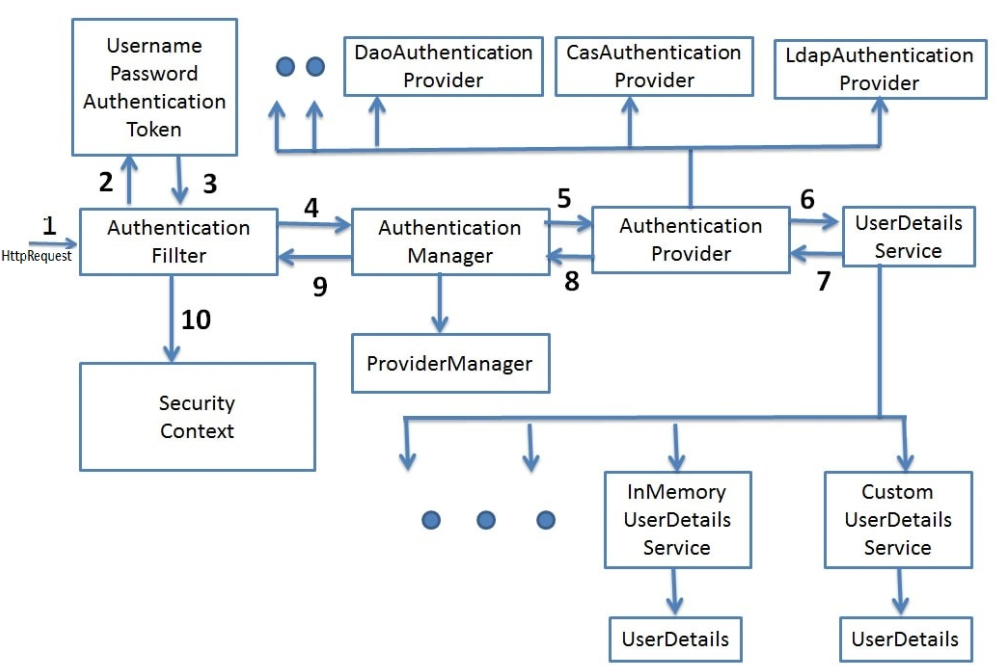
## Thông báo mới nhất răng phiên bảo 2.70 của spring nó báo hết hạn mẹ cái class WebSecurityConfigurerAdapter

* <https://spring.io/blog/2022/02/21/spring-security-without-the-websecurityconfigureradapter>

## Mô hình Authority trong spring

<https://stackoverflow.com/questions/2323377/spring-security-authenticationmanager-vs-authenticationprovider>

Có 2 loại đấy nhé AuthenProvider, AuthenManager



* **AuthenticationManager**

=>

Khi người dùng cố gắng truy cập vào một ứng dụng, yêu cầu http sẽ bị chặn bởi bộ lọc / chuỗi bộ lọc. Sử dụng Đối tượng xác thực đã tạo bộ lọc sau đó sẽ gọi phương thức xác thực của Trình quản lý xác thực. Trình quản lý xác thực chỉ là một giao diện và việc triển khai thực tế của phương thức xác thực được cung cấp bởi ProviderManager. ProviderManager có một danh sách các AuthenticationProviders. Từ phương thức xác thực nó gọi phương thức xác thực của AuthenticateProvider thích hợp. Để phản hồi, nó nhận được Đối tượng xác thực chính nếu xác thực thành công.

* AuthenticationProvider - AuthenicationProvider là một giao diện với phương thức xác thực và hỗ trợ. Nó có nhiều cách triển khai khác nhau như CasAuthenticationProvider hoặc DaoAuthenticationProvider. Tùy thuộc vào việc triển khai, triển khai AuthenicationProvider thích hợp được sử dụng. Nó nằm trong phương pháp xác thực triển khai AuthenticationProvider nơi tất cả các xác thực thực tế diễn ra.

## Tại sao cần @EnableWebSecurity

<https://stackoverflow.com/questions/44671457/what-is-the-use-of-enablewebsecurity-in-spring>

Thêm @Configurable nhưng không thêm ,@EnableWebSecurity

Nếu không thêm thì nó sẽ quét component thôi có gì đâu

* Nên thêm @EnableWebSecurity để spring có thể xác định nhanh hơn

## Cách giải quyết vấn đề tạo bean xoay vòng ( Bean phụ thuộc nhau)

<https://www.baeldung.com/circular-dependencies-in-spring>

* @Lazy vào bean tiêm bất kì trong 2 bean

Nó đua bean này vào poxy để làm gì đó còn chưa biết poxy là gì @@

@Component **public** **class** **CircularDependencyA** { **private** CircularDependencyB circB; @Autowired **public** **CircularDependencyA**(@Lazy CircularDependencyB circB) { this.circB = circB; } }

Chưa thực hành đc @@ nên xem lại

1. Các cách @Autowired và sự khác biệt của nó

<https://shareprogramming.net/cac-cach-su-dung-autowired-annotation-trong-spring/>

3 cách

* Tiêm vào thuộc tính (private, public,…)
* Tiêm vào hàm bất kì
* Tiêm vào constuctor

## Kế thừa Entity của Spring boot

* \*<https://docs.oracle.com/cd/E19798-01/821-1841/bnbqn/index.html>

Relational databases don't have a straightforward way to map class hierarchies onto database tables.

To address this, the JPA specification provides several strategies:

* *MappedSuperclass* – the parent classes, can't be entities
* Single Table – The entities from different classes with a common ancestor are placed in a single table.
* Joined Table – Each class has its table, and querying a subclass entity requires joining the tables.
* Table per Class – All the properties of a class are in its table, so no join is required.

## Hibernate – inheritance

<https://www.baeldung.com/hibernate-inheritance>

@Entity(name="products") @Inheritance(strategy = InheritanceType.SINGLE\_TABLE) @DiscriminatorColumn(name="product\_type", discriminatorType = DiscriminatorType.INTEGER)

**public** **class** **MyProduct**

{ // ... }

@Entity @DiscriminatorValue("1") // Loại của thằng này là 1

**public** **class** **Book** **extends** **MyProduct** { // ... }

Entity @DiscriminatorValue("2") // Loại của thằng này sẽ là 2

**class** **Pen** **extends** **MyProduct** { // ... }

Cái @DiscriminatorColumn giúp phân loại cho các bản con kế thừa nó