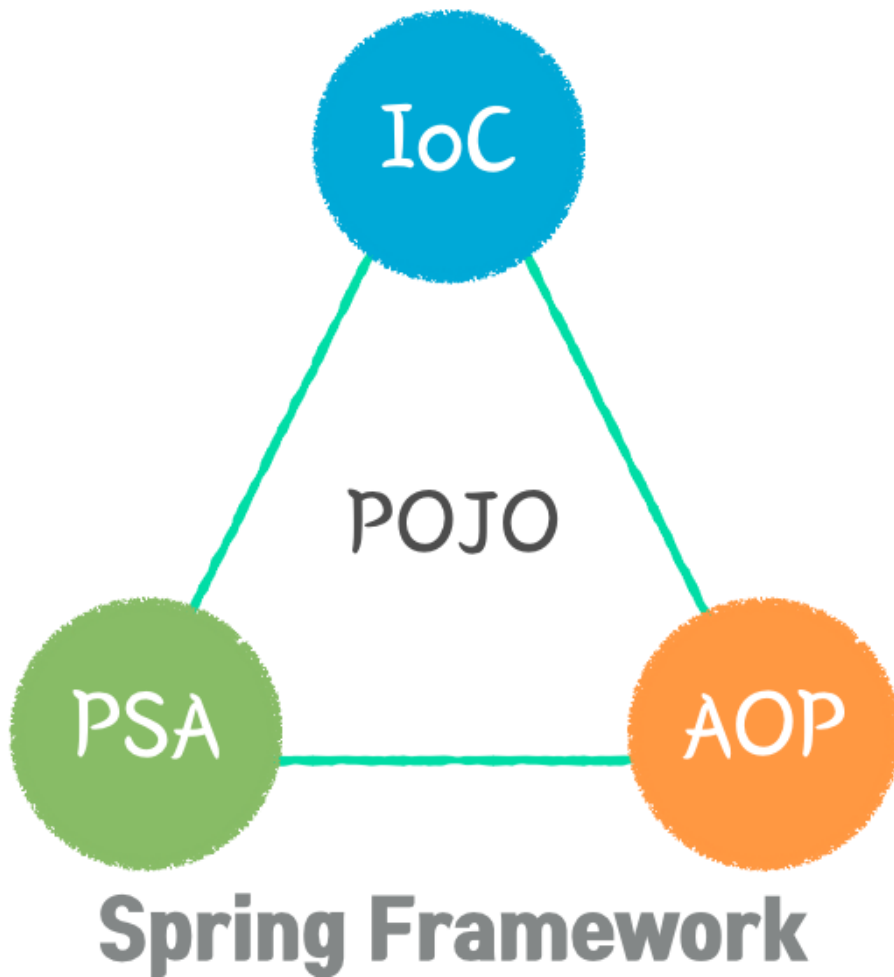




Dev.BackEnd/Spring

# Spring 의 시작, 프레임워크의 구성요소와 동작 원리

글쓰는 개발자 \_Jbee 2016.11.29 09:14



## Spring Framework의 구성요소와 동작원리

### POJO

스프링의 특징을 살펴보면 POJO라는 단어가 등장한다. POJO란 Plain Old Java Object로 직역하자면 평범한 옛날 자바 객체이다. 말 그대로 자바 객체인 것이다.



\_Jbee  
::Devlog::

CATEGORY

여기에는 자바 웹 애플리케이션을 개발하기 위해 Servlet 클래스를 상속받아 구현하였다. 이 Servlet 클래스는 POJO가 아닌 것이다. 개발자가 직접 Servlet 클래스를 작성하지 않고 POJO 만으로 웹 애플리케이션을 구축할 수 있다는 것이 스프링의 특징이다.

### 그러면 어떠한 장점이 있는가?

Servlet 클래스를 이용해서 자바 웹 애플리케이션을 구축하려면 반드시 Servlet에서 요구하는 규칙에 맞게 클래스를 만들어야 실행할 수 있다. 또한 이 Servlet 클래스를 이용하여 비즈니스 로직 외의 것들을 구축하는데 정말 많은 시간이 소요되었다. 하지만 스프링을 통해서라면 일반적인 자바 객체로 웹 애플리케이션을 구축할 수 있으며 비즈니스 로직에 집중할 수 있게 되는 것이다. 그렇다고 스프링에 Servlet 클래스가 없는 것은 아니다. 모두 추상화되어 라이브러리로 들어가 있으며 개발자는 XML 또는 다른 설정을 통해서 Servlet 을 이용할 수 있게 된다.

그렇다면 스프링을 감싸고 있는 3대 특징에 대해서 알아보자.

스프링을 한 줄로 요약하자면 다음과 같다.

"IoC와 AOP를 지원하는 경량의 컨테이너 프레임워크"

### 알아보기 앞서 컨테이너란 무엇인가?

컨테이너는 특정 객체의 생성과 관리를 담당하며 객체 운용에 필요한 다양한 기능을 제공한다. 애플리케이션 운용에 필요한 객체를 생성하고 객체 간의 의존관계를 관리한다는 점에서 스프링도 일종의 컨테이너라고 할 수 있다.

### 스프링 컨테이너의 종류

스프링에서는 **BeanFactory**와 이를 상속한 **ApplicationContext** 두 가지 유형의 컨테이너를 제공한다. BeanFactory는 스프링 설정 파일(applicationContext.xml)에 등록된 <bean> 객체를 생성하고 관리하는 가장 기본적인 컨테이너 기능만 제공한다. 그리고 컨테이너가 구동될 때 객체를 생성하는 것이 아니라 클라이언트로부터의 요청에 의해서만 객체를 생성한다. (lazy loading)

이를 확장한 ApplicationContext 컨테이너는 이 기능에 더해 트랜잭션 관리나 메시지 기반의 다국어 처리 등 다양한 기능을 지원한다. 또한 컨테이너가 구동되는 시점에 <bean>에 등록되어 있는 클래스들을 객체화하는 즉시 로딩 방식으로 동작한다. 대표적으로 이 ApplicationContext를 구현한 클래스는 GenericXmlApplicationContext 클래스이다.

정리하자면 스프링 컨테이너는 <bean> 저장소에 해당하는 XML 설정 파일을 참조하여 <bean>의 생명주기를 관리하고 여러 가지 서비스를 제공한다.



\_Jbee  
::Devlog::

CATEGORY

비즈니스 컴포넌트를 개발할 때 항상 신경쓰는 것이 바로 **낮은 결합도와 높은 응집도**이다.

스프링은 제어의 역행을 통해 애플리케이션을 구성하는 객체 간의 느슨한 결합 즉, 낮은 결합도를 유지한다. 스프링의 IoC는 객체 생성을 자바 코드로 직접 처리하는 것이 아니라 컨테이너가 대신 처리하게 한다. 그리고 객체와 객체 사이의 의존관계 역시 컨테이너가 처리한다.

## 의존성 주입(DI)은 무엇인가?

의존성 관계란 객체와 객체의 **결합 관계**이다.

하나의 객체에서 다른 객체의 변수나 메소드를 이용해야 한다면, 이용하려는 객체에 대한 객체 생성과 생성된 객체의 레퍼런스 정보가 필요하다. 즉, 쉽게 말해서 의존성은 new다. A라는 객체 생성자에서 new B();를 했다면 A는 B에 의존하게 된다. 주입이란 외부에서라는 뜻을 내포하고 있다. 즉 A라는 객체에서 B를 생성하는 것이 아니라 외부에서 생성된 B를 A에 주입함으로써 의존 관계를 없앨 수 있다. 이것을 의존성 주입이라고 한다.

의존성을 주입하는 방법은 여러 가지가 있다.

### 1. XML을 통한 의존성 주입

#### 1) 생성자를 통한 의존성 주입

생성자에 인자로 주입하고자 하는 객체를 넣어준다. 스프링 설정 파일에서는 <constructor-arg> 태그와 ref 속성을 이용한다.

#### 2) 속성을 통한 의존성 주입

내부적으로 set method 를 사용한다. 스프링 설정 파일에서는 <property> 태그를 사용해야 하며 name 속성값이 호출하고자 하는 메소드의 이름이어야 한다. name에는 변수명을 적어주면 스프링에서 name의 첫 글자를 대문자로 바꾸고 앞에 set키워드를 붙여 set method 를 실행한다.

### 2. 어노테이션을 통한 의존성 주입

스프링에서는 **@Autowired** 라는 어노테이션을 통해 의존성을 주입한다. @Autowired는 속성의 설정자 메서드에 해당하는 역할을 자동으로 수행한다. 이와 비슷한 역할을 하는 자바 어노테이션으로는 @Resource 어노테이션이 있다. 두 어노테이션의 차이는 bean을 탐색할 때 우선순위로 하는 기준이 어떤 것이냐이다.

## AOP (Aspect Oriented Programming)

직역하자면 **관점 지향 프로그래밍**이다.



\_Jbee  
::Devlog::

CATEGORY

모듈을 구성하고 있는 코드는 책임관심사와 평면관심사가 겹쳐진 것이다,

AOP는 횡단 관심사에 관심이 있다. 모듈마다 중복되는 부분을 걷어내는 것이 주 목적이다. AOP역시 하늘에서 뚝 떨어진 것이 아니라 중복을 제거해야 한다는 개발자의 숙명으로부터 탄생하게 된 것이다.

AOP를 이해하는 데에 가장 중요한 핵심 개념이 바로 **관심 분리(Separation of Concerns)**이다. 관점 지향 프로그래밍은 비즈니스 메소드를 개발할 때, 핵심 비즈니스 로직과 각 비즈니스 메소드마다 반복해서 등장하는 공통 로직을 분리함으로써 응집도가 높게 개발할 수 있도록 지원한다.

## 1. XML을 이용한 AOP

## 2. 어노테이션을 이용한 AOP

@Aspect 어노테이션은 클래스를 AOP에서 사용하겠다는 의미이며

@Before 어노테이션은 대상 메서드 실행 전에 이 메서드를 실행하겠다는 의미이다.

## PSA(Portable Service Abstraction)

직역하자면 **일관성 있는 서비스 추상화**이다.

JDBC처럼 어댑터 패턴을 적용하여 같은 일을 하는 다수의 기술을 공통의 인터페이스로 제어할 수 있게 한 것을 서비스 추상화라고 한다. 스프링 프레임워크에서는 서비스 추상화를 위해 다양한 어댑터를 제공한다. 스프링은 OXM, ORM, 캐시, 트랜잭션 등 다양한 기술에 대한 PSA 즉 API를 제공한다.

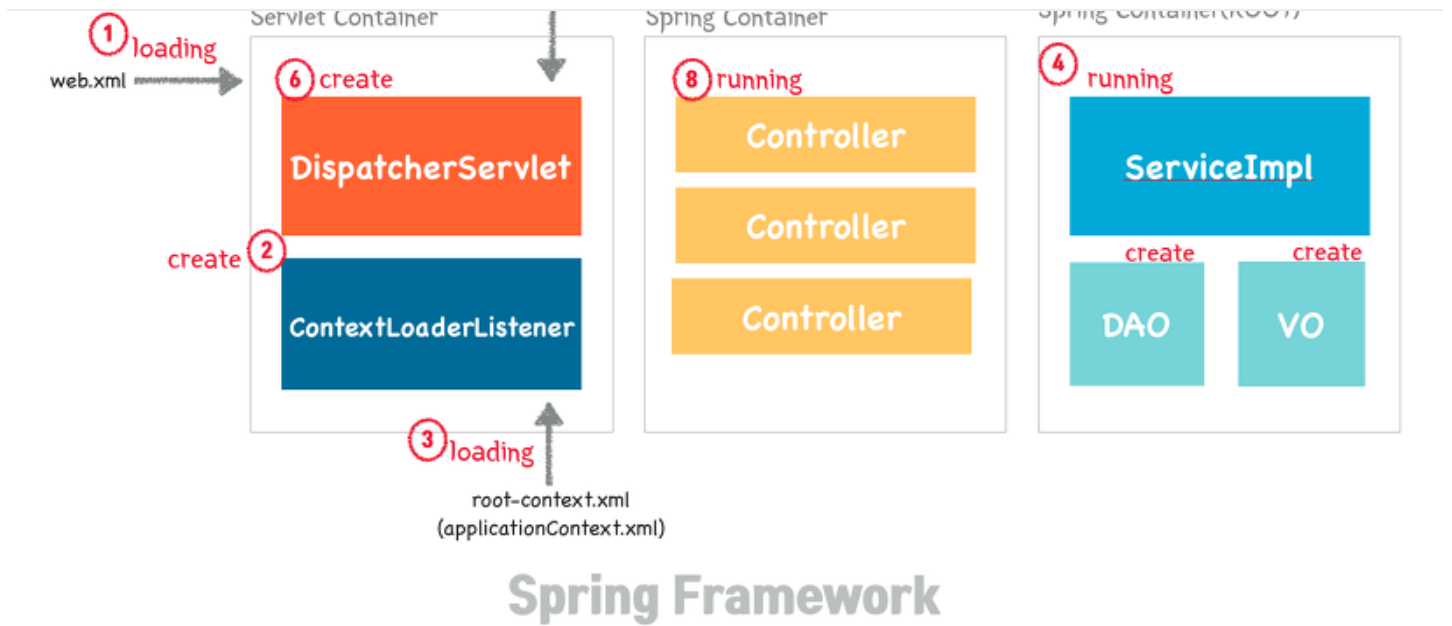
스프링 프레임워크는 POJO 만으로 어떻게 동작하는가?

내부에는 어떤 Servlet, Java Class들이 동작하고 있는가?



\_Jbee  
::Devlog::

CATEGORY



1. 웹 애플리케이션이 실행되면 Tomcat(WAS)에 의해 web.xml이 loading된다.
2. web.xml에 등록되어 있는 ContextLoaderListener(Java Class)가 생성된다. ContextLoaderListener 클래스는 ServletContextListener 인터페이스를 구현하고 있으며, ApplicationContext를 생성하는 역할을 수행한다.
3. 생성된 ContextLoaderListener는 root-context.xml을 loading한다.
4. root-context.xml에 등록되어 있는 Spring Container가 구동된다. 이 때 개발자가 작성한 비즈니스 로직에 대한 부분과 DAO, VO 객체들이 생성된다.
5. 클라이언트로부터 웹 애플리케이션이 요청이 온다.
6. DispatcherServlet(Servlet)이 생성된다. DispatcherServlet은 FrontController의 역할을 수행한다. 클라이언트로부터 요청 온 메시지를 분석하여 알맞은 PageController에게 전달하고 응답을 받아 요청에 따른 응답을 어떻게 할 지 결정만한다. 실질적 작업은 PageController에서 이루어지기 때문이다. 이러한 클래스들을 HandlerMapping, ViewResolver 클래스라고 한다.
7. DispatcherServlet은 servlet-context.xml을 loading 한다.
8. 두번째 Spring Container가 구동되며 응답에 맞는 PageController 들이 동작한다. 이 때 첫번째 Spring Container 가 구동되면서 생성된 DAO, VO, ServiceImpl 클래스들과 협업하여 알맞은 작업을 처리하게 된다.

end

## GoJS HTML5 Diagram Library

GoJS

Interactive diag  
JavaScript libra

11

구독하기

Jbee  
::Devlog::

CATEGORY

'Dev.BackEnd' &gt; Spring' 카테고리의 다른 글

[SpringBoot] WebSocket을 활용한 채팅 구현... (0)

Spring 의 시작, 프레임워크의 구성요소와 동작... (0)

TAG Spring

댓글 3건 Developer-jbee

박종억 ▾

♡ 추천 2

Tweet

f 공유

인기순 ▾



토론 참여하기



HongGyooKim • 4달 전

마지막에 간결하게 정리한 걸 보니 그 위의 내용들이 한 번에 정리되었습니다. 이해도 쏙쏙 되고 다른 포스팅에서 찾아볼 수 없던 개념 설명도 잘 와닿네요. 거의 완벽한 수준 ... !

정말 감사드립니다. ( \_ \_ )

^ | v • 답글 • 공유 &gt;



urong • 일년 전

정말 잘 읽었습니다. 감사합니다.

^ | v • 답글 • 공유 &gt;



Jbee 관리자 → urong • 일년 전

감사합니다! :D

^ | v • 답글 • 공유 &gt;

DEVELOPER-JBEE의 다른 댓글.

**[TCP] 3-way-handshake & 4-way-handshake**

댓글 한 건 • 2년 전



Alpaca — 잘 공부하고 갑니다!

**[Refactoring] 마틴 파울러, 리팩토링 정리 1편**

댓글 2건 • 2년 전



Jbee — 회신드렸습니다. 감사합니다 :)

타

**[OS] 멀티스레딩에 대해서**

댓글 2건 • 2년 전



Jbee — 상황에 따라, 컨텍스트에 따라 많이 달라질 수 있는 질문이라고 생각합니다 :) lock을 필요로 하는 환경은 멀티 스레드 환경입니다. 싱글 스레드 환경에서 ...

**[1장] 3. 전 세계의 DNS 서버가 연대한다**

댓글 2건 • 3년 전



이재호 — πππ 감동

타

구독 | 당신의 사이트에 Disqus 추가하기 Disqus 추가 | Disqus' Privacy Policy 개인정보 보호정책 개인정보 처리방침

1 ... 23 24 25 26 27 28 29 30 31 ... 341



\_Jbee  
::Devlog::



CATEGORY

2017년 지극히 개인적인 회고	cbr 언어라기 보다는 call by..	일	월	화	수	목	금	토	813,121			
					1	2	3	4				
[Refactoring] 마틴 파울러..	감사합니다! ㅎㅎ	5	6	7	8	9	10	11		Today	1,131	
[JS] JavaScript의 Event L..	요약 찰지게 정말 잘하시네요!	12	13	14	15	16	17	18		Yesterday	1,230	
		19	20	21	22	23	24	25				
SQL Injection이란 무엇인가	궁금한게 있습니다 그러면 cbr..	26	27	28	29	30	31					

Blog is powered by Tistory / Designed by Tistory