스프링

스프링의 주요 특징

1. POJO기반의 구성

자바코드를 이용해서 객체를 구성하는 방식을 그대로 스프링에서 사용가능

-->특정 라이브러리나 컨테이너의 기술에 종속적이지 않고 개발자는 일반 형태로 코드를 작성하고 실행할 수 있어서 생산성에서도 유리하고, 코드에 대한 테스트 작업 역시 좀 더 유연하게 할 수 있다는 장점이 있음.

2. 의존성 주입(DI)을 통한 객체 간의 관계 구성

스프링은 DI와 Inversion Of Control 제어의 역행이 핵심

IOC제어의 역행은 메소드나 객체의 호출작업을 개발자가 결정하는 것이 아니라, 외부에서 결정되는 것을 의미한다.

즉 개발자는 프레임워크에 필요한 부품을 개발하고ㅑ, 조립하는 방식의 개발을 한다. 이렇게 조립된 코드의 최종 호출은 개발자에 의해서 이뤄지지 않고, 프레임워크의 내부에서 결정된 대로 이뤄지게 되는데 이런 현상을 제어의 역행이라고 한다.

DI는 어떤 객체가 혼자 일을 처리할 수 없다는 것을 의미

(객체 A가 B에 의존적일때 자바는 인터페이스를 이용해서 이런 의존적인 객체의 관계를 최대한 유연하게 처리할 수 있도록 노력)

의존성 주입은 의존적인 객체를 직접 생성하거나, 제어하는 것이 아니라, 제어의 역행으로 특정 객체에 필요한 객체를 외부에서 결정해서 연결 시키는 것을 의미한다.

외부에서 필요한 객체를 결정하기 때문에 개발자는 의존적인 객체들과의 관계를 직접 처리할 필요가 없고, 인터페이스를 활용해서 유연한 구조를 사용할 수 있습니다.

즉 스프링은 의존성 주입을 프레임워크에서 처리하기 때문에 개발자는 자신이 만드는 객체나 클래스 외에는 신경 쓰지 않고 코드를 만들고, 자신의 코드에 필요한 객체는 스프링을 통해서 주입받는 구조로 작성

종류는 1)생성자를 통한 주입 2)set메소드를 이용한 주입 두 처리 모두 애노테이션만으로 처리 가능

3. AOP지원

개발자가 비즈니스 로직에만 집중하게 하기 위해서는 “반복적인 코드의 제거”

스프링에서도 위와 같은 방법을 제공

반드시 처리가 필요한 부분을 스프링에서는 횡단 관심사라고 하며, 스프링은 이러한 횡단 관심사를 분리해서 제작하는 것이 가능하다. AOP(Aspect Oriented Programming)은 이러한 횡단 관심사를 모듈로 분리하는 프로그래밍 패러다임이다.

스프링은 AOP를 AsptctJ의 문법을 통해서 작성할 수 있는데, 이를 통해서 개발자는 1)핵심 비즈니스 로직에만 집중해서 코드를 개발할 수 있게 되었고 2)각 프로젝트마다 다른 관심사를 적용할 때 코드의 수정을 최소화시킬 수 있었으며, 3)원하는 관심사의 유지보수가 수월한 코드를 구성할 수 있다.

트랜잭션의 지원(하나의 업무가 여러 작업으로 이뤄지는 경우)

스프링은 트랜잭션의 관리를 애노테이션이나 XML로 설정할 수 있기 때문에 개발자가 매번 상황에 맞는 코드를 작성할 필요가 없도록 설계되었다.

4. 편리한 MVC구조

5. WAS에 종속적이지 않은 개발 환경