

SportyShoes > src > main > java > com > myproject > sportyshoes > model > User > id

Project

SportyShoes [SportyShoes2] ~/Download:

- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - Category
 - Product
 - Purchase
 - User
 - repository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml

External Libraries

Scratches and Consoles

Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Bookmarks

SportyShoes

Dao.java x User.java x

```
1 package com.myproject.sportyshoes.model;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name = "Users")
7 public class User {
8
9     @Id
10    @GeneratedValue(strategy = GenerationType.IDENTITY)
11
12    private long id;
13
14    private String name;
15
16    private String email;
17
18    private String password;
19
20    private String role;
21
22    private String phoneNumber;
23
24    public long getId() { return id; }
25
26    public void setId(long id) {
27        this.id = id;
28    }
29
30    public String getName() { return name; }
31
32    public void setName(String name) { this.name = name; }
33
34    public String getEmail() {
35        return email;
36    }
37
38    public void setEmail(String email) { this.email = email; }
39
40
41
42
43 }
```

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (58 minutes ago)

13:1 LF UTF-8 4 spaces

Project

SportyShoes [SportyShoes2] ~/Download:

- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - Category
 - Product
 - Purchase
 - User
 - repository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml
 - External Libraries
 - Scratches and Consoles
 - Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Structure

Bookmarks

JPA Structure

Persistence

Bookmarks

SportyShoes

```
1 package com.myproject.sportyshoes.model;
2
3
4
5 import javax.persistence.Entity;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9
10 @Entity
11 public class Purchase {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private long order_id;
16
17     private String date;
18
19     private long id;
20
21     private long pro_id;
22
23
24     public long getOrder_id() { return order_id; }
25
26     public void setOrder_id(long order_id) { this.order_id = order_id; }
27
28     public String getDate() { return date; }
29
30     public void setDate(String date) { this.date = date; }
31
32     public long getId() { return id; }
33
34     public void setId(long id) { this.id = id; }
35
36     public long getPro_id() { return pro_id; }
37
38
39
40
41
42
43
44
45
46
47
48
49
```

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (58 minutes ago)

11:14 LF UTF-8 4 spaces

Project

SportyShoes [SportyShoes2] ~/Download:

- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - Category
 - Product
 - Purchase
 - User
 - repository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml
 - External Libraries
 - Scratches and Consoles
 - Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

```
13
14
15 @Id
16 @GeneratedValue(strategy = GenerationType.IDENTITY)
17
18 private long pro_id;
19 private String name;
20 private String description;
21 private int price;
22 private long cat_id;
23
24
25
26
27 public long getPro_id() { return pro_id; }
28
29
30
31 public void setPro_id(long pro_id) { this.pro_id = pro_id; }
32
33
34
35 public String getName() { return name; }
36
37
38
39 public void setName(String name) { this.name = name; }
40
41
42
43 public String getDescription() { return description; }
44
45
46
47 public void setDescription(String description) { this.description = description; }
48
49
50
51 public int getPrice() { return price; }
52
53
54
55 public void setPrice(int price) { this.price = price; }
56
57
58 /**
59  * @return the cat_id
60  */
61 public long getCat_id() { return cat_id; }
62
63
64
65 public void setCat_id(long cat_id) { this.cat_id = cat_id; }
66
67
```

Project

SportyShoes [SportyShoes2] ~/Download:

- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - Category
 - Product
 - Purchase
 - User
 - repository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml
 - External Libraries
 - Scratches and Consoles
 - Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Structure

Bookmarks

JPA Structure

Persistence

Bookmarks

Version Control

Find

Run

TODO

Problems

Profiler

Terminal

Endpoints

Build

Dependencies

Spring

Event Log

```
13     @Entity
14     @Table(name = "category")
15     public class Category {
16         @Id
17         @GeneratedValue(strategy = GenerationType.IDENTITY)
18         private long cat_id;
19
20         @Column(nullable = false)
21         private String name;
22
23         @Column(nullable = false)
24         private String type;
25
26         @OneToMany(targetEntity = Product.class, cascade = CascadeType.ALL)
27         @JoinColumn(name = "cp_fk", referencedColumnName = "cat_id")
28         private List<Product> products;
29
30
31         public long getCat_id() { return cat_id; }
32
33         public void setCat_id(long cat_id) { this.cat_id = cat_id; }
34
35         public String getName() { return name; }
36
37         public void setName(String name) { this.name = name; }
38
39         public String getType() { return type; }
40
41         public void setType(String type) { this.type = type; }
42
43
44         public List<Product> getProducts() { return products; }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

```
1 package com.myproject.sportyshoes.repository;
2
3 import com.myproject.sportyshoes.model.Category;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface CategoryRepository extends JpaRepository<Category, Long> {
7
8 }
9
10 }
11
12 }
```

The screenshot shows an IDE interface with the following details:

- Project Tree:** On the left, under the "Project" tab, the "SportyShoes [SportyShoes2]" project structure is visible. It includes ".idea", ".mvn", "src", "main" (containing "java", "resources", "test"), "target" (containing ".gitignore", "HELP.md", "mvnw", "mvnw.cmd", "pom.xml", "SportyShoes2.iml"), "External Libraries", and "Scratches and Consoles".
- Code Editor:** The main editor area displays the "CategoryRepository.java" file. The code implements a Spring Data JPA repository for the "Category" entity.
- Toolbars and Status Bar:** At the bottom, there are toolbars for Version Control, Find, Run, TODO, Problems, Profiler, Terminal, Endpoints, Build, Dependencies, and Spring. The status bar at the bottom right shows the time (9:1), file type (LF), encoding (UTF-8), and number of spaces (4 spaces).

Project

- SportyShoes [SportyShoes2] ~/Download:
- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - CategoryRepository
 - ProductRepository
 - PurchaseRepository
 - UserRepository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
- target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml
- External Libraries
- Scratches and Consoles
 - Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Bookmarks

- SportyShoes

```
1 package com.myproject.sportyshoes.repository;
2
3 import com.myproject.sportyshoes.model.Product;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface ProductRepository extends JpaRepository<Product, Long> {
7 }
```

Project

- SportyShoes [SportyShoes2] ~/Download:
 - .idea
 - .mvn
 - src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - CategoryRepository
 - ProductRepository
 - PurchaseRepository
 - UserRepository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml

Structure

Bookmarks

JPA Structure

Persistence

```
1 package com.myproject.sportyshoes.repository;
2
3 import com.myproject.sportyshoes.model.Purchase;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.data.jpa.repository.Query;
6
7
8 public interface PurchaseRepository extends JpaRepository<Purchase, Long> {
9
10    @Query("select d.date from Purchase d where d.date=:date")
11    public String findByDate(String date);
12
13    @Query("select p.pro_id from Purchase p where p.pro_id=:pro_id")
14    public String findByPro_id(long pro_id);
15
16
17
18 }
```

Project

- SportyShoes [SportyShoes2] ~/Download:
- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - CategoryRepository
 - ProductRepository
 - PurchaseRepository
 - UserRepository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
- target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml
- External Libraries
- Scratches and Consoles
 - Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Structure

Bookmarks

JPA Structure

Persistence

Bookmarks

- SportyShoes

```
1 package com.myproject.sportyshoes.repository;
2
3 import com.myproject.sportyshoes.model.User;
4 import org.springframework.data.jpa.repository.JpaRepository;
5
6 public interface UserRepository extends JpaRepository <User, Long> {
7
8 }
```

SportyShoes – SpringSecurityConfiguration.java

```
1 package com.myproject.sportyshoes.security;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
5 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
7 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
8 import org.springframework.security.crypto.password.NoOpPasswordEncoder;
9 import org.springframework.security.crypto.password.PasswordEncoder;
10
11
12 @EnableWebSecurity
13 public class SpringSecurityConfiguration extends WebSecurityConfigurerAdapter {
14
15     // Adding the roles
16     @Override
17     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
18         auth.inMemoryAuthentication()
19             .withUser(username: "natu") UserDetailsManagerConfigurer<AuthenticationManagerBuilder>
20                 .password("pass")
21                 .roles("admin_role")
22             .and()
23             .withUser(username: "user") UserDetailsManagerConfigurer<AuthenticationManagerBuilder>
24                 .password("pass1")
25                 .roles("user_role");
26     }
27
28     // Configuring the api
29     // according to the roles.
30     @Override
31     protected void configure(HttpSecurity http) throws Exception {
32         http.
33             httpBasic() HttpBasicConfigurer<HttpSecurity>
34             .and()
35             .authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
```

```
25     .roles("user_role");  
26 }  
27 // Configuring the api  
28 // according to the roles.  
29 @Override  
30 protected void configure(HttpSecurity http) throws Exception {  
31     http.  
32         httpBasic() HttpBasicConfigurer<HttpSecurity>  
33             .and() HttpSecurity  
34                 .authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry  
35             .antMatchers( ...antPatterns: "/api/admin/users/getAll").hasRole("admin_role")  
36             .antMatchers( ...antPatterns: "/api/admin/users/{id}").hasRole("admin_role")  
37             .antMatchers( ...antPatterns: "/api/admin/products/getAll").hasRole("admin_role")  
38             .antMatchers( ...antPatterns: "/api/admin/products/{id}").hasRole("admin_role")  
39             .antMatchers( ...antPatterns: "/api/admin/purchase/{date}").hasRole("admin_role")  
40             .antMatchers( ...antPatterns: "/api/admin/purchase/{pro_id}").hasRole("admin_role")  
41         .and()  
42             .formLogin();  
43 }  
44 // Function to encode the password  
45 // assign to the particular roles.  
46 @Bean  
47 public PasswordEncoder getPasswordEncoder() { return NeopPasswordEncoder.getInstance(); }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }
```

SportyShoes > src > main > java > com > myproject > sportyshoes > service > impl > AdminServiceImpl

Project

SportyShoes [SportyShoes2] ~/Downloads:

```
13
14     @Service
15     public class AdminServiceImpl implements AdminService {
16         private UserRepository userRepository;
17
18         public AdminServiceImpl(UserRepository userRepository) { this.userRepository = userRepository; }
19
20         @Override
21         public User saveUser(User user) { return userRepository.save(user); }
22
23         @Override
24         public List<User> getAllUsers() { return userRepository.findAll(); }
25
26         @Override
27         public User getUserById(long id) {
28             return userRepository.findById(id).orElseThrow(() ->
29                 new ResourceNotFoundException("User", "Id", id));
30         }
31
32         @Override
33         public User updateUser(User user, long id) {
34
35             // we need to check whether the user with given id exists in DB or not
36             User existingUser = userRepository.findById(id).orElseThrow(
37                 () -> new ResourceNotFoundException("User", "Id", id));
38
39             existingUser.setName(user.getName());
40             existingUser.setEmail(user.getEmail());
41
42             userRepository.save(existingUser);
43
44             return existingUser;
45         }
46
47
48
49
50
51
52
53
```

Structure

Bookmarks

Bookmarks

Persistence

External Libraries

Scratches and Consoles

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (1 hour ago)

27:1 (98 chars, 3 line breaks) LF UTF-8 4 spaces

```
1 package com.myproject.sportyshoes.service.impl;
2
3 import com.myproject.sportyshoes.Exception.ResourceNotFoundException;
4 import com.myproject.sportyshoes.model.User;
5 import com.myproject.sportyshoes.repository.UserRepository;
6 import com.myproject.sportyshoes.service.AdminService;
7 import org.springframework.stereotype.Service;
8
9 import java.sql.ResultSet;
10 import java.sql.SQLException;
11 import java.util.List;
12 import java.util.Optional;
13
14 @Service
15 public class AdminServiceImpl implements AdminService {
16     private UserRepository userRepository;
17
18     public AdminServiceImpl(UserRepository userRepository) { this.userRepository = userRepository; }
19
20     @Override
21     public User saveUser(User user) { return userRepository.save(user); }
22
23     @Override
24     public List<User> getAllUsers() { return userRepository.findAll(); }
25
26     @Override
27     public User getUserById(Long id) {
28
29         return userRepository.findById(id).orElseThrow(() ->
30             new ResourceNotFoundException("User", "Id", id));
31
32     }
33
34     @Override
35     public void deleteUserById(Long id) {
36
37         userRepository.deleteById(id);
38
39     }
40
41     @Override
```

SportyShoes [SportyShoes2] ~/Downloads: 16

- > .idea
- > .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - security
 - service
 - impl
 - AdminServiceImpl
 - CategoryServiceImpl
 - ProductServiceImpl
 - PurchaseServiceImpl

```
    public CategoryServiceImpl(CategoryRepository categoryRepository) { this.categoryRepository = categoryRepository; }

    @Override
    public Category saveCategory(Category category) { return categoryRepository.save(category); }

    @Override
    public List<Category> getAllCategories() { return categoryRepository.findAll(); }

    @Override
    public Category getCategoryById(long id) {
        return categoryRepository.findById(id).orElseThrow(() ->
            new ResourceNotFoundException("Category", "Id", id));
    }

    @Override
    public Category updateCategory(Category category, long id) {
        // we need to check whether the category with given id exists in DB or not
        Category existingCategory = categoryRepository.findById(id).orElseThrow(
            () -> new ResourceNotFoundException("Category", "Id", id));

        existingCategory.setName(category.getName());
        // save existing category to DB
        categoryRepository.save(existingCategory);
        return existingCategory;
    }

    @Override
    public void deleteCategory(long id) {

        // check whether a category exist in a DB or not
        categoryRepository.findById(id).orElseThrow(() ->
            new ResourceNotFoundException("Category", "Id", id));

        categoryRepository.deleteById(id);
    }
}
```

Project

SportyShoes [SportyShoes2] ~/Downloads

- .idea
- .mvn
- src
- main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - security
 - service
 - impl
 - AdminServiceImpl
 - CategoryServiceImpl
 - ProductServiceImpl
 - PurchaseServiceImpl
 - AdminService
 - CategoryService
 - ProductService
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml

External Libraries

Scrapes and Consoles

Bookmarks

SportyShoes

```
1 package com.myproject.sportyshoes.service.impl;
2
3 import com.myproject.sportyshoes.Exception.ResourceNotFoundException;
4 import com.myproject.sportyshoes.model.Product;
5 import com.myproject.sportyshoes.model.Product;
6 import com.myproject.sportyshoes.repository.CategoryRepository;
7 import com.myproject.sportyshoes.repository.ProductRepository;
8 import com.myproject.sportyshoes.repository.ProductRepository;
9 import com.myproject.sportyshoes.service.ProductService;
10 import org.springframework.stereotype.Service;
11
12 import java.util.List;
13
14 @Service
15 public class ProductServiceImpl implements ProductService {
16     private ProductRepository productRepository;
17
18     public ProductServiceImpl(ProductRepository productRepository) { this.productRepository = productRepository; }
19
20     @Override
21     public Product saveProduct(Product product) {
22         return productRepository.save(product);
23     }
24
25 }
26
27
28     @Override
29     public List<Product> getAllProducts() { return productRepository.findAll(); }
30
31
32     @Override
33     public Product getProductById(long id) {
34         return productRepository.findById(id).orElseThrow(() ->
35             new ResourceNotFoundException("Product", "Id", id));
36     }
37
38
39     @Override
```

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (1 hour ago)

28:14 LF UTF-8 4 spaces

Project

SportyShoes [SportyShoes2] ~/Downloads

- .idea
- .mvn
- src
- main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - security
 - service
 - impl
 - AdminServiceImpl
 - CategoryServiceImpl
 - ProductServiceImpl
 - PurchaseServiceImpl
 - AdminService
 - CategoryService
 - ProductService
 - SportyShoesApplication
 - resources
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - SportyShoes2.iml

External Libraries

Scrapes and Consoles

Bookmarks

SportyShoes

```
1 package com.myproject.sportyshoes.service.impl;
2
3 import com.myproject.sportyshoes.model.Purchase;
4
5 import com.myproject.sportyshoes.repository.PurchaseRepository;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8
9
10 @Service
11 public class PurchaseServiceImpl {
12
13     @Autowired
14     PurchaseRepository repo;
15
16     public Purchase savePurchase(Purchase purchase) { return repo.save(purchase); }
17
18     public String findByDate(String date) { return repo.findByDate(date); }
19
20     public String findByPro_id(long pro_id) { return repo.findByPro_id(pro_id); }
21
22 }
23
24 }
```

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (1 hour ago)

11:14 LF UTF-8 4 spaces

SportyShoes – CategoryController.java

```
import java.util.List;
import org.springframework.web.bind.annotation.*;
import com.myproject.sportyshoes.controller;
import com.myproject.sportyshoes.model.Category;
import com.myproject.sportyshoes.service.CategoryService;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.beans.factory.annotation.Autowired;
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/admin/category")
public class CategoryController {

    @Autowired
    private CategoryService categoryService;

    public CategoryController(CategoryService categoryService) { this.categoryService = categoryService; }

    // build get all categories REST API
    @GetMapping("/getAll")
    public List<Category> getAllCategory() { return categoryService.getAllCategories(); }

    // build get category by id REST API
    @GetMapping("/{id}")
    public ResponseEntity<Category> getCategoryById(@PathVariable("id") Long categoryId){
        return new ResponseEntity<Category>(categoryService.getCategoryById(categoryId), HttpStatus.OK);
    }

    // build update category REST API
    @PutMapping("/{id}")
    public ResponseEntity<Category> updateCategory(@PathVariable("id") long id
                                                    ,@RequestBody Category category){
        return new ResponseEntity<Category>(categoryService.updateCategory(category, id), HttpStatus.OK);
    }

    // build delete category REST API
    @DeleteMapping("/{id}")
    public ResponseEntity<String> deleteCategory(@PathVariable("id") long id){

        // delete category from DB
        categoryService.deleteCategory(id);

        return new ResponseEntity<String>("Category deleted successfully!", HttpStatus.OK);
    }
}
```

```
14
15     @RestController
16     @RequestMapping("/api/admin/products")
17     public class ProductController {
18
19         @Autowired
20         private ProductService productService;
21
22         public ProductController(ProductService productService) {
23             this.productService = productService;
24         }
25
26         //build create product REST API
27         @PostMapping("/save")
28         public ResponseEntity<Product> saveProduct(@RequestBody Product product){
29             return new ResponseEntity<Product>(productService.saveProduct(product), HttpStatus.CREATED);
30         }
31
32         // build get all products REST API
33         @GetMapping("/getAll")
34         public List<Product> getAllProducts(){
35             return productService.getAllProducts();
36
37             // build get product by id REST API
38             @GetMapping("/{id}")
39             public ResponseEntity<Product> getProductById(@PathVariable ("id")long productId){
40                 return new ResponseEntity<Product>(productService.getProductById(productId), HttpStatus.OK);
41
42             // build update product REST API
43             @PutMapping("/{id}")
44             public ResponseEntity<Product> updateProduct(@PathVariable("id") long id
45
46                 ,@RequestBody Product product){
47
48             return new ResponseEntity<Product>(productService.updateProduct(product,id), HttpStatus.OK);
49         }
50     }
51 }
```

Project

SportyShoes [SportyShoes2] ~/Download: 1 package com.myproject.sportyshoes.controller;

src

main

java

com.myproject.sportyshoes

controller

CategoryController

ProductController

PurchaseController

UserController

Dao

Exception

model

repository

security

service

SportyShoesApplication

resources

test

target

.gitignore

HELP.md

mvnw

mvnw.cmd

pom.xml

SportyShoes2.iml

External Libraries

Scratches and Consoles

Extensions

Database Tools and SQL

Java EE: Persistence (JPA)

Bookmarks

SportyShoes

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring Event Log

Flyaway – Dao.java

7 1 ^ v

```
1 package com.myproject.sportyshoes.controller;
2
3 import com.myproject.sportyshoes.model.Purchase;
4 import com.myproject.sportyshoes.service.impl.PurchaseServiceImpl;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.http.HttpStatus;
7 import org.springframework.http.ResponseEntity;
8 import org.springframework.web.bind.annotation.*;
9
10 @RestController
11 @RequestMapping("/api/admin/purchase")
12 public class PurchaseController {
13
14     @Autowired
15     PurchaseServiceImpl service;
16
17     @PostMapping("/save")
18     public ResponseEntity<Object> savePurchase(@RequestBody Purchase purchase){
19         Purchase purchase1= service.savePurchase(purchase);
20
21         if(purchase1!=null)
22             return new ResponseEntity<Object>(purchase1,HttpStatus.CREATED);
23         else
24             return new ResponseEntity<Object>( body: "Error while inserting data",
25                                         HttpStatus.INTERNAL_SERVER_ERROR);
26     }
27
28     @GetMapping("/{date}")
29     public ResponseEntity<String> findPurchase(@PathVariable String date){
30         String resp= service.findByDate(date);
31
32         if(resp!=null)
33             return new ResponseEntity<String>(resp,HttpStatus.FOUND);
34         else
35             return new ResponseEntity<String>( body: "Purchase not available",HttpStatus.NOT_FOUND);
```

Project

SportyShoes [SportyShoes2] ~/Download: 5 import com.myproject.sportyshoes.model.User;

src 6 import com.myproject.sportyshoes.service.AdminService;

main 7 import org.springframework.http.HttpStatus;

java 8 import org.springframework.http.ResponseEntity;

com.myproject.sportyshoes 9

controller 10 import java.util.List;

CategoryController 11

ProductController 12 @RestController

PurchaseController 13 @RequestMapping("/api/admin/users")

UserController 14 public class UserController {

15 @Autowired

16 private AdminService adminService;

17

18 public UserController(AdminService adminService) { this.adminService = adminService; }

19 //build create user REST API

20 @PostMapping("/save")

21 public ResponseEntity<User> saveUser(@RequestBody User user){

22 return new ResponseEntity<User>(adminService.saveUser(user), HttpStatus.CREATED);

23 }

24 // build get all users REST API

25 @GetMapping("/getAll")

26 public List<User> getAllUsers(){

27 return adminService.getAllUsers();

28 }

29

30 // build get user by id REST API

31 @GetMapping("/{id}")

32 public ResponseEntity<User> getUserId(@PathVariable("id") long userId) {

33 return new ResponseEntity<User>(adminService.getUserId(userId), HttpStatus.OK);

34 }

35

36 // build update user REST API

37 @PutMapping("/{id}")

38 public ResponseEntity<User> updateUser(@PathVariable("id") long id,

39

40

41

Bookmarks

SportyShoes

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

Flyaway – Dao.java

4 1 15:15 LF UTF-8 4 spaces

Project

SportyShoes [SportyShoes2] ~/Downloads:

- .idea
- .mvn
- src
 - main
 - java
 - com.myproject.sportyshoes
 - controller
 - Dao
 - Exception
 - model
 - repository
 - security
 - service
 - SportyShoesApplication
 - resources
 - test
 - target
- .gitignore
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml
- SportyShoes2.iml

External Libraries

Scratches and Consoles

- Extensions
 - Database Tools and SQL
 - Java EE: Persistence (JPA)

Bookmarks

- SportyShoes

Version Control Find Run TODO Problems Profiler Terminal Endpoints Build Dependencies Spring

SportyShoesApplication: Failed to retrieve application JMX service URL (today 4:40 AM)

9:14 LF UTF-8 4 spaces

SportyShoes [SportyShoes2] ~/Downloads:

- > .idea
- > .mvn
- ✓ src
 - ✓ main
 - ✓ java
 - ✓ com.myproject.sportyshoes
 - > controller
 - > Dao
 - > Exception
 - > model
 - > repository
 - > security
 - > service
 - SportyShoesApplication
 - > resources
 - > test
 - > target
 - ✓ .gitignore
 - ✓ HELP.md
 - ✓ mvnw
 - ✓ mvnw.cmd
 - ✓ pom.xml
 - ✓ SportyShoes2.iml

> External Libraries

✓ Scratches and Consoles

- ✓ Extensions
 - > Database Tools and SQL
 - > Java EE: Persistence (JPA)

Bookmarks

SportyShoes

```
<?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 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.2</version>
        <relativePath/> 
    </parent>
    <groupId>com.MyProject</groupId>
    <artifactId>SportyShoes</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>SportyShoes</name>
    <description>SportyShoes</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
            <version>2.7.3</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-security</artifactId>
            <version>2.7.3</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
    </dependencies>

```

project