequest);
}
@GetMapping
@ResponseStatus(HttpStatus.OK)
public List<ProductResponse> getAllProducts(){
}
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

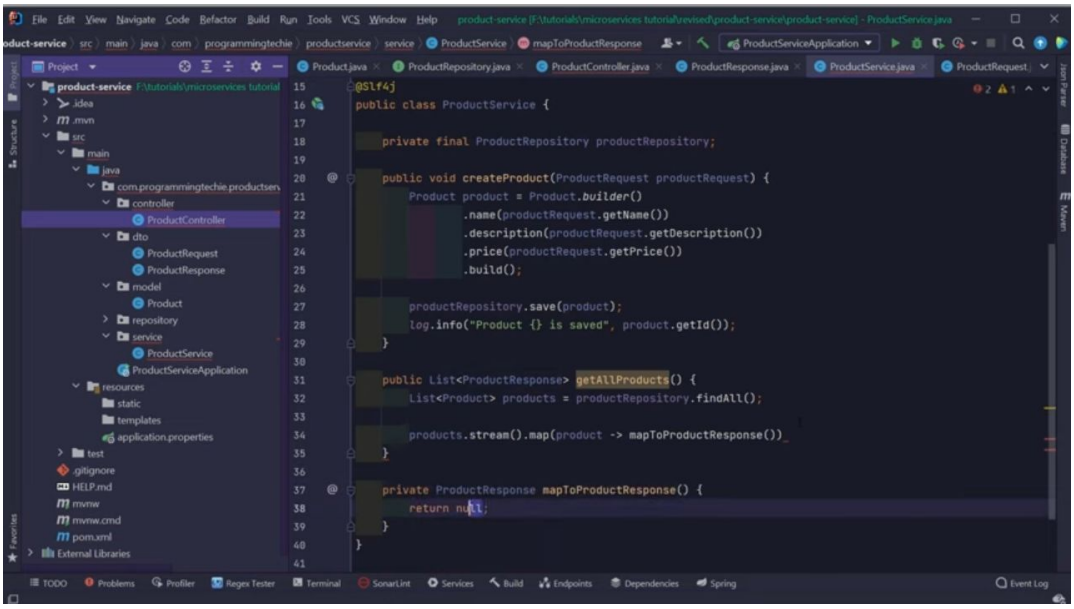
18:17


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService getAllProducts
ProductServiceApplication
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
import com.programmingtechie.productservice.model.Product;
import com.programmingtechie.productservice.repository.ProductRepository;
import lombok.RequiredArgsConstructor;
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
@Service
@RequiredArgsConstructor
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public void getAllProducts() {
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

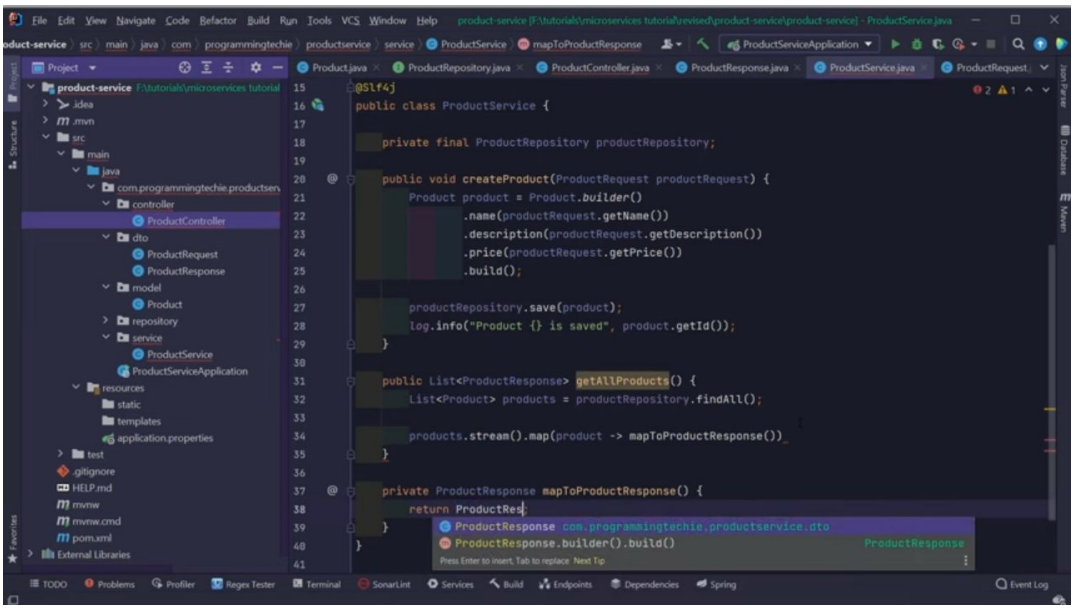
18:40


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService getAllProducts
ProductServiceApplication
Project
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;
import java.util.List;
@Service
@RequiredArgsConstructor
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
Declare final
Declare var type
public List<ProductResponse> getAllProducts() {
List<Product> all = productRepository.findAll();
}
all
productRepositoryAll
repositoryAll
products
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Spring
Event Log
Press Escape to remove the highlighting

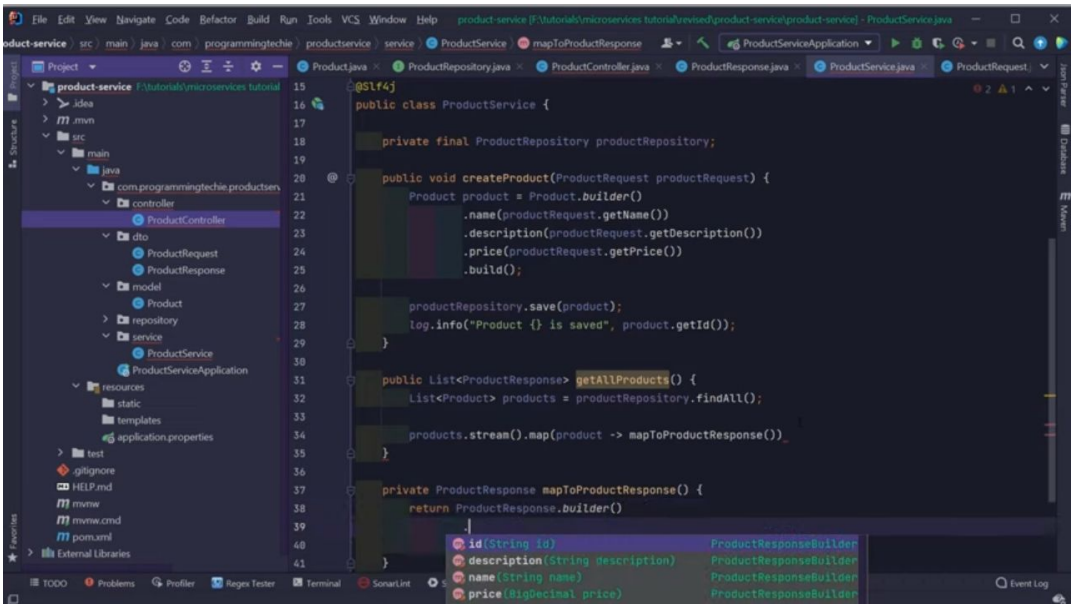
19:07


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService mapToProductResponse
ProductServiceApplication
Project
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse())
}
private ProductResponse mapToProductResponse() {
return null;
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

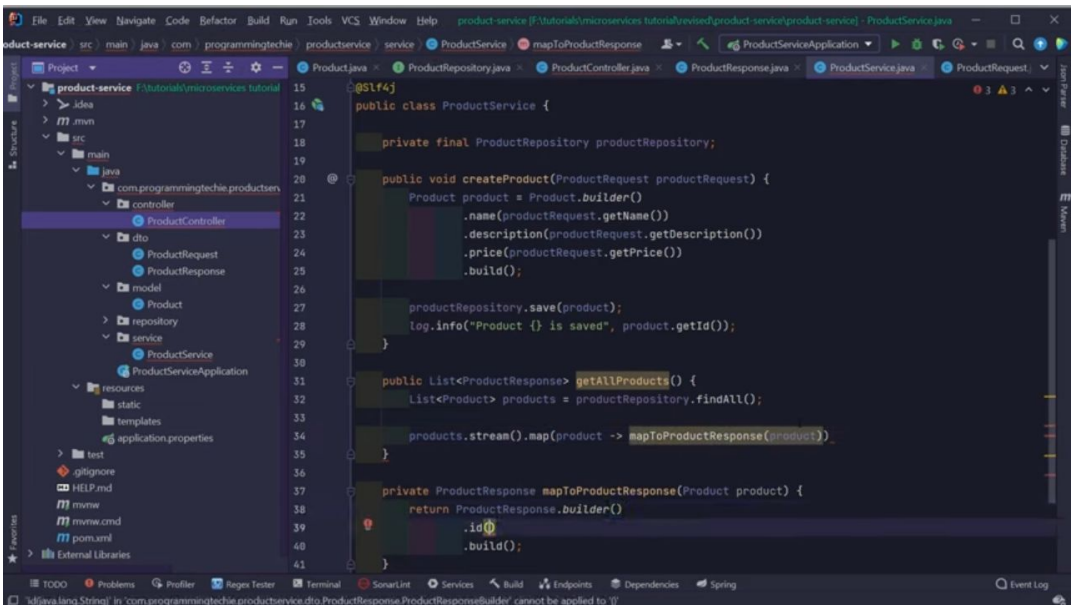
19:52


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse())
}
private ProductResponse mapToProductResponse() {
return ProductRes;
ProductResponse com.programmingtechie.productservice.dto
ProductResponse.builder().build()
ProductResponse
Press Enter to insert, Tab to replace Next Tip
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

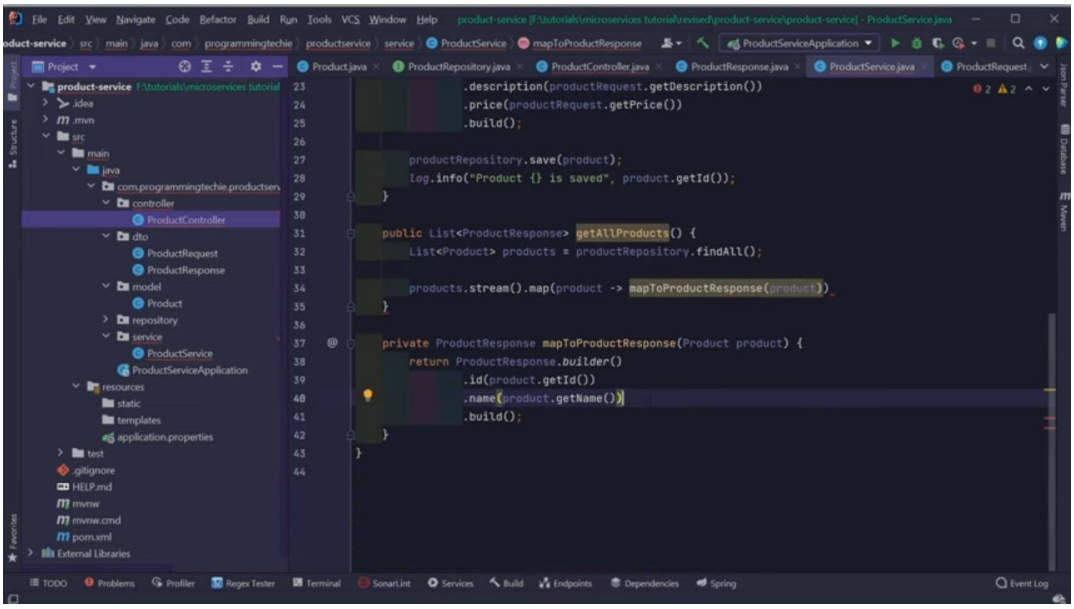
19:56


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
Project
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse())
}
private ProductResponse mapToProductResponse() {
return ProductResponse.builder()
.
id(String id) ProductResponseBuilder
description(String description) ProductResponseBuilder
name(String name) ProductResponseBuilder
price(BigDecimal price) ProductResponseBuilder
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Event Log

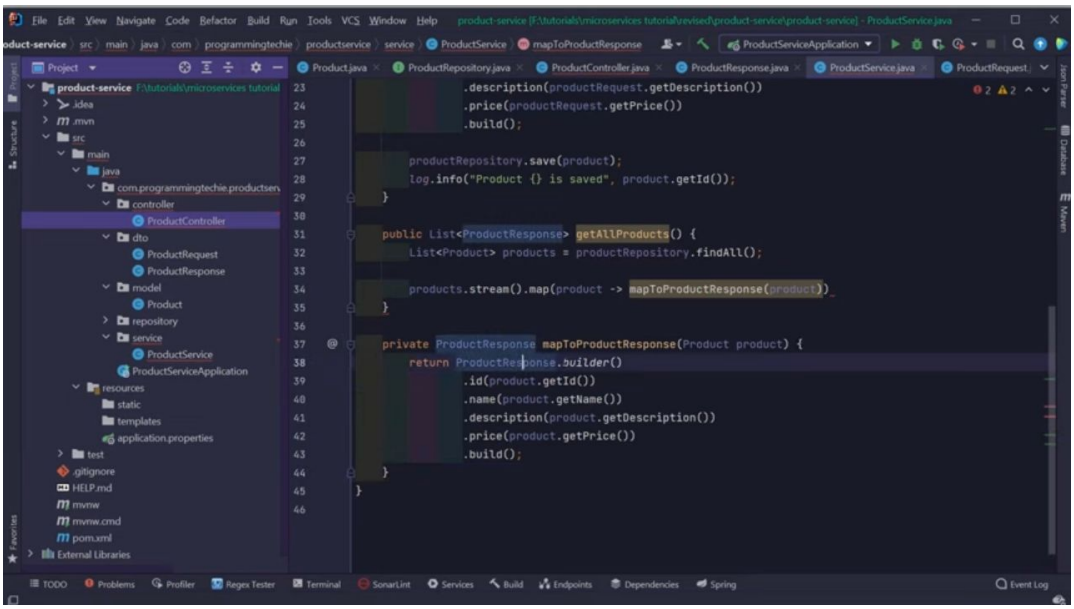
20:04


@Slf4j
public class ProductService {
private final ProductRepository productRepository;
public void createProduct(ProductRequest productRequest) {
Product product = Product.builder()
.name(productRequest.getName())
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse(product))
}
private ProductResponse mapToProductResponse(Product product) {
return ProductResponse.builder()
.id()
.build();
}

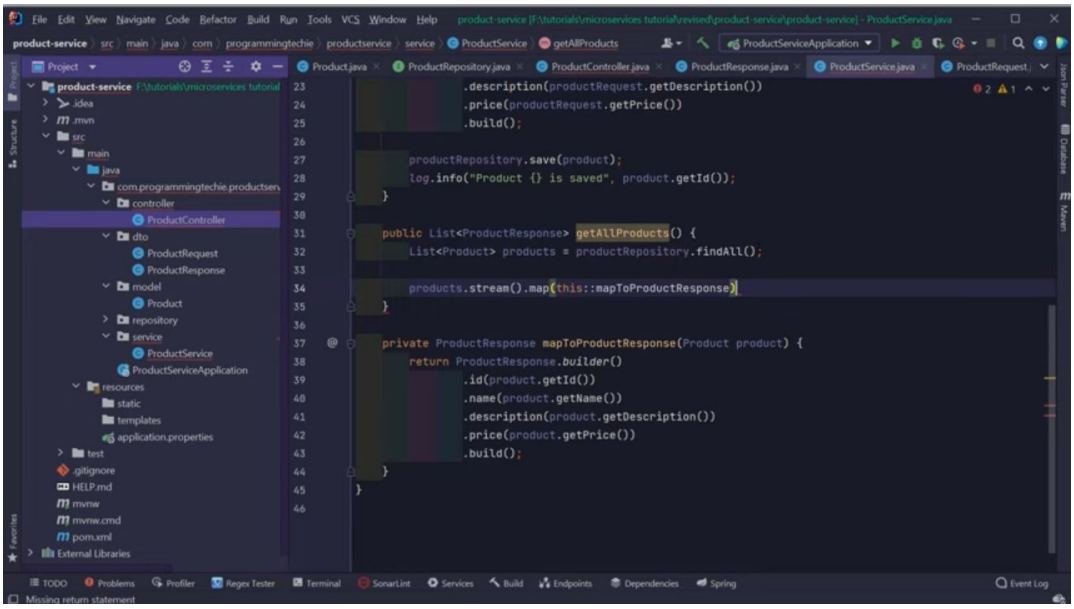
20:21


.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse(product))
}
private ProductResponse mapToProductResponse(Product product) {
return ProductResponse.builder()
.id(product.getId())
.name(product.getName())
.build();
}
}

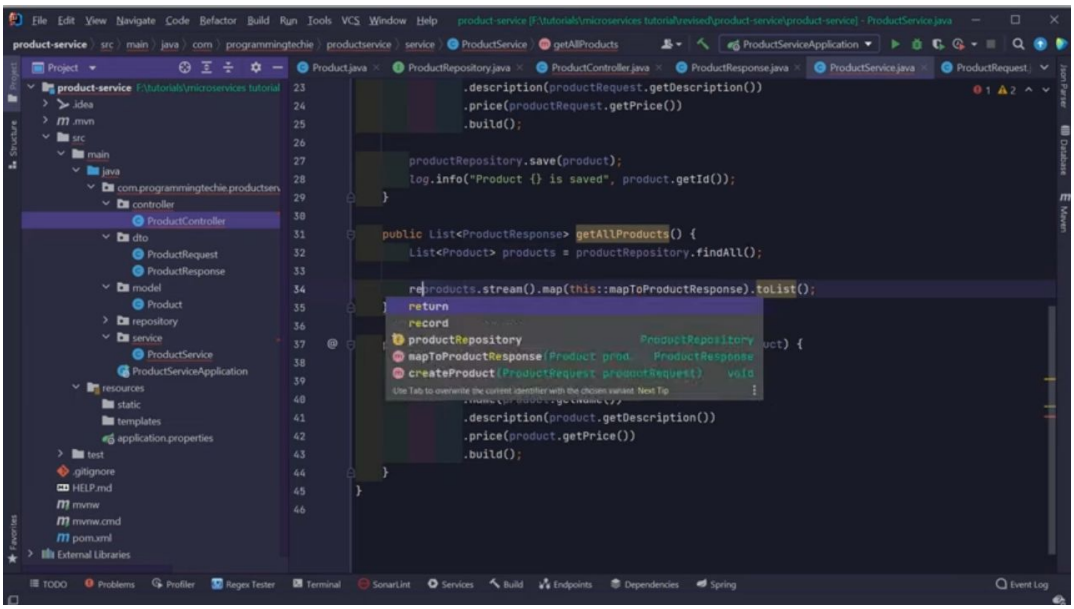
20:36


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService mapToProductResponse
ProductServiceApplication
Project
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(product -> mapToProductResponse(product))
private ProductResponse mapToProductResponse(Product product) {
return ProductResponse.builder()
.id(product.getId())
.name(product.getName())
.description(product.getDescription())
.price(product.getPrice())
.build();
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

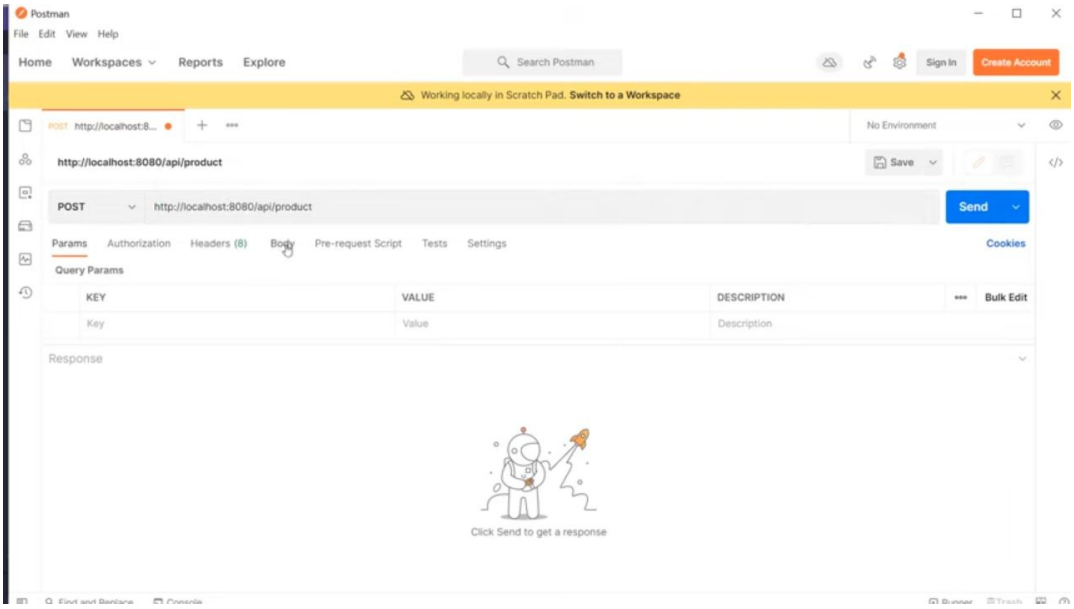
20:54


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService getAllProducts
ProductServiceApplication
Project
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
products.stream().map(this::mapToProductResponse)
private ProductResponse mapToProductResponse(Product product) {
return ProductResponse.builder()
.id(product.getId())
.name(product.getName())
.description(product.getDescription())
.price(product.getPrice())
.build();
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log
Missing return statement

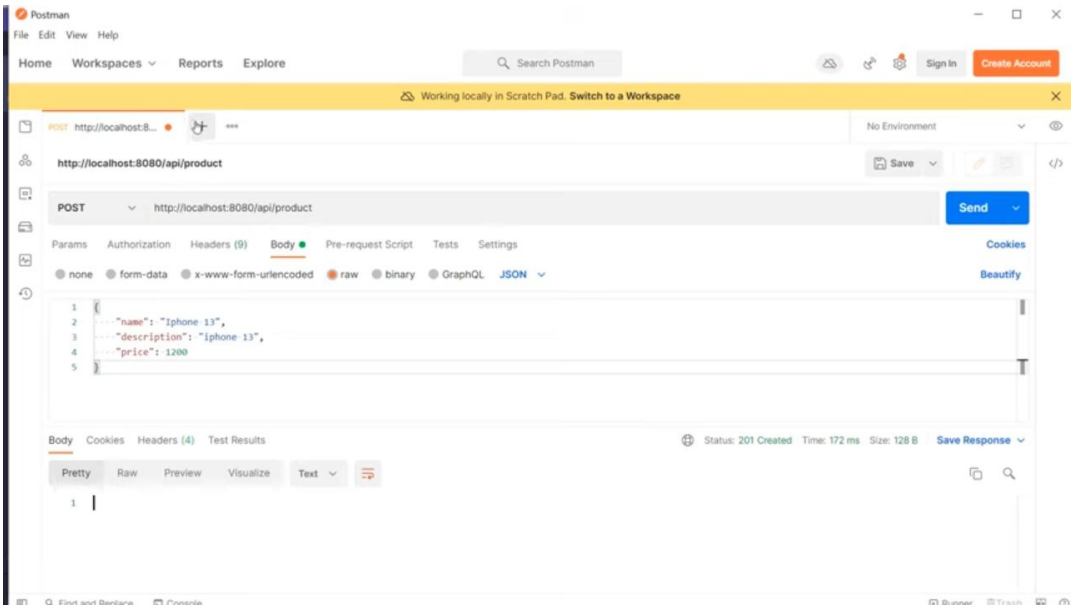
21:15


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service [F:\tutorials\microservices tutorial\revised\product-service\product-service] - ProductService.java
product-service src main java com programmingtechie productservice service ProductService getAllProducts
ProductServiceApplication
Project
Product.java
ProductRepository.java
ProductController.java
ProductResponse.java
ProductService.java
ProductRequest
product-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.productsen
controller
ProductController
dto
ProductRequest
ProductResponse
model
Product
repository
service
ProductService
ProductServiceApplication
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
.description(productRequest.getDescription())
.price(productRequest.getPrice())
.build();
productRepository.save(product);
log.info("Product {} is saved", product.getId());
}
public List<ProductResponse> getAllProducts() {
List<Product> products = productRepository.findAll();
return
record
productRepository ProductRepository
mapToProductResponse(Product prod ProductResponse
createProduct(ProductRequest productRequest) void
Use Tab to overwrite the current identifier with the chosen variant. Next Tip
.description(product.getDescription())
.price(product.getPrice())
.build();
}
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

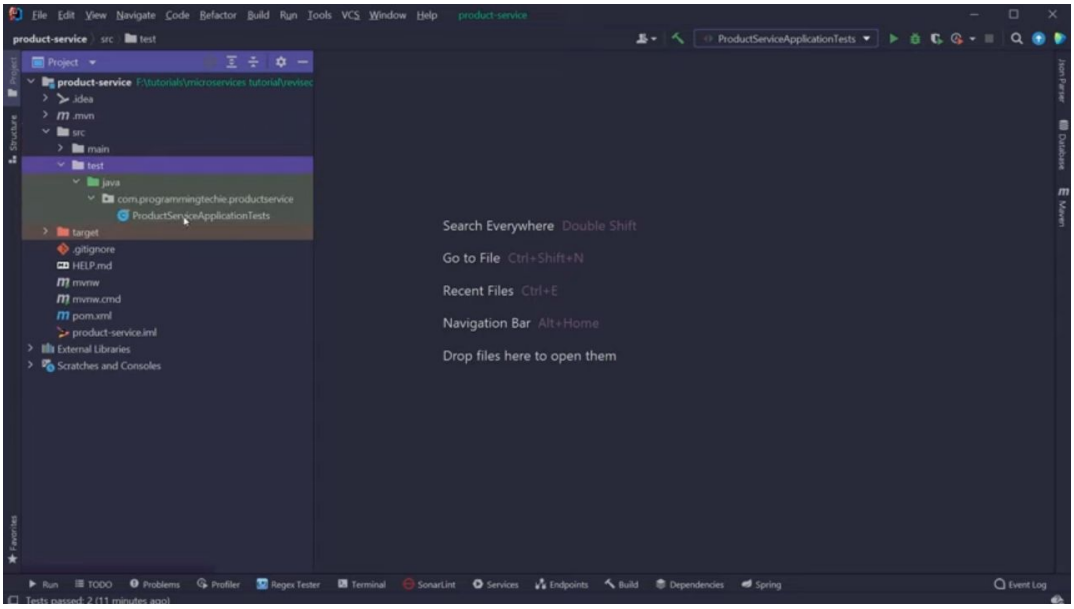
21:27


Postman
File Edit View Help
Home
Workspaces
Reports
Explore
Search Postman
Sign In
Create Account
Working locally in Scratch Pad. Switch to a Workspace
No Environment
http://localhost:8080/api/product
Save
POST
http://localhost:8080/api/product
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
Key
Value
Description
Response
Click Send to get a response

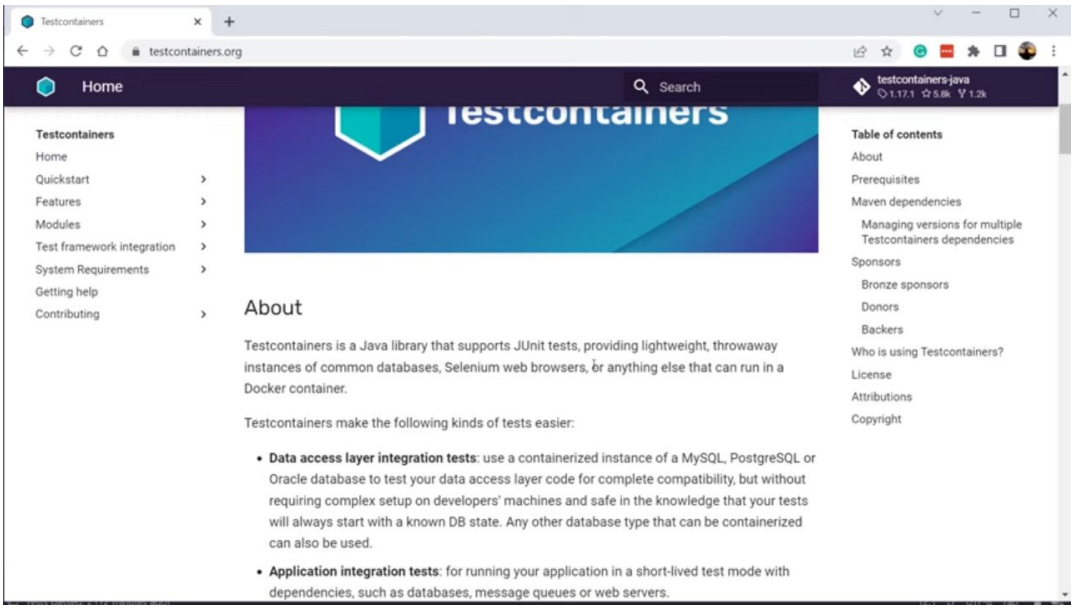
22:19


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Create Account
Working locally in Scratch Pad. Switch to a Workspace
No Environment
http://localhost:8080/api/product
Save
POST
http://localhost:8080/api/product
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"name": "Iphone 13",
"description": "iphone 13",
"price": 1200
}
Body Cookies Headers (4) Test Results
Status: 201 Created Time: 172 ms Size: 128 B
Save Response
Pretty Raw Preview Visualize Text

23:06


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service
ProductServiceApplicationTests
Project
product-service
.idea
.mvn
src
main
test
java
com.programmingtechie.productservice
ProductServiceApplicationTests
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
product-service.iml
External Libraries
Scratches and Consoles
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Navigation Bar Alt+Home
Drop files here to open them

23:38

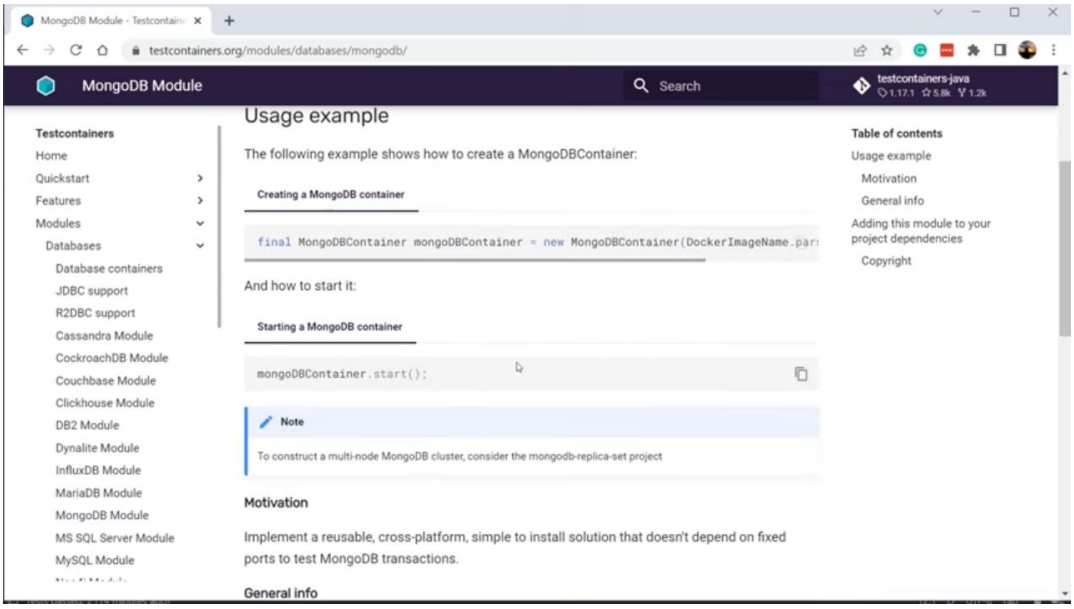


→ open

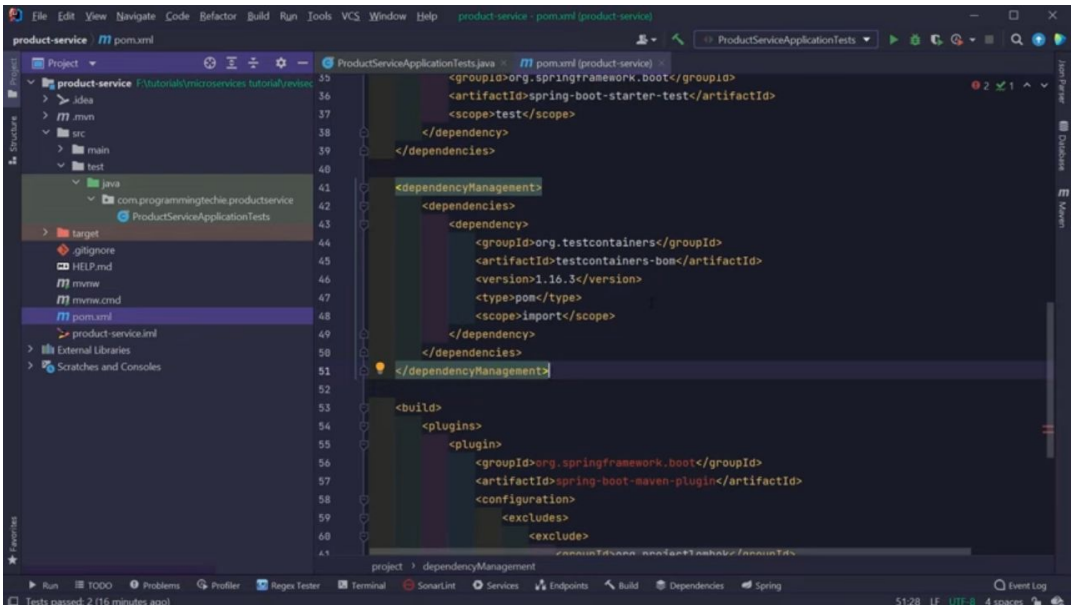
Jet

Containers

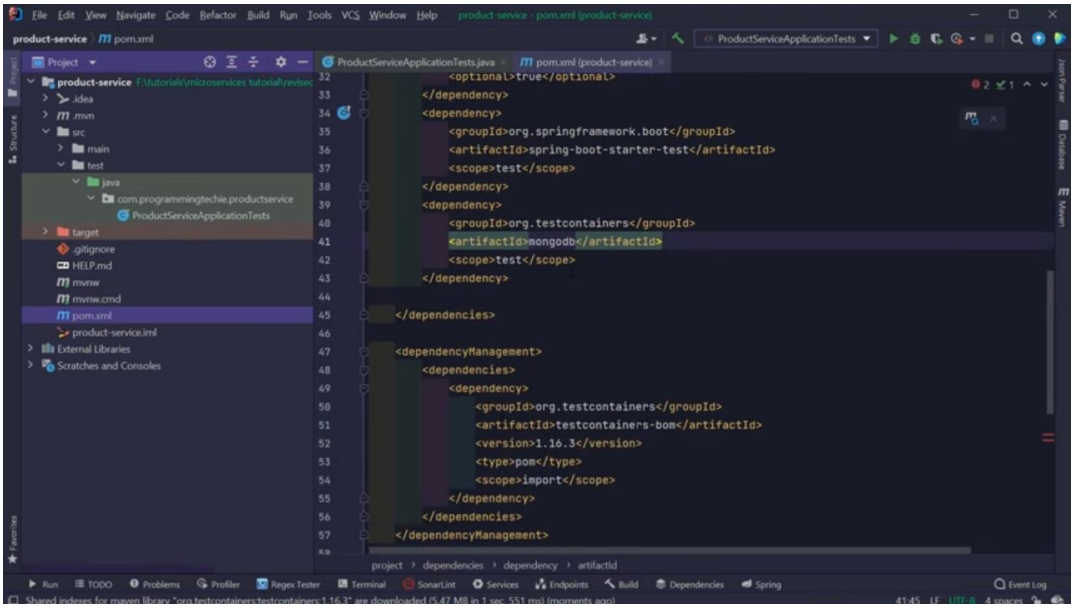
▷ 24:28



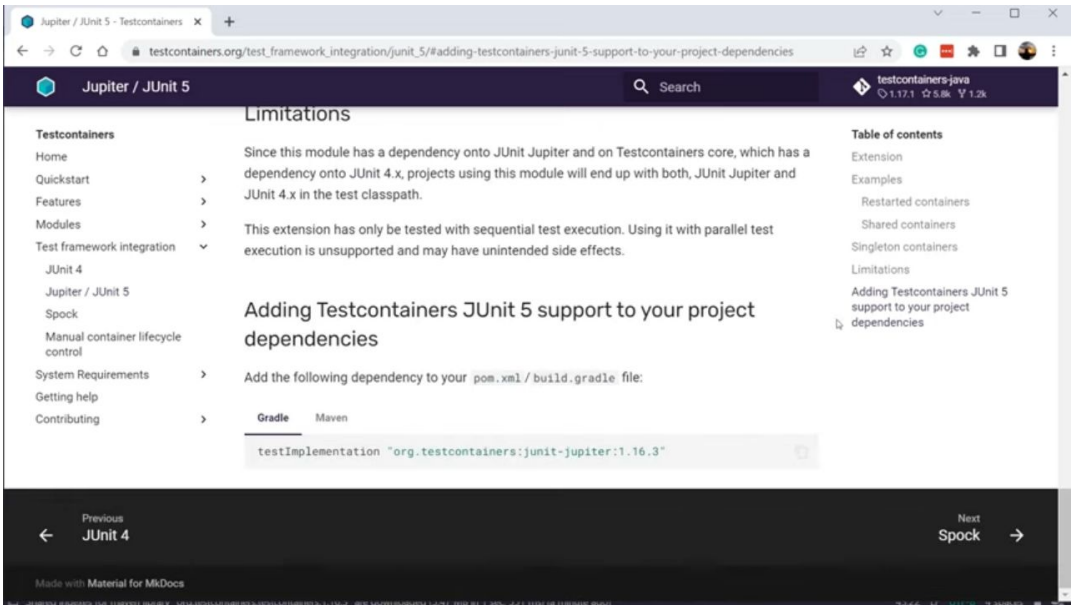
▷ 26:06


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - pom.xml (product-service)
product-service | pom.xml
ProductServiceApplicationTests
ProductServiceApplicationTests.java
pom.xml (product-service)
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers-bom</artifactId>
<version>1.16.3</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
<configuration>
<excludes>
<exclude>
project › dependencyManagement

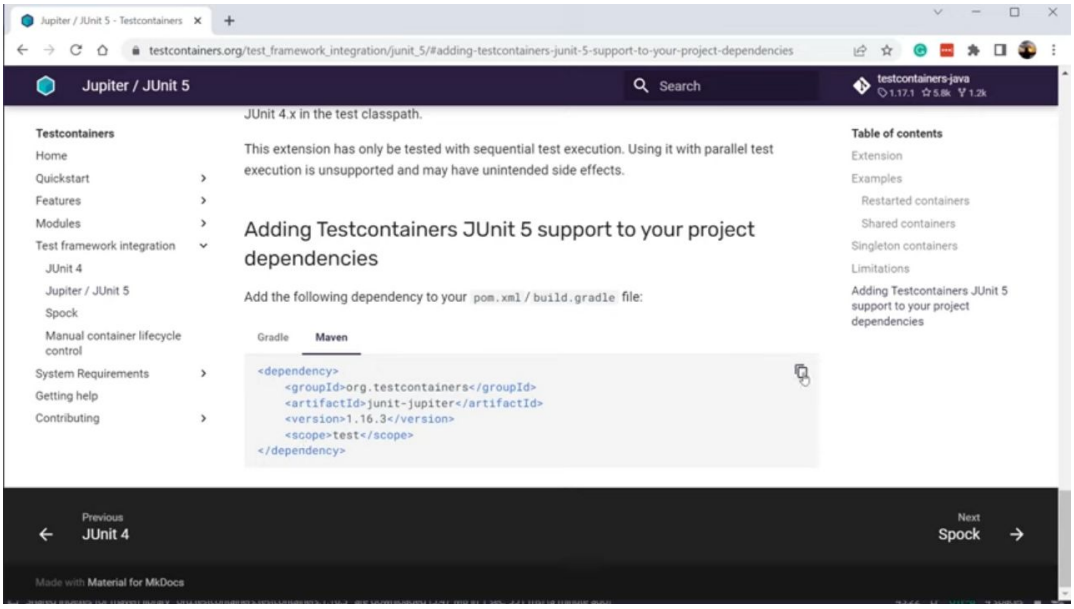
27:19


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - pom.xml (product-service)
product-service | pom.xml
ProductServiceApplicationTests
ProductServiceApplicationTests.java
pom.xml (product-service)
<optional>true</optional>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>mongodb</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers-bom</artifactId>
<version>1.16.3</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
project › dependencies › dependency › artifactId

28:09

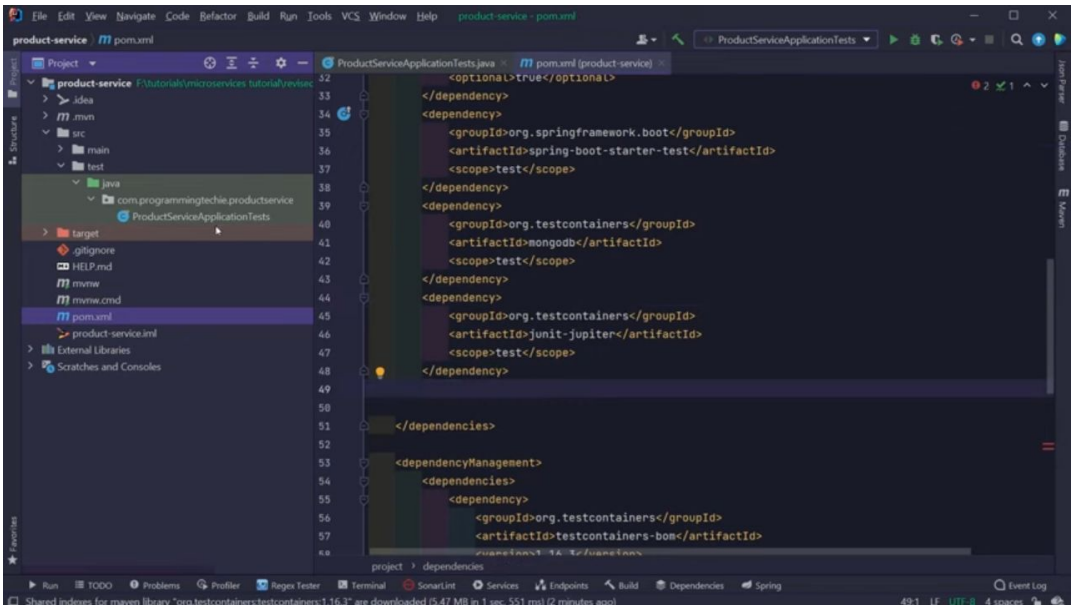


29:01

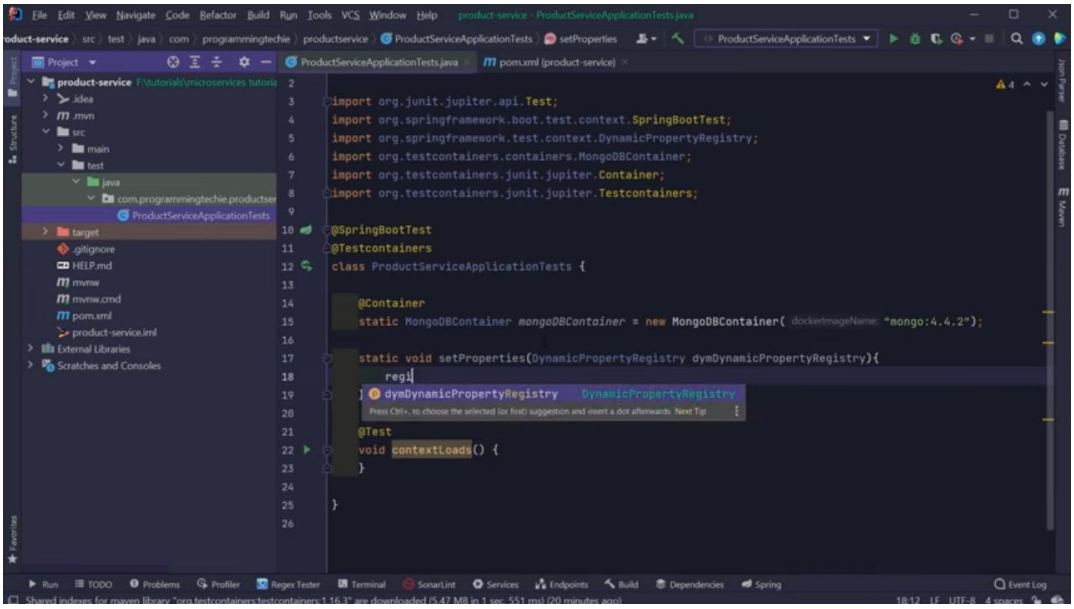


29:07

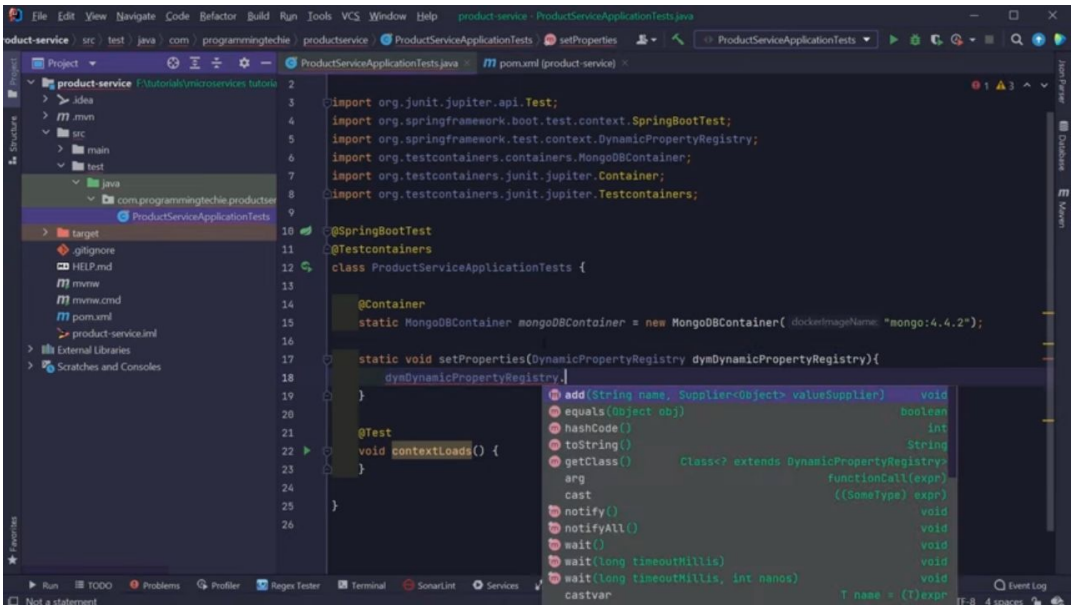
→ Copy the Maven dependency


product-service - pom.xml
ProductServiceApplicationTests.java
pom.xml (product-service)
<optional>true</optional>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>mongodb</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>junit-jupiter</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers-bom</artifactId>

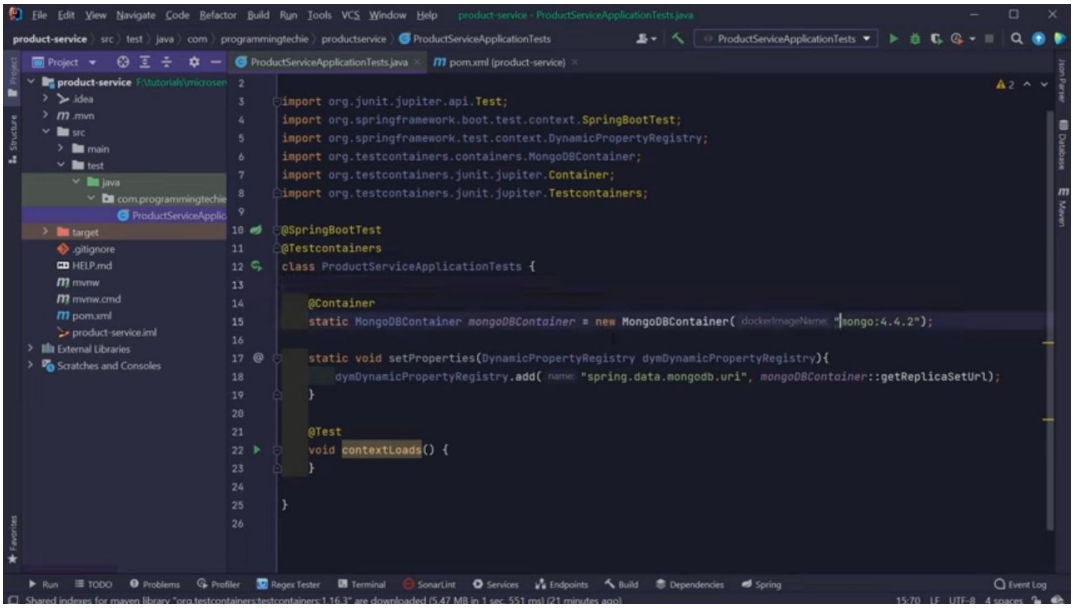
29:28


product-service - ProductServiceApplicationTests.java
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.DynamicPropertyRegistry;
import org.testcontainers.containers.MongoDBContainer;
import org.testcontainers.junit.jupiter.Container;
import org.testcontainers.junit.jupiter.Testcontainers;
@SpringBootTest
@Testcontainers
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry){
regi
dymDynamicPropertyRegistry DynamicPropertyRegistry
@Test
void contextLoads() {
}
}

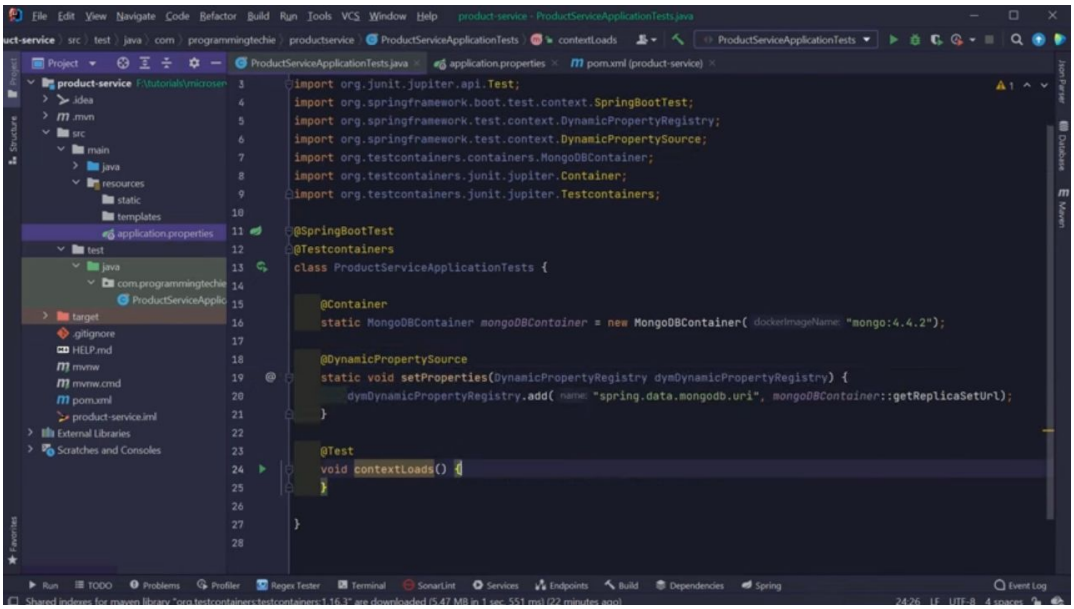
31:44


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - ProductServiceApplicationTests.java
ProductServiceApplicationTests.java
pom.xml (product-service)
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.DynamicPropertyRegistry;
import org.testcontainers.containers.MongoDBContainer;
import org.testcontainers.junit.jupiter.Container;
import org.testcontainers.junit.jupiter.Testcontainers;
@SpringBootTest
@Testcontainers
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry){
dymDynamicPropertyRegistry.
}
@Test
void contextLoads() {
}
}
add(String name, Supplier<Object> valueSupplier) void
equals(Object obj) boolean
hashCode() int
toString() String
getClass() Class<? extends DynamicPropertyRegistry>
arg functionCall(expr)
cast ((SomeType) expr)
notify() void
notifyAll() void
wait() void
wait(long timeoutMillis) void
wait(long timeoutMillis, int nanos) void
castvar
Run TODO Problems Profiler Regex Tester Terminal SonarLint Services
Event Log

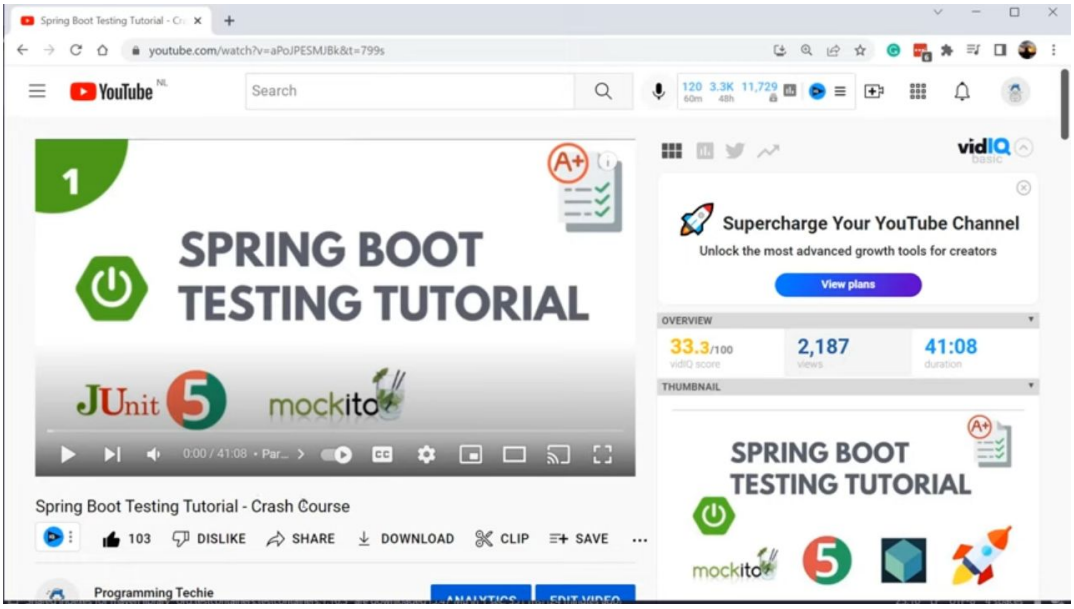
31:47


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - ProductServiceApplicationTests.java
ProductServiceApplicationTests.java
pom.xml (product-service)
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.DynamicPropertyRegistry;
import org.testcontainers.containers.MongoDBContainer;
import org.testcontainers.junit.jupiter.Container;
import org.testcontainers.junit.jupiter.Testcontainers;
@SpringBootTest
@Testcontainers
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry){
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void contextLoads() {
}
}
Run TODO Problems Profiler Regex Tester Terminal SonarLint Services Endpoints Build Dependencies Spring
Event Log

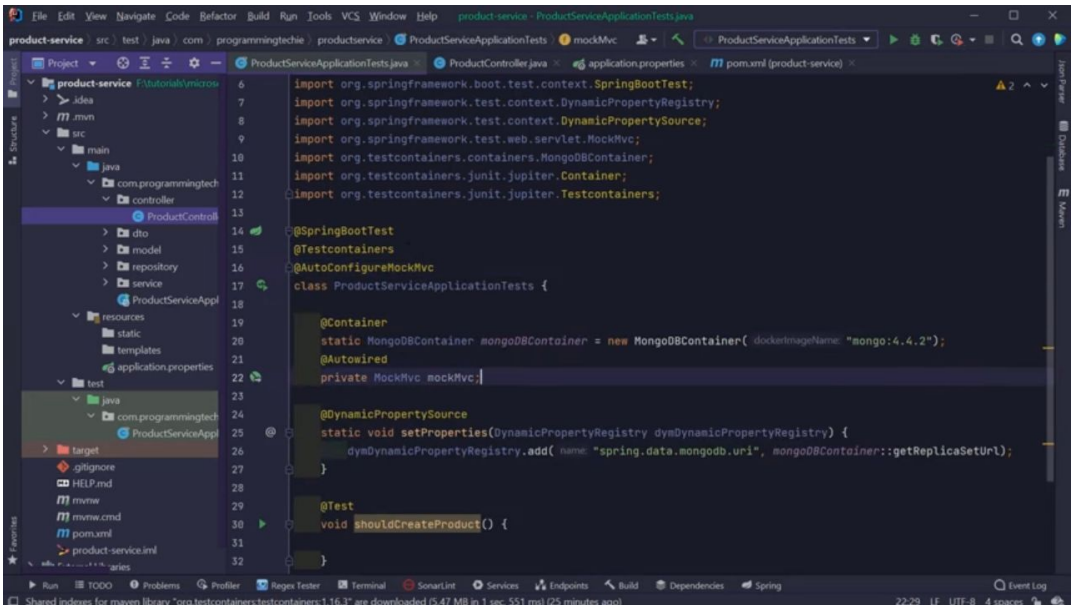
32:18


product-service - ProductServiceApplicationTests.java
ProductServiceApplicationTests.java
application.properties
pom.xml (product-service)
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.DynamicPropertyRegistry;
import org.springframework.test.context.DynamicPropertySource;
import org.testcontainers.containers.MongoDBContainer;
import org.testcontainers.junit.jupiter.Container;
import org.testcontainers.junit.jupiter.Testcontainers;
@SpringBootTest
@Testcontainers
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void contextLoads() {
}
}

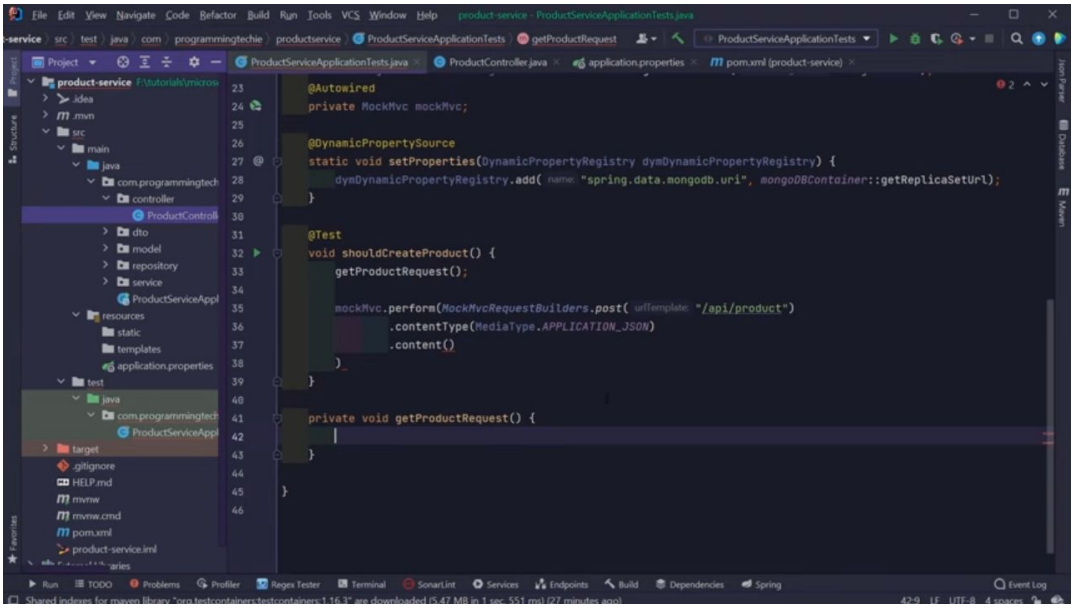
33:18


youtube.com/watch?v=aPoJPESMJBk&t=799s
YouTube
Search
1
SPRING BOOT
TESTING TUTORIAL
JUnit 5
mockito
Supercharge Your YouTube Channel
Unlock the most advanced growth tools for creators
View plans
OVERVIEW
33.3/100
2,187
41:08
THUMBNAIL
SPRING BOOT
TESTING TUTORIAL
Spring Boot Testing Tutorial - Crash Course
103
DISLIKE
SHARE
DOWNLOAD
CLIP
SAVE
Programming Techie

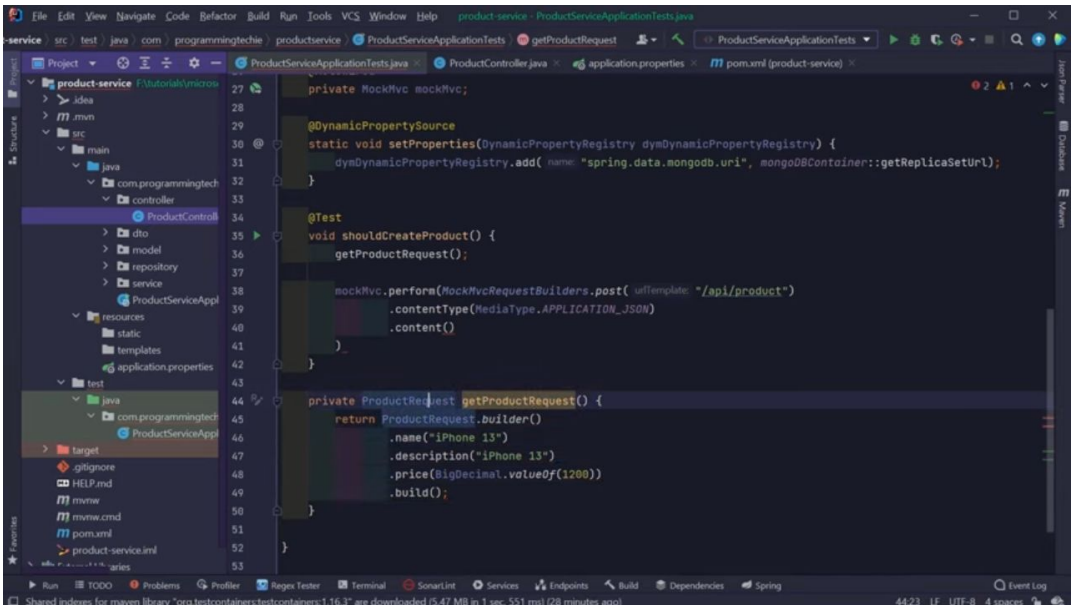
34:33


import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.DynamicPropertyRegistry;
import org.springframework.test.context.DynamicPropertySource;
import org.springframework.test.web.servlet.MockMvc;
import org.testcontainers.containers.MongoDBContainer;
import org.testcontainers.junit.jupiter.Container;
import org.testcontainers.junit.jupiter.Testcontainers;
@SpringBootTest
@Testcontainers
@AutoConfigureMockMvc
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@Autowired
private MockMvc mockMvc;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() {
}

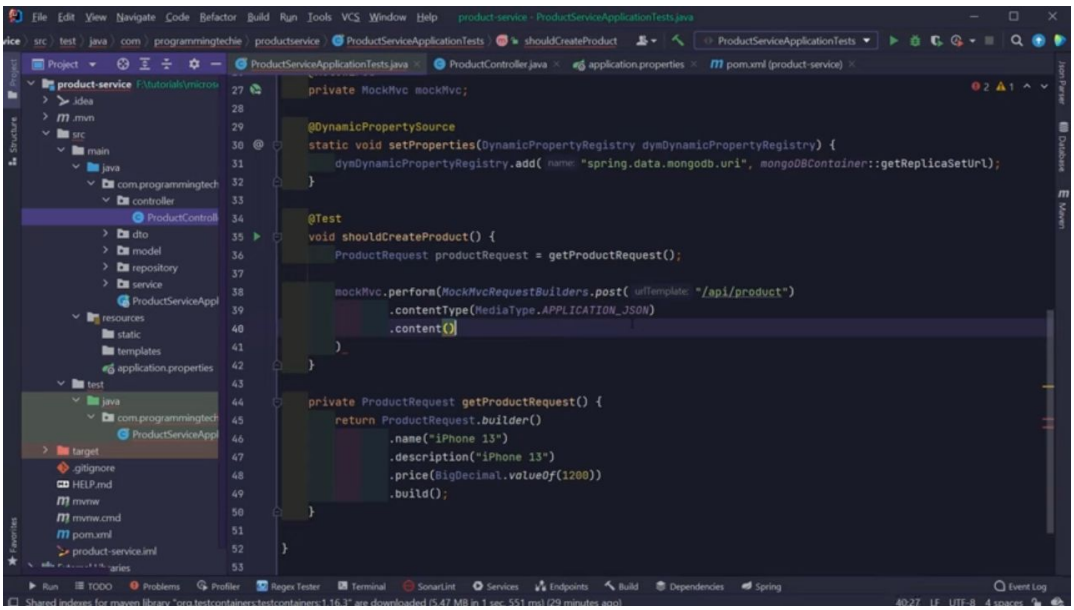
35:32


@Autowired
private MockMvc mockMvc;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() {
getProductRequest();
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private void getProductRequest() {
}
}

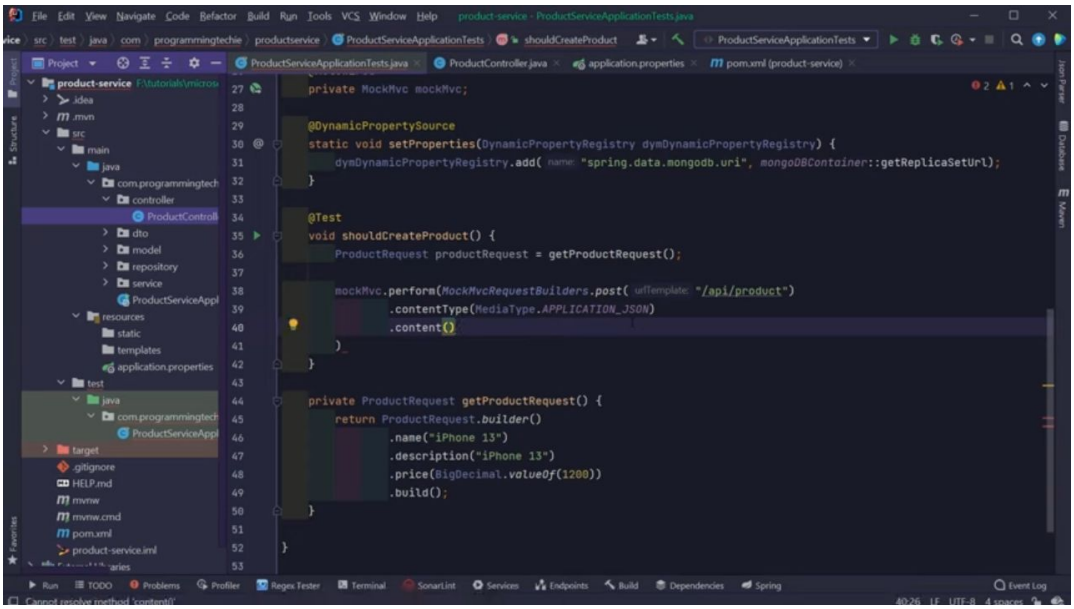
37:15


private MockMvc mockMvc;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() {
getProductRequest();
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()
.name("iPhone 13")
.description("iPhone 13")
.price(BigDecimal.valueOf(1200))
.build();
}
}

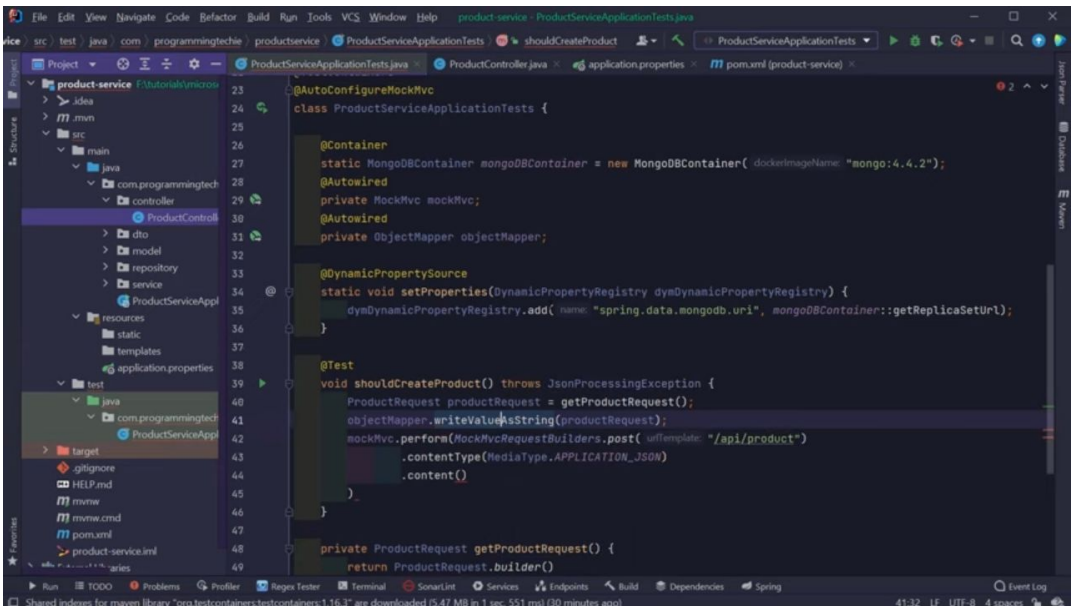
38:01


private MockMvc mockMvc;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() {
ProductRequest productRequest = getProductRequest();
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()
.name("iPhone 13")
.description("iPhone 13")
.price(BigDecimal.valueOf(1200))
.build();
}
}

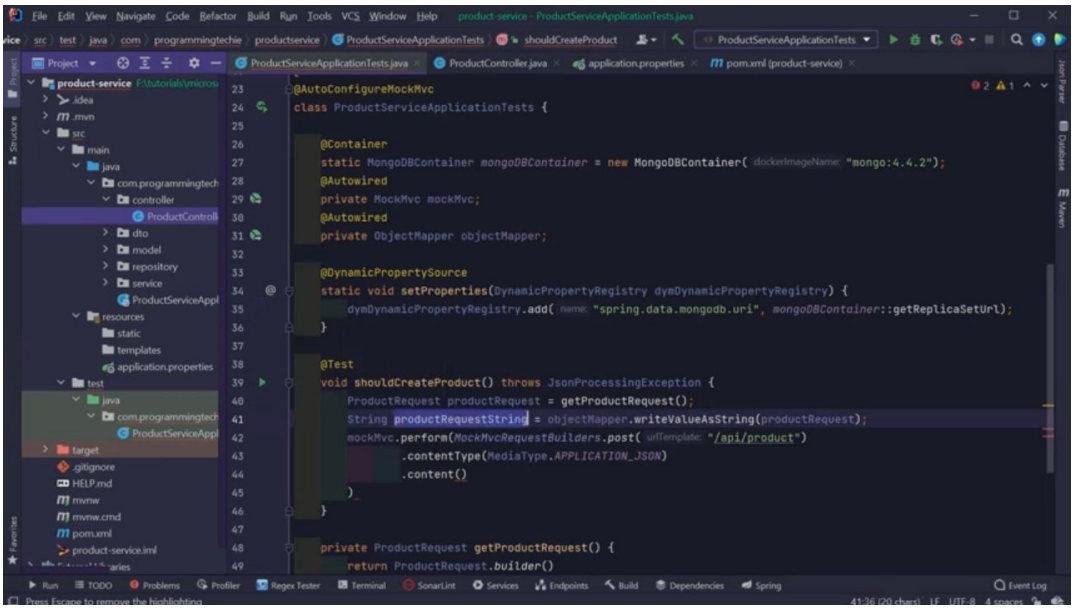
38:18


private MockMvc mockMvc;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() {
ProductRequest productRequest = getProductRequest();
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()
.name("iPhone 13")
.description("iPhone 13")
.price(BigDecimal.valueOf(1200))
.build();
}
}

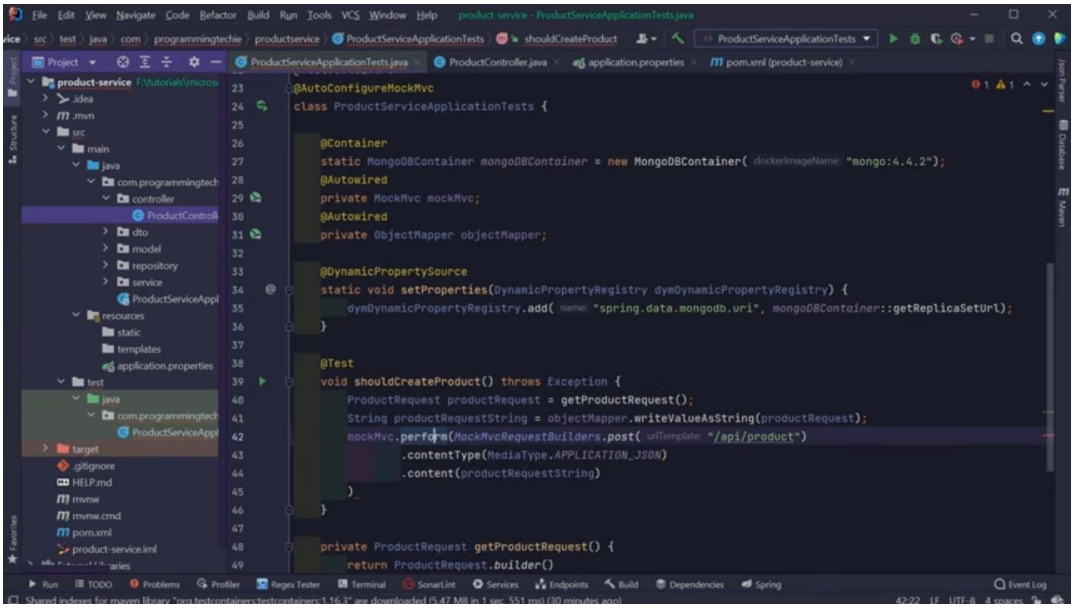
38:20


@AutoConfigureMockMvc
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@Autowired
private MockMvc mockMvc;
@Autowired
private ObjectMapper objectMapper;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() throws JsonProcessingException {
ProductRequest productRequest = getProductRequest();
objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()

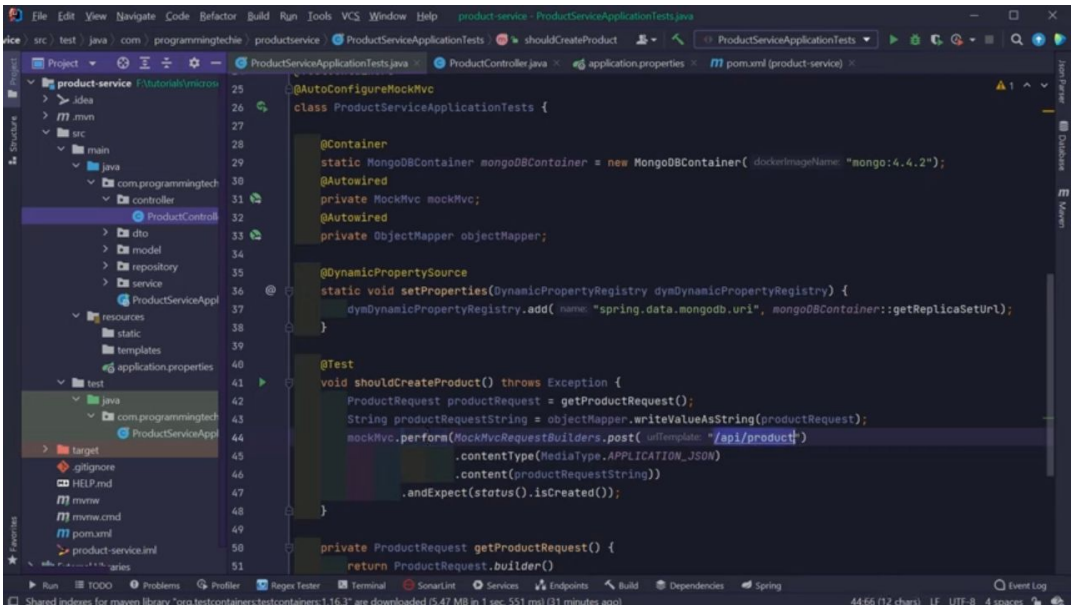
39:19


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - ProductServiceApplicationTests.java
@AutoConfigureMockMvc
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@Autowired
private MockMvc mockMvc;
@Autowired
private ObjectMapper objectMapper;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() throws JsonProcessingException {
ProductRequest productRequest = getProductRequest();
String productRequestString = objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content()
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()

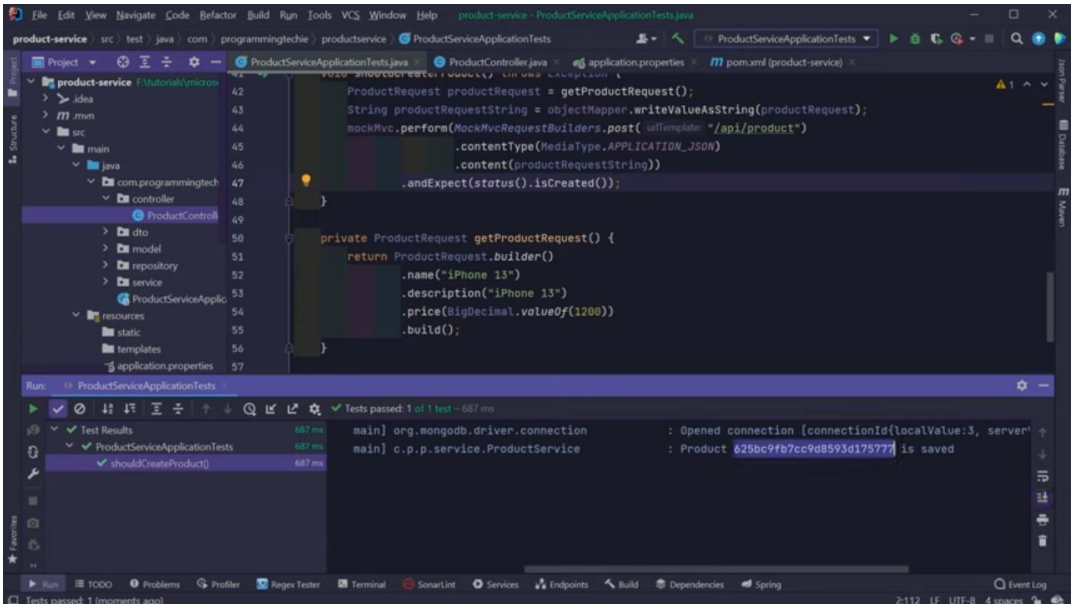
39:29


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
product-service - ProductServiceApplicationTests.java
@AutoConfigureMockMvc
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@Autowired
private MockMvc mockMvc;
@Autowired
private ObjectMapper objectMapper;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() throws Exception {
ProductRequest productRequest = getProductRequest();
String productRequestString = objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content(productRequestString)
)
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()

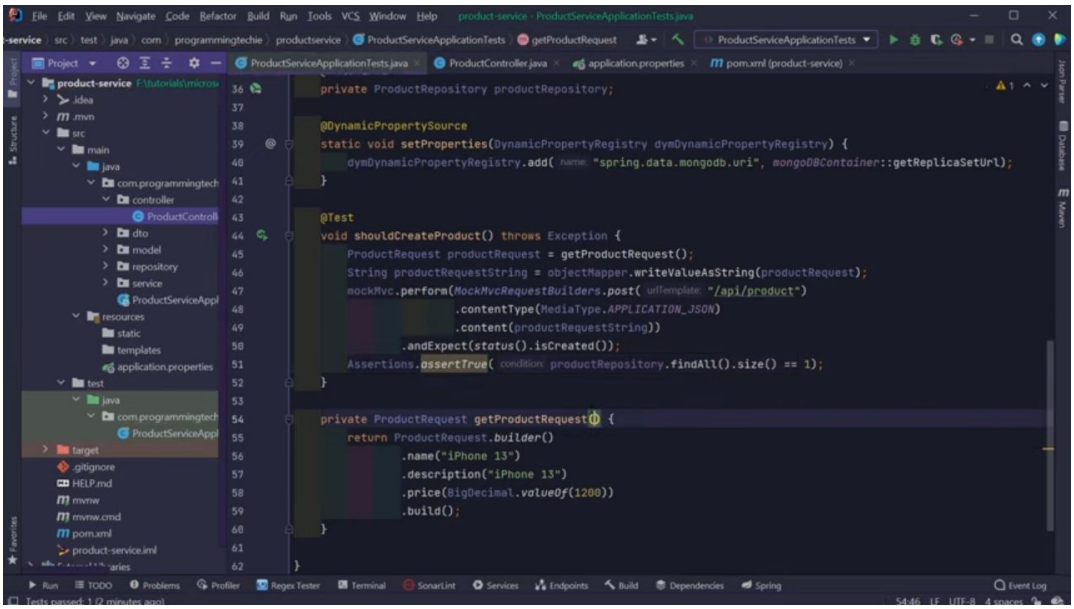
39:39


@AutoConfigureMockMvc
class ProductServiceApplicationTests {
@Container
static MongoDBContainer mongoDBContainer = new MongoDBContainer(dockerImageName: "mongo:4.4.2");
@Autowired
private MockMvc mockMvc;
@Autowired
private ObjectMapper objectMapper;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() throws Exception {
ProductRequest productRequest = getProductRequest();
String productRequestString = objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content(productRequestString))
.andExpect(status().isCreated());
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()

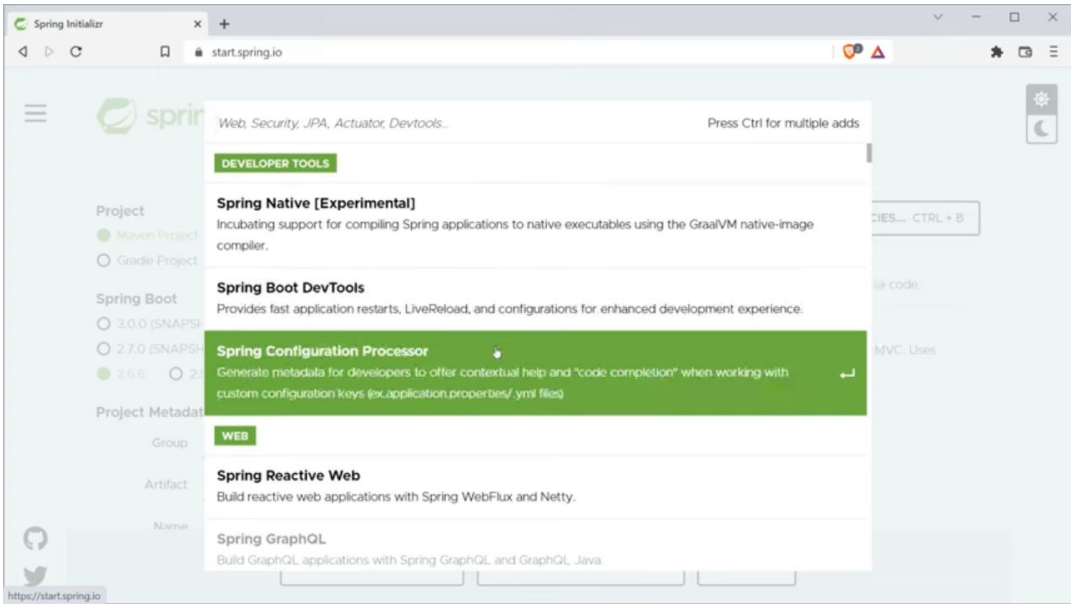
40:28


ProductRequest productRequest = getProductRequest();
String productRequestString = objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content(productRequestString))
.andExpect(status().isCreated());
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()
.name("iPhone 13")
.description("iPhone 13")
.price(BigDecimal.valueOf(1200))
.build();
}
Tests passed: 1 of 1 test – 687 ms
main] org.mongodb.driver.connection : Opened connection [connectionId{localValue:3, server
main] c.p.p.service.ProductService : Product 625bc9fb7cc9d8593d175777 is saved

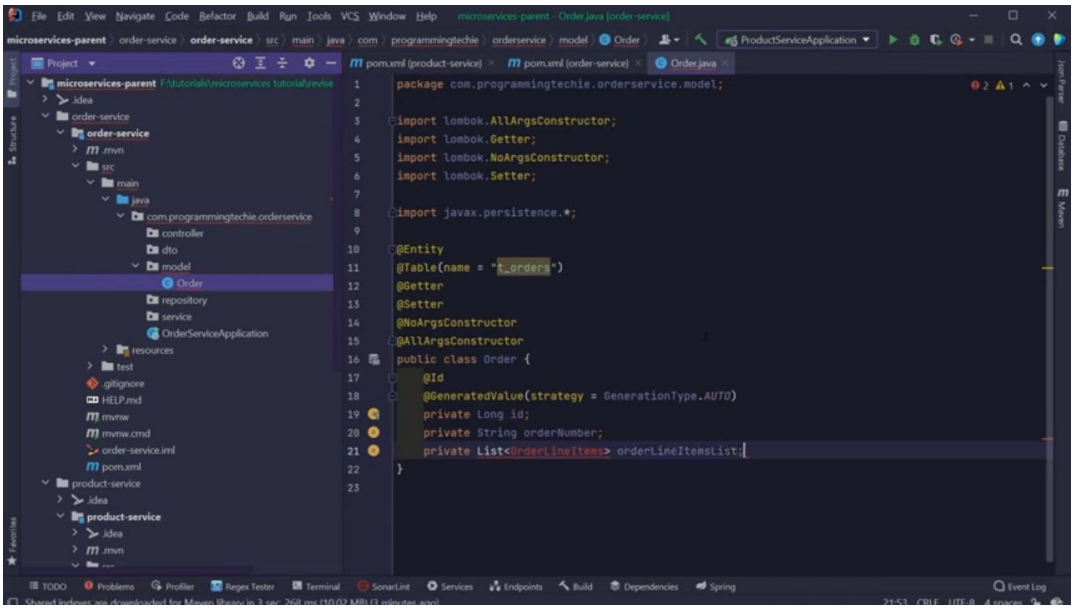
41:25


private ProductRepository productRepository;
@DynamicPropertySource
static void setProperties(DynamicPropertyRegistry dymDynamicPropertyRegistry) {
dymDynamicPropertyRegistry.add(name: "spring.data.mongodb.uri", mongoDBContainer::getReplicaSetUrl);
}
@Test
void shouldCreateProduct() throws Exception {
ProductRequest productRequest = getProductRequest();
String productRequestString = objectMapper.writeValueAsString(productRequest);
mockMvc.perform(MockMvcRequestBuilders.post(urlTemplate: "/api/product")
.contentType(MediaType.APPLICATION_JSON)
.content(productRequestString))
.andExpect(status().isCreated());
Assertions.assertTrue(condition: productRepository.findAll().size() == 1);
}
private ProductRequest getProductRequest() {
return ProductRequest.builder()
.name("iPhone 13")
.description("iPhone 13")
.price(BigDecimal.valueOf(1200))
.build();
}

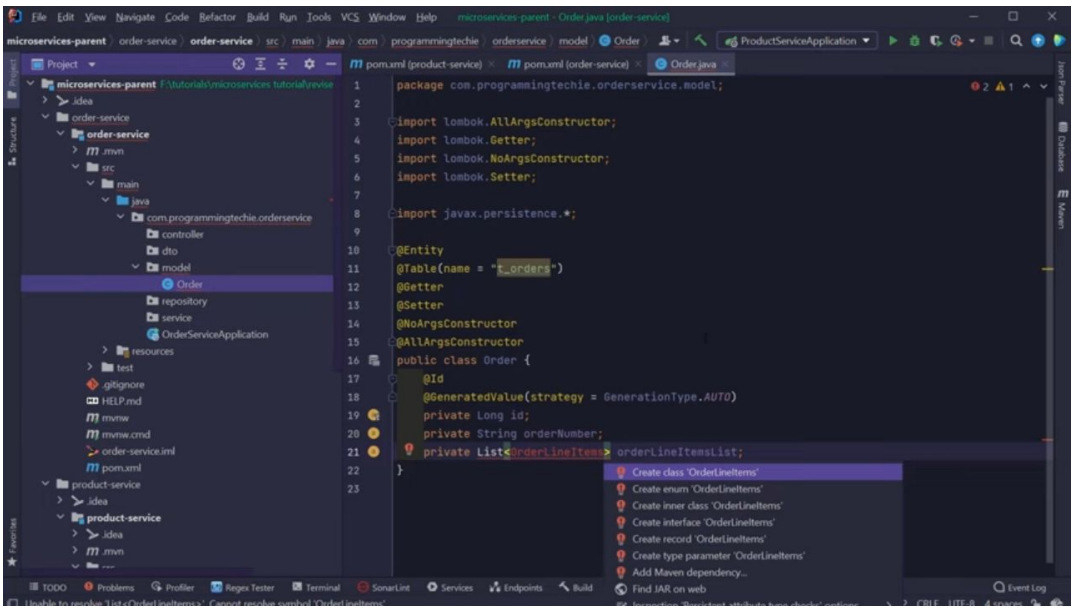
42:34


Spring Initializr
start.spring.io
Web, Security, JPA, Actuator, Devtools...
Press Ctrl for multiple adds
DEVELOPER TOOLS
Spring Native [Experimental]
Incubating support for compiling Spring applications to native executables using the GraalVM native-image compiler.
Spring Boot DevTools
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
Spring Configuration Processor
Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files)
WEB
Spring Reactive Web
Build reactive web applications with Spring WebFlux and Netty.
Spring GraphQL
Build GraphQL applications with Spring GraphQL and GraphQL Java.
Project
Spring Boot
Group
Artifact
Name
https://start.spring.io

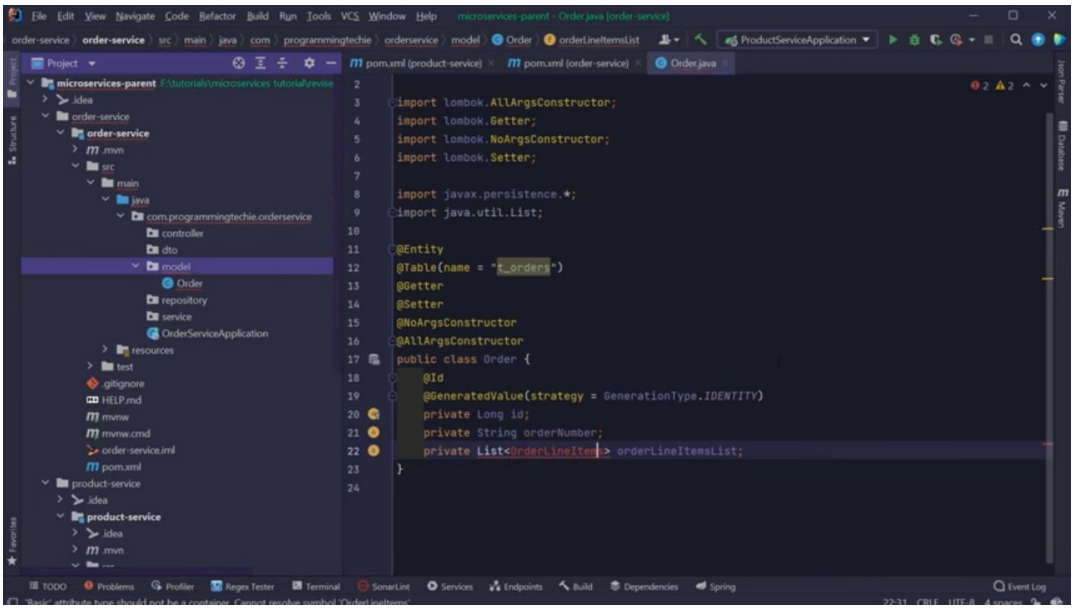
45:30


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - Order.java [order-service]
ProductServiceApplication
pom.xml (product-service)
pom.xml (order-service)
Order.java
package com.programmingtechie.orderservice.model;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
@Entity
@Table(name = "t_orders")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Order {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private Long id;
private String orderNumber;
private List<OrderLineItems> orderLineItemsList;
}

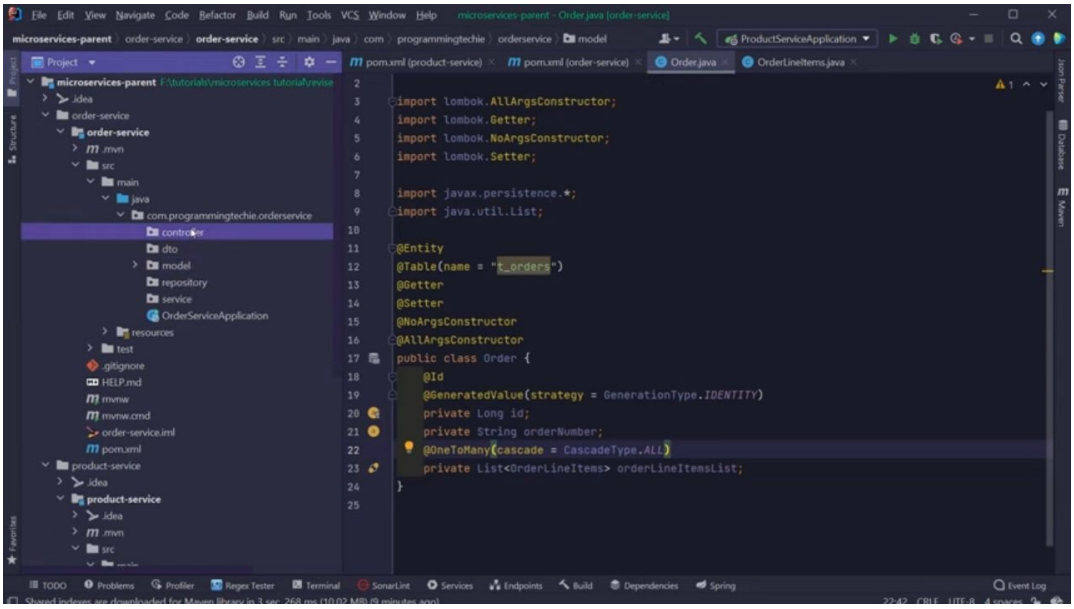
49:32


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - Order.java [order-service]
ProductServiceApplication
pom.xml (product-service)
pom.xml (order-service)
Order.java
package com.programmingtechie.orderservice.model;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
@Entity
@Table(name = "t_orders")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Order {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
private Long id;
private String orderNumber;
private List<OrderLineItems> orderLineItemsList;
}
Create class 'OrderLineItems'
Create enum 'OrderLineItems'
Create inner class 'OrderLineItems'
Create interface 'OrderLineItems'
Create record 'OrderLineItems'
Create type parameter 'OrderLineItems'
Add Maven dependency...
Find JAR on web

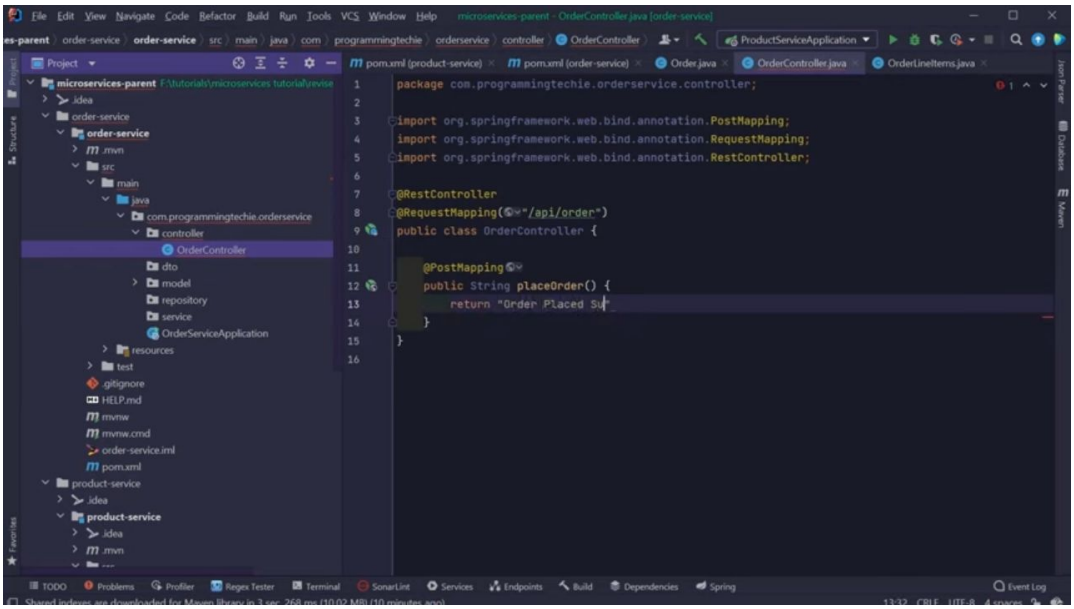
49:49


import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
import java.util.List;
@Entity
@Table(name = "t_orders")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Order {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
private String orderNumber;
private List<OrderLineItem> orderLineItemsList;
}

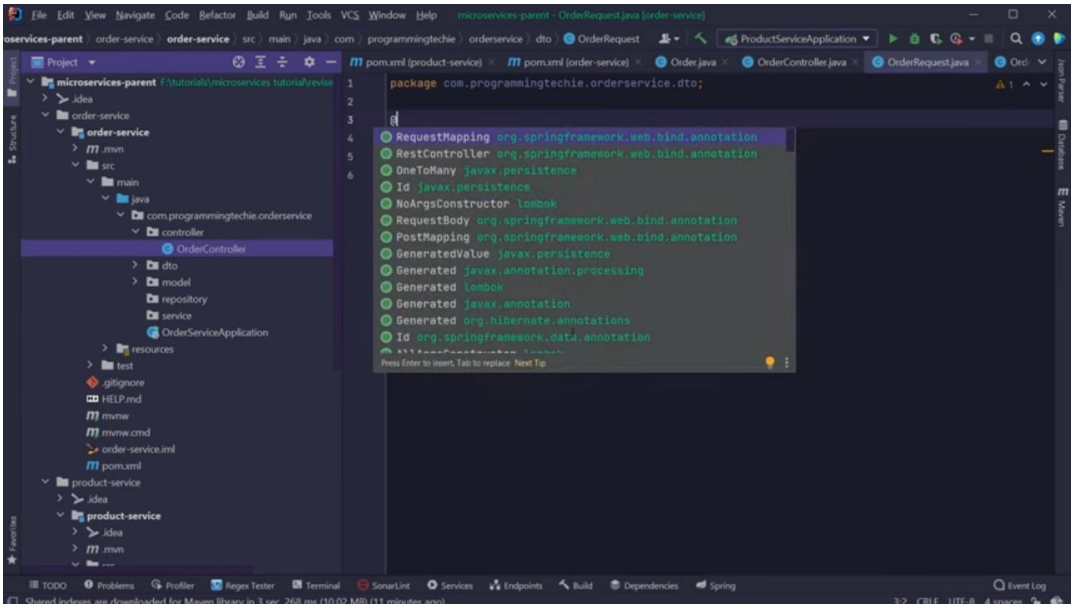
51:24


import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
import java.util.List;
@Entity
@Table(name = "t_orders")
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Order {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
private String orderNumber;
@OneToMany(cascade = CascadeType.ALL)
private List<OrderLineItems> orderLineItemsList;
}

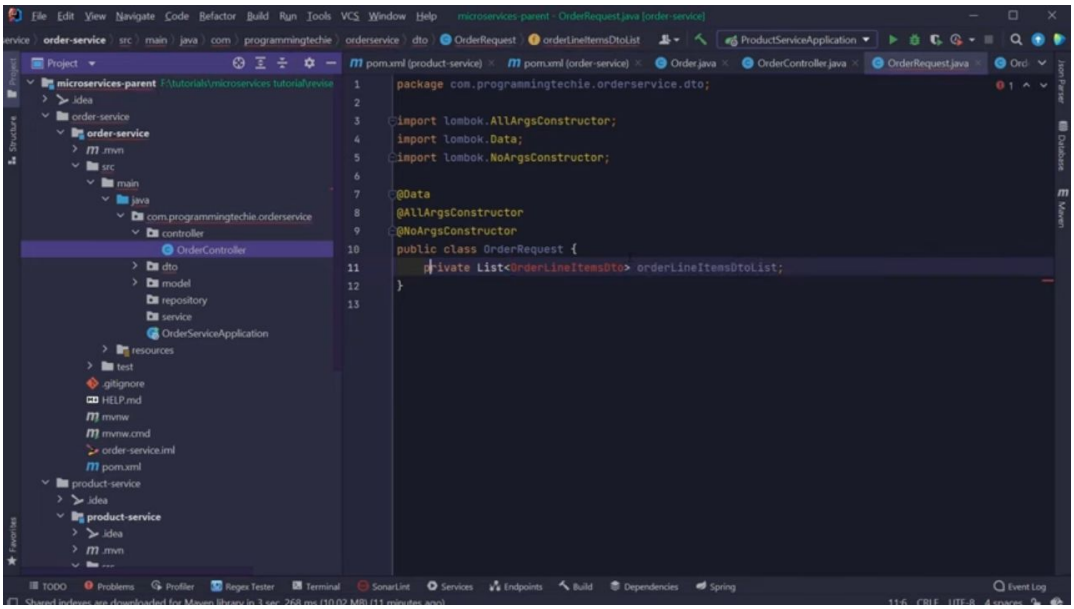
53:22


microservices-parent - OrderController.java [order-service]
package com.programmingtechie.orderservice.controller;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
@RequestMapping("/api/order")
public class OrderController {
@PostMapping
public String placeOrder() {
return "Order Placed Su"
}
}

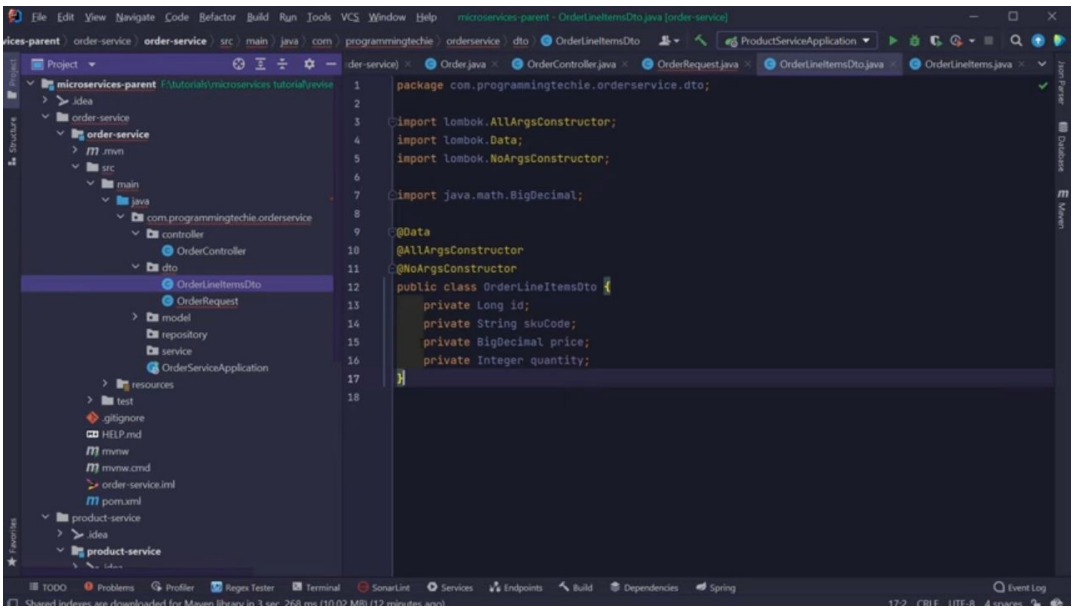
54:06


microservices-parent - OrderRequest.java [order-service]
package com.programmingtechie.orderservice.dto;
@
RequestMapping org.springframework.web.bind.annotation
RestController org.springframework.web.bind.annotation
OneToMany javax.persistence
Id javax.persistence
NoArgsConstructor lombok
RequestBody org.springframework.web.bind.annotation
PostMapping org.springframework.web.bind.annotation
GeneratedValue javax.persistence
Generated javax.annotation.processing
Generated lombok
Generated javax.annotation
Generated org.hibernate.annotations
Id org.springframework.data.annotation
Press Enter to insert, Tab to replace Next Tip

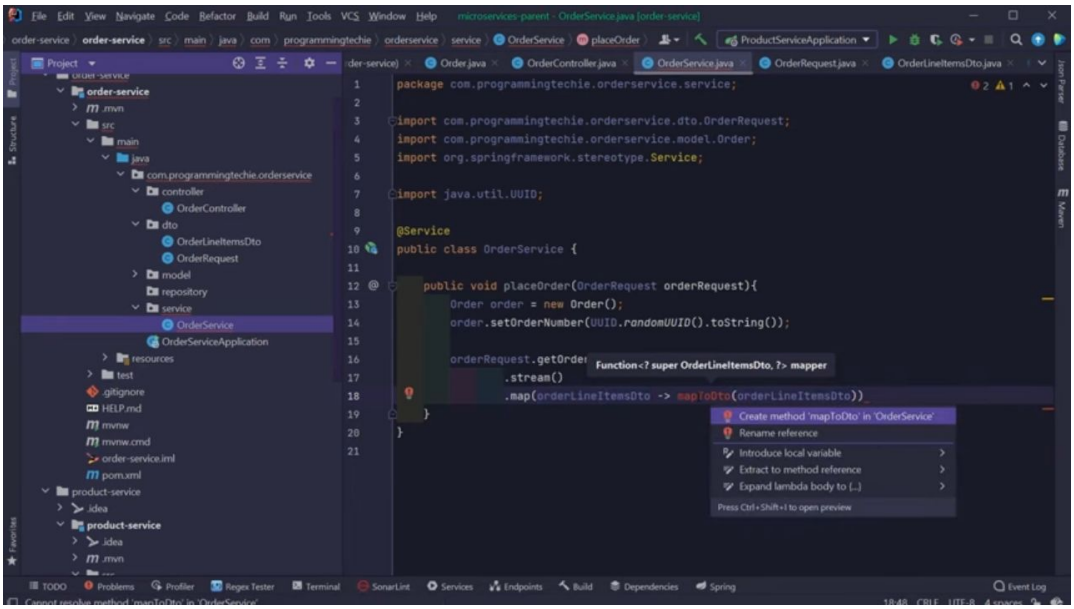
54:30


microservices-parent - OrderRequest.java [order-service]
pom.xml (product-service)
pom.xml (order-service)
Order.java
OrderController.java
OrderRequest.java
package com.programmingtechie.orderservice.dto;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
@Data
@AllArgsConstructor
@NoArgsConstructor
public class OrderRequest {
private List<OrderLineItemsDto> orderLineItemsDtoList;
}

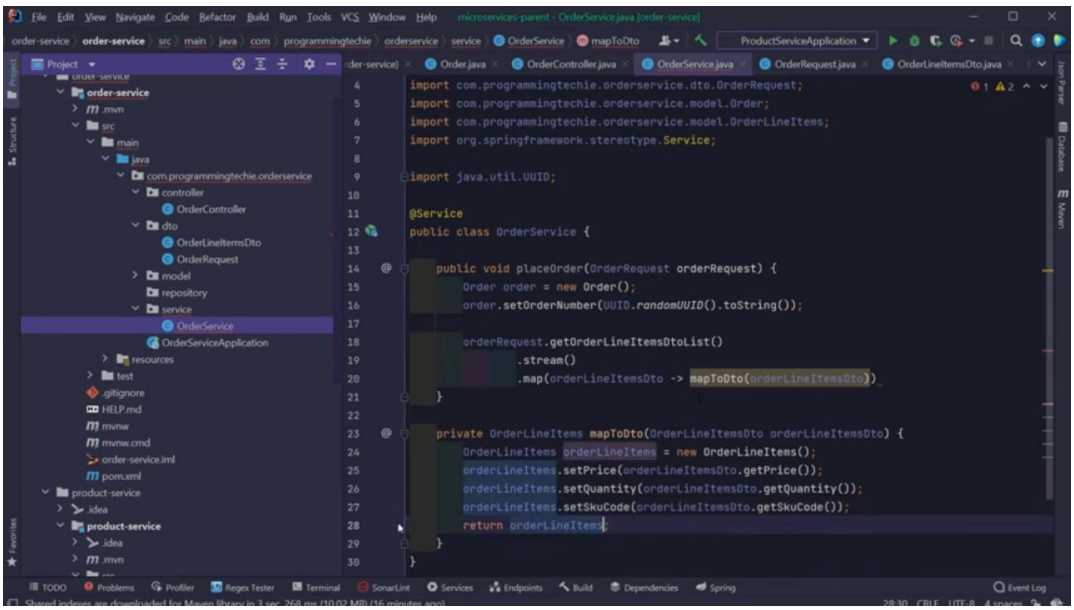
54:59


microservices-parent - OrderLineItemsDto.java [order-service]
Order.java
OrderController.java
OrderRequest.java
OrderLineItemsDto.java
OrderLineItems.java
package com.programmingtechie.orderservice.dto;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.math.BigDecimal;
@Data
@AllArgsConstructor
@NoArgsConstructor
public class OrderLineItemsDto {
private Long id;
private String skuCode;
private BigDecimal price;
private Integer quantity;
}

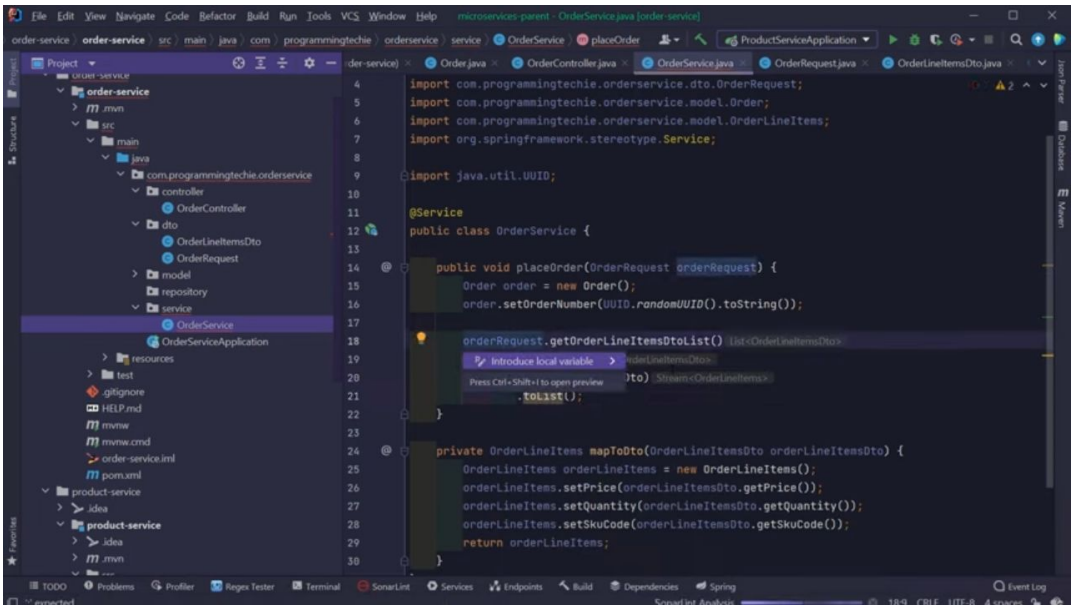
55:40


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderService.java [order-service]
order-service order-service src main java com programmingtechie orderservice service OrderService placeOrder
ProductServiceApplication
Project
order-service
.mvn
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
.idea
.mvn
Order.java
OrderController.java
OrderService.java
OrderRequest.java
OrderLineItemsDto.java
package com.programmingtechie.orderservice.service;
import com.programmingtechie.orderservice.dto.OrderRequest;
import com.programmingtechie.orderservice.model.Order;
import org.springframework.stereotype.Service;
import java.util.UUID;
@Service
public class OrderService {
public void placeOrder(OrderRequest orderRequest){
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
orderRequest.getOrder
Function<? super OrderLineItemsDto, ?> mapper
.stream()
.map(orderLineItemsDto -> mapToDto(orderLineItemsDto))
}
}
Create method 'mapToDto' in 'OrderService'
Rename reference
Introduce local variable
Extract to method reference
Expand lambda body to {...}
Press Ctrl+Shift+I to open preview
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

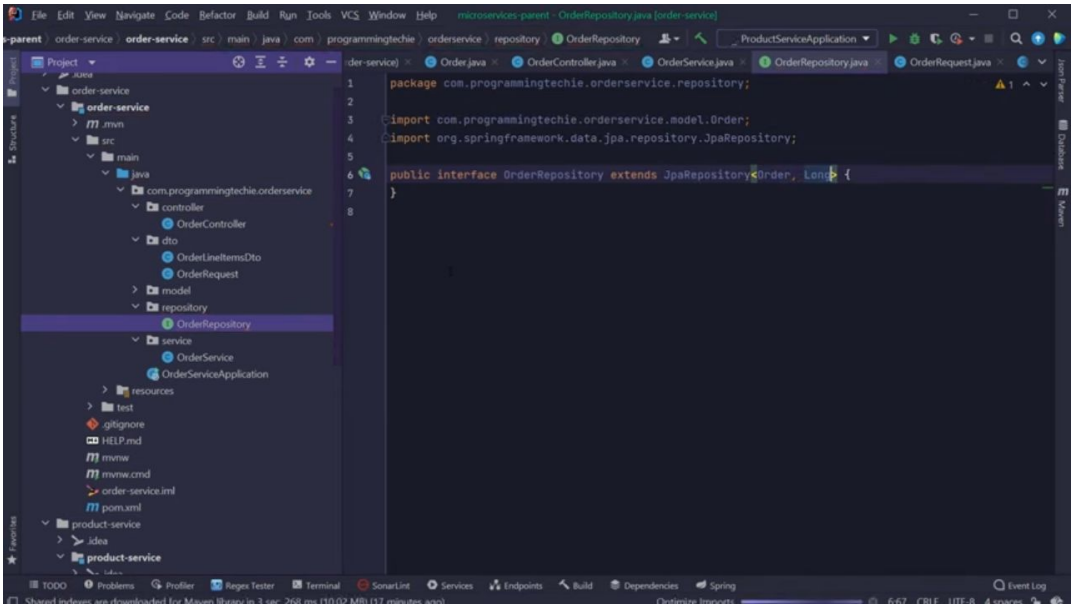
57:35


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderService.java [order-service]
order-service order-service src main java com programmingtechie orderservice service OrderService mapToDto
ProductServiceApplication
Project
order-service
.mvn
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
.idea
.mvn
Order.java
OrderController.java
OrderService.java
OrderRequest.java
OrderLineItemsDto.java
import com.programmingtechie.orderservice.dto.OrderRequest;
import com.programmingtechie.orderservice.model.Order;
import com.programmingtechie.orderservice.model.OrderLineItems;
import org.springframework.stereotype.Service;
import java.util.UUID;
@Service
public class OrderService {
public void placeOrder(OrderRequest orderRequest) {
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
orderRequest.getOrderLineItemsDtoList()
.stream()
.map(orderLineItemsDto -> mapToDto(orderLineItemsDto))
}
private OrderLineItems mapToDto(OrderLineItemsDto orderLineItemsDto) {
OrderLineItems orderLineItems = new OrderLineItems();
orderLineItems.setPrice(orderLineItemsDto.getPrice());
orderLineItems.setQuantity(orderLineItemsDto.getQuantity());
orderLineItems.setSkuCode(orderLineItemsDto.getSkuCode());
return orderLineItems;
}
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

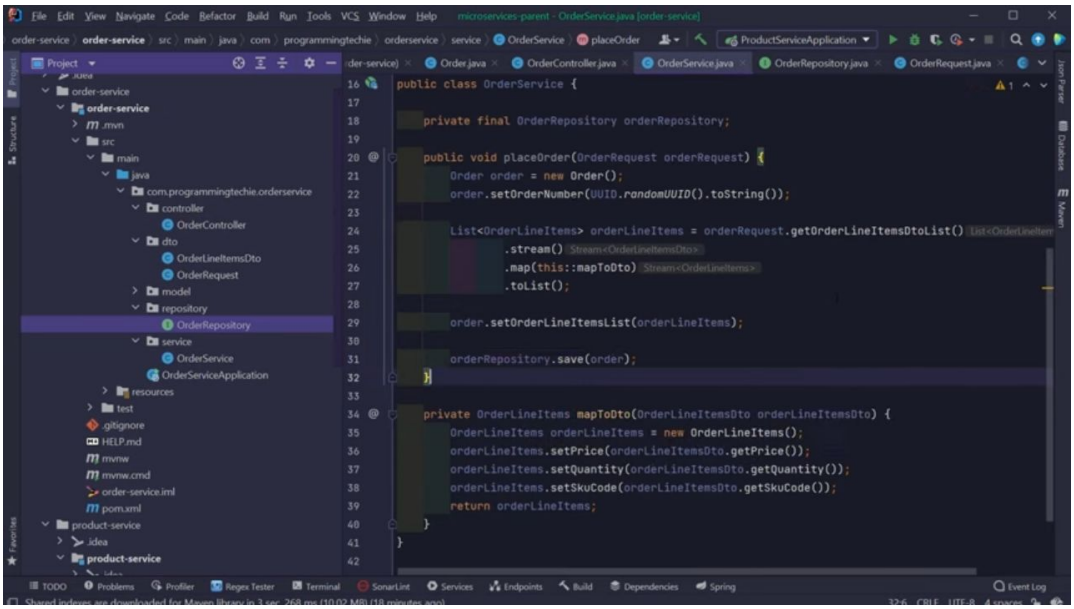
58:12


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderService.java [order-service]
order-service > order-service > src > main > java > com > programmingtechie > orderservice > service > OrderService > placeOrder
ProductServiceApplication
Project
order-service
.mvn
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
.idea
.mvn
Order.java
OrderController.java
OrderService.java
OrderRequest.java
OrderLineItemsDto.java
import com.programmingtechie.orderservice.dto.OrderRequest;
import com.programmingtechie.orderservice.model.Order;
import com.programmingtechie.orderservice.model.OrderLineItems;
import org.springframework.stereotype.Service;
import java.util.UUID;
@Service
public class OrderService {
public void placeOrder(OrderRequest orderRequest) {
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
orderRequest.getOrderLineItemsDtoList()
Introduce local variable
Press Ctrl+Shift+I to open preview
.toList();
}
private OrderLineItems mapToDto(OrderLineItemsDto orderLineItemsDto) {
OrderLineItems orderLineItems = new OrderLineItems();
orderLineItems.setPrice(orderLineItemsDto.getPrice());
orderLineItems.setQuantity(orderLineItemsDto.getQuantity());
orderLineItems.setSkuCode(orderLineItemsDto.getSkuCode());
return orderLineItems;
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

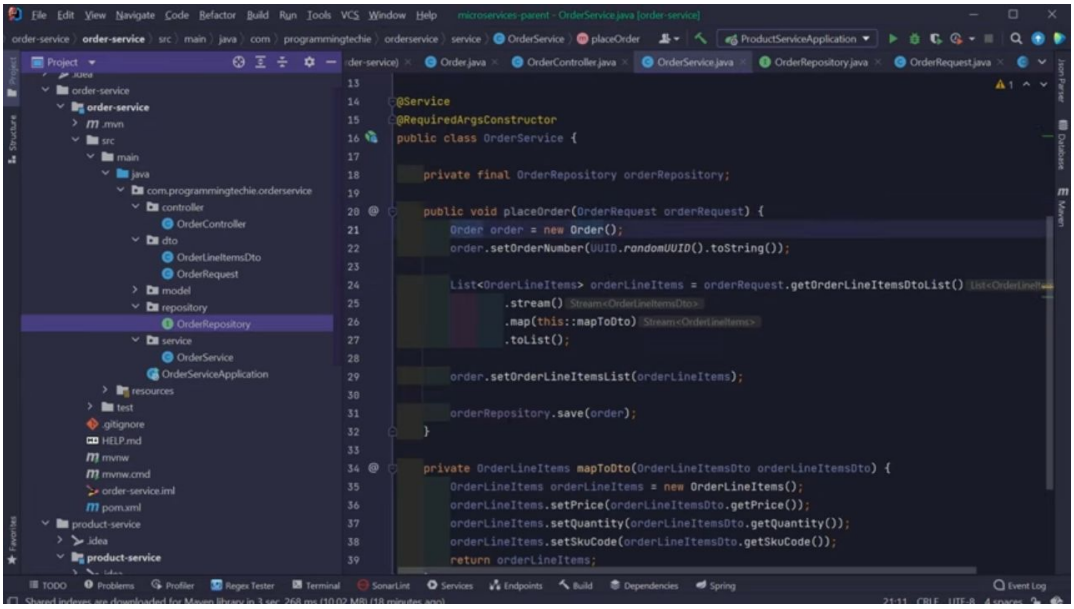
58:29


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderRepository.java [order-service]
order-service > order-service > src > main > java > com > programmingtechie > orderservice > repository > OrderRepository
ProductServiceApplication
Project
order-service
.mvn
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
OrderRepository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
Order.java
OrderController.java
OrderService.java
OrderRepository.java
OrderRequest.java
package com.programmingtechie.orderservice.repository;
import com.programmingtechie.orderservice.model.Order;
import org.springframework.data.jpa.repository.JpaRepository;
public interface OrderRepository extends JpaRepository<Order, Long> {
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

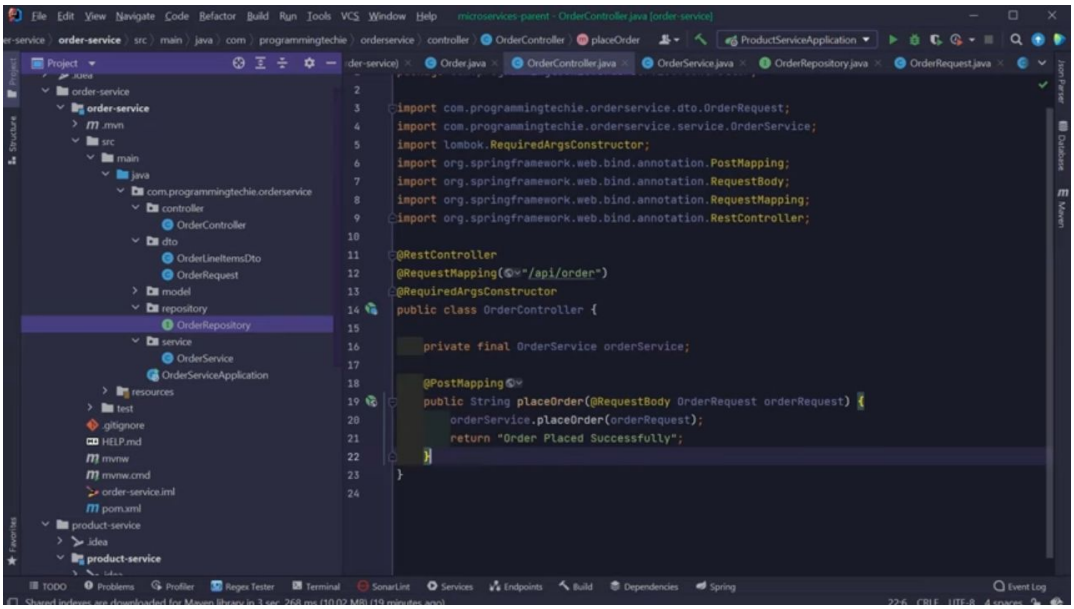
59:21


public class OrderService {
private final OrderRepository orderRepository;
public void placeOrder(OrderRequest orderRequest) {
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
List<OrderLineItems> orderLineItems = orderRequest.getOrderLineItemsDtoList()
.stream()
.map(this::mapToDto)
.toList();
order.setOrderLineItemsList(orderLineItems);
orderRepository.save(order);
}
private OrderLineItems mapToDto(OrderLineItemsDto orderLineItemsDto) {
OrderLineItems orderLineItems = new OrderLineItems();
orderLineItems.setPrice(orderLineItemsDto.getPrice());
orderLineItems.setQuantity(orderLineItemsDto.getQuantity());
orderLineItems.setSkuCode(orderLineItemsDto.getSkuCode());
return orderLineItems;
}
}

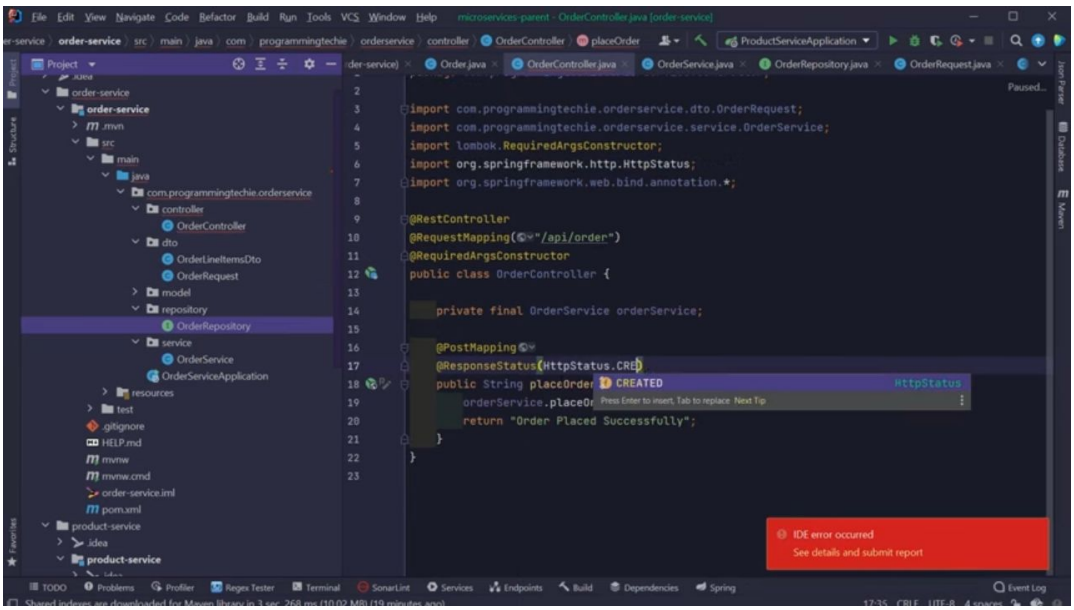
1:00:08


@Service
@RequiredArgsConstructor
public class OrderService {
private final OrderRepository orderRepository;
public void placeOrder(OrderRequest orderRequest) {
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
List<OrderLineItems> orderLineItems = orderRequest.getOrderLineItemsDtoList()
.stream()
.map(this::mapToDto)
.toList();
order.setOrderLineItemsList(orderLineItems);
orderRepository.save(order);
}
private OrderLineItems mapToDto(OrderLineItemsDto orderLineItemsDto) {
OrderLineItems orderLineItems = new OrderLineItems();
orderLineItems.setPrice(orderLineItemsDto.getPrice());
orderLineItems.setQuantity(orderLineItemsDto.getQuantity());
orderLineItems.setSkuCode(orderLineItemsDto.getSkuCode());
return orderLineItems;

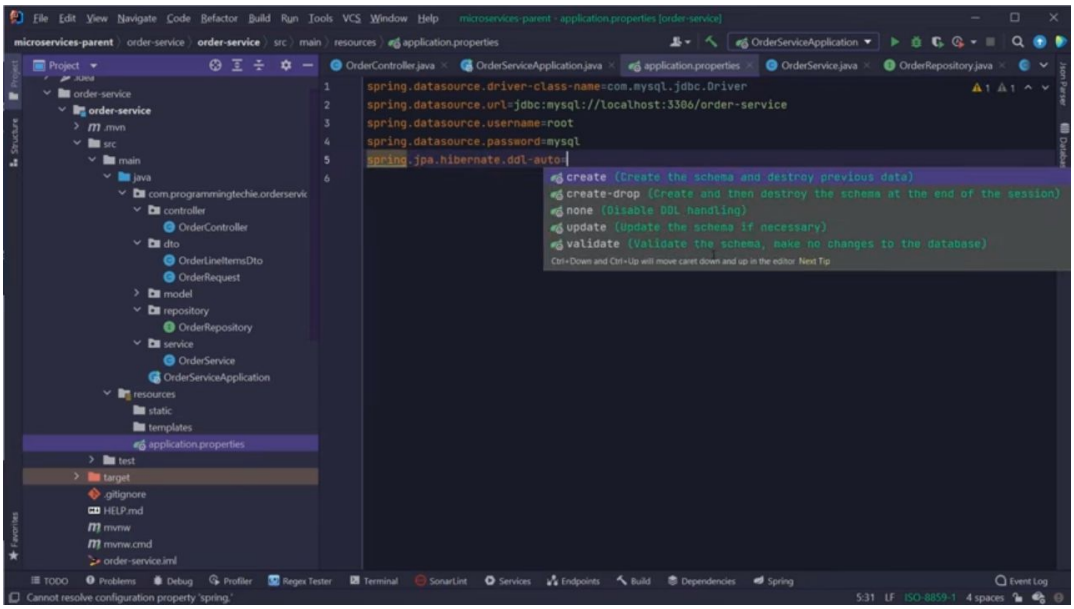
1:00:23


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderController.java [order-service]
ProductServiceApplication
Project
order-service
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
OrderRepository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
Order.java
OrderController.java
OrderService.java
OrderRepository.java
OrderRequest.java
import com.programmingtechie.orderservice.dto.OrderRequest;
import com.programmingtechie.orderservice.service.OrderService;
import lombok.RequiredArgsConstructor;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
@RequestMapping("/api/order")
@RequiredArgsConstructor
public class OrderController {
private final OrderService orderService;
@PostMapping
public String placeOrder(@RequestBody OrderRequest orderRequest) {
orderService.placeOrder(orderRequest);
return "Order Placed Successfully";
}
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

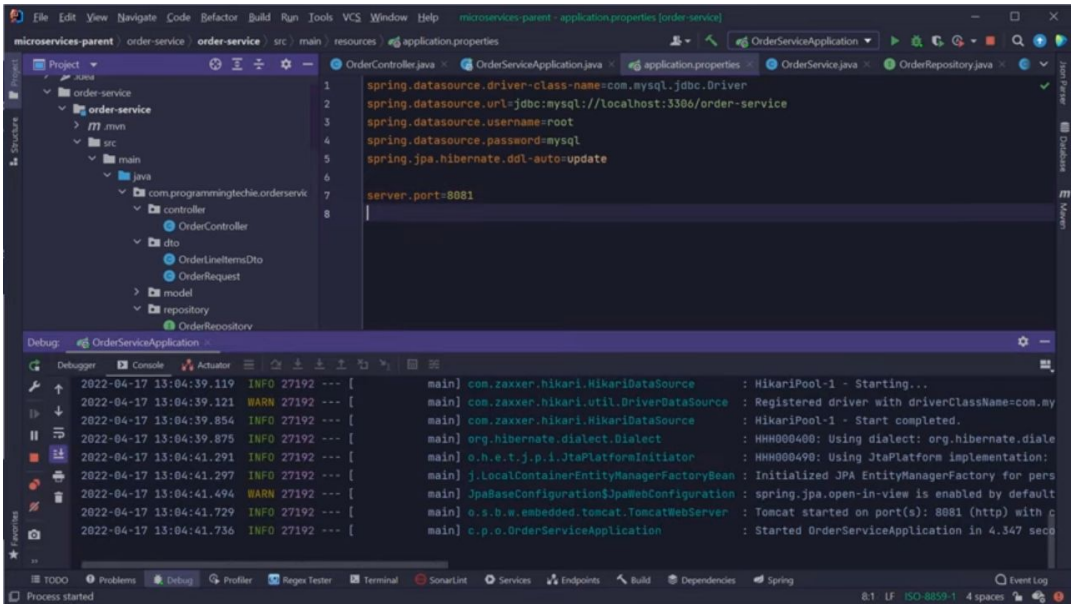
▷ 1:01:06


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent - OrderController.java [order-service]
ProductServiceApplication
Project
order-service
src
main
java
com.programmingtechie.orderservice
controller
OrderController
dto
OrderLineItemsDto
OrderRequest
model
repository
OrderRepository
service
OrderService
OrderServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
order-service.iml
pom.xml
product-service
.idea
product-service
Order.java
OrderController.java
OrderService.java
OrderRepository.java
OrderRequest.java
Paused...
import com.programmingtechie.orderservice.dto.OrderRequest;
import com.programmingtechie.orderservice.service.OrderService;
import lombok.RequiredArgsConstructor;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.*;
@RestController
@RequestMapping("/api/order")
@RequiredArgsConstructor
public class OrderController {
private final OrderService orderService;
@PostMapping
@ResponseStatus(HttpStatus.CRE
public String placeOrder
CREATED
HttpStatus
Press Enter to insert, Tab to replace Next Tip
orderService.placeOr
return "Order Placed Successfully";
}
}
IDE error occurred
See details and submit report
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Endpoints
Build
Dependencies
Spring
Event Log

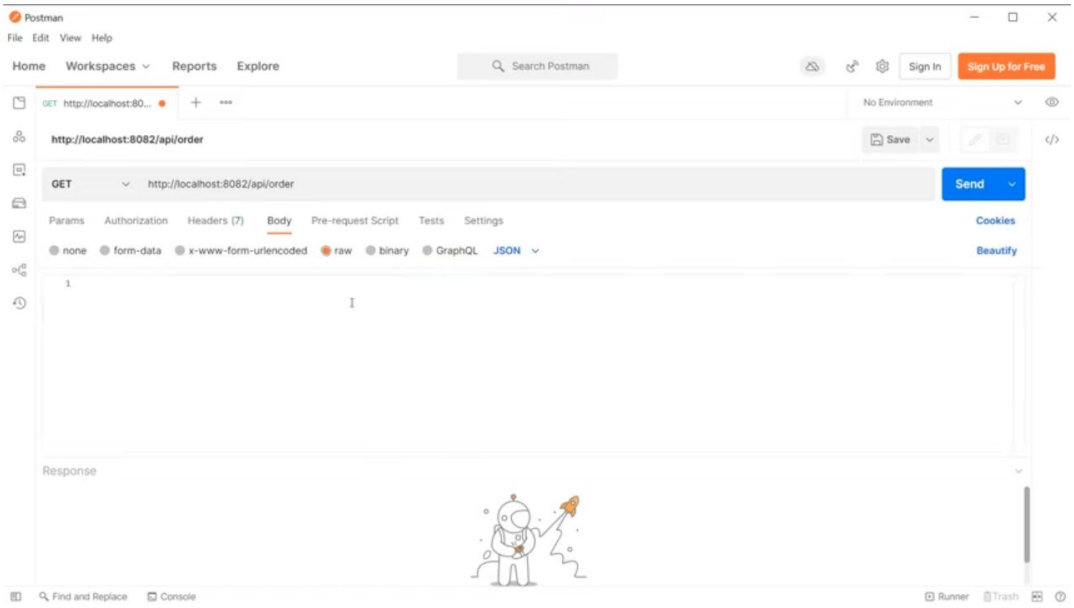
▷ 1:01:17


spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/order-service
spring.datasource.username=root
spring.datasource.password=mysql
spring.jpa.hibernate.ddl-auto=
create (Create the schema and destroy previous data)
create-drop (Create and then destroy the schema at the end of the session)
none (Disable DDL handling)
update (Update the schema if necessary)
validate (Validate the schema, make no changes to the database)

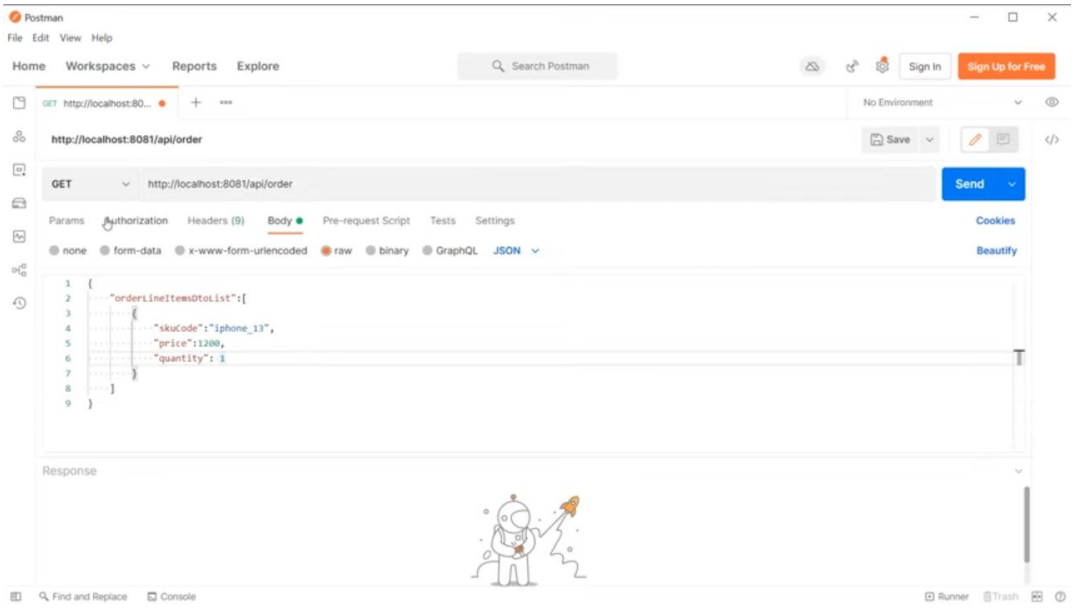
1:02:44


spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/order-service
spring.datasource.username=root
spring.datasource.password=mysql
spring.jpa.hibernate.ddl-auto=update
server.port=8081
Started OrderServiceApplication in 4.347 seco

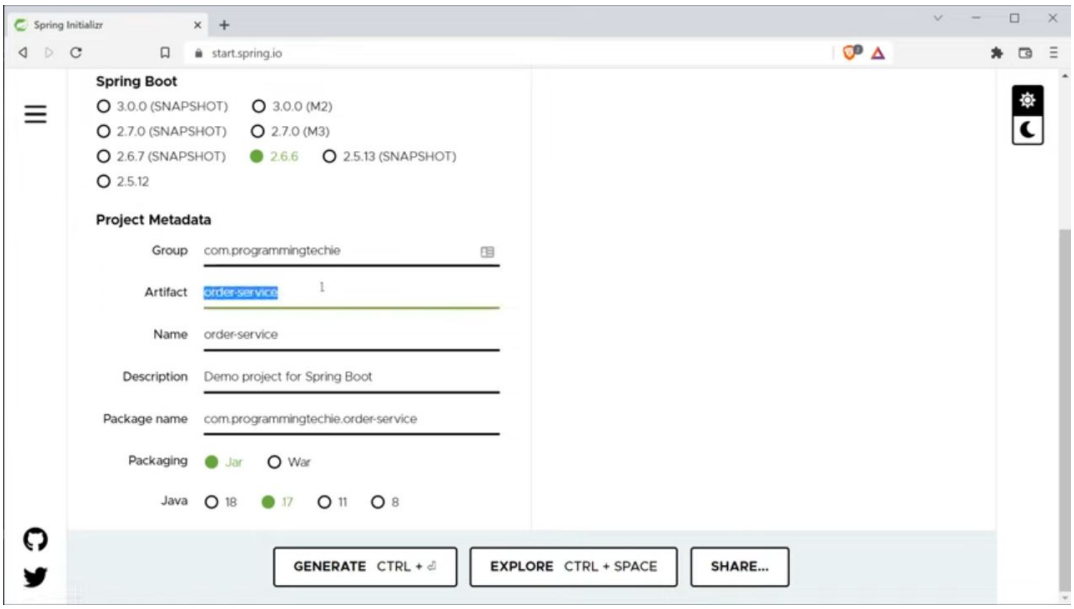
1:03:04


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Sign Up for Free
No Environment
http://localhost:8082/api/order
Save
GET
http://localhost:8082/api/order
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
Response
Find and Replace
Console
Runner
Trash

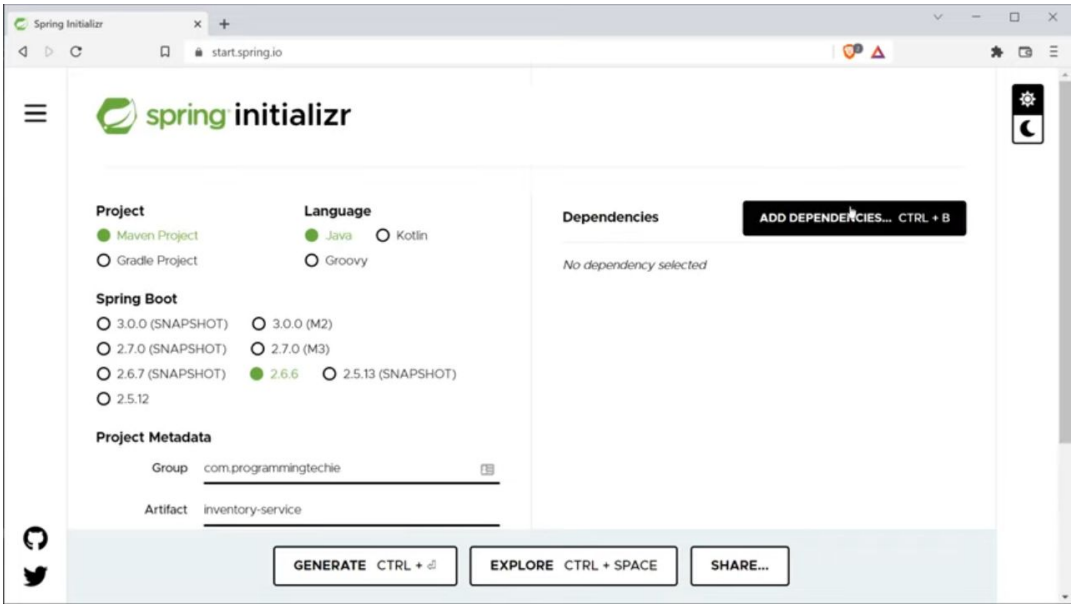
1:03:56


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Sign In
Sign Up for Free
No Environment
http://localhost:8081/api/order
Save
GET
http://localhost:8081/api/order
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"orderLineItemsDtoList":[
{
"skuCode":"iphone_13",
"price":1200,
"quantity": 1
}
]
}
Response
Find and Replace
Console
Runner
Trash

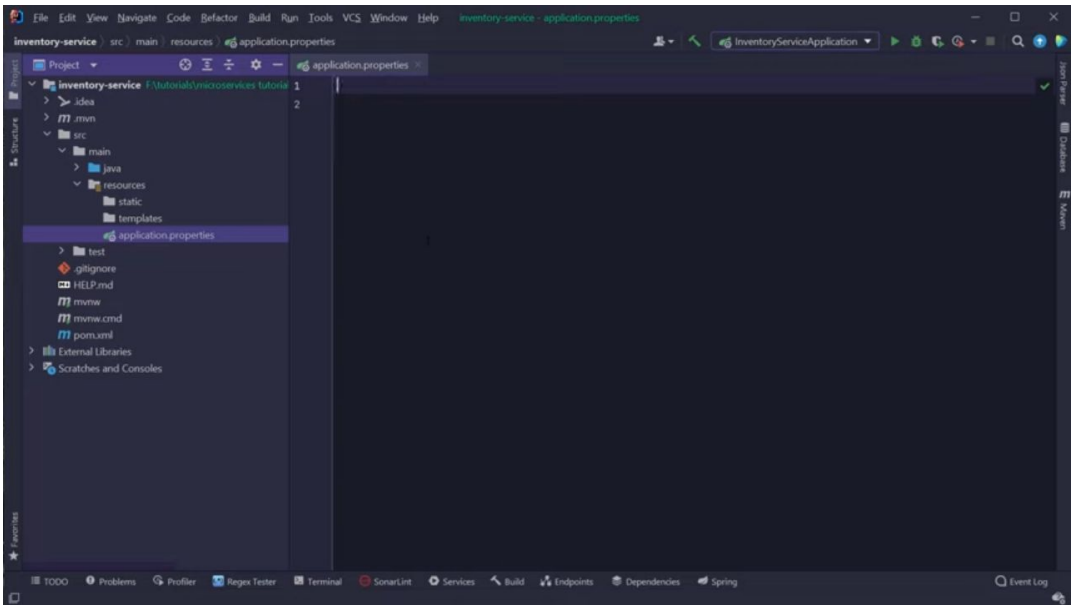
1:05:31


Spring Initializr
start.spring.io
Spring Boot
3.0.0 (SNAPSHOT)
3.0.0 (M2)
2.7.0 (SNAPSHOT)
2.7.0 (M3)
2.6.7 (SNAPSHOT)
2.6.6
2.5.13 (SNAPSHOT)
2.5.12
Project Metadata
Group
com.programmingtechie
Artifact
order-service
Name
order-service
Description
Demo project for Spring Boot
Package name
com.programmingtechie.order-service
Packaging
Jar
War
Java
18
17
11
8
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
SHARE...

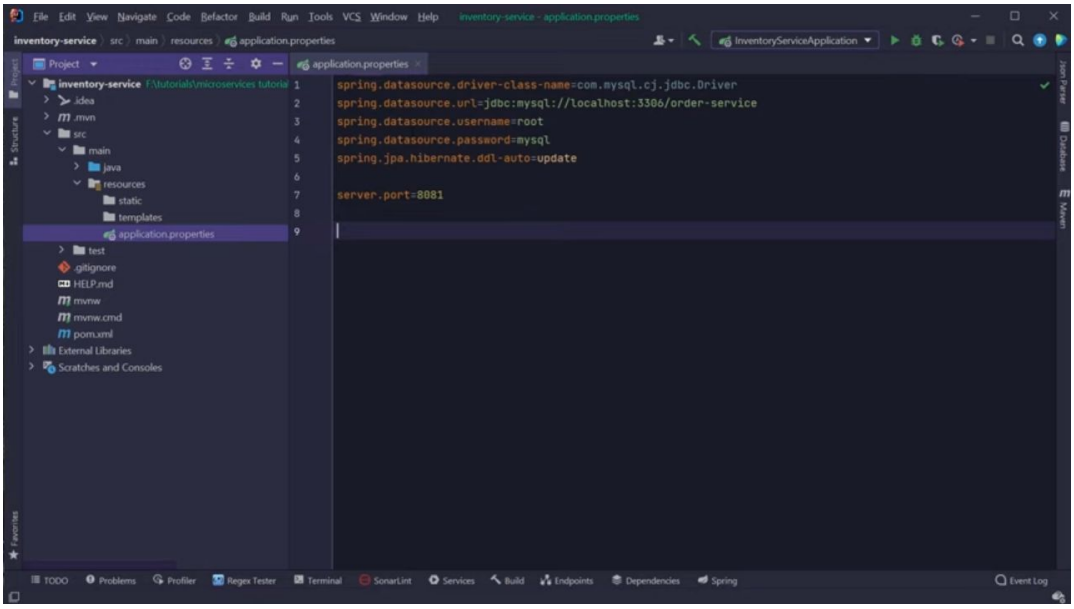
1:06:18


Spring Initializr
start.spring.io
spring initializr
Project
Maven Project
Gradle Project
Language
Java
Kotlin
Groovy
Dependencies
ADD DEPENDENCIES... CTRL + B
No dependency selected
Spring Boot
3.0.0 (SNAPSHOT)
3.0.0 (M2)
2.7.0 (SNAPSHOT)
2.7.0 (M3)
2.6.7 (SNAPSHOT)
2.6.6
2.5.13 (SNAPSHOT)
2.5.12
Project Metadata
Group
com.programmingtechie
Artifact
inventory-service
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
SHARE...

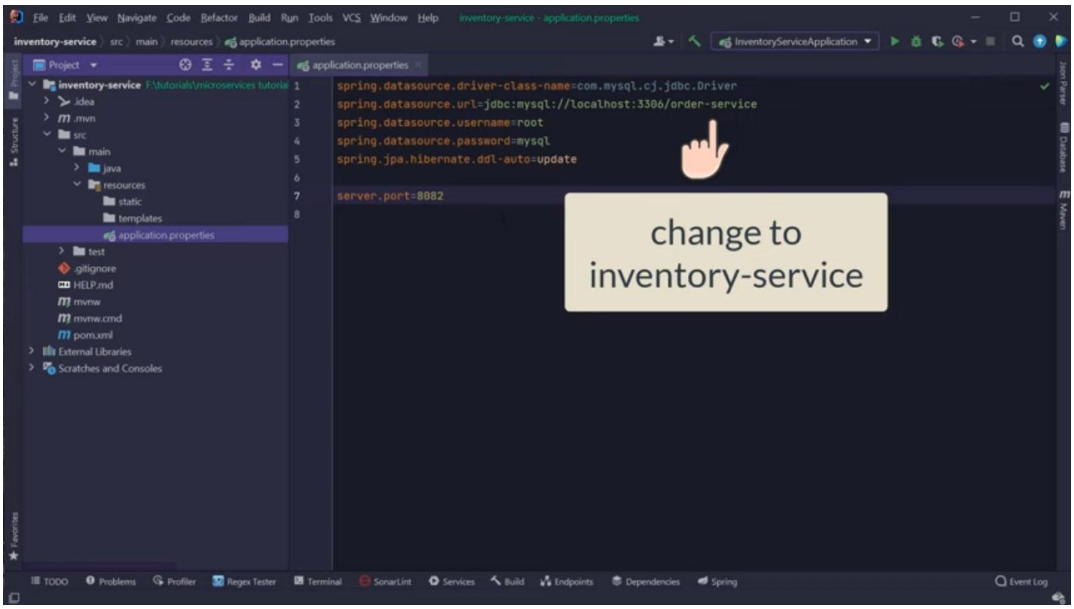
1:06:23


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - application.properties
inventory-service > src > main > resources > application.properties
InventoryServiceApplication
Project
application.properties
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

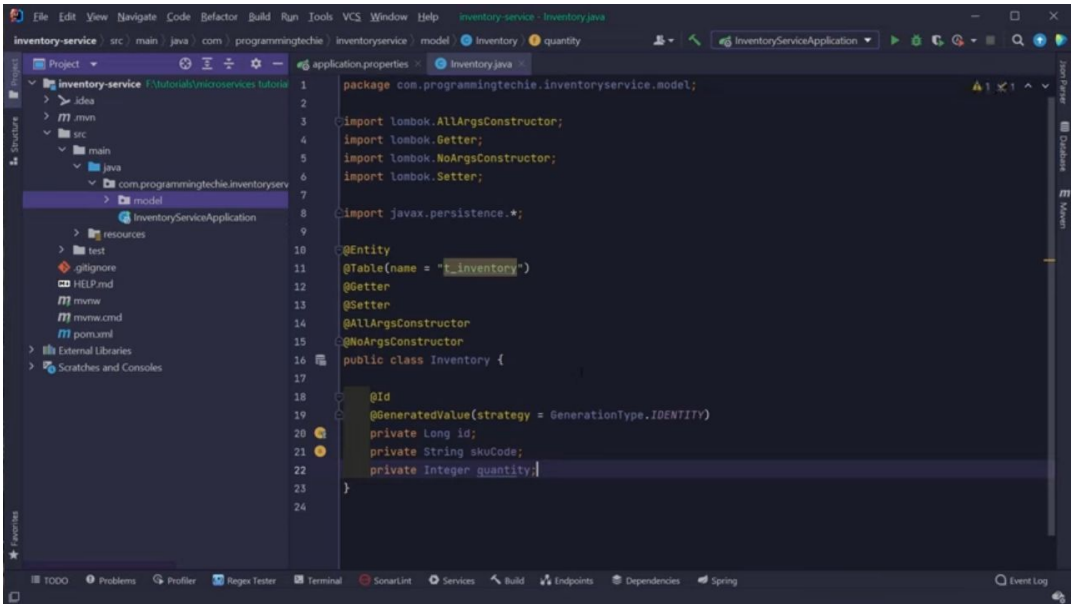
1:07:00


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - application.properties
inventory-service > src > main > resources > application.properties
InventoryServiceApplication
Project
application.properties
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
resources
static
templates
application.properties
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
2 spring.datasource.url=jdbc:mysql://localhost:3306/order-service
3 spring.datasource.username=root
4 spring.datasource.password=mysql
5 spring.jpa.hibernate.ddl-auto=update
6
7 server.port=8081
8
9
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

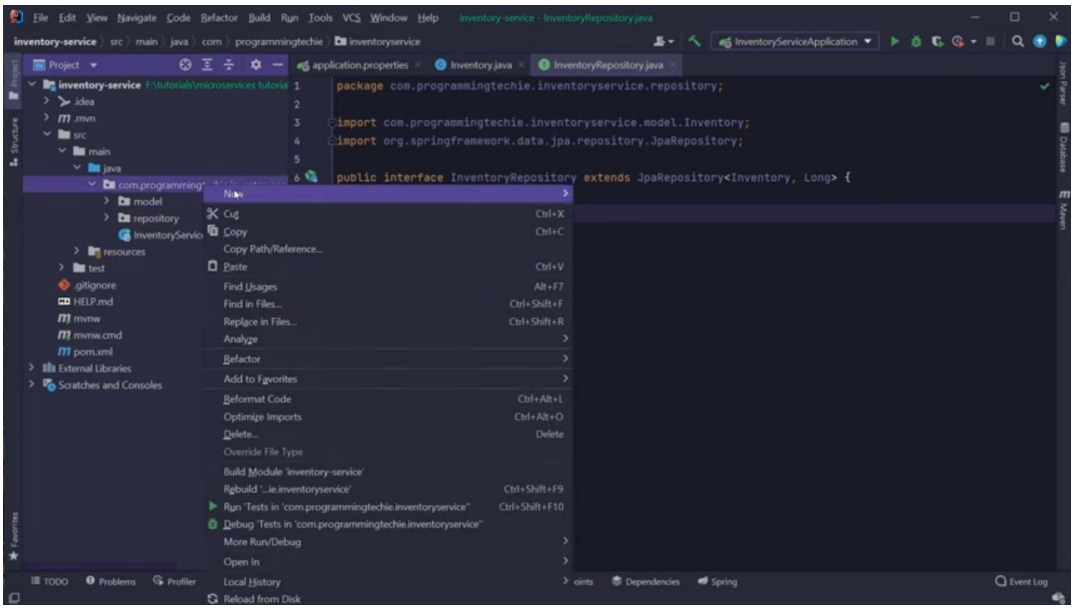
1:07:30


spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/order-service
spring.datasource.username=root
spring.datasource.password=mysql
spring.jpa.hibernate.ddl-auto=update
server.port=8082
change to
inventory-service

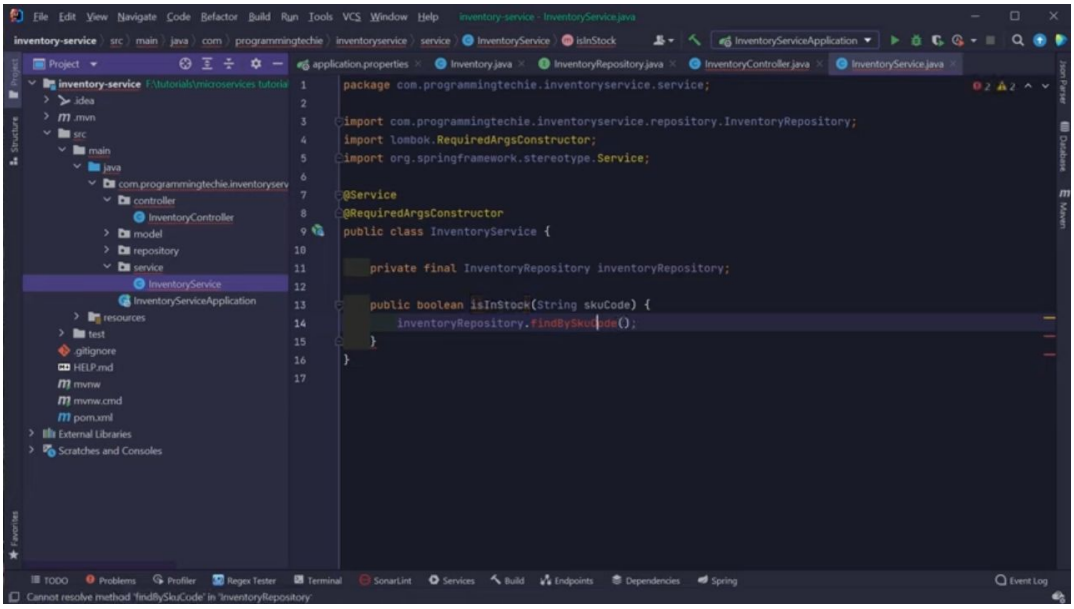
1:08:14


package com.programmingtechie.inventoryservice.model;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
@Entity
@Table(name = "t_inventory")
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class Inventory {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;
private String skuCode;
private Integer quantity;
}

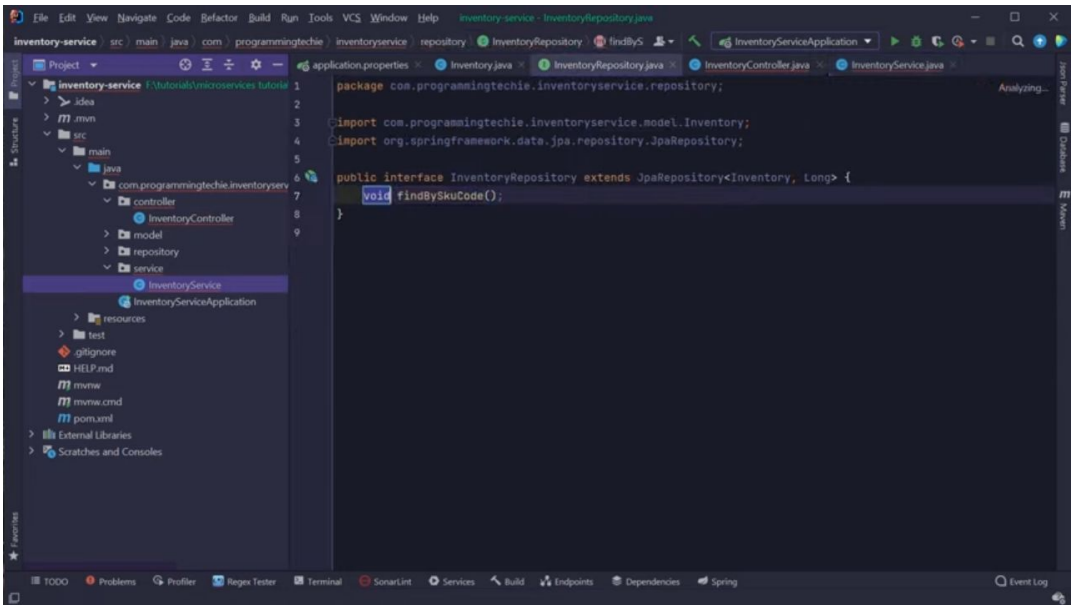
1:09:36


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryRepository.java
inventory-service src main java com programmingtechie inventoryservice
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie
model
repository
InventoryServic
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 package com.programmingtechie.inventoryservice.repository;
2
3 import com.programmingtechie.inventoryservice.model.Inventory;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface InventoryRepository extends JpaRepository<Inventory, Long> {
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Override File Type
Build Module 'inventory-service'
Rebuild '...ie.inventoryservice' Ctrl+Shift+F9
Run 'Tests in 'com.programmingtechie.inventoryservice'' Ctrl+Shift+F10
Debug 'Tests in 'com.programmingtechie.inventoryservice''
More Run/Debug
Open In
Local History
Reload from Disk
TODO
Problems
Profiler
Dependencies
Spring
Event Log

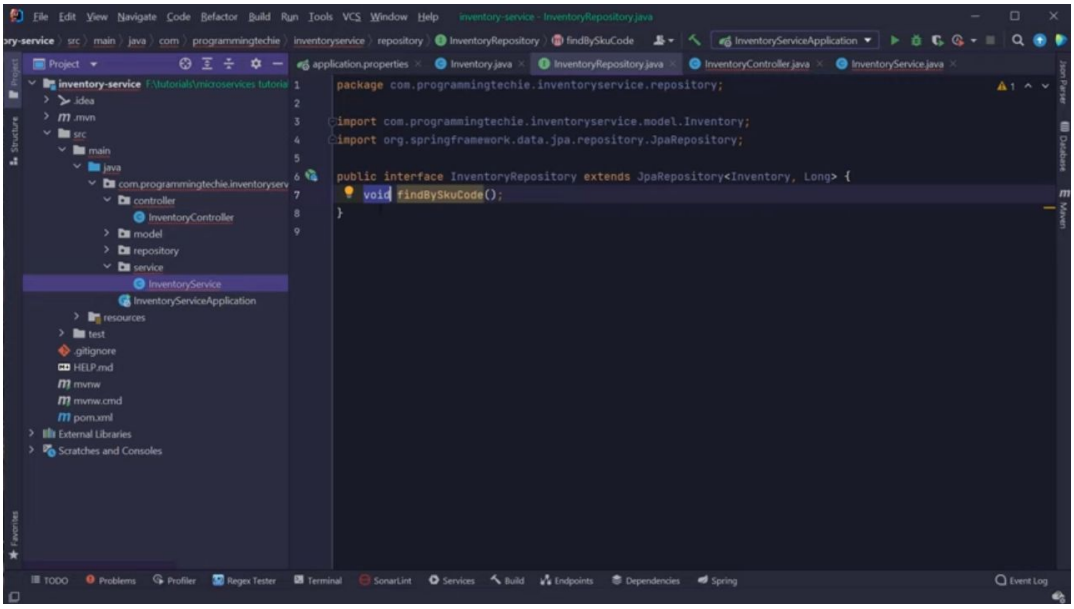
1:10:41


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryService.java
inventory-service src main java com programmingtechie inventoryservice service InventoryService isInStock
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryService.java
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.inventoryserv
controller
InventoryController
model
repository
service
InventoryService
InventoryServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 package com.programmingtechie.inventoryservice.service;
2
3 import com.programmingtechie.inventoryservice.repository.InventoryRepository;
4 import lombok.RequiredArgsConstructor;
5 import org.springframework.stereotype.Service;
6
7 @Service
8 @RequiredArgsConstructor
9 public class InventoryService {
10
11 private final InventoryRepository inventoryRepository;
12
13 public boolean isInStock(String skuCode) {
14 inventoryRepository.findBySkuCode();
15 }
16 }
17
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log
Cannot resolve method 'findBySkuCode' in 'InventoryRepository'

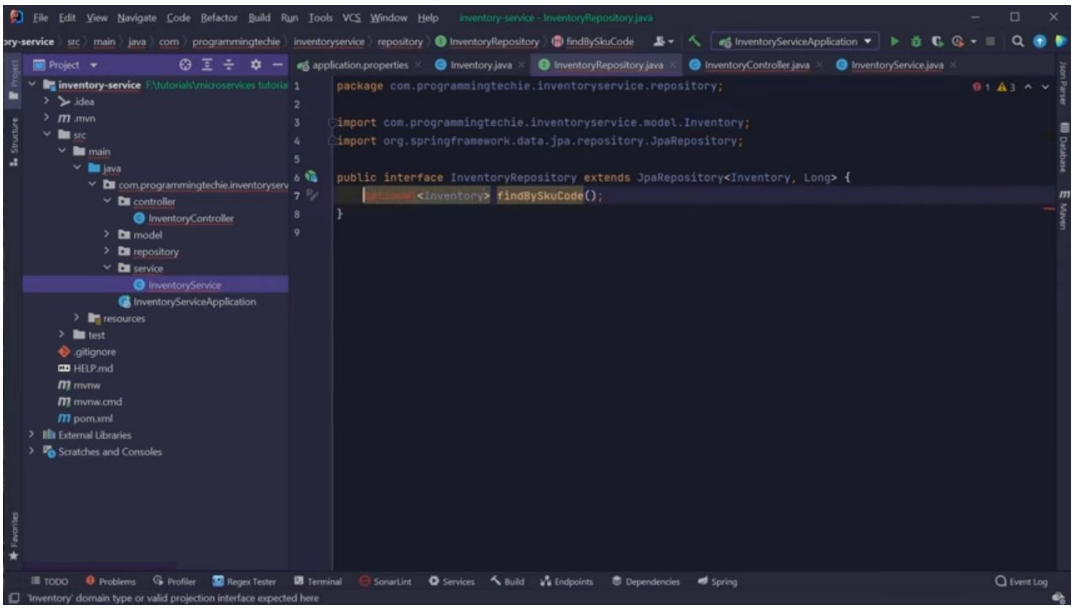
1:14:02


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryRepository.java
inventory-service src main java com programmingtechie inventoryservice repository InventoryRepository findBySkuCode
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryService.java
Analyzing...
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.inventoryservice
controller
InventoryController
model
repository
service
InventoryService
InventoryServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.programmingtechie.inventoryservice.repository;
import com.programmingtechie.inventoryservice.model.Inventory;
import org.springframework.data.jpa.repository.JpaRepository;
public interface InventoryRepository extends JpaRepository<Inventory, Long> {
void findBySkuCode();
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

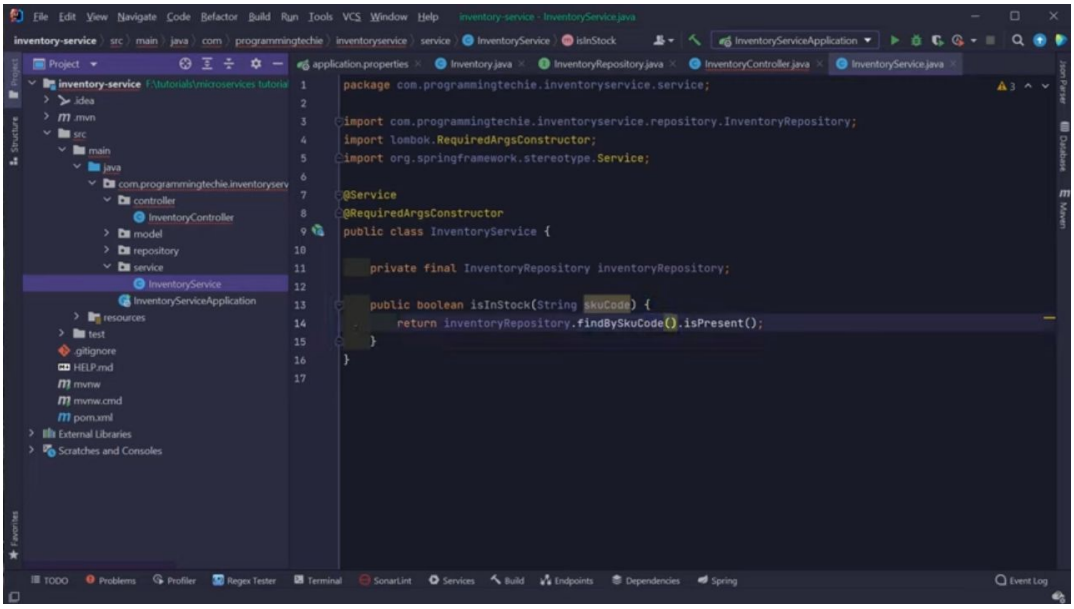
1:14:07


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryRepository.java
inventory-service src main java com programmingtechie inventoryservice repository InventoryRepository findBySkuCode
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryService.java
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.inventoryservice
controller
InventoryController
model
repository
service
InventoryService
InventoryServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.programmingtechie.inventoryservice.repository;
import com.programmingtechie.inventoryservice.model.Inventory;
import org.springframework.data.jpa.repository.JpaRepository;
public interface InventoryRepository extends JpaRepository<Inventory, Long> {
void findBySkuCode();
}
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Endpoints
Dependencies
Spring
Event Log

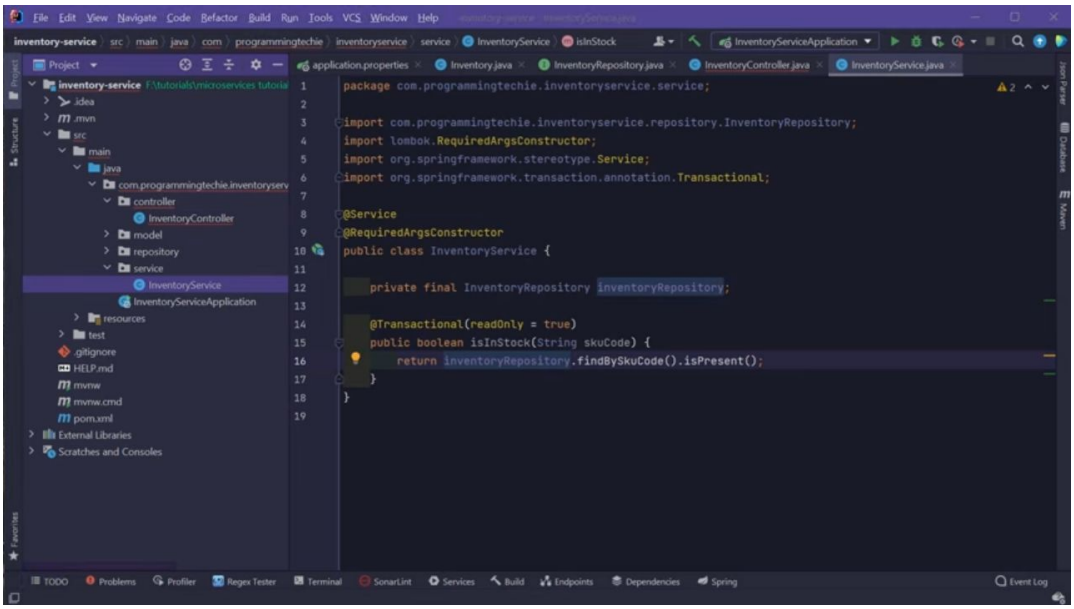
1:14:09


package com.programmingtechie.inventoryservice.repository;
import com.programmingtechie.inventoryservice.model.Inventory;
import org.springframework.data.jpa.repository.JpaRepository;
public interface InventoryRepository extends JpaRepository<Inventory, Long> {
Optional<Inventory> findBySkuCode();
}

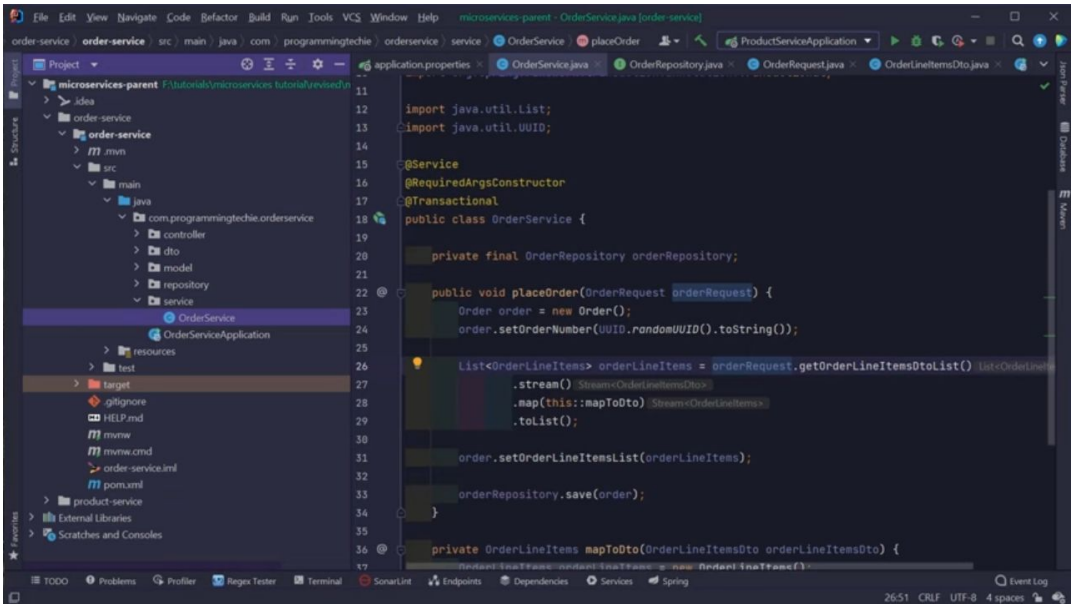
1:14:33


package com.programmingtechie.inventoryservice.service;
import com.programmingtechie.inventoryservice.repository.InventoryRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
@Service
@RequiredArgsConstructor
public class InventoryService {
private final InventoryRepository inventoryRepository;
public boolean isInStock(String skuCode) {
return inventoryRepository.findBySkuCode().isPresent();
}
}

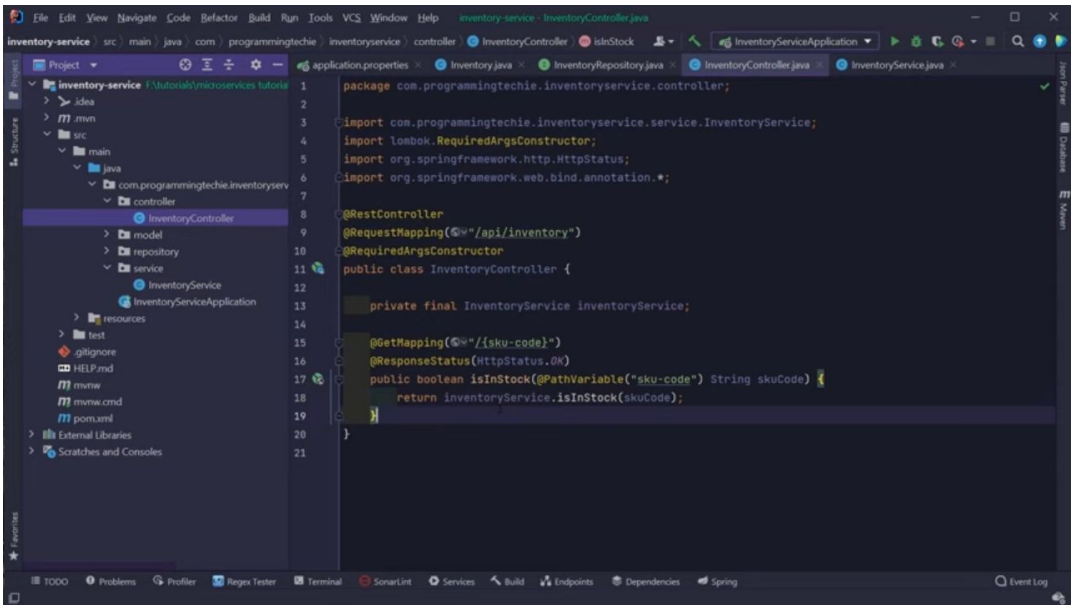
1:14:58


package com.programmingtechie.inventoryservice.service;
import com.programmingtechie.inventoryservice.repository.InventoryRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
@Service
@RequiredArgsConstructor
public class InventoryService {
private final InventoryRepository inventoryRepository;
@Transactional(readOnly = true)
public boolean isInStock(String skuCode) {
return inventoryRepository.findBySkuCode().isPresent();
}
}

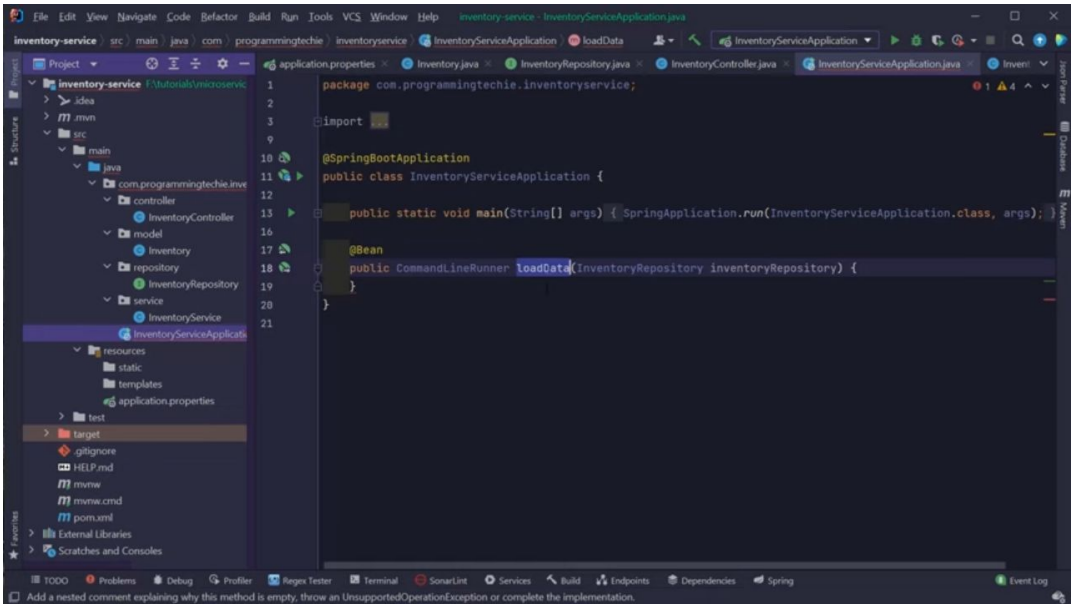
1:15:29


import java.util.List;
import java.util.UUID;
@Service
@RequiredArgsConstructor
@Transactional
public class OrderService {
private final OrderRepository orderRepository;
public void placeOrder(OrderRequest orderRequest) {
Order order = new Order();
order.setOrderNumber(UUID.randomUUID().toString());
List<OrderLineItems> orderLineItems = orderRequest.getOrderLineItemsDtoList()
.stream()
.map(this::mapToDto)
.toList();
order.setOrderLineItemsList(orderLineItems);
orderRepository.save(order);
}
private OrderLineItems mapToDto(OrderLineItemsDto orderLineItemsDto) {

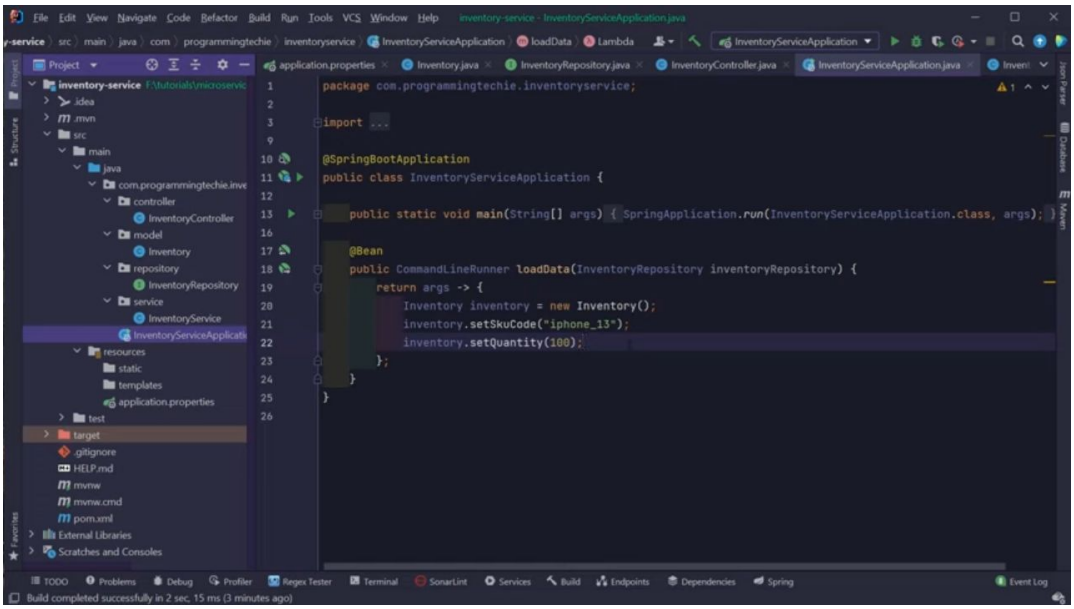
1:15:57


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryController.java
inventory-service src main java com programmingtechie inventoryservice controller InventoryController isInStock
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryService.java
inventory-service F:\tutorials\microservices tutorial
.idea
.mvn
src
main
java
com.programmingtechie.inventoryserv
controller
InventoryController
model
repository
service
InventoryService
InventoryServiceApplication
resources
test
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 package com.programmingtechie.inventoryservice.controller;
2
3 import com.programmingtechie.inventoryservice.service.InventoryService;
4 import lombok.RequiredArgsConstructor;
5 import org.springframework.http.HttpStatus;
6 import org.springframework.web.bind.annotation.*;
7
8 @RestController
9 @RequestMapping("/api/inventory")
10 @RequiredArgsConstructor
11 public class InventoryController {
12
13 private final InventoryService inventoryService;
14
15 @GetMapping("/{sku-code}")
16 @ResponseStatus(HttpStatus.OK)
17 public boolean isInStock(@PathVariable("sku-code") String skuCode) {
18 return inventoryService.isInStock(skuCode);
19 }
20 }
21
TODO Problems Profiler Regex Tester Terminal SonarLint Services Build Endpoints Dependencies Spring Event Log

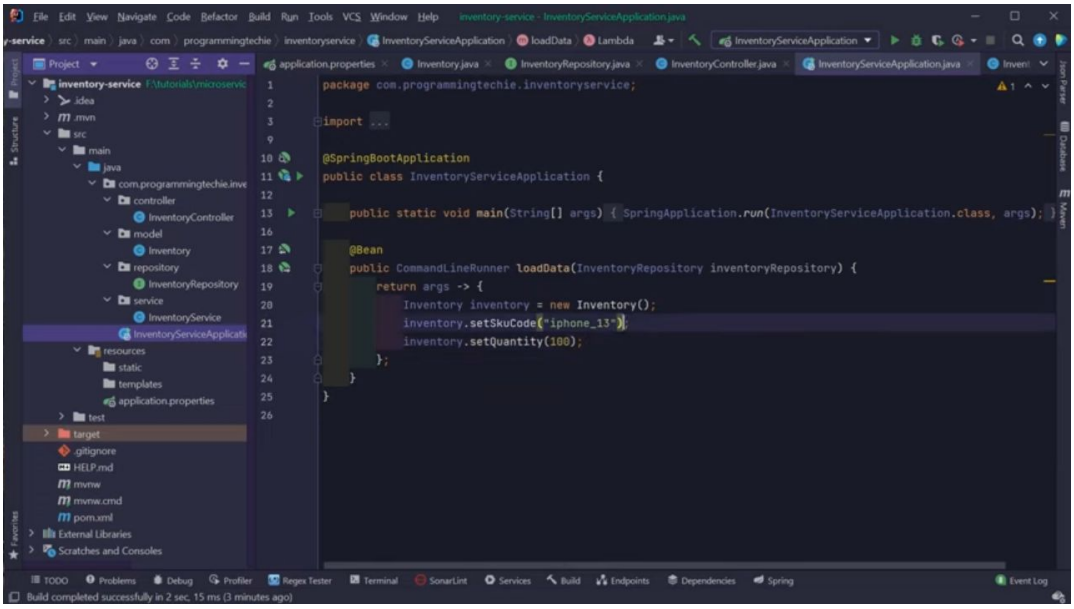
1:16:33


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryServiceApplication.java
inventory-service src main java com programmingtechie inventoryservice InventoryServiceApplication loadData
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryServiceApplication.java
inventory-service F:\tutorials\microservic
.idea
.mvn
src
main
java
com.programmingtechie.inve
controller
InventoryController
model
Inventory
repository
InventoryRepository
service
InventoryService
InventoryServiceApplicati
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
1 package com.programmingtechie.inventoryservice;
2
3 import ...
9
10 @SpringBootApplication
11 public class InventoryServiceApplication {
12
13 public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }
16
17 @Bean
18 public CommandLineRunner loadData(InventoryRepository inventoryRepository) {
19 }
20 }
21
TODO Problems Debug Profiler Regex Tester Terminal SonarLint Services Build Endpoints Dependencies Spring Event Log
Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.

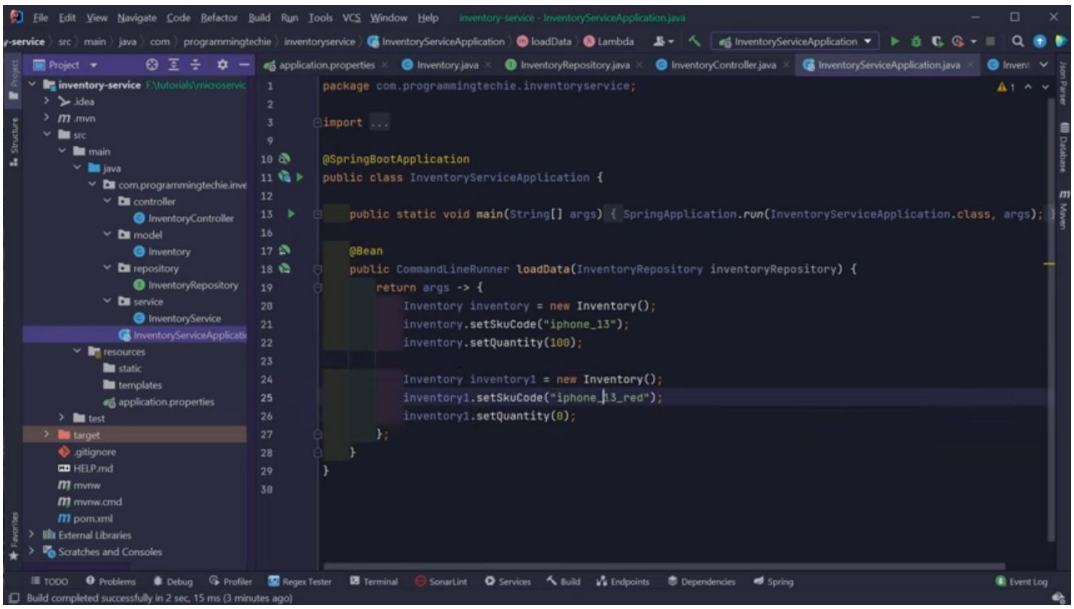
1:17:31


package com.programmingtechie.inventoryservice;
import ...
@SpringBootApplication
public class InventoryServiceApplication {
public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }
@Bean
public CommandLineRunner loadData(InventoryRepository inventoryRepository) {
return args -> {
Inventory inventory = new Inventory();
inventory.setSkuCode("iphone_13");
inventory.setQuantity(100);
};
}
}

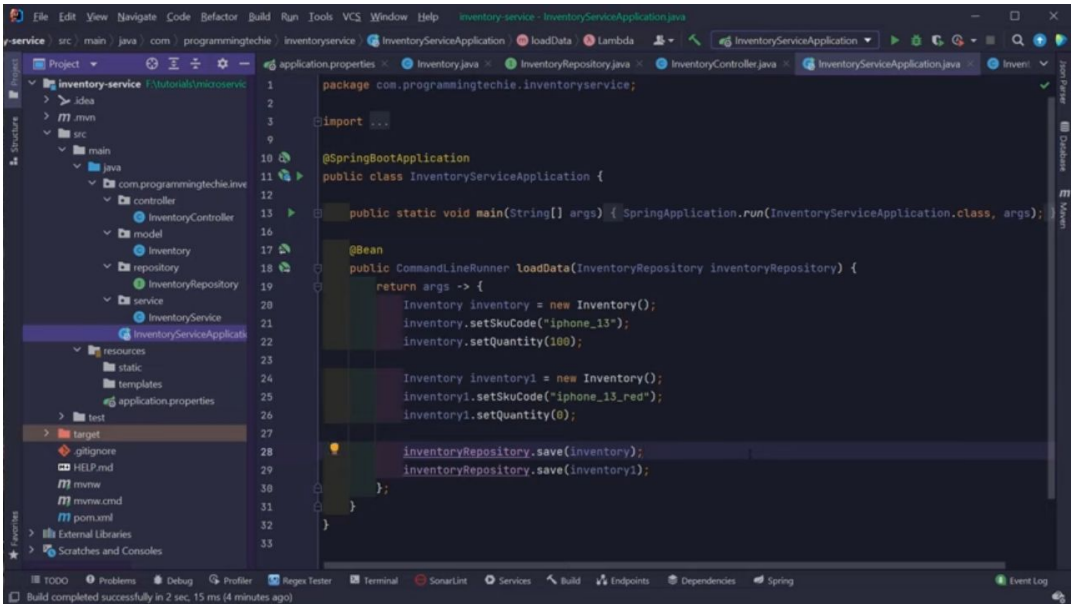
1:17:55


package com.programmingtechie.inventoryservice;
import ...
@SpringBootApplication
public class InventoryServiceApplication {
public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }
@Bean
public CommandLineRunner loadData(InventoryRepository inventoryRepository) {
return args -> {
Inventory inventory = new Inventory();
inventory.setSkuCode("iphone_13");
inventory.setQuantity(100);
};
}
}

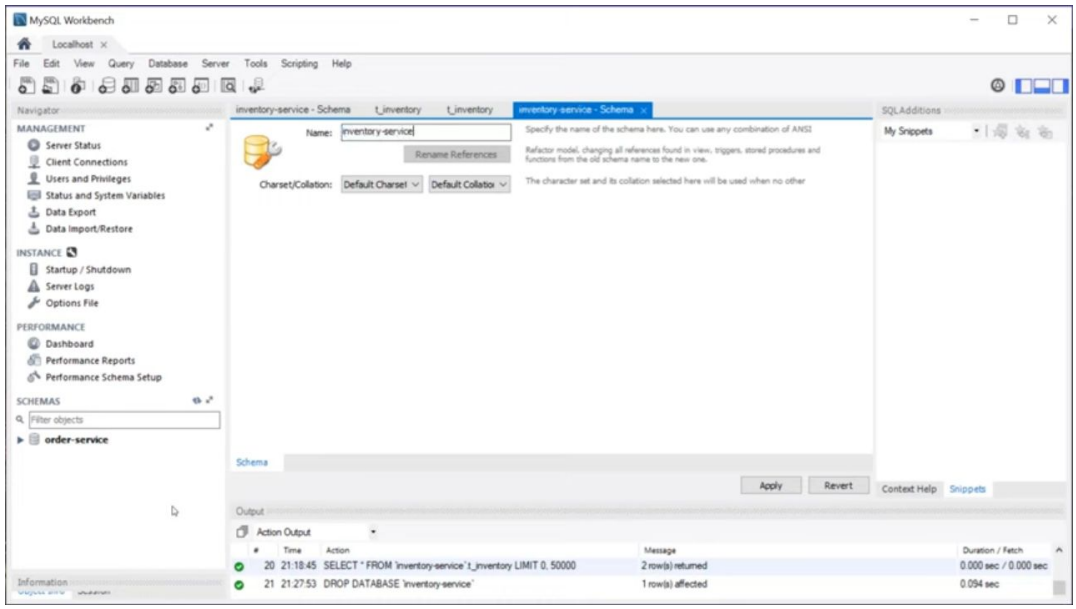
1:18:01


package com.programmingtechie.inventoryservice;
import ...
@SpringBootApplication
public class InventoryServiceApplication {
public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }
@Bean
public CommandLineRunner loadData(InventoryRepository inventoryRepository) {
return args -> {
Inventory inventory = new Inventory();
inventory.setSkuCode("iphone_13");
inventory.setQuantity(100);
Inventory inventory1 = new Inventory();
inventory1.setSkuCode("iphone_13_red");
inventory1.setQuantity(0);
};
}
}

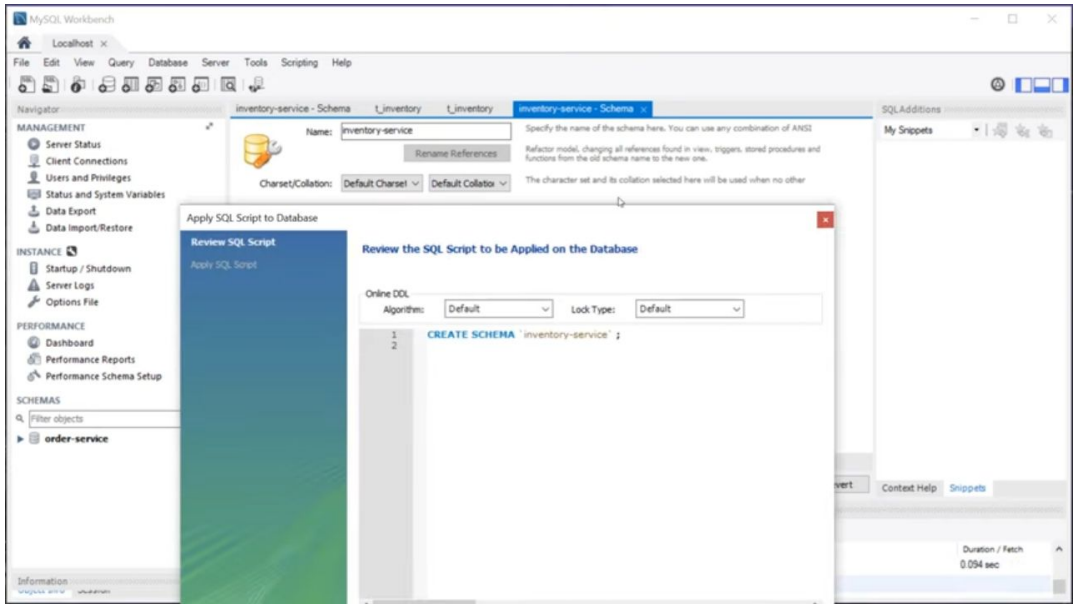
1:18:12


package com.programmingtechie.inventoryservice;
import ...
@SpringBootApplication
public class InventoryServiceApplication {
public static void main(String[] args) { SpringApplication.run(InventoryServiceApplication.class, args); }
@Bean
public CommandLineRunner loadData(InventoryRepository inventoryRepository) {
return args -> {
Inventory inventory = new Inventory();
inventory.setSkuCode("iphone_13");
inventory.setQuantity(100);
Inventory inventory1 = new Inventory();
inventory1.setSkuCode("iphone_13_red");
inventory1.setQuantity(0);
inventoryRepository.save(inventory);
inventoryRepository.save(inventory1);
};
}
}

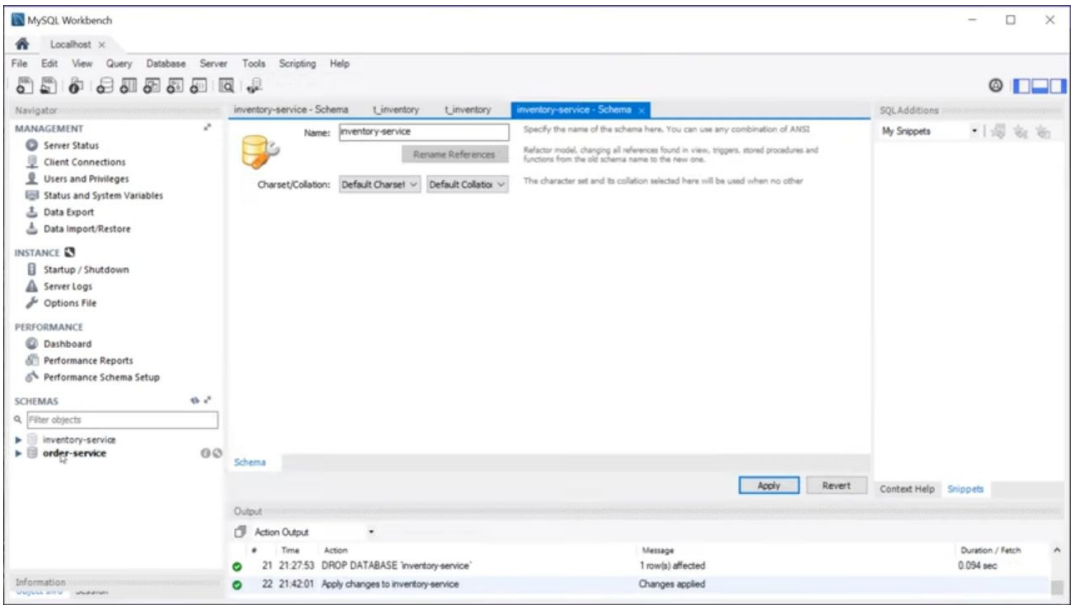
1:18:31


MySQL Workbench
Localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
order-service
inventory-service - Schema
t_inventory
t_inventory
inventory-service - Schema
Name: inventory-service
Rename References
Charset/Collation: Default Charset Default Collatio
SQLAdditions
My Snippets
Schema
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
20 21:18:45 SELECT * FROM `inventory-service`.t_inventory LIMIT 0, 50000 2 row(s) returned 0.000 sec / 0.000 sec
21 21:27:53 DROP DATABASE `inventory-service` 1 row(s) affected 0.094 sec
Information

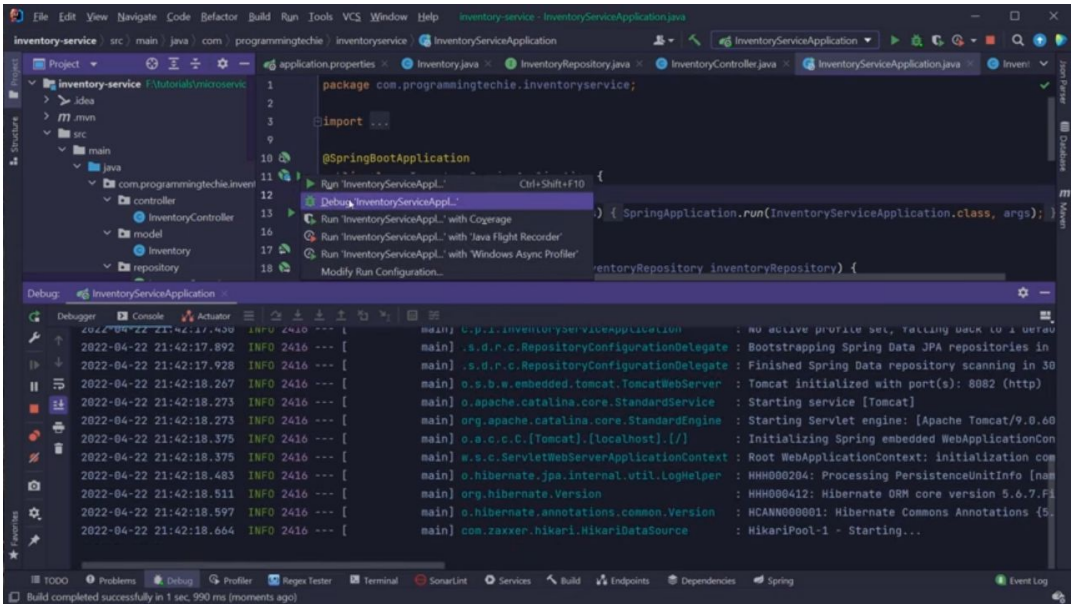
1:18:59


MySQL Workbench
Localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
order-service
inventory-service - Schema
t_inventory
t_inventory
inventory-service - Schema
Name: inventory-service
Rename References
Charset/Collation: Default Charset Default Collatio
SQLAdditions
My Snippets
Apply SQL Script to Database
Review SQL Script
Apply SQL Script
Review the SQL Script to be Applied on the Database
Online DDL
Algorithm: Default
Lock Type: Default
1 CREATE SCHEMA `inventory-service` ;
2
Context Help
Snippets
Duration / Fetch
0.094 sec
Information

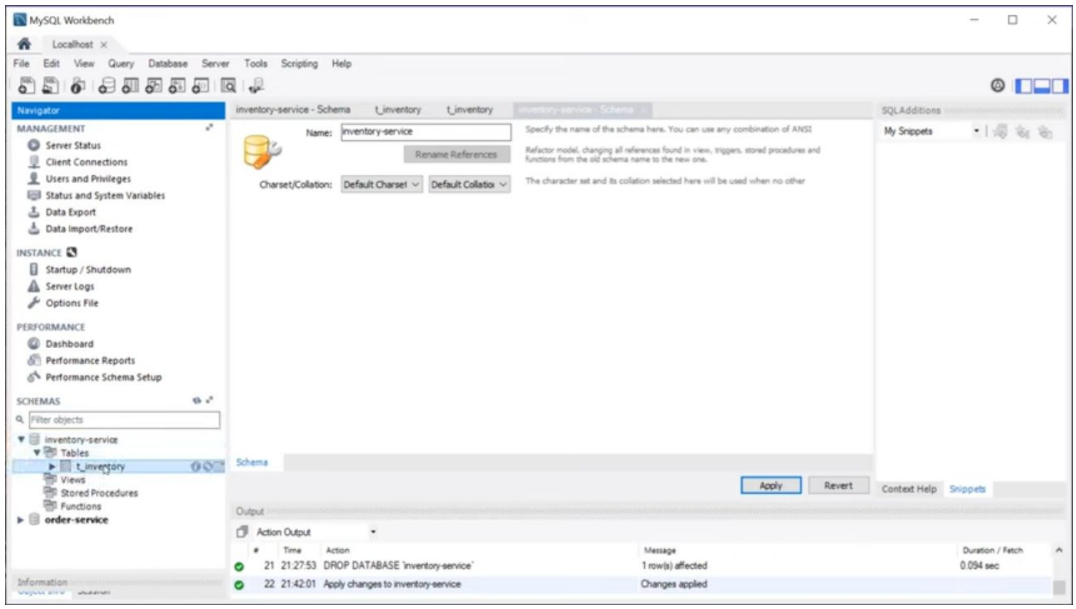
1:19:02


MySQL Workbench
Localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
inventory-service
order-service
inventory-service - Schema
t_inventory
t_inventory
inventory-service - Schema
Name: inventory-service
Rename References
Charset/Collation: Default Charset Default Collatio
SQLAdditions
My Snippets
Schema
Apply
Revert
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
21 21:27:53 DROP DATABASE 'inventory-service' 1 row(s) affected 0.094 sec
22 21:42:01 Apply changes to inventory-service Changes applied
Information

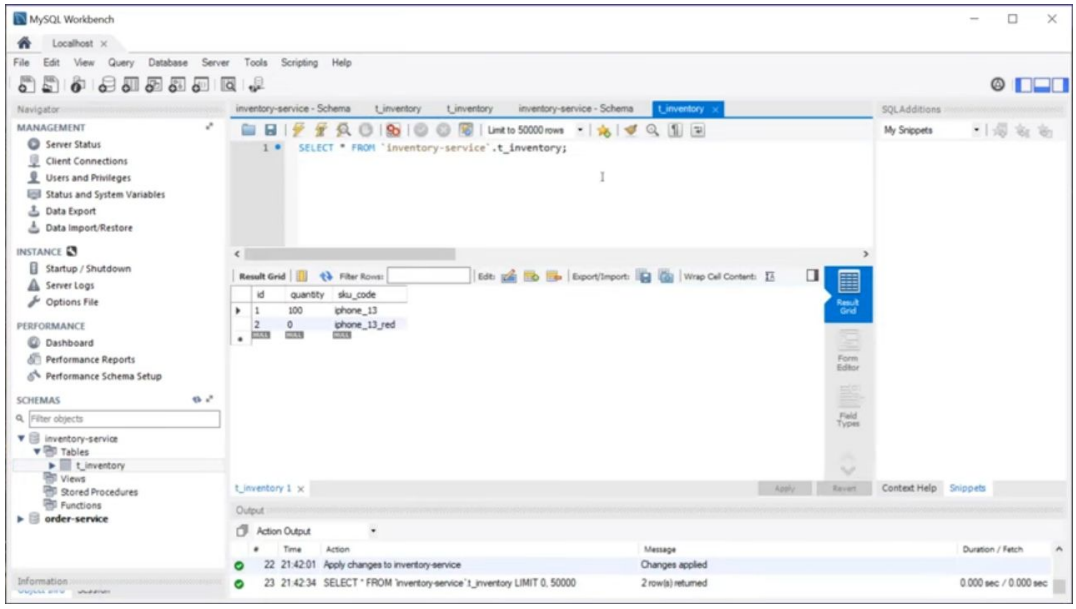
1:19:06


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
inventory-service - InventoryServiceApplication.java
inventory-service src main java com programmingtechie inventoryservice InventoryServiceApplication
InventoryServiceApplication
Project
application.properties
Inventory.java
InventoryRepository.java
InventoryController.java
InventoryServiceApplication.java
inventory-service F:\tutorials\microservic
.idea
.mvn
src
main
java
com.programmingtechie.invent
controller
InventoryController
model
Inventory
repository
package com.programmingtechie.inventoryservice;
import ...
@SpringBootApplication
Run 'InventoryServiceAppl...' Ctrl+Shift+F10
Debug 'InventoryServiceAppl...'
Run 'InventoryServiceAppl...' with Coverage
Run 'InventoryServiceAppl...' with 'Java Flight Recorder'
Run 'InventoryServiceAppl...' with 'Windows Async Profiler'
Modify Run Configuration...
SpringApplication.run(InventoryServiceApplication.class, args);
Debug: InventoryServiceApplication
Debugger Console Actuator
2022-04-22 21:42:17.892 INFO 2416 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in
2022-04-22 21:42:17.928 INFO 2416 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 30
2022-04-22 21:42:18.267 INFO 2416 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8082 (http)
2022-04-22 21:42:18.273 INFO 2416 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-04-22 21:42:18.273 INFO 2416 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.60
2022-04-22 21:42:18.375 INFO 2416 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationCon
2022-04-22 21:42:18.375 INFO 2416 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization com
2022-04-22 21:42:18.483 INFO 2416 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [nam
2022-04-22 21:42:18.511 INFO 2416 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.7.Fi
2022-04-22 21:42:18.597 INFO 2416 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.
2022-04-22 21:42:18.664 INFO 2416 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
TODO Problems Debug Profiler Regex Tester Terminal SonarLint Services Build Endpoints Dependencies Spring Event Log
Build completed successfully in 1 sec, 990 ms (moments ago)

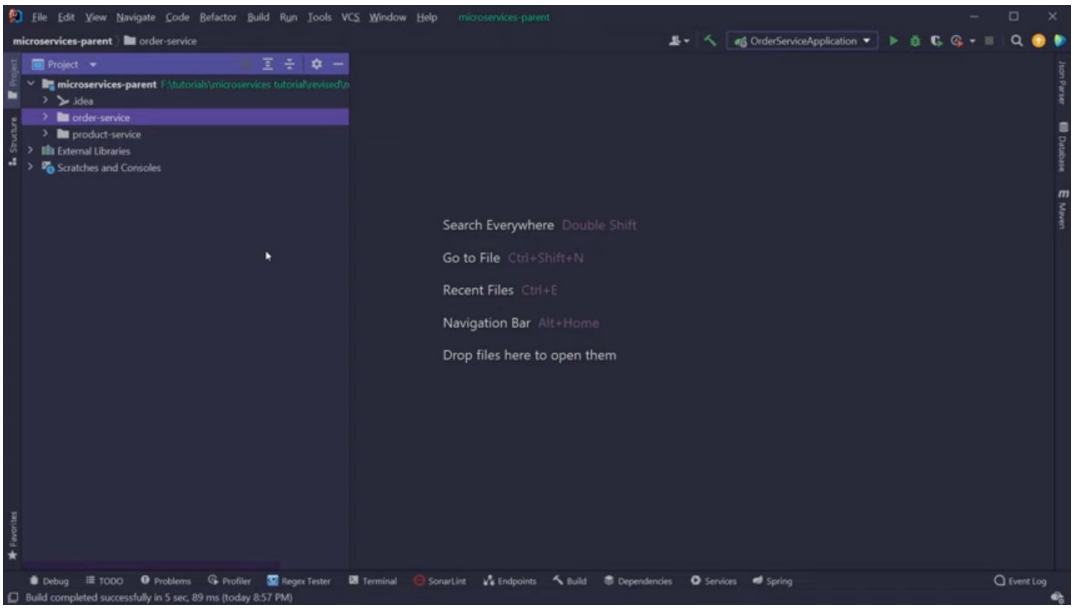
1:19:19


MySQL Workbench
Localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
inventory-service
Tables
t_inventory
Views
Stored Procedures
Functions
order-service
Information
inventory-service - Schema
t_inventory
t_inventory
Name: inventory-service
Rename References
Charset/Collation: Default Charset
Default Collatio
Schema
Apply
Revert
SQLAdditions
My Snippets
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
21 21:27:53 DROP DATABASE `inventory-service` 1 row(s) affected 0.094 sec
22 21:42:01 Apply changes to inventory-service Changes applied

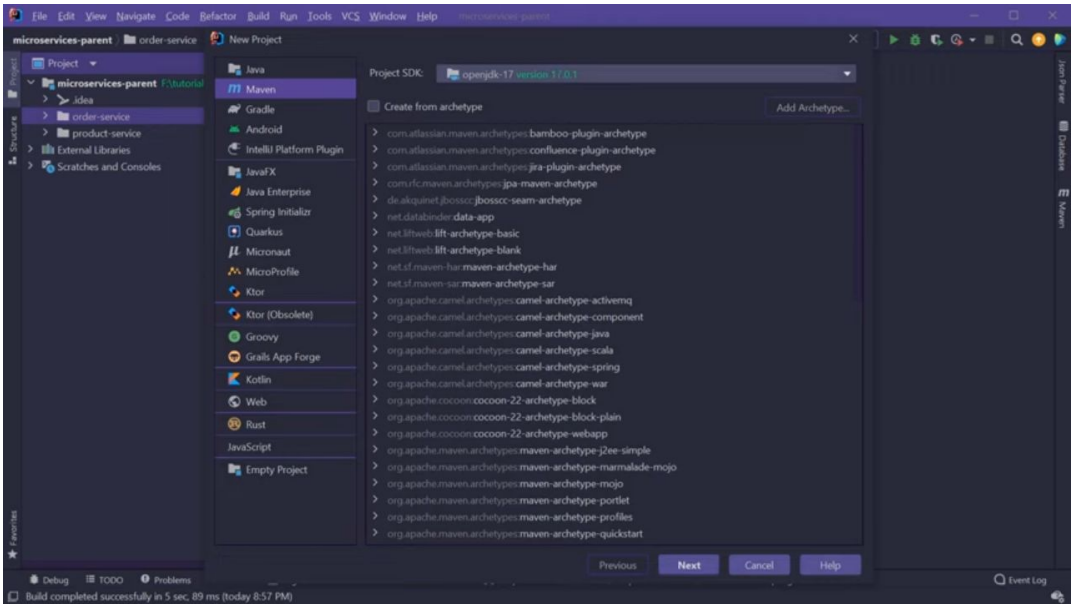
1:19:31


MySQL Workbench
Localhost
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
inventory-service
Tables
t_inventory
Views
Stored Procedures
Functions
order-service
Information
inventory-service - Schema
t_inventory
t_inventory
inventory-service - Schema
t_inventory
Limit to 50000 rows
1 SELECT * FROM `inventory-service`.t_inventory;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
id quantity sku_code
1 100 iphone_13
2 0 iphone_13_red
Result Grid
Form Editor
Field Types
t_inventory 1
Apply
Revert
SQLAdditions
My Snippets
Context Help
Snippets
Output
Action Output
Time Action Message Duration / Fetch
22 21:42:01 Apply changes to inventory-service Changes applied
23 21:42:34 SELECT * FROM `inventory-service`.t_inventory LIMIT 0, 50000 2 row(s) returned 0.000 sec / 0.000 sec

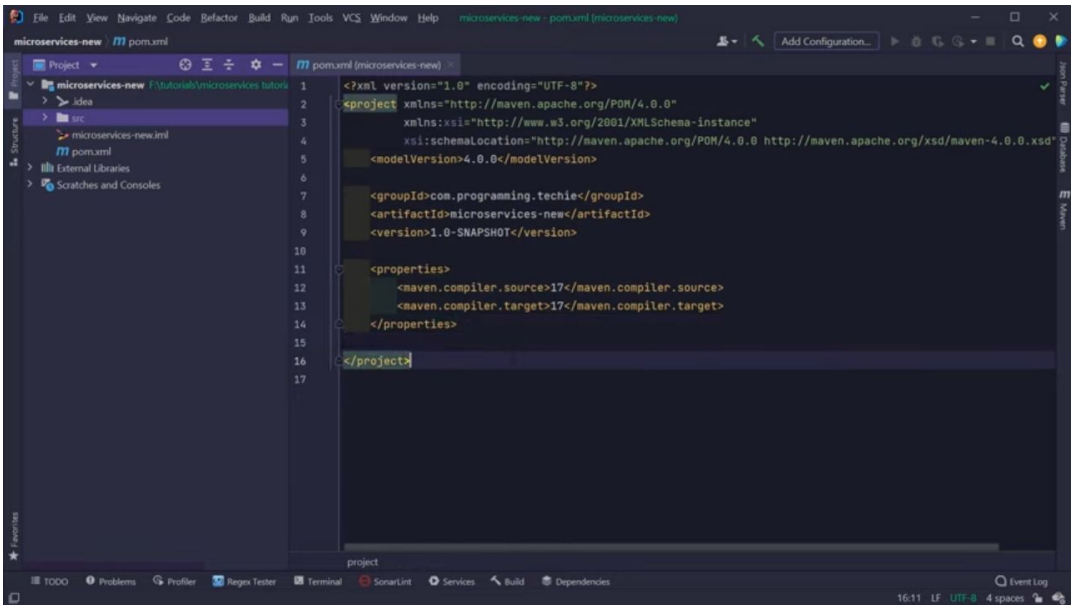
1:19:36


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent
order-service
OrderServiceApplication
Project
microservices-parent
.idea
order-service
product-service
External Libraries
Scratches and Consoles
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Navigation Bar Alt+Home
Drop files here to open them
Debug
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Endpoints
Build
Dependencies
Services
Spring
Event Log
Build completed successfully in 5 sec, 89 ms (today 8:57 PM)

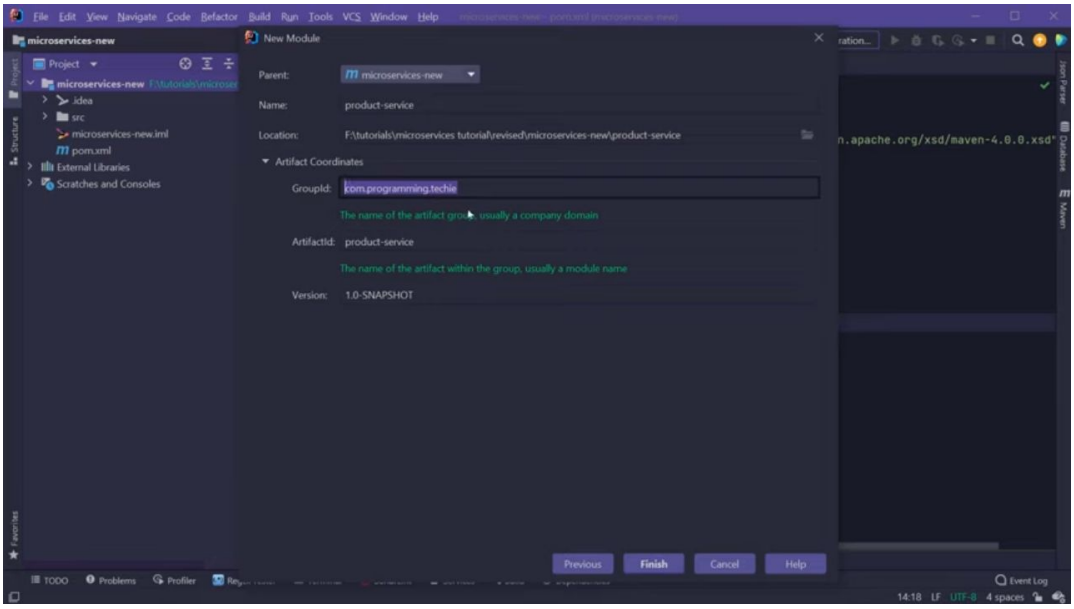
1:19:52


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-parent
order-service
New Project
Project
microservices-parent
.idea
order-service
product-service
External Libraries
Scratches and Consoles
Java
Maven
Gradle
Android
IntelliJ Platform Plugin
JavaFX
Java Enterprise
Spring Initializr
Quarkus
Micronaut
MicroProfile
Ktor
Ktor (Obsolete)
Groovy
Grails App Forge
Kotlin
Web
Rust
JavaScript
Empty Project
Project SDK:
openjdk-17 version 17.0.1
Create from archetype
Add Archetype...
com.atlassian.maven.archetypes:bamboo-plugin-archetype
com.atlassian.maven.archetypes:confluence-plugin-archetype
com.atlassian.maven.archetypes:jira-plugin-archetype
com.rfc.maven.archetypes:jpa-maven-archetype
de.akquinet.jbosscc:jbosscc-seam-archetype
net.databinder:data-app
net.liftweb:lift-archetype-basic
net.liftweb:lift-archetype-blank
net.sf.maven-har:maven-archetype-har
net.sf.maven-sar:maven-archetype-sar
org.apache.camel.archetypes:camel-archetype-activemq
org.apache.camel.archetypes:camel-archetype-component
org.apache.camel.archetypes:camel-archetype-java
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
org.apache.cocoon:cocoon-22-archetype-block
org.apache.cocoon:cocoon-22-archetype-block-plain
org.apache.cocoon:cocoon-22-archetype-webapp
org.apache.maven.archetypes:maven-archetype-j2ee-simple
org.apache.maven.archetypes:maven-archetype-marmalade-mojo
org.apache.maven.archetypes:maven-archetype-mojo
org.apache.maven.archetypes:maven-archetype-portlet
org.apache.maven.archetypes:maven-archetype-profiles
org.apache.maven.archetypes:maven-archetype-quickstart
Previous
Next
Cancel
Help
Debug
TODO
Problems
Event Log
Build completed successfully in 5 sec, 89 ms (today 8:57 PM)

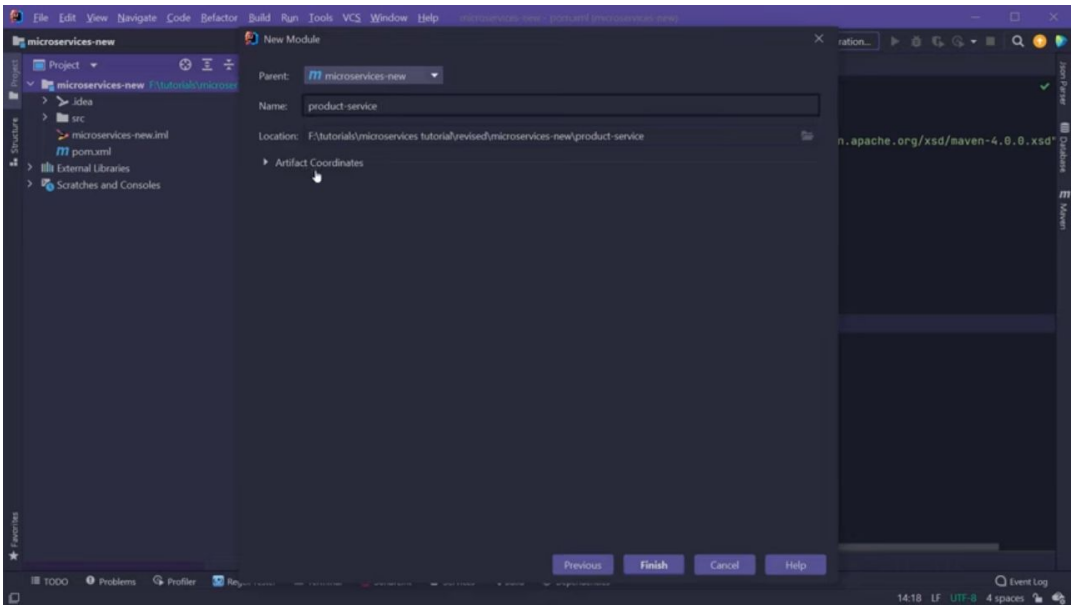
1:20:20


microservices-new - pom.xml (microservices-new)
Add Configuration...
pom.xml (microservices-new)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.programming.techie</groupId>
<artifactId>microservices-new</artifactId>
<version>1.0-SNAPSHOT</version>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
</project>
project

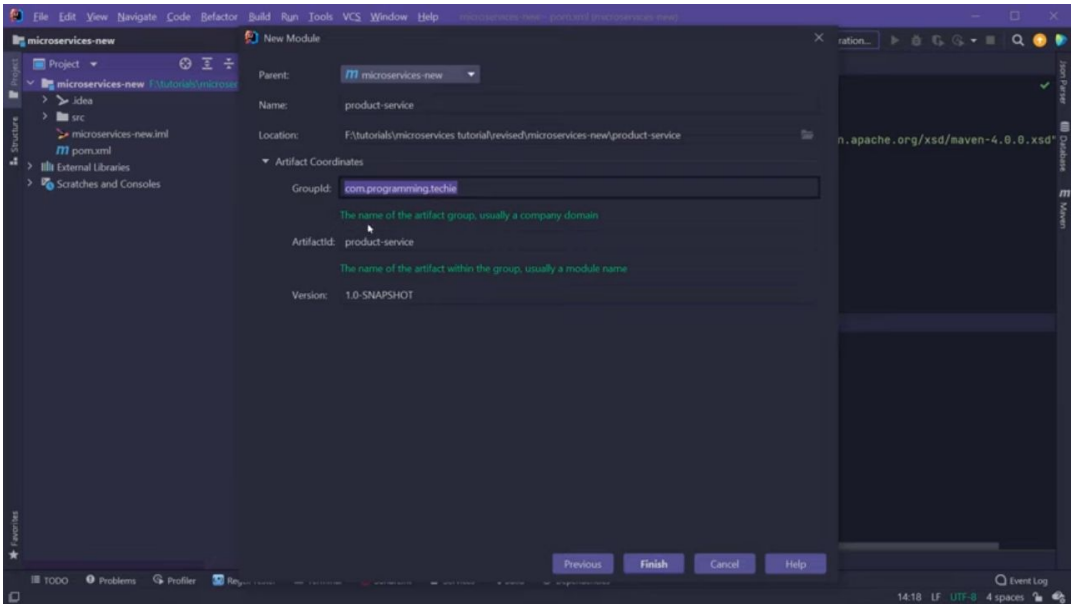
1:21:22


New Module
Parent: microservices-new
Name: product-service
Location: F:\tutorials\microservices tutorial\revised\microservices-new\product-service
Artifact Coordinates
GroupId: com.programming.techie
The name of the artifact group, usually a company domain
ArtifactId: product-service
The name of the artifact within the group, usually a module name
Version: 1.0-SNAPSHOT
Previous
Finish
Cancel
Help

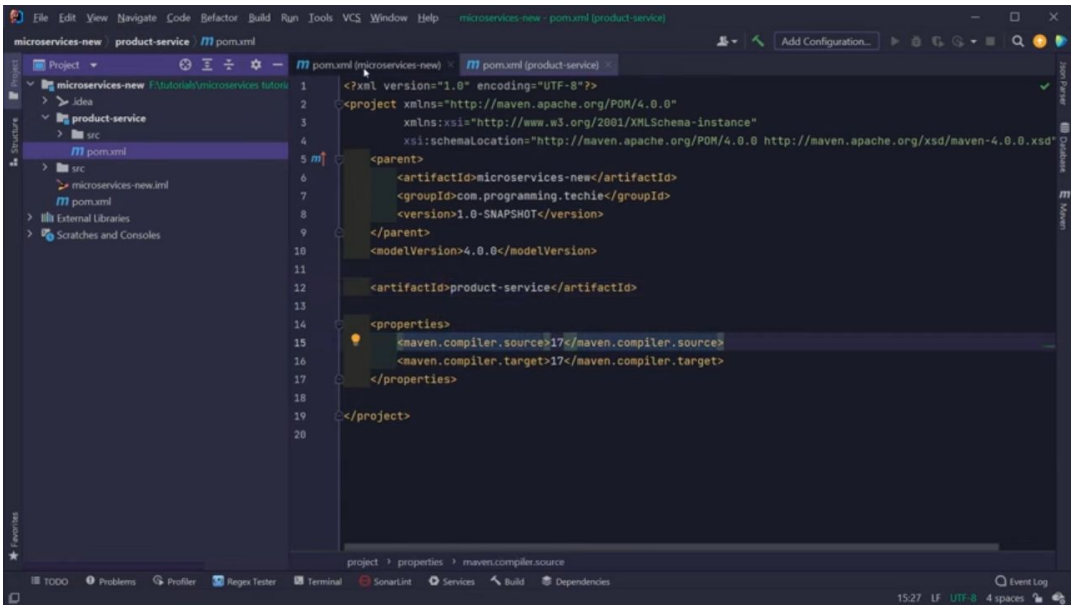
1:21:50


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-new
New Module
Parent: microservices-new
Name: product-service
Location: F:\tutorials\microservices tutorial\revised\microservices-new\product-service
Artifact Coordinates
Project
microservices-new
.idea
src
microservices-new.iml
pom.xml
External Libraries
Scratches and Consoles
Previous
Finish
Cancel
Help
TODO
Problems
Profiler
Event Log
14:18 LF UTF-8 4 spaces

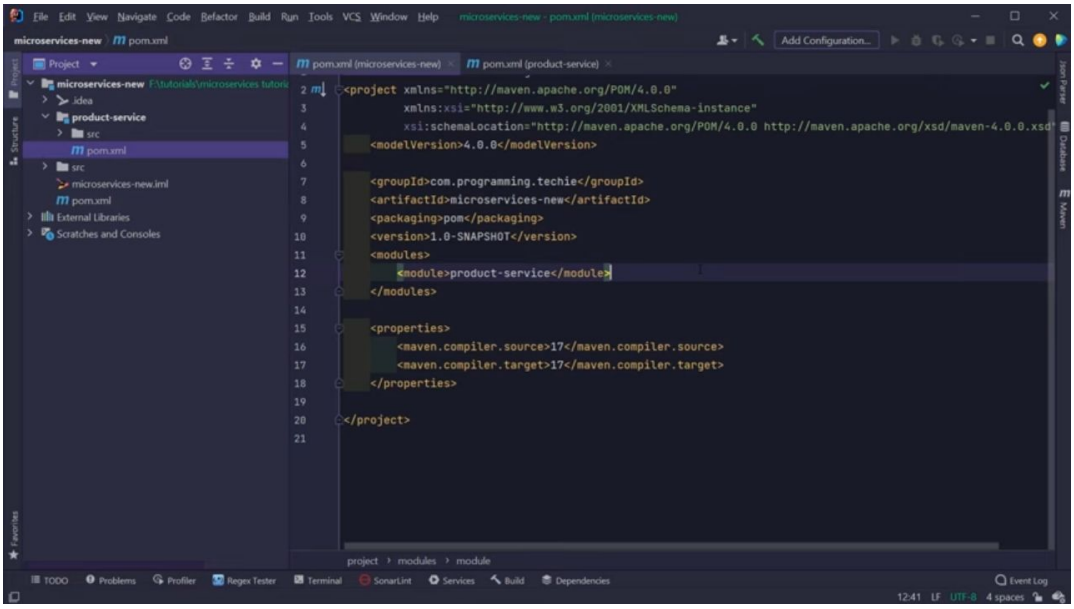
1:21:53


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-new
New Module
Parent: microservices-new
Name: product-service
Location: F:\tutorials\microservices tutorial\revised\microservices-new\product-service
Artifact Coordinates
GroupId: com.programming.techie
The name of the artifact group, usually a company domain
ArtifactId: product-service
The name of the artifact within the group, usually a module name
Version: 1.0-SNAPSHOT
Project
microservices-new
.idea
src
microservices-new.iml
pom.xml
External Libraries
Scratches and Consoles
Previous
Finish
Cancel
Help
TODO
Problems
Profiler
Event Log
14:18 LF UTF-8 4 spaces

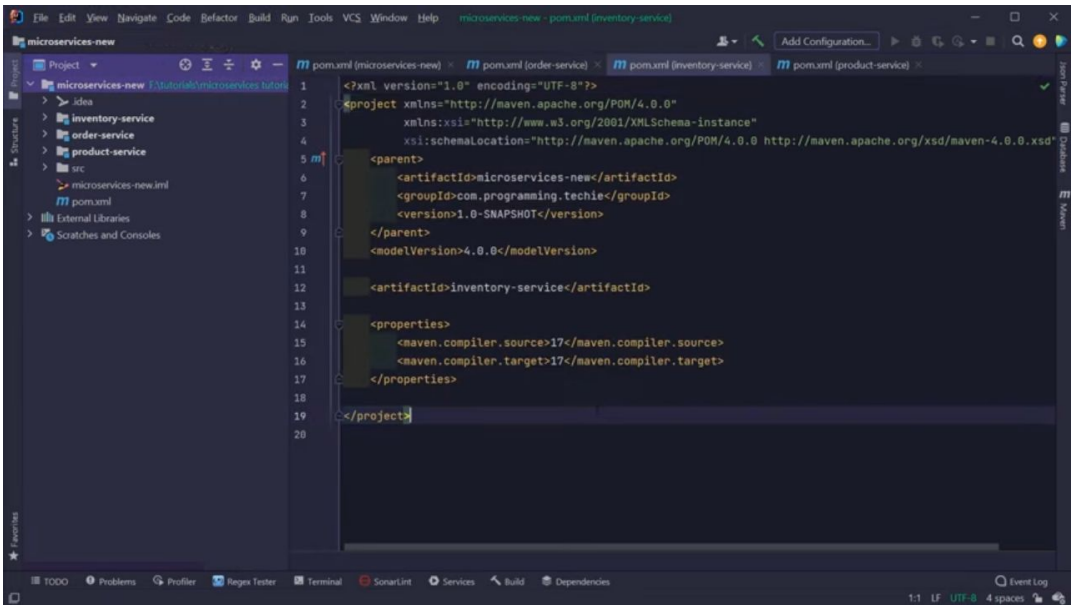
1:21:59


microservices-new - pom.xml (product-service)
pom.xml (microservices-new)
pom.xml (product-service)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<parent>
<artifactId>microservices-new</artifactId>
<groupId>com.programming.techie</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>product-service</artifactId>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
</project>
project › properties › maven.compiler.source

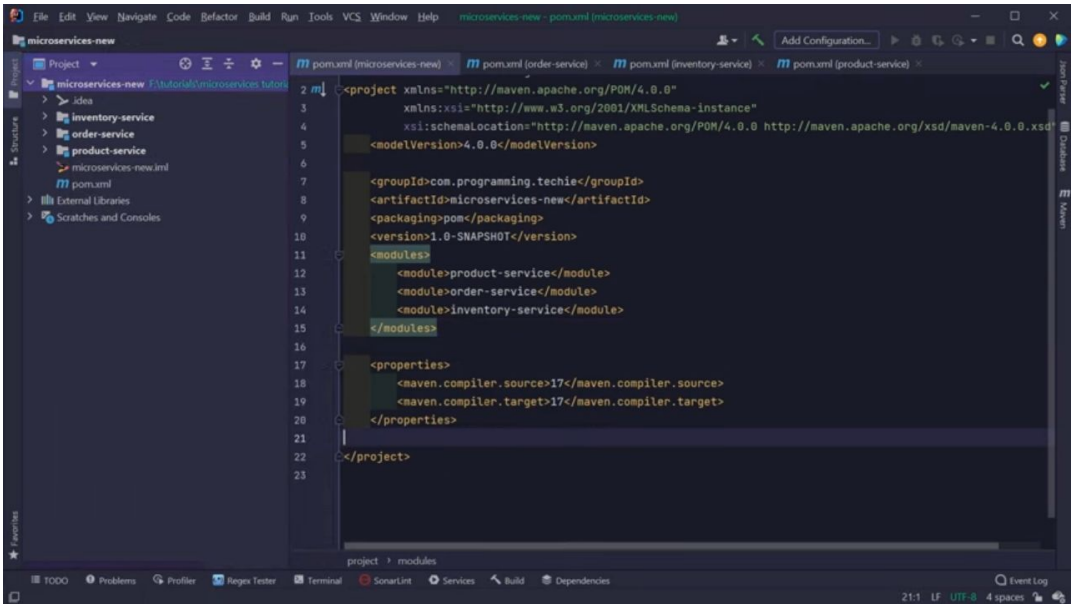
1:22:16


microservices-new - pom.xml (microservices-new)
pom.xml (microservices-new)
pom.xml (product-service)
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.programming.techie</groupId>
<artifactId>microservices-new</artifactId>
<packaging>pom</packaging>
<version>1.0-SNAPSHOT</version>
<modules>
<module>product-service</module>
</modules>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
</project>
project › modules › module

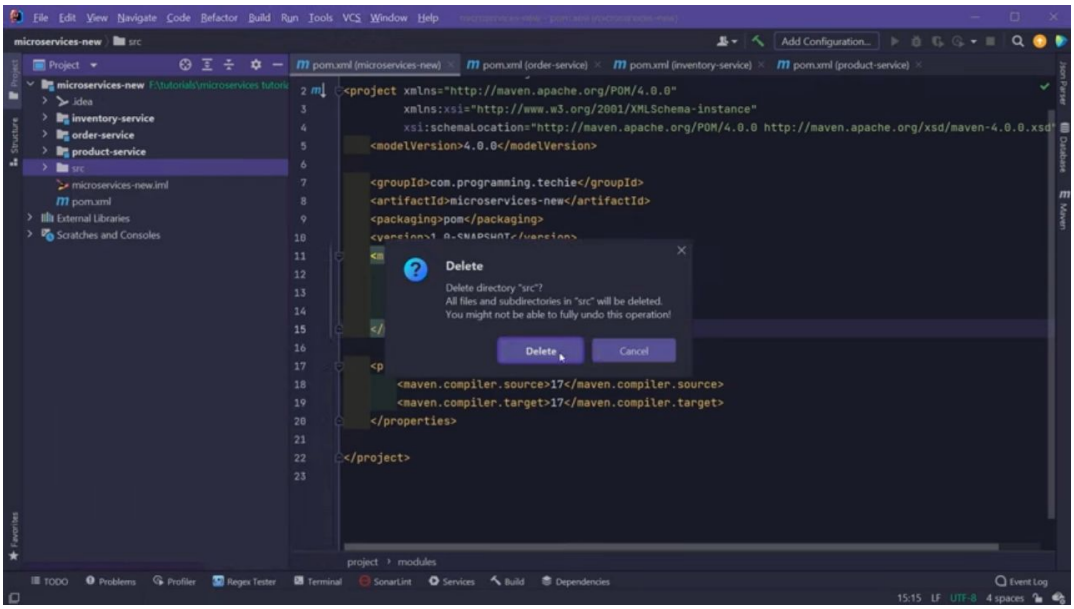
1:22:24


microservices-new - pom.xml (inventory-service)
pom.xml (microservices-new)
pom.xml (order-service)
pom.xml (inventory-service)
pom.xml (product-service)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<parent>
<artifactId>microservices-new</artifactId>
<groupId>com.programming.techie</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>inventory-service</artifactId>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
</project>

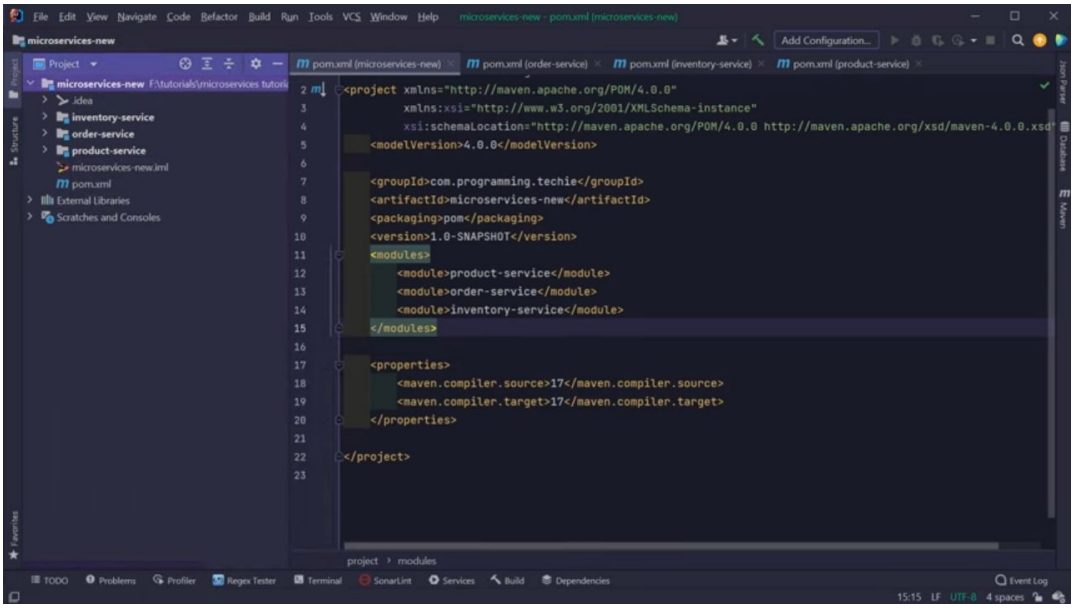
1:22:42


microservices-new - pom.xml (microservices-new)
pom.xml (microservices-new)
pom.xml (order-service)
pom.xml (inventory-service)
pom.xml (product-service)
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.programming.techie</groupId>
<artifactId>microservices-new</artifactId>
<packaging>pom</packaging>
<version>1.0-SNAPSHOT</version>
<modules>
<module>product-service</module>
<module>order-service</module>
<module>inventory-service</module>
</modules>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
</project>

1:22:49



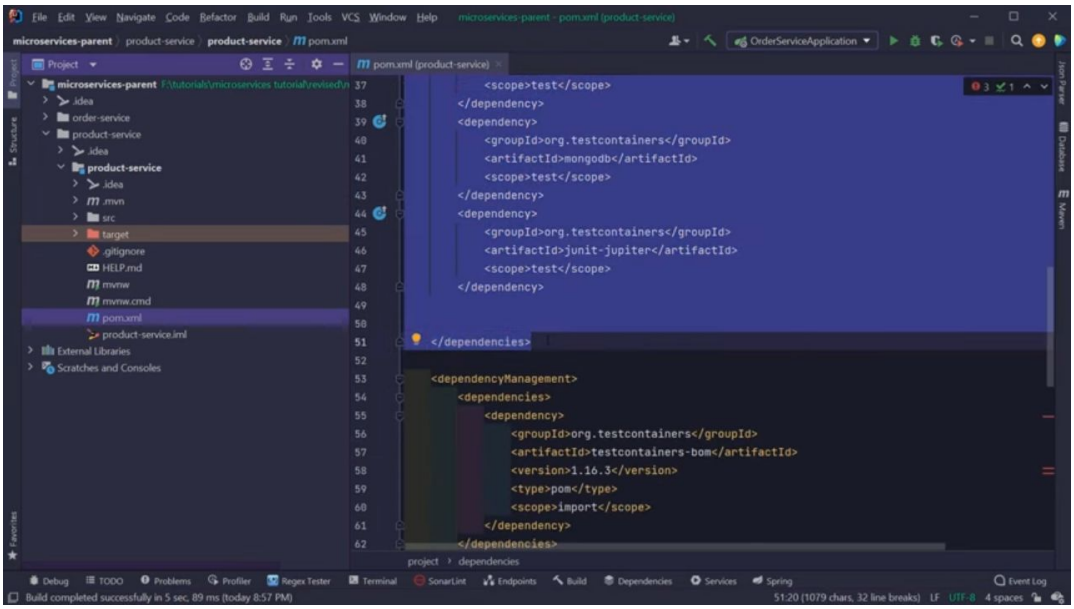
1:22:57



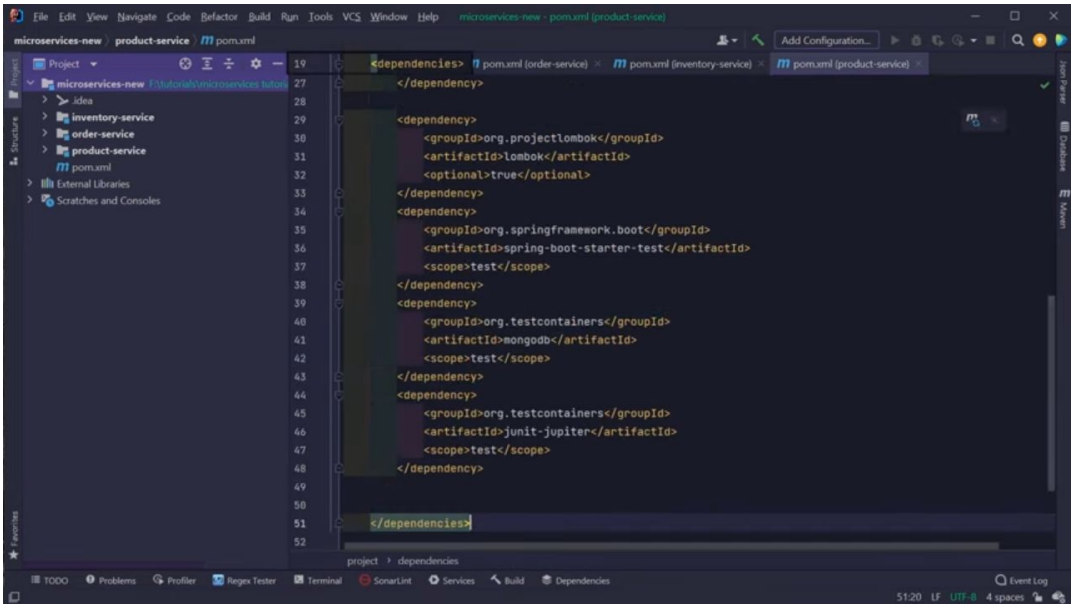
1:22:58

delete src folder

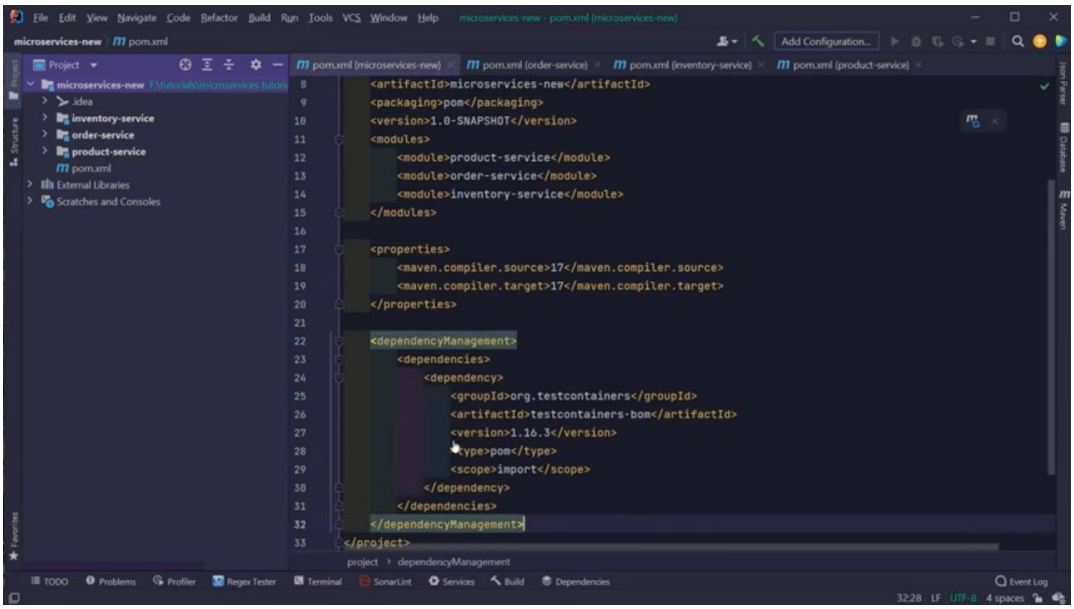
1:23:03


microservices-parent - pom.xml (product-service)
pom.xml (product-service)
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>mongodb</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>junit-jupiter</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers-bom</artifactId>
<version>1.16.3</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>

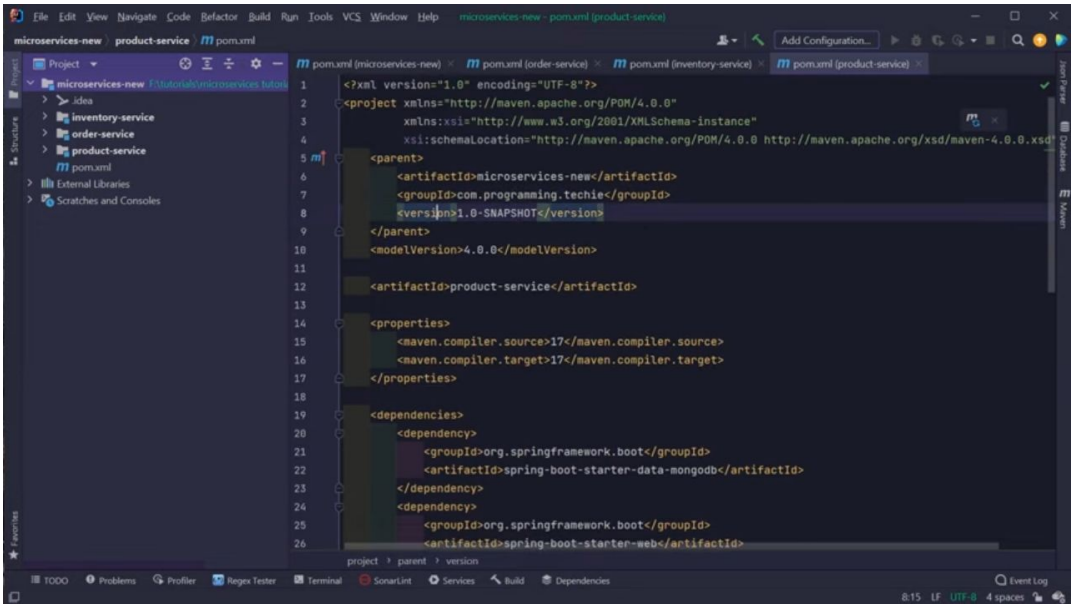
1:23:43


microservices-new - pom.xml (product-service)
pom.xml (order-service)
pom.xml (inventory-service)
pom.xml (product-service)
</dependency>
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>mongodb</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>junit-jupiter</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

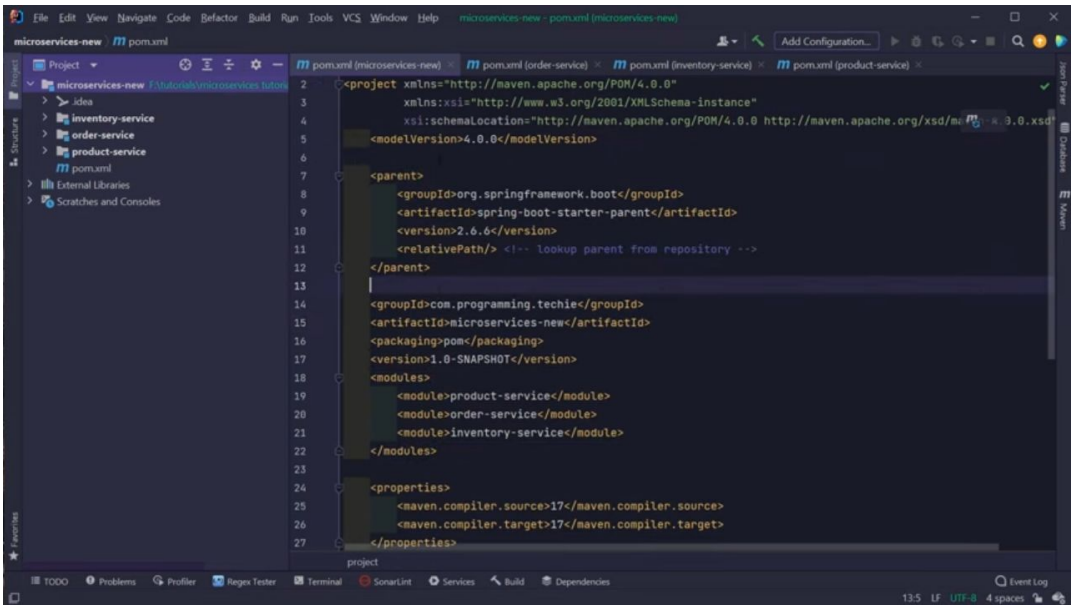
1:23:54


microservices-new - pom.xml (microservices-new)
pom.xml (microservices-new)
pom.xml (order-service)
pom.xml (inventory-service)
pom.xml (product-service)
<artifactId>microservices-new</artifactId>
<packaging>pom</packaging>
<version>1.0-SNAPSHOT</version>
<modules>
<module>product-service</module>
<module>order-service</module>
<module>inventory-service</module>
</modules>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
<dependencyManagement>
<dependencies>
<dependency>
<groupId>org.testcontainers</groupId>
<artifactId>testcontainers-bom</artifactId>
<version>1.16.3</version>
<type>pom</type>
<scope>import</scope>
</dependency>
</dependencies>
</dependencyManagement>
</project>
project › dependencyManagement

1:25:50


microservices-new - pom.xml (product-service)
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
<parent>
<artifactId>microservices-new</artifactId>
<groupId>com.programming.techie</groupId>
<version>1.0-SNAPSHOT</version>
</parent>
<modelVersion>4.0.0</modelVersion>
<artifactId>product-service</artifactId>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-mongodb</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
project › parent › version

1:27:05


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
microservices-new - pom.xml (microservices-new)
microservices-new / pom.xml
Add Configuration...
Project
microservices-new F:\tutorials\microservices tutori
.idea
inventory-service
order-service
product-service
pom.xml
External Libraries
Scratches and Consoles
pom.xml (microservices-new)
pom.xml (order-service)
pom.xml (inventory-service)
pom.xml (product-service)
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.6.6</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.programming.techie</groupId>
<artifactId>microservices-new</artifactId>
<packaging>pom</packaging>
<version>1.0-SNAPSHOT</version>
<modules>
<module>product-service</module>
<module>order-service</module>
<module>inventory-service</module>
</modules>
<properties>
<maven.compiler.source>17</maven.compiler.source>
<maven.compiler.target>17</maven.compiler.target>
</properties>
project
TODO
Problems
Profiler
Regex Tester
Terminal
SonarLint
Services
Build
Dependencies
Event Log
13:5 LF UTF-8 4 spaces

1:27:26