

스프링 Spring

PRESENTATIONPRESENTATION

02. Spring?

JAVA의 웹 프레임워크로 JAVA 언어를 기반으로 사용한다.

JAVA로 다양한 애플리케이션을 만들기 위한 프로그래밍 틀이라 할 수 있다.

Spring은 중복코드의 사용률을 줄여주고, 비즈니스 로직을 더 간단하게 해줄 수 있다.

결론적으로 Spring이란 JAVA 기술들을 더 쉽게 사용할 수 있게 해주는 오픈 소스 프레임 워크

03. Spring 특징

스프링(Spring)의 특징

스프링의 가장 큰 특징은 아마도

제어의 역전(IOC)

의존성 주입(DI)

관점 지향 프로그래밍(AOP)

이렇게 3가지가 아닐까 싶습니다.

이런 특징으로 인해 결합도를 낮춰(Loose coupling)

유연한 개발이 가능해졌습니다.

그럼 특징들을

하나하나씩 살펴보겠습니다. 😊

03. Spring 특징

✅ 제어의 역전(IOC : Inversion of Control)

제어의 역전 패턴이 인기를 끄는 이유는 프로그램의 생명주기에 대한 제어권이 웹 애플리케이션 컨테이너에 있기 때문입니다.

즉, 사용자가 직접 new 연산자를 통해 인스턴스를 생성하고 메서드를 호출하는 일련의 생명주기에 대한 작업들을 스프링에 위임할 수 있게 되는 것입니다.

IOC는 직관적이지 못하기 때문에 IOC를 구현하는 방법으로는 의존성 주입(DI)이 있습니다.

03. Spring 특징

✅ 의존성 주입(DI : Dependency Injection)

객체 사이에 필요한 의존 관계에 대해서 스프링 컨테이너가 자동으로 연결해 주는 것을 말합니다
스프링 컨테이너는 DI를 이용하여 빈(Beans) 객체를 관리하며, 스프링 컨테이너에 클래스를 등록하면 스프링이 클래스의 인스턴스를 관리해 줍니다

의존성이란 무엇일까요? 🤔

간단히 생각하면 스마트폰을 사용하기 위해서는 배터리가 필요합니다. 배터리가 없으면 스마트폰은 사용할 수 없죠. 즉 스마트폰은 배터리에 의존한다고 보면 됩니다. 만약 그 배터리가 스마트폰과 일체형으로 되어있으면 어떨까요?

배터리 교체를 하기 위해선 분리형과 비교해서 가볍게 바꿀 수가 없겠죠?

03. Spring 특징

3. AOP(Aspect Object Programming, 관점 지향 프로그래밍)

- 로깅, 트랜잭션, 보안 등 여러 모듈에서 공통적으로 사용하는 **기능을 분리하여 관리** 할 수 있다.
- 각각의 클래스가 있다고 가정하자. 각 클래스들은 서로 코드와 기능들이 중복되는 부분이 많다. 코드가 중복될 경우 실용성과 가독성 및 개발 속도에 좋지 않다. 중복된 코드를 최대한 배제하는 방법은 중복되는 기능들을 전부 빼놓은 뒤 그 기능이 필요할때만 호출하여 쓰면 훨씬 효율성이 좋다.
- 즉, AOP는 **여러 객체에 공통으로 적용할 수 있는 기능을 구분함으로써 재사용성을 높여주는 프로그래밍 기법**이다.

Spring?

스프링 프레임워크(Spring Framework)는 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로서 간단히 **스프링**(Spring)이라고도 불린다. 동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있다. 대한민국 공공기관의 웹 서비스 개발 시 사용을 권장하고 있는 전자정부 표준프레임워크의 기반 기술로서 쓰이고 있다.

출처: 위키백과

Spring Boot?

스프링 프레임워크는 기능이 많은만큼 환경설정이 복잡한 편이다. 이에 어려움을 느끼는 사용자들을 위해 나온 것이 바로 **스프링 부트**다. 스프링 부트는 스프링 프레임워크를 사용하기 위한 설정의 많은 부분을 자동화하여 사용자가 정말 편하게 스프링을 활용할 수 있도록 돕는다. 스프링 부트 starter 디펜던시만 추가해주면 바로 API를 정의하고, 내장된 톰캣이나 제티로 웹 애플리케이션 서버를 실행할 수 있다. 심지어 스프링 홈페이지의 이니셜라이저를 사용하면 바로 실행 가능한 코드를 만들어준다. 실행환경이나 의존성 관리 등의 인프라 관련 등은 신경쓸 필요 없이 바로 **코딩**을 시작하면 된다. 그리고 바로 그것이 스프링의 키 포인트이다.

04. Spring VS Spring Boot

Spring Boot는 Spring framework와 몇 가지면에서 차이가 있다.

차이점

- 1) Embed Tomcat을 사용하기 때문에, (Spring Boot 내부에 Tomcat이 포함되어있다.) 따로 Tomcat을 설치하거나 매번 버전을 관리해 주어야 하는 수고로움을 덜어준다.
- 2) starter를 통한 dependency 자동화 :
아마 Spring 유저들이 가장 열광한 기능이 아닐까 싶다. 과거 Spring framework에서는 각각의 dependency들의 호환되는 버전을 일일이 맞추어 주어야 했고, 때문에 하나의 버전을 올리하고자 하면 다른 dependency에 까지 영향을 미쳐 version관리에 어려움이 많았다. 하지만, 이제 starter가 대부분의 dependency를 관리해주기 때문에 이러한 걱정을 많이 덜게 되었다.
- 3) XML설정을 하지 않아도 된다.
- 4) jar file을 이용해 자바 옵션만으로 손쉽게 배포가 가능하다.
Spring Actuator를 이용한 애플리케이션의 모니터링과 관리를 제공한다.

PART 02

해결하고자 하는 것

Spring

- ✓ Dependency Injection(DI) - 의존성 주입
- ✓ 중복된 코드 제거
- ✓ 다른 프레임워크와의 통합

Spring Boot

- ✓ Auto Configuration 자동 설정
- ✓ 쉬운 의존성 관리
- ✓ 내장 서버

뽀나스 ++

PRESENTATIONPRESENTATION

05. Boot 장점

Spring Boot의 장점

1) 라이브러리 관리 자동화

- 기존 스프링 자바 프로젝트에서는 메이븐이나 그라들을 이용해서 라이브러리 의존성을 관리해왔다. 하지만 스프링 부트에서는 스타터(Starter)라는 것을 이용해 특정 기능에 필요한 라이브러리 의존성을 더욱 간단히 처리 할 수 있다.

2) 설정의 자동화

- 스프링 부트에서는 프로젝트에 추가된 라이브러리를 기반으로 실행에 필요한 환경을 자동으로 설정해준다.
- 개발에 필요한 라이브러리들을 추가하면 스프링 부트가 이 라이브러리들을 인지해서 관련된 스프링 설정을 자동으로 처리해주기 때문에 개발자들은 복잡한 설정을 하지 않고도 개발이 가능하다.

05. Boot 장점

3) 라이브러리 버전 자동 관리

- 개발시 가장 신경쓰이는 부분이 라이브러리와 버전 관리이다. 기존의 스프링은 스프링 라이브러리만 사용하여 개발할 수 없으며, 의존관계에 있는 서드파티 라이브러리들도 사용한다. 스프링 부트를 사용하면 스프링 부트 버전에 해당하는 스프링 라이브러리뿐만 아니라 서드파티 라이브러리들도 호환되는 버전으로 다운로드해준다.
- 라이브러리 버전이 달라 정상적으로 동작하지 않는 상황을 겪을 필요가 없고, XML설정을 이용해서 라이브러리를 매번 설정하는 과정을 줄이고 개발에만 집중할 수 있는 환경을 제공한다.

4) 테스트 환경과 내장 Tomcat

- JUnit을 비롯한 테스트 관련 라이브러리들이 기본적으로 포함되어 있기 때문에 컨트롤러를 비롯한 다양한 계층의 클래스들에 대해서 테스트 케이스를 쉽게 작성할 수 있다.
- Tomcat 서버를 내장하고 있기 때문에 단지 main() 메소드를 가진 클래스를 실행하는 방식으로 서버를 구동하기 때문에 실행결과를 빠르게 확인할 수 있다.

5) 독립적으로 실행 가능한 JAR

- 애플리케이션을 개발하고 테스트까지 마쳤으면 애플리케이션을 실제 운영 서버에 배포하기 위해서 패키징을 해야하는데, 프로젝트가 일반 자바 프로젝트라면 간단하게 JAR파일로 패키징하면 되지만 웹 프로젝트라면 WAR 파일로 패키징 해야한다.
- 스프링 부트는 독립적으로 실행 가능한 애플리케이션을 빠르게 개발하는 것을 목표로 하기 때문에 웹 애플리케이션도 WAR가 아닌 JAR파일로 패키징 하여 사용할 수 있다.



THANK YOU

2021.06.28