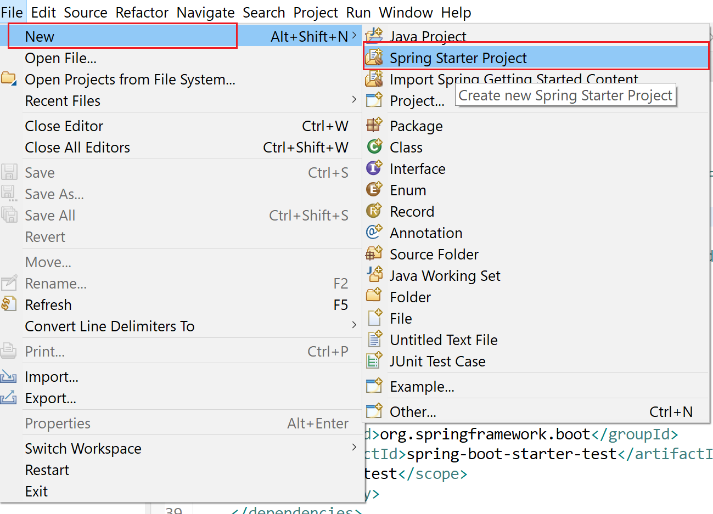
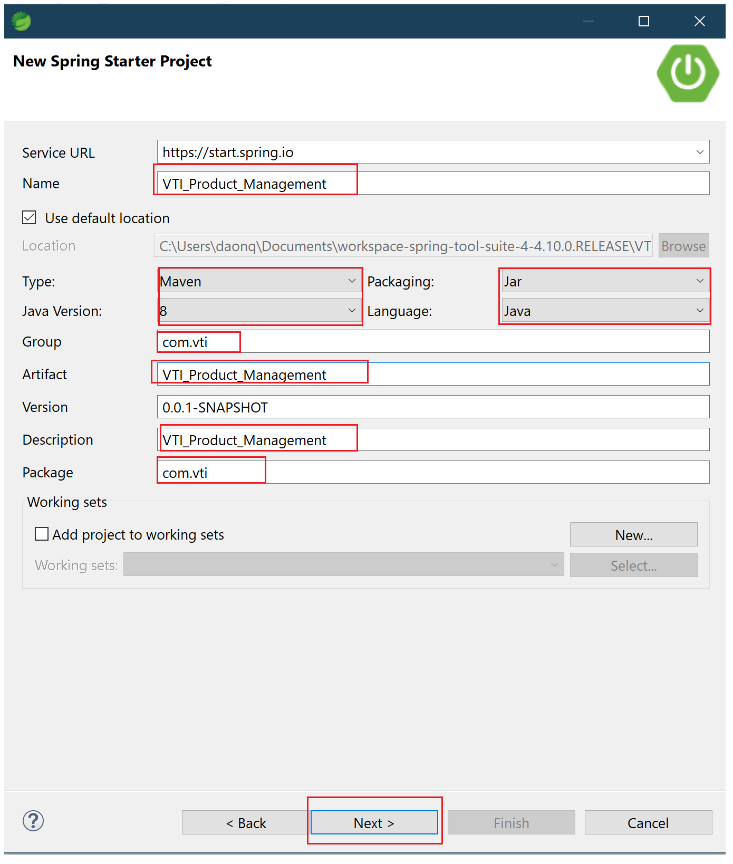
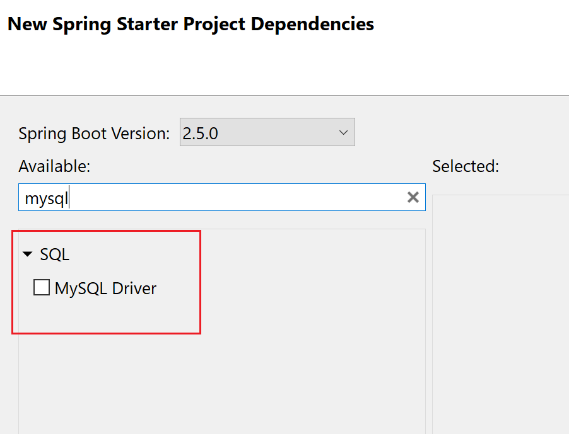
# **Đẩy API cho các bảng dữ liệu trong DB Product:**

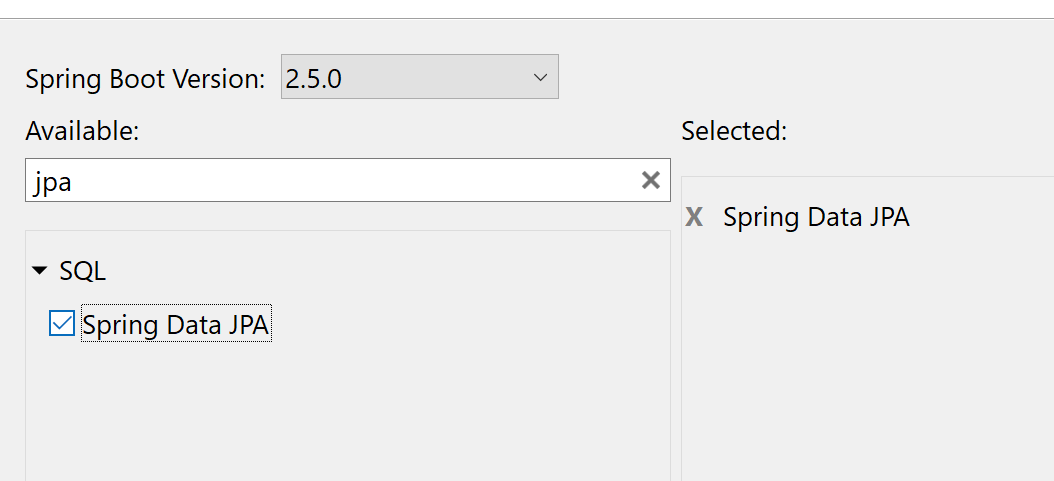
# **Thông tin chung:**

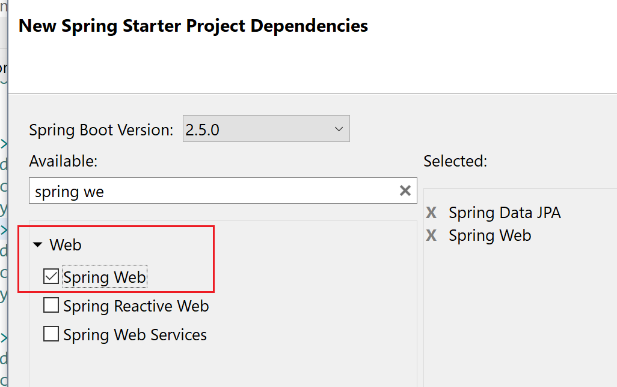
* Thực hiện đẩy API cho bảng dữ liệu.
* Sử dụng mô hình 3 Layer đã học trong học phần JavaCore.

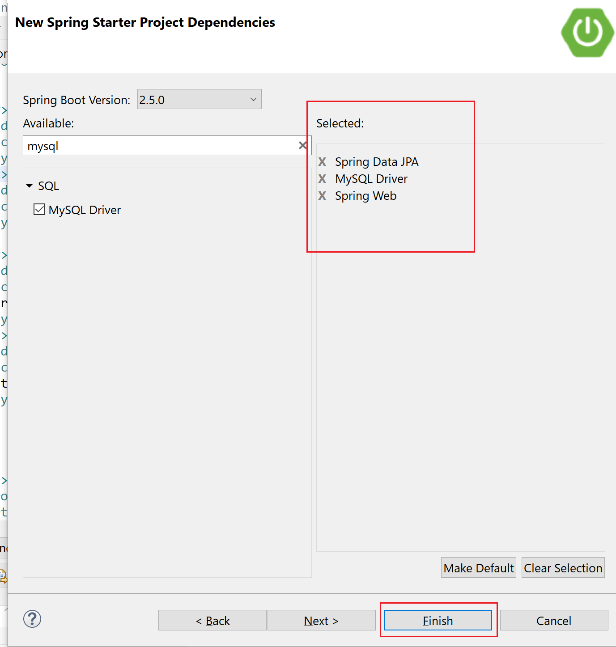
# **Tạo mới 1 Project Spring Boot :**

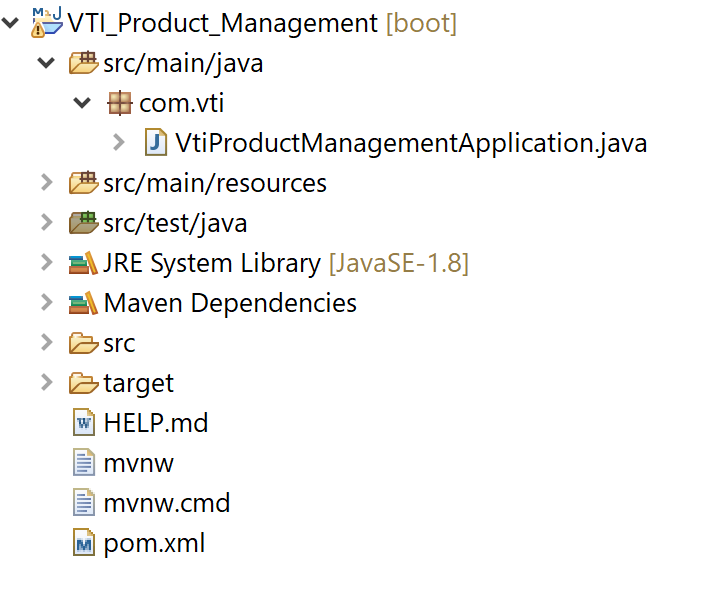
* Tạo mới Project Spring: **VTI\_Product\_Management**
* Tạo mới Project có tên: **VTI\_Product\_Management**
* Lựa chọn các công nghệ để thêm vào Project:



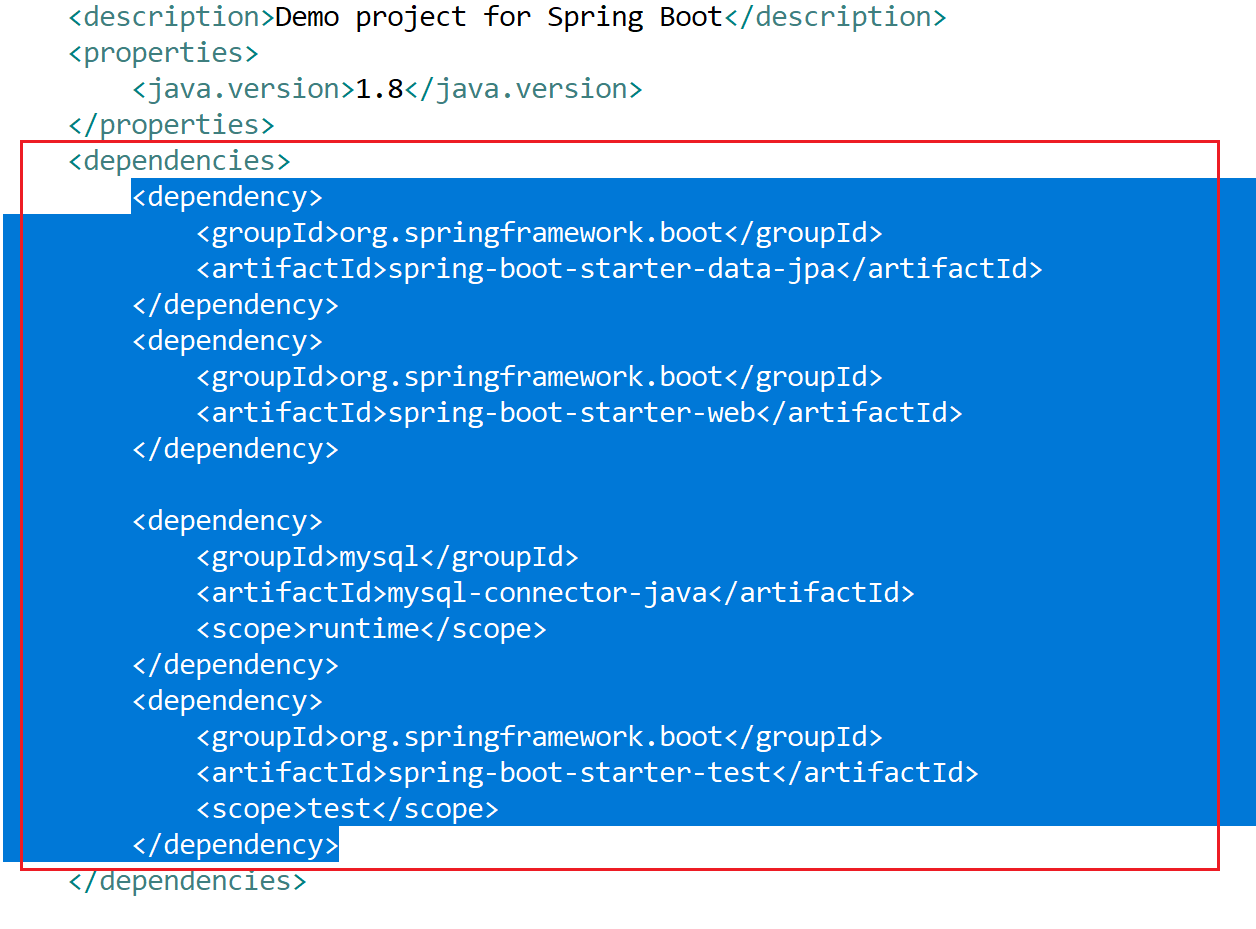




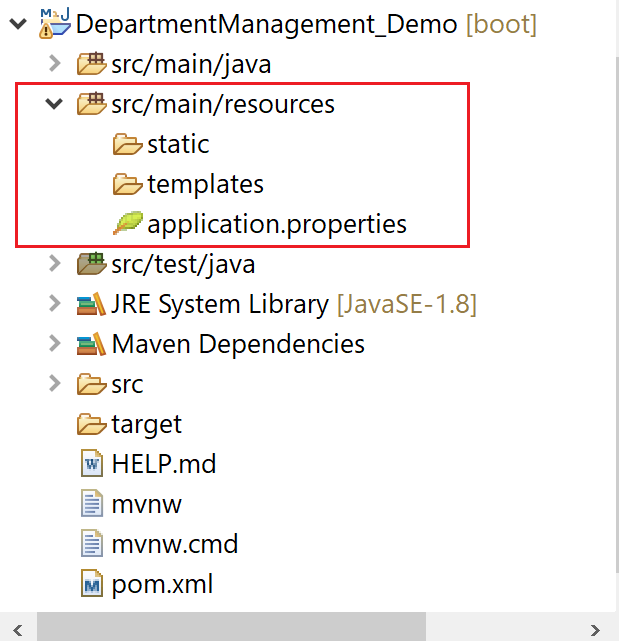




* Kiểm tra lại các Dependecy đã được thêm vào file pom.xml



* Cấu hình lại file **application.properties** để lưu các thông tin kết nối tới file Database, khi tạo mới project, file này nội dung sẽ null.



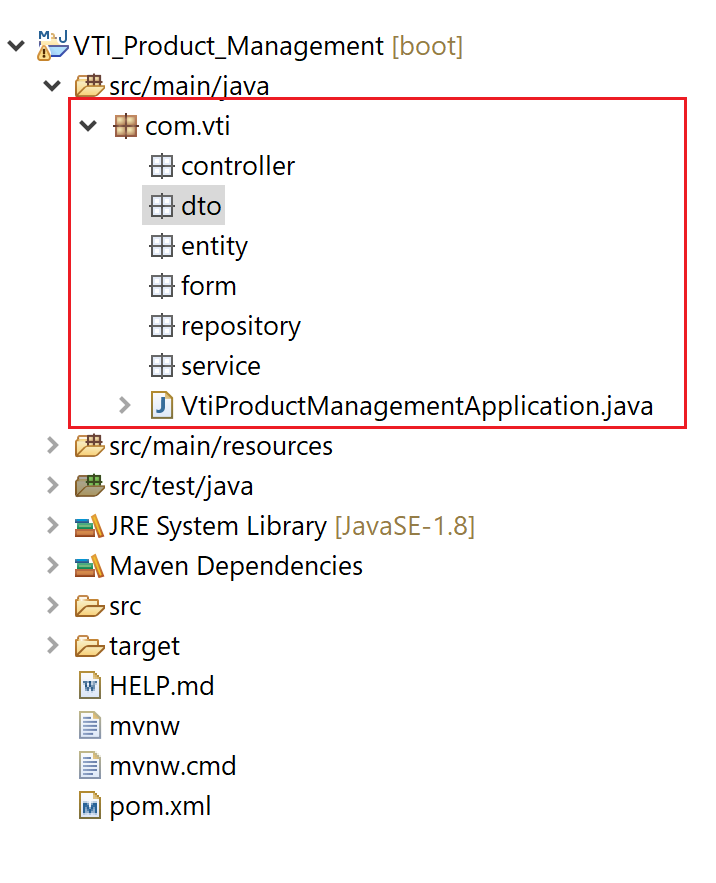
|  |
| --- |
| # -----------------------  # Database  # -----------------------  spring.datasource.url=jdbc:mysql://localhost:3306/ **ProductManagement**?createDatabaseIfNotExist=true  spring.datasource.username=root  spring.datasource.password=root  # -----------------------  # Naming entity mapping table in SQL  # -----------------------  spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl  # -----------------------  # Show sql  # -----------------------  spring.jpa.properties.hibernate.format\_sql=true  logging.level.org.hibernate.SQL=DEBUG  logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE |

# **Tạo DB ProductManagement trong MySqlWorkbench**

* Thực hiện chạy file DB bên dưới trên MySqlWorkbench

|  |
| --- |
|  |

# **Tạo cấu trúc Project**



* Tạo các Package: **com.vti.entity, com.vti.repository, com.vti.service, com.vti.controller, com.vti.form, com.vti.dto.**

# **Tạo Enum Status trong entity**

* Lưu thông tin trạng thái của Account

|  |
| --- |
| **package** com.vti.entity;  **public** **enum** Status {  ***NOT\_ACTIVE***, ***ACTIVE***;  } |

# **Tạo Class ManufacturerName trong entity**

* Lưu thông tin tên Manufacturer.

|  |
| --- |
| **package** com.vti.entity;  **public** **enum** ManufacturerName {  ***SAMSUNG***, ***APPLE***, ***XIAOMI***, ***OPPO***  } |

# **Tạo Class Category trong entity**

|  |
| --- |
| **package** com.vti.entity;  **import** java.io.Serializable;  **import** java.util.List;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.OneToMany;  **import** javax.persistence.Table;  @Entity  @Table(name = "Category", catalog = "ProductManagement")  **public** **class** Category **implements** Serializable {  @Column(name = "CategoryId")  @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **short** id;  @Column(name = "CategoryName", length = 30, nullable = **false**, unique = **true**)  **private** String name;  @OneToMany(mappedBy = "category")  **private** List<Product> products;  **public** Category() {  **super**();  }  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** List<Product> getProducts() {  **return** products;  }  **public** **void** setProducts(List<Product> products) {  **this**.products = products;  }  } |

# **Tạo Class Manufacturer trong entity**

|  |
| --- |
| **package** com.vti.entity;  **import** java.io.Serializable;  **import** java.util.List;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.EnumType;  **import** javax.persistence.Enumerated;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.OneToMany;  **import** javax.persistence.Table;  @Entity  @Table(name = "Manufacturer", catalog = "ProductManagement")  **public** **class** Manufacturer **implements** Serializable {  @Column(name = "ManufacturerId")  @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **short** id;  @Column(name = "ManufacturerName", nullable = **false**, unique = **true**)  @Enumerated(EnumType.***STRING***)  **private** ManufacturerName name;  @OneToMany(mappedBy = "manufacturer")  **private** List<Product> products;  **public** Manufacturer() {  **super**();  }  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** ManufacturerName getName() {  **return** name;  }  **public** **void** setName(ManufacturerName name) {  **this**.name = name;  }  **public** List<Product> getProducts() {  **return** products;  }  **public** **void** setProducts(List<Product> products) {  **this**.products = products;  }  } |

# **Tạo Class Product trong entity**

|  |
| --- |
| **package** com.vti.entity;  **import** java.io.Serializable;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.JoinColumn;  **import** javax.persistence.ManyToOne;  **import** javax.persistence.Table;  @Entity  @Table(name = "Product", catalog = "ProductManagement")  **public** **class** Product **implements** Serializable {  @Column(name = "ProductId")  @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **short** id;  @Column(name = "ProductName", length = 50, nullable = **false**, unique = **true**)  **private** String name;  @Column(name = "ProductPrice", length = 50, nullable = **false**)  **private** String price;  @Column(name = "ProductInfo", length = 200, nullable = **false**)  **private** String info;  @Column(name = "ProductDetail", length = 500)  **private** String detail;  @Column(name = "RatingStar")  **private** **short** ratingStar;  @Column(name = "ProductImageName", length = 50, nullable = **false**)  **private** String imageName;  @ManyToOne  @JoinColumn(name = "ManufacturerId", nullable = **false**)  **private** Manufacturer manufacturer;  @ManyToOne  @JoinColumn(name = "CategoryId", nullable = **false**)  **private** Category category;  **public** Product() {  **super**();  }  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** String getPrice() {  **return** price;  }  **public** **void** setPrice(String price) {  **this**.price = price;  }  **public** String getInfo() {  **return** info;  }  **public** **void** setInfo(String info) {  **this**.info = info;  }  **public** String getDetail() {  **return** detail;  }  **public** **void** setDetail(String detail) {  **this**.detail = detail;  }  **public** **short** getRatingStar() {  **return** ratingStar;  }  **public** **void** setRatingStar(**short** ratingStar) {  **this**.ratingStar = ratingStar;  }  **public** String getImageName() {  **return** imageName;  }  **public** **void** setImageName(String imageName) {  **this**.imageName = imageName;  }  **public** Manufacturer getManufacturer() {  **return** manufacturer;  }  **public** **void** setManufacturer(Manufacturer manufacturer) {  **this**.manufacturer = manufacturer;  }  **public** Category getCategory() {  **return** category;  }  **public** **void** setCategory(Category category) {  **this**.category = category;  }  } |

# **Tạo Class Account trong entity**

|  |
| --- |
| **package** com.vti.entity;  **import** java.io.Serializable;  **import** java.util.Date;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.EnumType;  **import** javax.persistence.Enumerated;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  **import** javax.persistence.Temporal;  **import** javax.persistence.TemporalType;  **import** org.hibernate.annotations.CreationTimestamp;  @Entity  @Table(name = "`Account`", catalog = "ProductManagement")  **public** **class** Account **implements** Serializable {  @Column(name = "AccountID")  @Id  @GeneratedValue(strategy = GenerationType.***IDENTITY***)  **private** **short** id;  @Column(name = "Email", length = 50, nullable = **false**, unique = **true**)  **private** String email;  @Column(name = "Username", length = 50, nullable = **false**, unique = **true**)  **private** String username;  @Column(name = "FullName", length = 50, nullable = **false**)  **private** String fullname;  @Column(name = "AvatarImageName", length = 50, unique = **true**)  **private** String avatarImageName;  @Column(name = "Mobile", length = 50, nullable = **false**, unique = **true**)  **private** String mobile;  @Column(name = "Address", length = 100, nullable = **false**, unique = **true**)  **private** String address;  @Column(name = "CreateDate")  @Temporal(TemporalType.***TIMESTAMP***)  @CreationTimestamp  **private** Date createDate;  @Column(name = "Password", length = 100, nullable = **false**)  **private** String password;  @Column(name = "`Status`")  @Enumerated(EnumType.***ORDINAL***)  **private** Status status = Status.***NOT\_ACTIVE***;  **public** Account() {  **super**();  }  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getEmail() {  **return** email;  }  **public** **void** setEmail(String email) {  **this**.email = email;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getFullname() {  **return** fullname;  }  **public** **void** setFullname(String fullname) {  **this**.fullname = fullname;  }  **public** String getAvatarImageName() {  **return** avatarImageName;  }  **public** **void** setAvatarImageName(String avatarImageName) {  **this**.avatarImageName = avatarImageName;  }  **public** String getMobile() {  **return** mobile;  }  **public** **void** setMobile(String mobile) {  **this**.mobile = mobile;  }  **public** String getAddress() {  **return** address;  }  **public** **void** setAddress(String address) {  **this**.address = address;  }  **public** Date getCreateDate() {  **return** createDate;  }  **public** **void** setCreateDate(Date createDate) {  **this**.createDate = createDate;  }  **public** String getPassword() {  **return** password;  }  **public** **void** setPassword(String password) {  **this**.password = password;  }  **public** Status getStatus() {  **return** status;  }  **public** **void** setStatus(Status status) {  **this**.status = status;  }  } |

# **Tạo Class ProductDto**

* Class này dùng để chuyển đổi giữa dữ liệu lấy được từ DB sang dữ hiển thị đẩy lên Frontend.

|  |
| --- |
| **package** com.vti.dto;  **public** **class** ProductDto {  **private** **short** id;  **private** String name;  **private** String price;  **private** String info;  **private** String detail;  **private** **short** ratingStar;  **private** String imageName;  **private** String manufacturerName;  **private** String categoryName;  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** String getPrice() {  **return** price;  }  **public** **void** setPrice(String price) {  **this**.price = price;  }  **public** String getInfo() {  **return** info;  }  **public** **void** setInfo(String info) {  **this**.info = info;  }  **public** String getDetail() {  **return** detail;  }  **public** **void** setDetail(String detail) {  **this**.detail = detail;  }  **public** **short** getRatingStar() {  **return** ratingStar;  }  **public** **void** setRatingStar(**short** ratingStar) {  **this**.ratingStar = ratingStar;  }  **public** String getImageName() {  **return** imageName;  }  **public** **void** setImageName(String imageName) {  **this**.imageName = imageName;  }  **public** String getManufacturerName() {  **return** manufacturerName;  }  **public** **void** setManufacturerName(String manufacturerName) {  **this**.manufacturerName = manufacturerName;  }  **public** String getCategoryName() {  **return** categoryName;  }  **public** **void** setCategoryName(String categoryName) {  **this**.categoryName = categoryName;  }  } |

# **Tạo Class CategoryDto**

* Class này dùng để chuyển đổi giữa dữ liệu lấy được từ DB sang dữ hiển thị đẩy lên Frontend.

|  |
| --- |
| **package** com.vti.dto;  **public** **class** CategoryDto {  **private** **short** id;  **private** String name;  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }    } |

# **Tạo Class ManufacturerDto**

* Class này dùng để chuyển đổi giữa dữ liệu lấy được từ DB sang dữ hiển thị đẩy lên Frontend.

|  |
| --- |
| **package** com.vti.dto;  **public** **class** ManufacturerDto {  **private** **short** id;  **private** String name;  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  } |

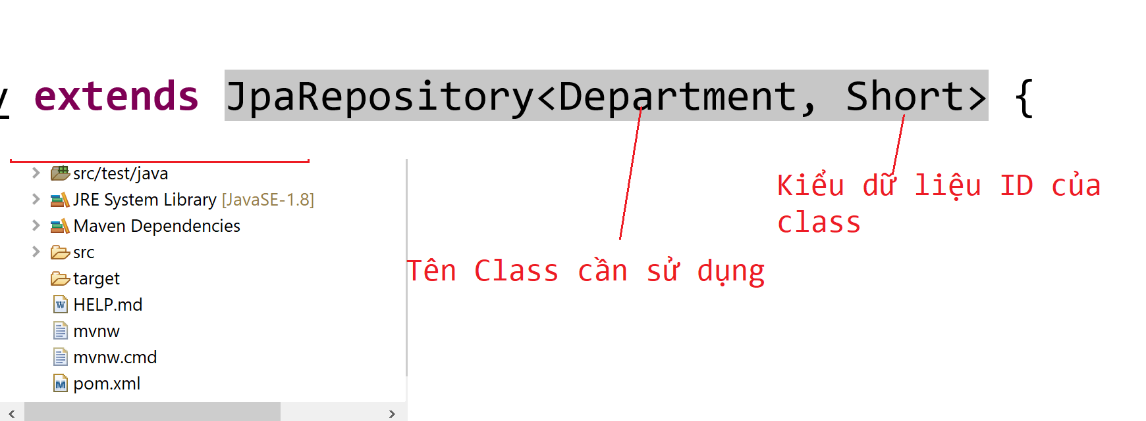
# **Tạo Class AccountDto**

* Class này dùng để chuyển đổi giữa dữ liệu lấy được từ DB sang dữ hiển thị đẩy lên Frontend.

|  |
| --- |
| **package** com.vti.dto;  **import** java.util.Date;  **import** com.fasterxml.jackson.annotation.JsonFormat;  **public** **class** AccountDto {  **private** **short** id;  **private** String email;  **private** String username;  **private** String fullname;  **private** String avatarImageName;  **private** String mobile;  **private** String address;  @JsonFormat(pattern = "yyyy-MM-dd")  **private** Date createDate;  **private** String status;  **public** **short** getId() {  **return** id;  }  **public** **void** setId(**short** id) {  **this**.id = id;  }  **public** String getEmail() {  **return** email;  }  **public** **void** setEmail(String email) {  **this**.email = email;  }  **public** String getUsername() {  **return** username;  }  **public** **void** setUsername(String username) {  **this**.username = username;  }  **public** String getFullname() {  **return** fullname;  }  **public** **void** setFullname(String fullname) {  **this**.fullname = fullname;  }  **public** String getAvatarImageName() {  **return** avatarImageName;  }  **public** **void** setAvatarImageName(String avatarImageName) {  **this**.avatarImageName = avatarImageName;  }  **public** String getMobile() {  **return** mobile;  }  **public** **void** setMobile(String mobile) {  **this**.mobile = mobile;  }  **public** String getAddress() {  **return** address;  }  **public** **void** setAddress(String address) {  **this**.address = address;  }  **public** Date getCreateDate() {  **return** createDate;  }  **public** **void** setCreateDate(Date createDate) {  **this**.createDate = createDate;  }  **public** String getStatus() {  **return** status;  }  **public** **void** setStatus(String status) {  **this**.status = status;  }  } |

# **Tạo Interface IManufacturerRepository**

* Trong phần này Extend JpaRepository của Spring data JPA, chỉ cần Extend mà k cần làm thêm gì, do 1 số phương thức hay dùng JPA đã hỗ trợ sẵn.



|  |
| --- |
| **package** com.vti.repository;  **import** org.springframework.data.jpa.repository.JpaRepository;  **import** com.vti.entity.Manufacturer;  **public** **interface** IManufacturerRepository **extends** JpaRepository<Manufacturer, Short> {  } |

# **Tạo Interface ICategoryRepository**

* Trong phần này chú ý Extend JpaRepository của Spring data JPA, chỉ cần Extend mà k cần làm thêm gì, do 1 số phương thức hay dùng JPA đã hỗ trợ sẵn.

|  |
| --- |
| **package** com.vti.repository;  **import** org.springframework.data.jpa.repository.JpaRepository;  **import** com.vti.entity.Category;  **public** **interface** ICategoryRepository **extends** JpaRepository<Category, Short> {  } |

# **Tạo Interface IProductRepository**

* Trong phần này chú ý Extend JpaRepository của Spring data JPA, chỉ cần Extend mà k cần làm thêm gì, do 1 số phương thức hay dùng JPA đã hỗ trợ sẵn.

|  |
| --- |
| **package** com.vti.repository;  **import** org.springframework.data.jpa.repository.JpaRepository;  **import** com.vti.entity.Product;  **public** **interface** IProductRepository **extends** JpaRepository<Product, Short> {  } |

# **Tạo Interface IAccountRepository**

* Trong phần này chú ý Extend JpaRepository của Spring data JPA, chỉ cần Extend mà k cần làm thêm gì, do 1 số phương thức hay dùng JPA đã hỗ trợ sẵn.

|  |
| --- |
| **package** com.vti.repository;  **import** org.springframework.data.jpa.repository.JpaRepository;  **import** com.vti.entity.Account;  **public** **interface** IAccountRepository **extends** JpaRepository<Account, Short> {  } |

# **Tạo Interface IManufacturerService**

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** com.vti.entity.Manufacturer;  **public** **interface** IManufacturerService {  **public** List<Manufacturer> getAllManufacturers();  } |

# **Tạo Interface ICategoryService**

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** com.vti.entity.Category;  **public** **interface** ICategoryService {  **public** List<Category> getAllCategorys();  } |

# **Tạo Interface IProductService**

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** com.vti.entity.Product;  **import** com.vti.form.ProductFormForCreating;  **import** com.vti.form.ProductFormForUpdating;  **public** **interface** IProductService {  **public** List<Product> getAllProducts();  **public** Product getProductById(**short** id);  **public** Product createNewProduct(ProductFormForCreating productNewForm);  **public** Product updateProduct(**short** id, ProductFormForUpdating productUpdateForm);  **public** **void** deleteProductById(**short** id);  } |

# **Tạo Interface IAccountService**

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** com.vti.entity.Account;  **public** **interface** IAccountService {  **public** List<Account> getAllAccounts();  } |

# **Tạo Class ManufacturerService**

* Trong Class này sẽ gọi các phương thức của JPA đã xây dựng sẵn thông qua Repository.

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Service;  **import** com.vti.entity.Manufacturer;  **import** com.vti.repository.IManufacturerRepository;  @Service  **public** **class** ManufacturerService **implements** IManufacturerService {  @Autowired  **private** IManufacturerRepository manufacturerRepository;  @Override  **public** List<Manufacturer> getAllManufacturers() {  **return** manufacturerRepository.findAll();  }  } |

# **Tạo Class CategoryService**

* Trong Class này sẽ gọi các phương thức của JPA đã xây dựng sẵn thông qua: Repository.

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Service;  **import** com.vti.entity.Category;  **import** com.vti.repository.ICategoryRepository;  @Service  **public** **class** CategoryService **implements** ICategoryService {  @Autowired  **private** ICategoryRepository categoryRepository;  @Override  **public** List<Category> getAllCategorys() {  **return** categoryRepository.findAll();  }  } |

# **Tạo Class ProductService**

* Trong Class này sẽ gọi các phương thức của JPA đã xây dựng sẵn thông qua: Repository.

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Service;  **import** com.vti.entity.Category;  **import** com.vti.entity.Manufacturer;  **import** com.vti.entity.Product;  **import** com.vti.form.ProductFormForCreating;  **import** com.vti.form.ProductFormForUpdating;  **import** com.vti.repository.ICategoryRepository;  **import** com.vti.repository.IManufacturerRepository;  **import** com.vti.repository.IProductRepository;  @Service  **public** **class** ProductService **implements** IProductService {  @Autowired  **private** IProductRepository productRepository;  @Autowired  **private** IManufacturerRepository manufacturerRepository;  @Autowired  **private** ICategoryRepository categoryRepository;  @Override  **public** List<Product> getAllProducts() {  **return** productRepository.findAll();  }  @Override  **public** Product getProductById(**short** id) {  **return** productRepository.getById(id);  }  @Override  **public** **void** deleteProductById(**short** id) {  productRepository.deleteById(id);  }  @Override  **public** Product createNewProduct(ProductFormForCreating productNewForm) {  // Tìm manufacturer theo id  Manufacturer manufacturer = manufacturerRepository.getById(productNewForm.getManufacturerId());  // Tìm manufacturer theo id  Category category = categoryRepository.getById(productNewForm.getCategoryId());  Product product = **new** Product();  product.setName(productNewForm.getName());  product.setPrice(productNewForm.getPrice());  product.setInfo(productNewForm.getInfo());  product.setDetail(productNewForm.getDetail());  product.setRatingStar(productNewForm.getRatingStar());  product.setImageName(productNewForm.getImageName());  product.setManufacturer(manufacturer);  product.setCategory(category);  Product productNew = productRepository.save(product);  **return** productNew;  }  @Override  **public** Product updateProduct(**short** id, ProductFormForUpdating productUpdateForm) {  Product product = productRepository.getById(id);  // Tìm manufacturer theo id  Manufacturer manufacturer = manufacturerRepository.getById(productUpdateForm.getManufacturerId());  // Tìm manufacturer theo id  Category category = categoryRepository.getById(productUpdateForm.getCategoryId());  product.setName(productUpdateForm.getName());  product.setPrice(productUpdateForm.getPrice());  product.setInfo(productUpdateForm.getInfo());  product.setDetail(productUpdateForm.getDetail());  product.setRatingStar(productUpdateForm.getRatingStar());  product.setImageName(productUpdateForm.getImageName());  product.setManufacturer(manufacturer);  product.setCategory(category);  Product productUpdate = productRepository.save(product);  return productUpdate;  }  } |

# **Tạo Class AccountService**

* Trong Class này sẽ gọi các phương thức của JPA đã xây dựng sẵn thông qua: Repository.

|  |
| --- |
| **package** com.vti.service;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Service;  **import** com.vti.entity.Account;  **import** com.vti.repository.IAccountRepository;  @Service  **public** **class** AccountService **implements** IAccountService {  @Autowired  **private** IAccountRepository accountRepository;  @Override  **public** List<Account> getAllAccounts() {  **return** accountRepository.findAll();  }  } |

# **Tạo Class ProductFormForCreating trong Package form**

|  |
| --- |
| **package** com.vti.form;  **public** **class** ProductFormForCreating {  **private** String name;  **private** String price;  **private** String info;  **private** String detail;  **private** **short** ratingStar;  **private** String imageName;  **private** **short** manufacturerId;  **private** **short** categoryId;  **public** ProductFormForCreating() {  **super**();  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** String getPrice() {  **return** price;  }  **public** **void** setPrice(String price) {  **this**.price = price;  }  **public** String getInfo() {  **return** info;  }  **public** **void** setInfo(String info) {  **this**.info = info;  }  **public** String getDetail() {  **return** detail;  }  **public** **void** setDetail(String detail) {  **this**.detail = detail;  }  **public** **short** getRatingStar() {  **return** ratingStar;  }  **public** **void** setRatingStar(**short** ratingStar) {  **this**.ratingStar = ratingStar;  }  **public** String getImageName() {  **return** imageName;  }  **public** **void** setImageName(String imageName) {  **this**.imageName = imageName;  }  **public** **short** getManufacturerId() {  **return** manufacturerId;  }  **public** **void** setManufacturerId(**short** manufacturerId) {  **this**.manufacturerId = manufacturerId;  }  **public** **short** getCategoryId() {  **return** categoryId;  }  **public** **void** setCategoryId(**short** categoryId) {  **this**.categoryId = categoryId;  }  } |

# **Tạo Class ProductFormForUpdating trong Package form**

|  |
| --- |
| **package** com.vti.form;  **public** **class** ProductFormForUpdating {  **private** String name;  **private** String price;  **private** String info;  **private** String detail;  **private** **short** ratingStar;  **private** String imageName;  **private** **short** manufacturerId;  **private** **short** categoryId;  **public** ProductFormForUpdating() {  **super**();  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  **public** String getPrice() {  **return** price;  }  **public** **void** setPrice(String price) {  **this**.price = price;  }  **public** String getInfo() {  **return** info;  }  **public** **void** setInfo(String info) {  **this**.info = info;  }  **public** String getDetail() {  **return** detail;  }  **public** **void** setDetail(String detail) {  **this**.detail = detail;  }  **public** **short** getRatingStar() {  **return** ratingStar;  }  **public** **void** setRatingStar(**short** ratingStar) {  **this**.ratingStar = ratingStar;  }  **public** String getImageName() {  **return** imageName;  }  **public** **void** setImageName(String imageName) {  **this**.imageName = imageName;  }  **public** **short** getManufacturerId() {  **return** manufacturerId;  }  **public** **void** setManufacturerId(**short** manufacturerId) {  **this**.manufacturerId = manufacturerId;  }  **public** **short** getCategoryId() {  **return** categoryId;  }  **public** **void** setCategoryId(**short** categoryId) {  **this**.categoryId = categoryId;  }  } |

# **Tạo Class ManufacturerController**

|  |
| --- |
| **package** com.vti.controller;  **import** java.util.ArrayList;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.http.HttpStatus;  **import** org.springframework.http.ResponseEntity;  **import** org.springframework.web.bind.annotation.CrossOrigin;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.vti.dto.CategoryDto;  **import** com.vti.dto.ManufacturerDto;  **import** com.vti.entity.Category;  **import** com.vti.entity.Manufacturer;  **import** com.vti.service.IManufacturerService;  @RestController  @RequestMapping(value = "api/v1/manufacturers")  @CrossOrigin("\*")  **public** **class** ManufacturerController {  @Autowired  **private** IManufacturerService manufacturerService;  @GetMapping()  **public** ResponseEntity<?> getAllManufacturers() {  List<Manufacturer> manufacturerListDB = manufacturerService.getAllManufacturers();  List<ManufacturerDto> manufacturerListDto = **new** ArrayList<>();  // convert manufacturerListDB --> manufacturerListDto  **for** (Manufacturer manufacturerDB : manufacturerListDB) {  ManufacturerDto manufacturerDto = **new** ManufacturerDto();  manufacturerDto.setId(manufacturerDB.getId());  manufacturerDto.setName(manufacturerDB.getName().toString());  manufacturerListDto.add(manufacturerDto);  }  **return** **new** ResponseEntity<>(manufacturerListDto, HttpStatus.OK);  }  } |

# **Tạo Class CategoryController**

|  |
| --- |
| **package** com.vti.controller;  **import** java.util.ArrayList;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.http.HttpStatus;  **import** org.springframework.http.ResponseEntity;  **import** org.springframework.web.bind.annotation.CrossOrigin;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.vti.dto.CategoryDto;  **import** com.vti.entity.Category;  **import** com.vti.service.ICategoryService;  @RestController  @RequestMapping(value = "api/v1/categorys")  @CrossOrigin("\*")  **public** **class** CategoryController {  @Autowired  **private** ICategoryService categoryService;  @GetMapping()  **public** ResponseEntity<?> getAllCategorys() {  List<Category> categoryListDB = categoryService.getAllCategorys();  List<CategoryDto> categoryListDto = **new** ArrayList<>();    // convert categoryListDB --> categoryListDto  **for** (Category categoryDB : categoryListDB) {  CategoryDto categoryDto = **new** CategoryDto();  categoryDto.setId(categoryDB.getId());  categoryDto.setName(categoryDB.getName());    categoryListDto.add(categoryDto);  }  **return** **new** ResponseEntity<>(categoryListDto, HttpStatus.OK);  }  } |

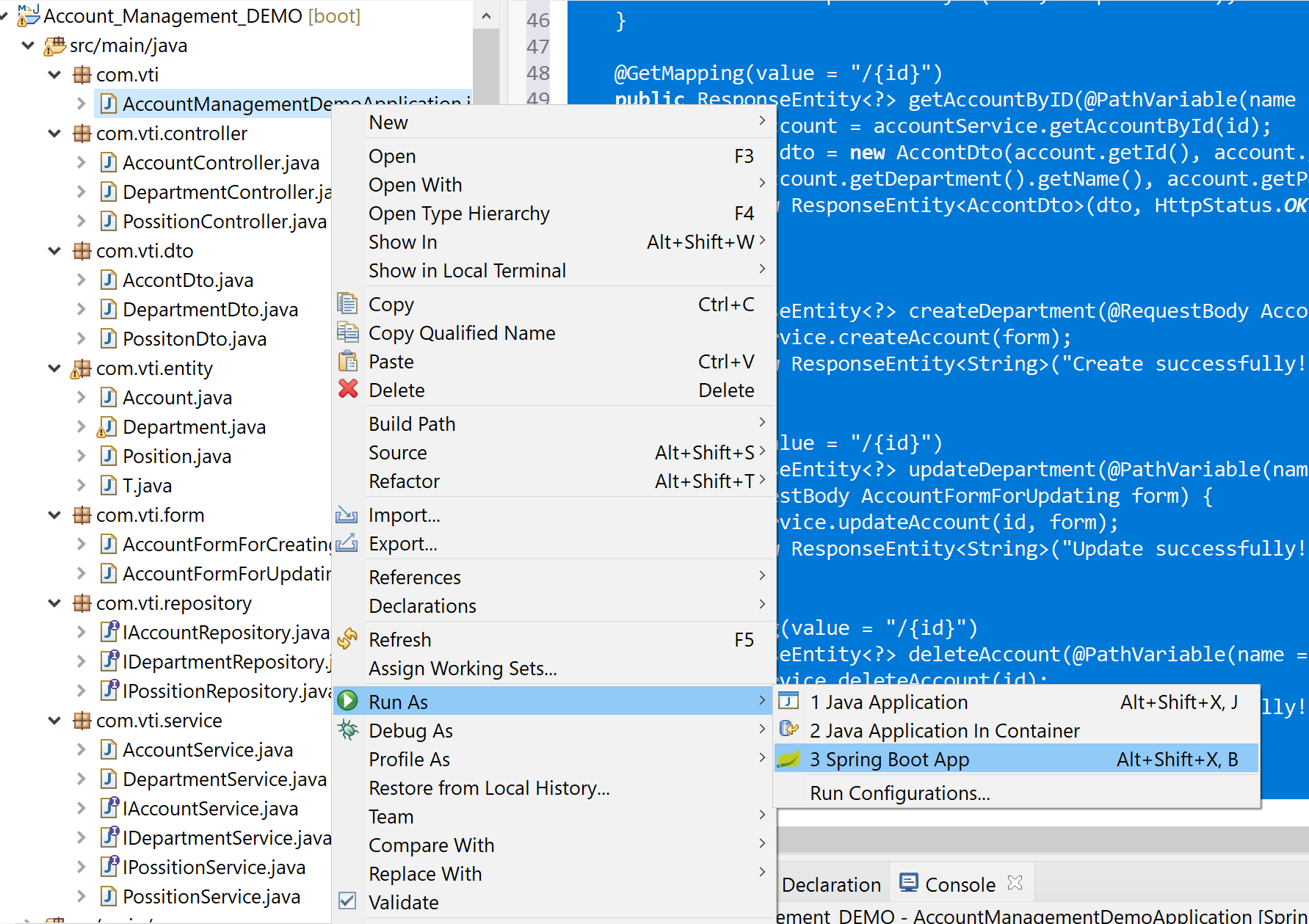
# **Tạo Class ProductController**

|  |
| --- |
| **package** com.vti.controller;  **import** java.util.ArrayList;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.http.HttpStatus;  **import** org.springframework.http.ResponseEntity;  **import** org.springframework.web.bind.annotation.CrossOrigin;  **import** org.springframework.web.bind.annotation.DeleteMapping;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.PathVariable;  **import** org.springframework.web.bind.annotation.PostMapping;  **import** org.springframework.web.bind.annotation.PutMapping;  **import** org.springframework.web.bind.annotation.RequestBody;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.vti.dto.ProductDto;  **import** com.vti.entity.Product;  **import** com.vti.form.ProductFormForCreating;  **import** com.vti.form.ProductFormForUpdating;  **import** com.vti.service.IProductService;  @RestController  @RequestMapping(value = "api/v1/products")  @CrossOrigin("\*")  **public** **class** ProductController {  @Autowired  **private** IProductService productService;  // Lấy All danh sách sản phẩm  @GetMapping()  **public** ResponseEntity<?> getAllProducts() {  List<Product> productListDB = productService.getAllProducts();  List<ProductDto> productListDto = **new** ArrayList<>();  // convert productListDB --> productListDto  **for** (Product productDB : productListDB) {  ProductDto productDto = **new** ProductDto();  productDto.setId(productDB.getId());  productDto.setName(productDB.getName());  productDto.setPrice(productDB.getPrice());  productDto.setInfo(productDB.getInfo());  productDto.setDetail(productDB.getDetail());  productDto.setRatingStar(productDB.getRatingStar());  productDto.setImageName(productDB.getImageName());  productDto.setManufacturerName(productDB.getManufacturer().getName().toString());  productDto.setCategoryName(productDB.getCategory().getName());  productListDto.add(productDto);  }  **return** **new** ResponseEntity<>(productListDto, HttpStatus.OK);  }  // Tìm sản phẩm theo id  @GetMapping(value = "/{id}")  **public** ResponseEntity<?> getProductByID(@PathVariable(name = "id") **short** id) {  **try** {  Product productDB = productService.getProductById(id);  ProductDto productDto = **new** ProductDto();  productDto.setId(productDB.getId());  productDto.setName(productDB.getName());  productDto.setPrice(productDB.getPrice());  productDto.setInfo(productDB.getInfo());  productDto.setDetail(productDB.getDetail());  productDto.setRatingStar(productDB.getRatingStar());  productDto.setImageName(productDB.getImageName());  productDto.setManufacturerName(productDB.getManufacturer().getName().toString());  productDto.setCategoryName(productDB.getCategory().getName());  **return** **new** ResponseEntity<>(productDto, HttpStatus.OK);  } **catch** (Exception e) {  **return** **new** ResponseEntity<>("Not found", HttpStatus.NOT\_FOUND);  }  }  // Thêm mới sản phẩm  @PostMapping()  **public** ResponseEntity<?> createNewproduct(@RequestBody ProductFormForCreating productNewForm) {  **try** {  // Thêm mới Product  // Sau khi thêm mới, trả về thông tin Product vừa thêm  Product productNew = productService.createNewProduct(productNewForm);  // Convert  ProductDto productNewDto = **new** ProductDto();  productNewDto.setId(productNew.getId());  productNewDto.setName(productNew.getName());  productNewDto.setPrice(productNew.getPrice());  productNewDto.setInfo(productNew.getInfo());  productNewDto.setDetail(productNew.getDetail());  productNewDto.setRatingStar(productNew.getRatingStar());  productNewDto.setImageName(productNew.getImageName());  productNewDto.setManufacturerName(productNew.getManufacturer().getName().toString());  productNewDto.setCategoryName(productNew.getCategory().getName());  **return** **new** ResponseEntity<>(productNewDto, HttpStatus.CREATED);  } **catch** (Exception e) {  **return** **new** ResponseEntity<>("Not OK", HttpStatus.BAD\_REQUEST);  }  }  // Update sản phẩm  @PutMapping(value = "/{id}")  **public** ResponseEntity<?> updateProduct(@PathVariable(name = "id") **short** id,  @RequestBody ProductFormForUpdating productUpdateForm) {  **try** {  // Thực hiện Update Product  // Sau khi Update, trả về thông tin Product vừa Update  Product productUpdate = productService.updateProduct(id, productUpdateForm);  // Convert  ProductDto productUpdateDto = **new** ProductDto();  productUpdateDto.setId(productUpdate.getId());  productUpdateDto.setName(productUpdate.getName());  productUpdateDto.setPrice(productUpdate.getPrice());  productUpdateDto.setInfo(productUpdate.getInfo());  productUpdateDto.setDetail(productUpdate.getDetail());  productUpdateDto.setRatingStar(productUpdate.getRatingStar());  productUpdateDto.setImageName(productUpdate.getImageName());  productUpdateDto.setManufacturerName(productUpdate.getManufacturer().getName().toString());  productUpdateDto.setCategoryName(productUpdate.getCategory().getName());  **return** **new** ResponseEntity<>(productUpdateDto, HttpStatus.OK);  } **catch** (Exception e) {  **return** **new** ResponseEntity<>("Not found", HttpStatus.NOT\_FOUND);  }  }  // Xóa sản phẩm theo id  @DeleteMapping(value = "/{id}")  **public** ResponseEntity<?> deleteProductById(@PathVariable(name = "id") **short** id) {  **try** {  // Thực hiện lấy thông tin chi tiết của Product cần xóa, sau khi xóa xong thực hiện trả về thông tin chi tiết  // của Product vừa xóa  Product productDelete = productService.getProductById(id);  // Convert  ProductDto productDeleteDto = **new** ProductDto();  productDeleteDto.setId(productDelete.getId());  productDeleteDto.setName(productDelete.getName());  productDeleteDto.setPrice(productDelete.getPrice());  productDeleteDto.setInfo(productDelete.getInfo());  productDeleteDto.setDetail(productDelete.getDetail());  productDeleteDto.setRatingStar(productDelete.getRatingStar());  productDeleteDto.setImageName(productDelete.getImageName());  productDeleteDto.setManufacturerName(productDelete.getManufacturer().getName().toString());  productDeleteDto.setCategoryName(productDelete.getCategory().getName());  // Xóa Product  productService.deleteProductById(id);  **return** **new** ResponseEntity<>(productDeleteDto, HttpStatus.OK);  } **catch** (Exception e) {  **return** **new** ResponseEntity<>("Not found", HttpStatus.NOT\_FOUND);  }  }  } |

# **Tạo Class AccountController**

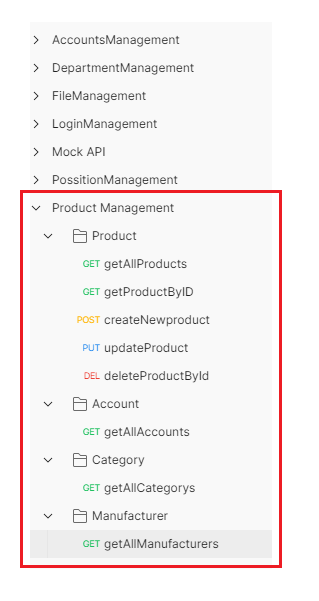
|  |
| --- |
| **package** com.vti.controller;  **import** java.util.ArrayList;  **import** java.util.List;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.http.HttpStatus;  **import** org.springframework.http.ResponseEntity;  **import** org.springframework.web.bind.annotation.CrossOrigin;  **import** org.springframework.web.bind.annotation.GetMapping;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.RestController;  **import** com.vti.dto.AccountDto;  **import** com.vti.entity.Account;  **import** com.vti.service.IAccountService;  @RestController  @RequestMapping(value = "api/v1/accounts")  @CrossOrigin("\*")  **public** **class** AccountController {  @Autowired  **private** IAccountService accountService;  @GetMapping()  **public** ResponseEntity<?> getAllAccounts() {  List<Account> accountListDB = accountService.getAllAccounts();  List<AccountDto> accountListDto = **new** ArrayList<>();  // convert accountListDB --> accountListDto  **for** (Account accountDB : accountListDB) {  AccountDto accountDto = **new** AccountDto();  accountDto.setId(accountDB.getId());  accountDto.setEmail(accountDB.getEmail());  accountDto.setUsername(accountDB.getUsername());  accountDto.setFullname(accountDB.getFullname());  accountDto.setAvatarImageName(accountDB.getAvatarImageName());  accountDto.setMobile(accountDB.getMobile());  accountDto.setAddress(accountDB.getAddress());  accountDto.setCreateDate(accountDB.getCreateDate());  accountDto.setStatus(accountDB.getStatus().toString());  accountListDto.add(accountDto);  }  **return** **new** ResponseEntity<>(accountListDto, HttpStatus.OK);  }  } |

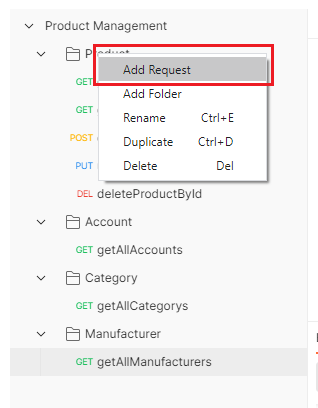
# Chạy chương trình Demo kết quả



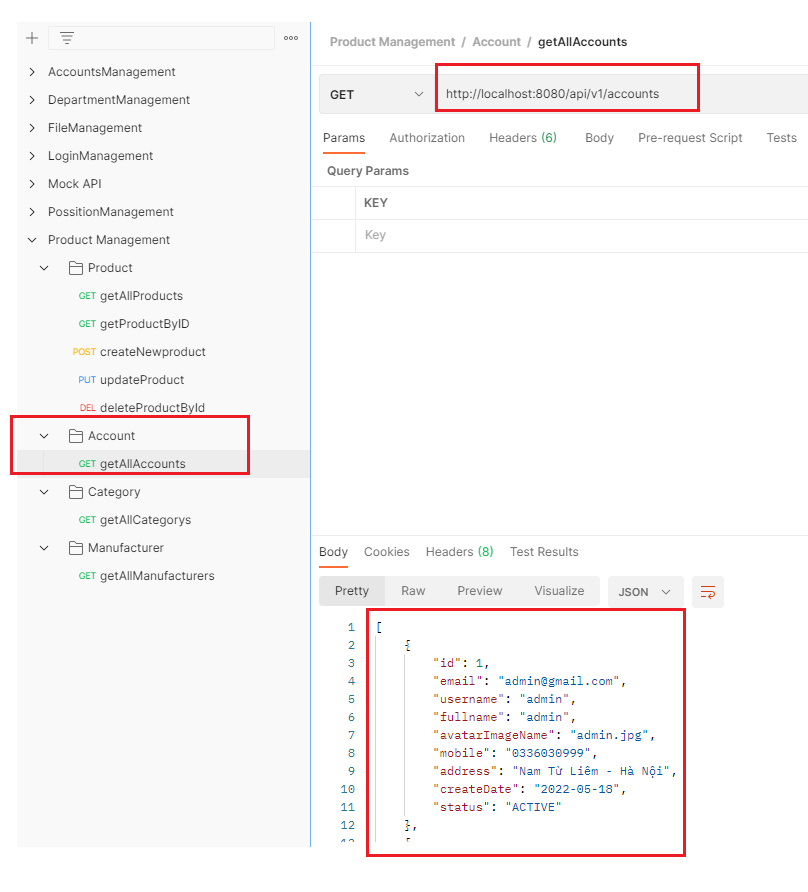
# Sử dụng Postman để Demo kết quả API với Account

* Tạo cấu trúc thư mục và add các Request như bên dưới trên Postman.





* VD: getAllAccounts:



|  |
| --- |
| http://localhost:8080/api/v1/accounts |

* Thực hiện tương tự để tạo các Request khác, với thông tin như bảng bên dưới.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group** | **Request Name** | **Methor** | **Link** | **Data** |
| **Product** | getAllProducts | GET | http://localhost:8080/api/v1/products |  |
| getProductByID | GET | http://localhost:8080/api/v1/products/9 |  |
| createNewproduct | POST | http://localhost:8080/api/v1/products | {  "name": "OPPO Reno7 NEW",  "price": "8.490.000₫ NEW",  "info": "Thiết kế mạnh mẽ, lịch lãm NEW",  "detail": "detail Reno7 NEW",  "ratingStar": "4",  "imageName": "Reno7\_NEW.jpg",  "manufacturerId": "4",  "categoryId": "1"  } |
| updateProduct | PUT | http://localhost:8080/api/v1/products/3 | {  "name": "Name Product Update",  "price": "8.490.000₫ Update",  "info": "Thiết kế mạnh mẽ, lịch lãm Update",  "detail": "detail Update",  "ratingStar": "5",  "imageName": "Update.jpg",  "manufacturerId": "2",  "categoryId": "2"  } |
| deleteProductById | DELETE | http://localhost:8080/api/v1/products/8 |  |
| **Account** | getAllAccounts | GET | http://localhost:8080/api/v1/accounts |  |
| **Category** | getAllCategorys | GET | http://localhost:8080/api/v1/categorys |  |
| **Manufacturer** | getAllManufacturers | GET | http://localhost:8080/api/v1/manufacturers |  |

* Với yêu cầu thêm mới hoặc update cần tạo thêm thông tin Account thì tạo tương tự như dưới (Hình minh họa)

