**@Value**

**@Value** cho phép chúng ta tiêm giá trị từ các nguồn khác nhau vào các thành phần Spring như các thuộc tính của bean, tham số của phương thức và các trường. Các giá trị có thể được cung cấp trực tiếp hoặc tham chiếu đến các nguồn khác nhau như các tệp cấu hình, biến môi trường hoặc các bean khác trong Spring ApplicationContext

A screenshot of a computer code

Description automatically generated with medium confidence

**@ConfigurationProperties**

**@ConfigurationProperties annotation**

* **@ConfigurationProperties** cho phép truy xuất các thuộc tính được đặt trong các tệp cấu hình một cách tự động và mạnh mẽ
* Demo: Spring sẽ tự động tìm kiếm các thuộc tính được đặt bên trong application.properties hay application.yml có tiền tố là email và đem chúng tiêm vào các thuộc tính tương ứng trong ConfigProperties class:

**A screen shot of a computer code

Description automatically generated**

Như vậy ConfigProperties bean của chúng ta sẽ có giá trị hostName tương ứng là *host@mail.com*, port là *9000* và from là [*mailer@mail.com*](mailto:mailer@mail.com)

**@ConfigurationProperties từ Spring Boot 2.2**

* Kể tử spring boot 2.2, nó sẽ tự động tìm kiếm và đăng ký các **@ConfigurationProperties** class, do vậy chúng ta không cần chú thích **@Configuration** hay sử dụng **@EnableConfigurationProperties** trên main class nữa

A screenshot of a computer code

Description automatically generated

* Ngoài ra chúng ta có thể sử dụng **@ConfigurationPropertiesScan** để tuỳ biến vị trí mà spring sẽ tìm kiếm các **@ConfigurationProperties** class:

A computer code with black text

Description automatically generated

**Nested Properties**

* Chúng ta có thể có các thuộc tính lồng nhau hoặc thậm chí là các thuộc tính có kiểu List, map v.v

A screen shot of a computer code

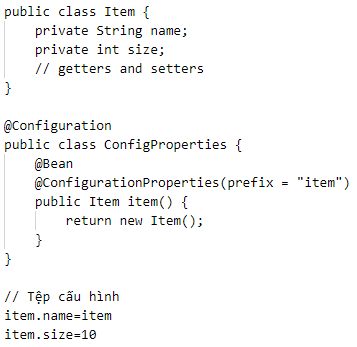
Description automatically generated

A computer screen shot of a computer code

Description automatically generated

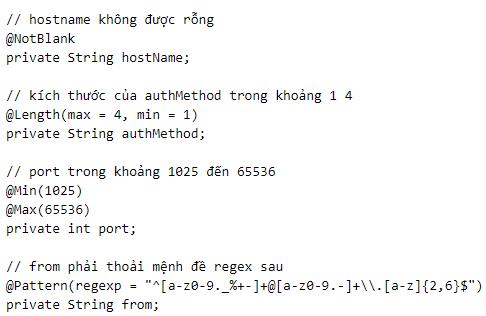
**@ConfigurationProperties với bean method**

* Chúng ta cũng có thể sử dụng **@ConfigurationProperties** annotation trên các bean method. Điều này sẽ khiến spring tìm kiếm các thuộc tính trong tệp cấu hình và tiêm nó vào bean sắp được khởi tạo bởi bean method



**Property Validation**

* **@ConfigurationProperties** cung cấp cơ chế kiểm định dữ liệu được cấu hình theo chuẩn JSR-303



Khi chúng ta định nghĩa các thuộc tính trong file cấu hình mà không thoả các điều kiện trên đây thì ứng dụng sẽ không thể khởi động và nó sẽ ném ra IllegalStateException

**Immutable @ConfigurationProperties Binding**

* Đối với các immutable object thì chúng ta sẽ không thể định nghĩa các setter method và vì vậy mà spring sẽ không thể tiêm các dữ liệu từ tệp cấu hình vào, do đó chúng ta có thể sử dụng **@ConstructorBinding** để spring có thể tiêm chúng thông qua constructor

A screenshot of a computer program

Description automatically generated