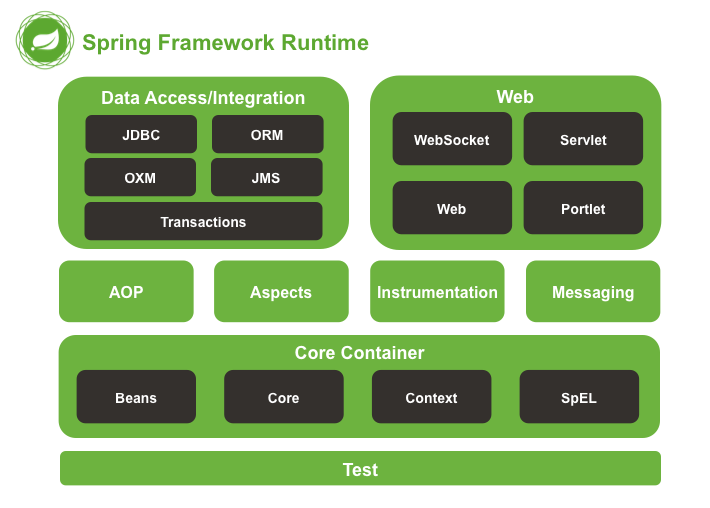
SPRING

I, Overview Spring

* Spring là mã nguồn mở, đc phát triển, chia sẻ, và có cộng đồng người sử dụng lớn
* Spring Framework đc xây dựng trên 2 nguyên tắc design chính là : Dependency injection và Aspect Oriented Programming
* Các tính năng core cốt lõi đc sử dụng để ptr : Java Desk, Mobie app, Java Web
* Mục tiêu chính của Spring là giúp ptr các ứng dụng J2EE dựa trên mô hình sử dụng POJO(Plan Old Java Object)
* Structure + Module của Spring :



* + Test : provide JUnit , TestNG
  + Spring Core Container :
    - Spring Core + Bean : provide IOC + Dependency Injection
    - Spring Context : support internationalization(multi language)
    - EL(Expresion Language) : support get/set, method cải tiến, cho phép truy cập vào Collection, index, toán tử logic…
  + AOP, Aspect,Intrumentaton : Hỗ trợ lập trình hướng khía cạnh(Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.
  + Data Access/Intergation : Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database.
  + Web : Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

II, Overview Spring boot

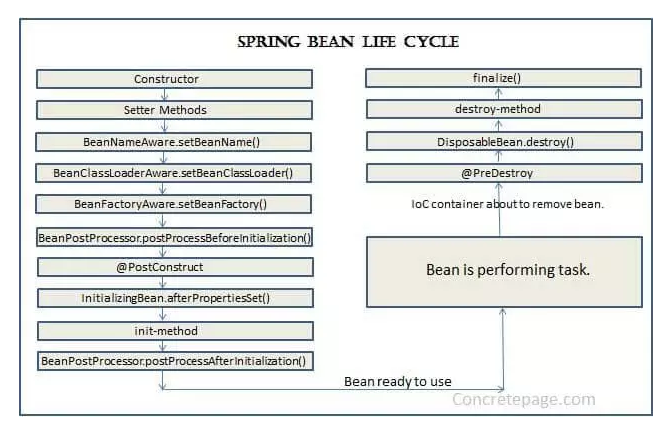
* Là 1 java framework mạnh mẽ, được cải tiến nhiều hơn so với Spring framework thông thường :
  + AutoConfig : Tự động cấu hình
  + Dựa trên các Anotation để tạo ra các bean thay vì dung XML
  + Server Tomcat có thể đc nhúng ngay trong file JAR bulid ra và dùng ở bất kì môi trường nào java chạy được

III,Overview Sping Bean

* Define : Các đối tượng hay là những thành phần chính được viết sẵn để tạo nên ứng dụng và được quản lý(khởi tạo, kết nối, quản lý) bởi Spring IoC container được gọi là Bean.
* Địng nghĩa 1 bean thông qua 2 anotation :
  + @Component : Chú thích cho 1 class
  + @Bean : Chú thích trên method
* Trong ứng dụng Spring luôn sử dụng @Configuration và @ComponentScan để load config và scan những package trước khi start ứng dụng
  + @ComponentScan : dùng để hướng dẫn IOC container tìm kiếm và khởi tạo 1 bean mới trong package

IV, Life Cycle

* Vòng đời (life cycle) của bean được hiểu là từ khi bean được tạo ra cho tới khi chết đi, sẽ có những sự kiện (event) khác nhau xảy ra.
* Sơ đồ :



* + IoC container(ApplicationContext) tạo bean bằng cách tạo constructer
  + Gọi method setter để inject các bean và thực hiện các quá trình cài đặt khác vào Bean như setBeanName, setBeanClassLoader, v.v..
  + @PostConstruct được gọi (Hàm đánh dấu)
  + Init Method được gọi
  + Bean sẵn sang hoạt động
  + Nếu IoC Container không quản lý bean nữa hoặc bị shutdown nó sẽ gọi hàm @PreDestroy trong Bean
  + Xóa bean

V, Spring annotations

* Annotation được hiểu là một dạng chú thích hoặc một dạng siêu dữ liệu (metadata)
* bắt đầu với ký hiệu @ và sau đó là tên của chú thích, Ví dụ: @Deprecated
* Chức năng :
  + chỉ dẫn cho trình biên dịch (Compiler) : @Deprecated, @Override, @SuppressWarnings
  + chỉ dẫn trong thời điểm biên dịch (Build-time)
  + chỉ dẫn trong thời gian chạy (Runtime)
* Các annotations thường sử dụng :
  + @Autowire : Tạo + inject 1 bean
  + @SpringApplication : Đánh dấu phương thức main
  + @Component : Đánh dấu class là 1 thành phần của Spring
  + @Bean : Cách Spring tạo bean
  + @RequestMapping : Khai báo đường dẫn api
  + @RestController : Khai báo class là 1 source api
  + @Service,@Repository , @Entity: Các thành phần của 1 Spring dùng để khai báo,kết nối db,xử lí code logic
  + @GetMapping,@PostMapping,@PutMapping,@DeleteMaping : Các phương thức được sử dụng cho method
  + @Pathvariable,@Path : Truyền data cho method
  + @RequestBody : Truyền data cho method dưới dạng JSON
  + @Query,@NameQuery : Khai báo 1 câu truy vấn SQL