[**https://o7planning.org/vi/11267/huong-dan-lap-trinh-spring-boot-cho-nguoi-moi-bat-dau**](https://o7planning.org/vi/11267/huong-dan-lap-trinh-spring-boot-cho-nguoi-moi-bat-dau)

* **Khi tạo project**

Cài spring tool suit (sts): <https://o7planning.org/vi/10249/cai-dat-spring-tool-suite-vao-eclipse>

Trên **Eclipse** tạo một project sử dụng **Spring Tool Suite (STS)**

File pom.xml: 3 phần: parent, dependencies, build

**(phần parent) spring-boot-starter-parent** là một project sẵn có trong **Spring Boot**. Các thư viện phụ thuộc cơ bản đã được khai báo trong **spring-boot-starter-parent**, project của bạn chỉ cần thừa kế nó. Chỉ cần khai báo **<parent>** trong file **pom.xml** của project của bạn.

**(phần dependencies) spring-boot-starter-web**

Các **"Starter"** khác chỉ đơn giản là cung cấp phụ thuộc mà bạn có khả năng cần thiết khi phát triển một loại hình cụ thể của ứng dụng. Chẳng hạn khi bạn phát triển một ứng dụng web, bạn cần một phụ thuộc  **spring-boot-starter-web**.

Hay khi dùng themyleaf sẽ có thêm: spring-boot-starter-thymeleaf

**(phần build)Spring-boot-maven-plugin** là plugin cung cấp các thư viện cần thiết giúp project của bạn có thể chạy trực tiếp mà không cần triển khai trên một **Web Server**.

Chú thích **@SpringBootApplication** là tương đương với việc sử dụng **@Configuration**, **@EnableAutoConfiguration** và**@ComponentScan** với các thuộc tính mặc định của chúng.

Như vậy **@SpringBootApplication** giúp bạn tự động cấu hình **Spring**, và tự động quét (Scan) toàn bộ project để tìm ra các thành phần Spring (Controller, Bean, Service,...)

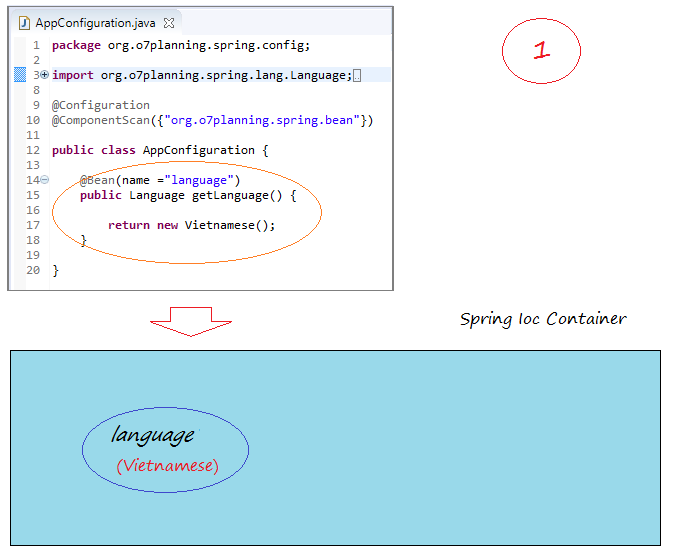
## **Nguyên tắc hoạt động của Spring**

Bạn tạo một đối tượng **ApplicationContext** bằng cách đọc các cấu hình trong class **AppConfiguration**, giống như đoạn code dưới đây.

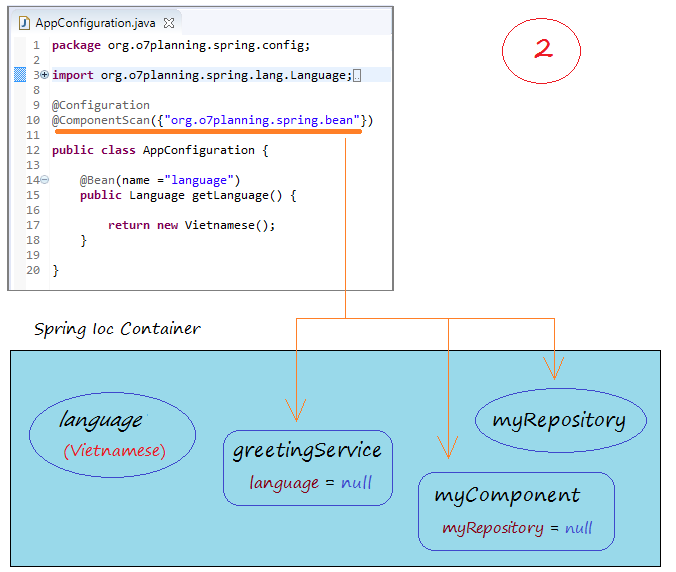
?

|  |  |
| --- | --- |
| 1 | ApplicationContext context = new AnnotationConfigApplicationContext(AppConfiguration.class); |

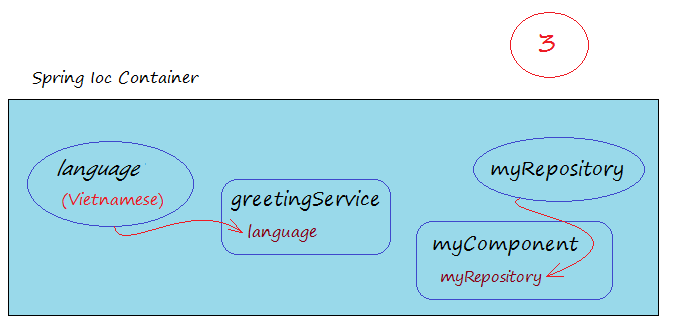
**Spring** sẽ tạo các **Spring BEAN,** theo các định nghĩa trong class **AppConfiguration**, (Chú ý: Class **AppConfiguration** phải được chú thích bởi **@Configuration**).



Tiếp theo **Spring**sẽ tìm kiếm trong package **"org.o7planning.spring.bean"** để tạo các **Spring BEAN** khác, (Tạo các đối tượng từ các class được chú thích bởi **@Service**, **@Component** hoặc **@Repository**).



Lúc này các **Spring BEAN** đã được tạo ra, và được chứa trong **Spring IoC**. Các trường của các **Spring BEAN** có chú thích bởi**@Autowired** sẽ được tiêm các giá trị vào, giống hình minh họa dưới đây:



*Trở về với câu hỏi****"IoC là gì?"****.*

*Theo cách truyền thống một đối tượng được tạo ra từ một class, các trường (field) của nó sẽ được gán giá trị từ chính bên trong class đó.****Spring****đã làm ngược lại với cách truyền thống, các đối tượng được tạo ra và một vài trường của nó được tiêm giá trị từ bên ngoài vào bởi một cái được gọi là****IoC****.****IoC****viết tắt của****"Inversion of Control"****- Có nghĩa là****"Đảo ngược của sự điều khiển"****.****IoC Container****là bộ chứa tất cả các****Spring BEAN****được sử dụng trong ứng dụng.*