

스프링 컨테이너 : 스프링은 객체 관리 컨테이너를 제공한다.

컨테이너에 객체를 담아 두고 필요할 때 컨테이너로 부터 객체를 가져와 사용 할 수 있도록 한다.

BeanFactory => ApplicationContext => WebApplicationContext

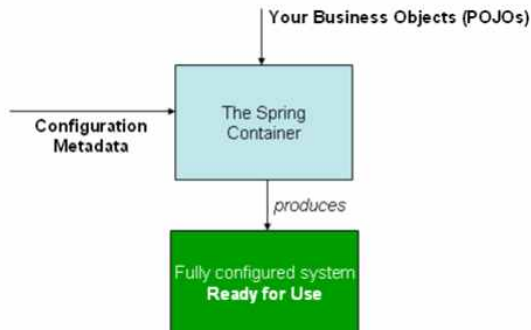
자바에서는 ApplicationContext 사용 // Web WebApplicationContext 사용

❑ Container

- Spring IoC Container는 객체를 생성하고, 객체 간의 의존성을 이어주는 역할을 한다.

❑ 설정 정보(Configuration Metadata)

- Spring IoC container가 “객체를 생성하고, 객체간의 의존성을 이어줄 수 있도록” 필요한 정보를 제공한다. 설정 정보는 기본적으로 XML 형태로 작성되며, 추가적으로 Java Annotation을 이용하여서도 설정이 가능하다.



❑ Bean

- Spring IoC Container에 의해 생성되고 관리되는 객체를 의미한다.

❑ IoC(Inversion of Control)이란?

- IoC는 Inversion of Control의 약자로 한글로 “제어의 역전” 또는 “역제어”라고 부른다. 어떤 모듈이 제어를 가진다는 것은 “어떤 모듈을 사용할 것인지”, “모듈의 함수는 언제 호출할 것인지” 등을 스스로 결정한다는 것을 의미한다. 이러한 제어가 역전되었다는 것은, 어떤 모듈이 사용할 모듈을 스스로 결정하는 것이 아니라 다른 모듈에게 선택권을 넘겨준다는 것을 의미한다.

❑ DI(Dependency Injection)이란?

- Dependency Injection이란 모듈간의 의존성을 모듈의 외부(컨테이너)에서 주입시켜주는 기능으로 Inversion of Control의 한 종류이다.
- 런타임시 사용하게 될 의존대상과의 관계를 Spring Framework 이 총체적으로 결정하고 그 결정된 의존특징을 런타임시 부여한다.

❑ Non-IoC/DI vs IoC/DI

