**Mục Lục**

[1. Chuẩn bị : 1](#_Toc511543056)

[2. Phân tích Database : 2](#_Toc511543057)

[2.1. Categories : Danh mục sản phẩm 2](#_Toc511543058)

[2.2. Products : Sản Phẩm 3](#_Toc511543059)

[2.3. ProductImages : Hình ảnh của sản phẩm 4](#_Toc511543060)

[2.4. ProductInStock : Kho 4](#_Toc511543061)

[2.5. Orders : Hóa đơn 5](#_Toc511543062)

[2.6. OrderDetails : Chi Tiết Hóa Đơn 5](#_Toc511543063)

[2.7. Customers : Đây là User 5](#_Toc511543064)

[2.8. Tags : Từ khóa của sản phẩm 6](#_Toc511543065)

[2.9. ProductTags 6](#_Toc511543066)

[3. Lưu ý : 6](#_Toc511543067)

[4. Tổ chức dự án : 6](#_Toc511543068)

[5. Tổ chức layout : Step by step 7](#_Toc511543069)

[6. Xử lý Trang Home : 8](#_Toc511543070)

[7. Tạo BaseController dùng chung : 8](#_Toc511543071)

[8. Tiến hành Mapping model => table : 10](#_Toc511543072)

[9. Tạo Component ( có tên là @Repository ) để sau này cần dùng ta tiêm vào : Xây dựng DAO ( package com.nghiepuit.dao ) 17](#_Toc511543073)

[10. Tạo component Service xử lý nghiệp vụ : ( @Service và @Transactional để quản lý transaction ) 18](#_Toc511543074)

1. **Chuẩn bị :**

\_ Phần 1 : Website phần cho khách hàng

+ Thiết kế :

* Giao diện.
* Cơ sở dữ liệu ( dùng Hibernate ).
* Chuẩn bị : thư viện, cấu hình, dự án mẫu.

+ Trưng bày hàng hóa : để khách hàng dễ tìm

* Phân loại theo nhà cung cấp.
* Phân loại theo 1 số tiêu chí đặc biệt : mới nhất, yêu thích, mua nhiều
* Tìm kiếm
* Chi tiết sản phẩm

+ Giỏ hàng :

* Mô hình hóa giỏ hàng.
* Hiển thị thông tin tổng hợp giỏ hàng.
* Quản lý giỏ hàng.
* Quản lý thành viên : để đặt hàng.

+ Quản lý thành viên :

* Đăng ký, kích hoạt
* Đăng nhập, đăng xuất
* Đổi mật khẩu, quên mật khẩu
* Sửa đổi thông tin tài khoản

+ Security & Đặt hàng

* Bảo vệ tài nguyên riêng tư.
* Đặt hàng.
* Quản lý đơn hàng :
  + Chưa giao dịch được hủy
  + Giao dịch rồi => chỉ xem
* Quản lý mặt hàng đã đặt

+ Đa ngôn ngữ :

+ Validation

\_ Phần 2 : dành cho người quản trị Admin

+ Quản lý dữ liệu ( cấu hình )

+ Xử lý đơn hàng

+ Thống kê, vẽ biểu đồ

+ Kiểm kê hàng tồn

+ Kiểm kê doanh thu, lợi nhuận

1. **Phân tích Database :**
   1. **Categories : Danh mục sản phẩm**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer | Primary key |
| Name | String |  |
| Description | String |  |
| Image | String |  |
| HomeFlag | Boolean |  |
| HotFlag | Boolean |  |
| OrderStatus | Integer | Sắp xếp |
| Status | 1 : True, 0 False |  |
| SeoPageTitle | String | SEO |
| SeoAlias | String | SEO |
| SeoKeywords | String | SEO |
| SeoDescription | String | SEO |
| CreatedAt | DateTime |  |
| UpdatedAt | DateTime |  |
| CreatedBy | String |  |
| CreatedBy | String |  |
| ParentId | Integer | **Khóa Ngoại** |

* 1. **Products : Sản Phẩm**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer | Primary key |
| Name | String |  |
| Description | String | Mô tả ngắn |
| Detail | String | Mô tả sản phẩm |
| Image | String |  |
| HomeFlag | Boolean |  |
| HotFlag | Boolean |  |
| OrderStatus | Integer | Sắp xếp |
| Price | Double | Giá Gốc |
| PromotionPrice | Double | Giá khuyến mãi |
| ViewCount | Long | Lượt xem |
| Special | Boolean |  |
| Latest | Boolean |  |
| Unit | String | Đơn vị tính |
| Status | 1 : True, 0 False |  |
| SeoPageTitle | String | SEO |
| SeoAlias | String | SEO |
| SeoKeywords | String | SEO |
| SeoDescription | String | SEO |
| CreatedAt | DateTime |  |
| UpdatedAt | DateTime |  |
| CreatedBy | String |  |
| CreatedBy | String |  |
| CategoryId | Integer | **Khóa Ngoại** |

* 1. **ProductImages : Hình ảnh của sản phẩm**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer |  |
| Url | String | Đường dẫn hình ảnh đến trang website |
| Title | String | Caption |
| Alt | String | Caption |
| ProductId | Integer | **Khóa Ngoại** |

* 1. **ProductInStock : Kho**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer |  |
| ProductId | Integer |  |
| Quantity | Integer | Tồn kho |
|  |  |  |

* 1. **Orders : Hóa đơn**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer |  |
| CustomerName | String |  |
| CustomerAddress | String |  |
| CustomerPhone | String |  |
| CustomerDescription | String |  |
| PaymentMethod | 0 : COD, 1 : OnlineBanking, 2 : Visa, 3: Paypal, … | ENUM |
| Status | 1 : True, 0 False | ENUM |
| OrderStatus | New, InProgress, Returned, Cancelled, Completed | ENUM |
| CreatedAt | DateTime |  |
| UpdatedAt | DateTime |  |
| CreatedBy | String |  |
| CreatedBy | String |  |
| CustomerID | String | **Khóa Ngoại** |

* 1. **OrderDetails : Chi Tiết Hóa Đơn**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer |  |
| ProductId | Integer |  |
| Price | Double |  |
| Quantity | Integer |  |
| Discount | Double |  |
| OrderId | Integer | **Khóa Ngoại** |

* 1. **Customers : Đây là User**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | String | Duy nhất = username |
| Password | String |  |
| Fullname | String |  |
| Phone | String |  |
| Email | String |  |
| BirthDay | DateTime |  |
| Avatar | String |  |
| Status | 1 : True, 0 : False |  |
| Activated | 1 : True, 0 : False | **Kích hoạt tài khoản** |
| CreatedAt | DateTime |  |
| UpdatedAt | DateTime |  |
| CreatedBy | String |  |
| CreatedBy | String |  |
| isAdmin | Boolean | **Mở rộng sau** |

* 1. **Tags : Từ khóa của sản phẩm**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | String |  |
| Name | String |  |
| Type | String |  |

**Products & Tags có bảng quan hệ nhiều nhiều là ProductTags**

* 1. **ProductTags**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** |  |
| Id | Integer |  |
| ProductId | Integer |  |
| TagId | String |  |

1. **Lưu ý :**

<https://stackoverflow.com/questions/38161180/auditing-changes-in-entity-jpa?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>

\_ Trường created\_at và updated\_at sử dụng tự động gọi.

**\_** Giá sản phẩm & giá trong hóa đơn phải khác. Vì sau này giá sp cập nhật thì giá hóa đơn không thay đổi.

1. **Tổ chức dự án :**

\_ Từng bước cấu hình : Dispatcher Servlet, Filter.

+ Có thể cấu hình bằng code java. ( thay thế XML )

+ Đặt tên thư mục configs ( ngắn gọn ).

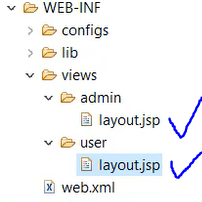
+ Khai báo View Resolver : thư mục views.

\_ Cấu hình : Hibernate

1. **Tổ chức layout :** Step by step

\_ Chọn views -> Tạo 2 thư mục nhỏ : **user** & **admin**

\_ Tạo layout vào **user** và layout vào trong **admin** ( tên quy luật camel hoặc pascal )



\_ Trong user, admin có thư mục shared dùng chung.

\_ Trong layout bổ sung chỉ thị layout vào :



\_ Thêm thẻ **base :**



\_ Khai báo các css, js cần thiết.

+ jquery-ui : bay vào giỏ hàng, calendar.

\_ Gọi biến view vào layout **<jsp:include page=”${view}” />**

\_ Tách từng phần riêng ra cho dễ kiểm soát :

+ header : phần thẻ head chứa title và các file css.

+ menu

+ cart\_info : chứa phần giỏ hàng

+ search

+ category

+ supplier : chứa danh sách nhà cung cấp ( nếu có )

+ special : sản phẩm đặc biệt

\_ Gọi vào bằng **jsp:include page=”shared/menu|head”**

*\* \* Chia nhỏ ra để cho lập trình – chức năng nào với task nào vào đúng 1 file lập trình*

1. **Xử lý Trang Home :**

**\_** Tạo HomeController ( map /home ) : có action index. ( /home/index )



* home/index : view này phải nằm cùng thư mục với user/layout.

\_ Tương tự ta tạo **home/about home/contact**

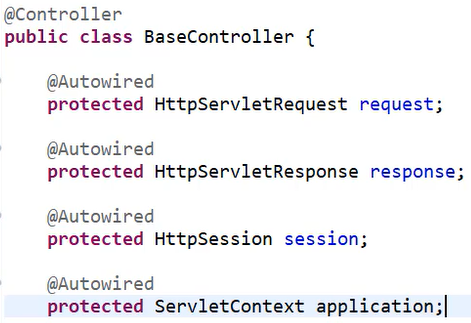
+ Xem xét tại từng trang view return về cả HTML ( bao gồm cả title, body )

\_ Tạo file index.jsp ở WebContent để khi chạy project.

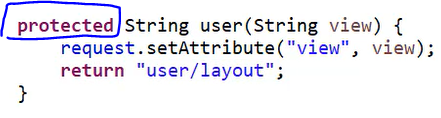
\_ Chạy dự án.

1. **Tạo BaseController dùng chung :**

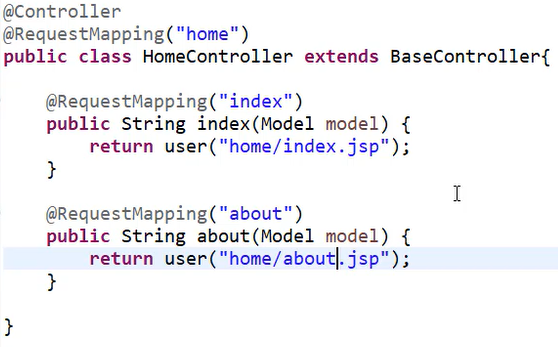
\_ Tiêm : HttpServletRequest, HttpServletResponse, HttpSession, ServletContext ( phạm vi truy cập **protected** )



\_ Tạo phương thức trả về **layout** – có tham số là thiết lập **view :**



\_ Các controller khác sẽ extends lại :



1. **Tiến hành Mapping model => table :**

\_ Khai báo cấu hình hibernate, chỉ ra package chứa các entity.

\_ Mapping :

1 Entity Class ⬄ 1 Table => @Table

1 Field ⬄ 1 Column => @Column

Thực thể kết hợp ⬄ 1 Relationship ( Quan hệ trong DB )

( 2 thực thể kết hợp : @OneToMany & @ManyToOne )

\_ Lưu ý : bỏ @Annotation Column khi tên field và tên column giống nhau.

\_ Cột ID : @Id

\_ Cột Thời gian :

@DateTimeFormat(pattern = "MM/dd/yyyy")

@Temporal(TemporalType.DATE)

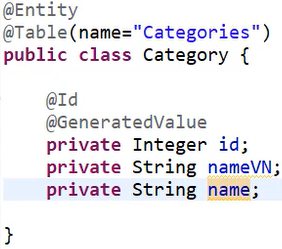
\_ Tự tang : @GeneratedValue

\_ Kiểu dữ liệu

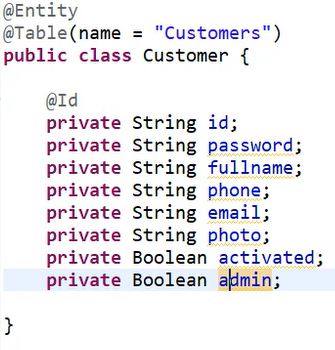
|  |  |
| --- | --- |
| **JAVA** | **SQL** |
| String | Char, Varchar, Text |
| Integer | Int |
| Long | BigInt |
| Double | Float |
| Boolean | Bit |
| Byte[] | Binary, Image |

\_ Thực hiện bước 1 DB : tạo package trùng với scan trong file cấu hình ( com.nghiepuit.entity )

Categories :



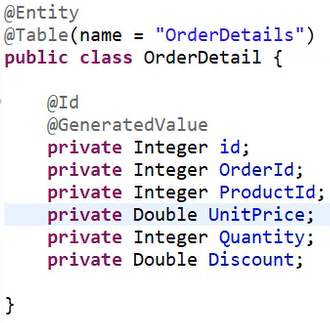
Customers :



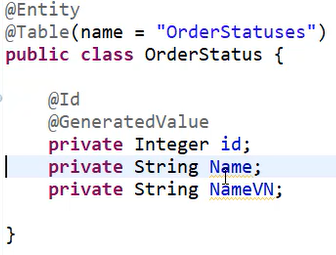
Orders :



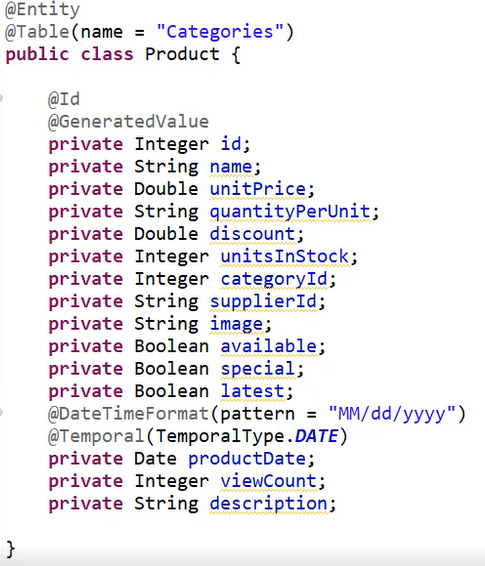
OrderDetails :



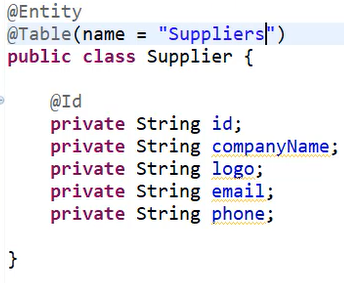
OrderStatus :



Products



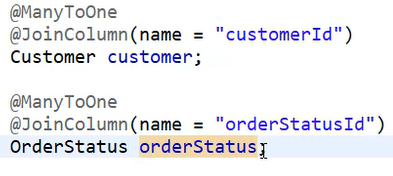
Suppliers :



\_ Bước 2 : Tìm khóa ngoại để ánh xạ : bỏ khóa ngoại và thay bằng thực thể kết hợp.

Để từ đó ta có thể lấy được thông tin của khóa ngoại ( đầy đủ )

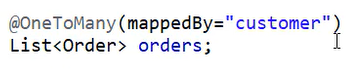
+ Order có 2 khóa ngoại : **customerId** và **orderStatusId** – nhìn trên sơ đồ quan hệ => bỏ đi ( comment lại ) => Dùng **@ManyToOne + @JoinColumn(name=”tên\_khóa\_ngoại\_vừa\_xóa”)**



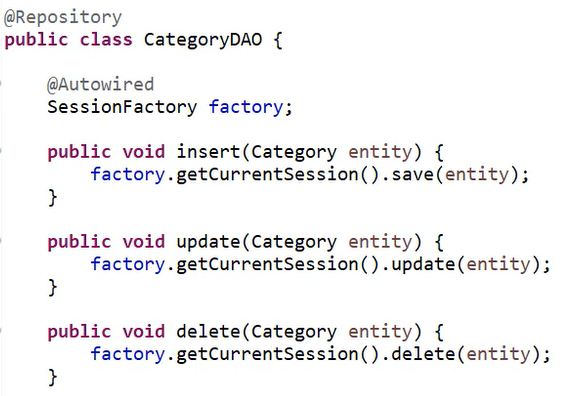
* Đổi khóa ngoại thay bằng thực thể kết hợp.

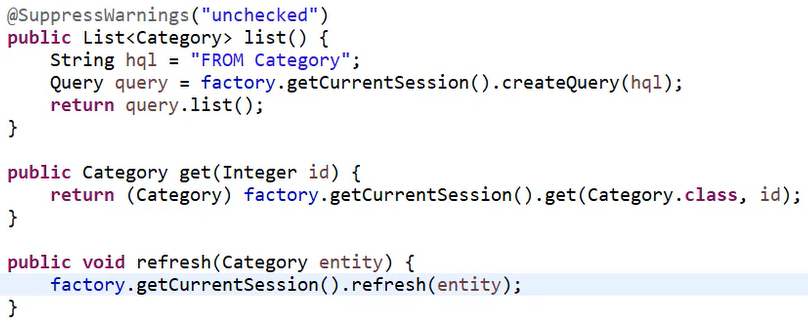
\_ Bước 3 : làm ngược lại – Đến các thực thể của các khóa ngoại này ta bổ sung :

+ Dùng List hoặc Collection có @Annotation là **@OneToMany(mappedBy=”customer”)** => ở trên tên biến là gì thì ở **mappedBy** gọi như vậy

 *Lưu ý : mappedBy là tên biến.*

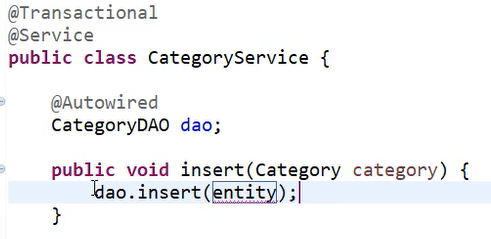
1. **Tạo Component ( có tên là @Repository )** để sau này cần dùng ta tiêm vào : Xây dựng DAO ( package com.nghiepuit.dao )

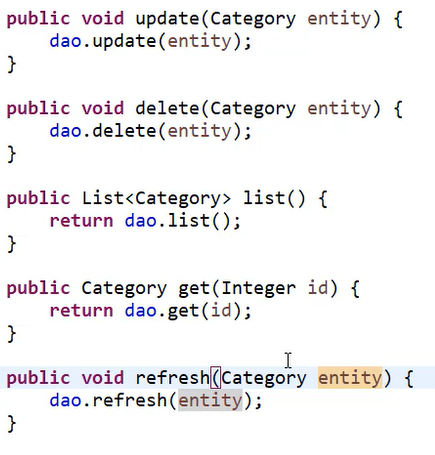




**Tương tự cho những bảng còn lại . => Tìm giải pháp viết 1 lần.**

1. **Tạo component Service** xử lý nghiệp vụ : ( @Service và @Transactional để quản lý transaction )





**Tương tự các bảng còn lại.**