Cake

Ngày 1

* #25 để biết cách đặt tên
* Khởi tạo dự án tại [Spring Initializr](https://start.spring.io/)
* Ôn tập chương 2 ,3,4

Những cái không hiêu

Bean #28

Ngày 2

* Ôn tập chương 5 , 6
* Trong file **application.properties:** setup properties
* 1. Authentication

Trả lời cho câu hỏi: Who are you ? Bạn là ai ?

Đối với ứng dụng Facebook, bạn cần "đăng nhập" thì mới có thể "định danh" bạn là ai.

Nếu bạn không đăng nhập, bạn là "guess" (khách), ai cũng giống nhau, không thể biết bạn là ai.

**Vì vậy, quá trình "authentication" thường gắn liền với bước đăng nhập vào hệ thống.**

Các cách authentication hay gặp:

- sử dụng **username/password** (bạn dùng email/số điện thoại + mật khẩu để đăng nhập facebook)

- sử dụng biometric: xác thực vân tay, vẽ pattern, xác thực khuôn mặt...

2. Authorization

Sau khi bạn hệ thống đã biết bạn là ai, câu hỏi tiếp theo cần trả lời, là What can I do ? (bạn có thể/không thể làm gì)

Ví dụ, bạn đã đăng nhập thành công vào tài khoản facebook (authentication):

+ Bạn có thể tạo mới/update/delete bài post/comments của chính bạn

+ Bạn có thể xem (read) post/comments của friends

+ Bạn "không thể" update/delete post/comments của friends

**=> Quá trình xác định "bạn có thể/không thể" làm gì, gọi là authorization.**

Thông thường, quá trình này cũng gắn liền với việc login vào hệ thống.

Khi bạn đã "login thành công", hệ thống đã biết được bạn "được phép" làm gì :v

\* Nếu muôn connect với database khác thì search với từ khóa ( set up mysql with java spring )

Ngày 3

* Ôn tập chương 7
* Config security user , password

config spring security

spring.security.user.name=hoidanit

spring.security.user.password=123456

* Xem log security

logging.level.org.springframework.security=DEBUG

* Disable security (key word để tham khảo là Disable sping security
* Anotation là gì Annotation (chú thích), là cách chúng ta “trang trí” class của Java. (decorator)

Mục đích là “tăng thêm sức mạnh” cho class đấy (meta-data)

Khi sử dụng Java Spring, thông thường, rất hiếm khi chúng ta định nghĩa thêm annotation, đa phần là sử dụng lại các annotation mà framework đã định nghĩa sẵn.

Khi sử dụng annotation, code của chúng ta ngắn hơn, từng minh và rõ ràng (dấu toàn bộ logic/magic phía sau annotation)

* **#46. Render view HTML** cách để tạo file lưu tài nguyên

Mặc định nó sẽ tìm file html trong thư mục static

* **#47. View Engine là gì** Học View engine JSP

File static để lưu các thư mục không thay đỗi người dùng có thể truy cập được

File Webapp lứu các thư mục thay đổi (dynamic) người dùng không thể truy cập được

* JSP , JSTL
* **#49. JSTL trong View**

Ngày 4

**<%@page contentType="text/html" pageEncoding="UTF-8"%>**

Tag hổ trợ tiếng việt

sets the content type returned by the page.

**<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>**

Tag để sài thư viện jsp và jstl

Tag library directives (<%@taglib ... %>) import custom tag libraries.\

* **Tích hợp Bootstrap và JQuery**

1. Sử dụng online

<!-- Latest compiled and minified CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

<!-- Latest compiled JavaScript -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

Trong webapp tạo thư mục **resources** để lưu tài nguyên động

Không lưu code trong file static vì khi này nó sẽ lấy code từ phần webapp

Resources bên ngoài để config

Resources bên trong dùng để code

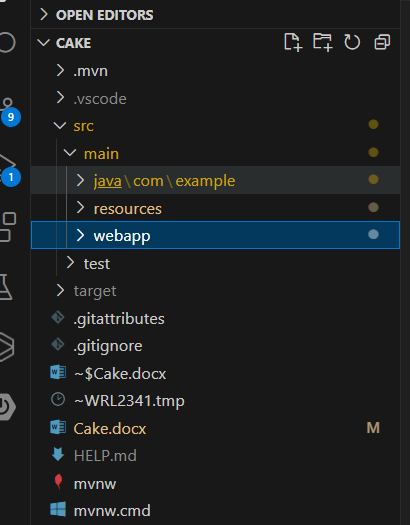
Cấu hình (config) tài nguyên css/js/images trong resources

@Override

    public void addResourceHandlers(ResourceHandlerRegistry registry) {

        registry.addResourceHandler("/css/\*\*").addResourceLocations("/resources/css/");

    }



Phải tạo folder đúng vị trí

* Tạo form create User

Tạo thêm folder, đặt tên là **domain -> User.java**

Với view, chia thành 2 folder: client/admin

Folder admin: lưu giao diện xử lý tại admin

Folder client: lưu giao diện xử lý tại user (client)

Lỗi không truy cập được tài nguyên resource

Ngày 5

* Sendata from HTML form to sever

Action , method

* [%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%](mailto:%25@taglib%20uri=%22http://www.springframework.org/tags/form%22%20prefix=%22form%22%25) (tag) để làm việc với form
* Thêm hậu tố form: vào thẻ form và input (lưu ý : nên đóng tag lại )
* Thêm path cho mỗi thẻ input (tên của path phải trùng với tên của đối tượng (enty)
* **@ModelAttribute** là một biến object dùng để nhận giá trị trong form (các path) (lưu ý  modelAttribute viết thường chữ đầu )
* Spring DATA : là công cụ để đơn giản hóa quá trình truy cập và sữ dụng dụng dữ liệu (thêm , sữa , xóa )
* Sữ dụng ORM (Object Relational Mapping ) : là cách để truy cập sữ dụng dữ liệu thông qua object định nghĩa code
* Hibernate >< JPA Hinernate để viết ORM còn JPA để thao tác với data