

삼성 청년 SW 아카데미

Front-End Programming

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

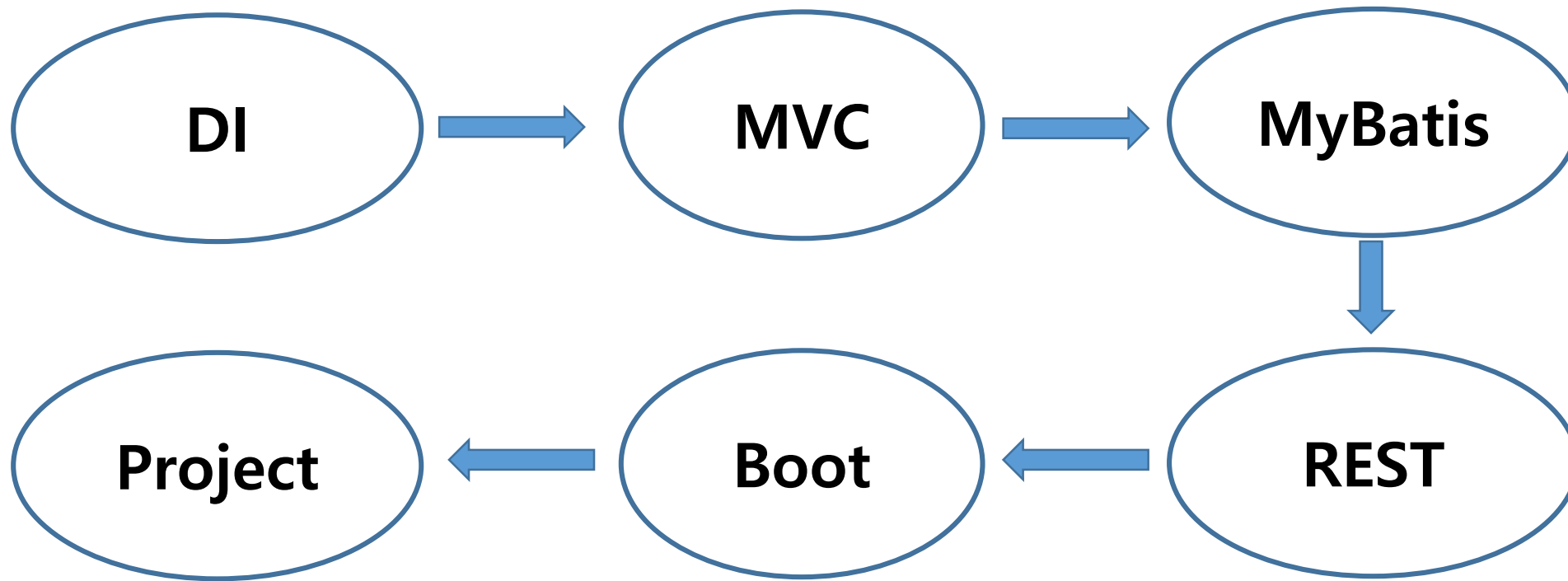
Confidential

**Spring 라이브
강의**

Confidential



교육스케줄



학습목표

- 스프링개발에 필요한 환경을 설정하면서 협업에 필요한 요소를 설명할 수 있다
- IoC에 대해 설명할 수 있다.
- DI에 대해 이해하고 적용하는 방법 습득과 응용 할 수 있다.

자바의 끝은? EJB

EJB : 자바 엔터프라이즈 애플리케이션 개발 표준

EJB 를 사용하면 애플리케이션 작성을 쉽게 할 수 있다.

저수준의 트랜잭션이나 상태관리, 멀티 쓰레딩, 리소스 풀링과 같은 복잡한 저수준의 API 따위를 이해하지 못하더라도 아무 문제 없이 애플리케이션을 개발할 수 있다.

그러나, 현실은 너무 어려웠다. EJB의 대안 : **스프링 프레임워크**



Rod Johnson은 자신의 저서에서 EJB를 사용하지 않고 엔터프라이즈 애플리케이션을 개발하는 방법을 소개

스프링 프레임워크의 모태

스프링 프레임워크란?

자바 엔터프라이즈 개발을 위한 오픈소스 경량 애플리케이션 프레임워크다.
공통 프로그래밍 모델 및 Configuration 모델을 제공한다.



프레임워크가 애플리케이션 수준의 인프라 구조를 제공
개발자가 귀찮은 일에 신경 쓰지 않고 비즈니스 로직 개발에 전념

참고)
엔터프라이즈 시스템이란 서버환경에서 동작하며 기업과
조직의 업무를 처리해주는 시스템

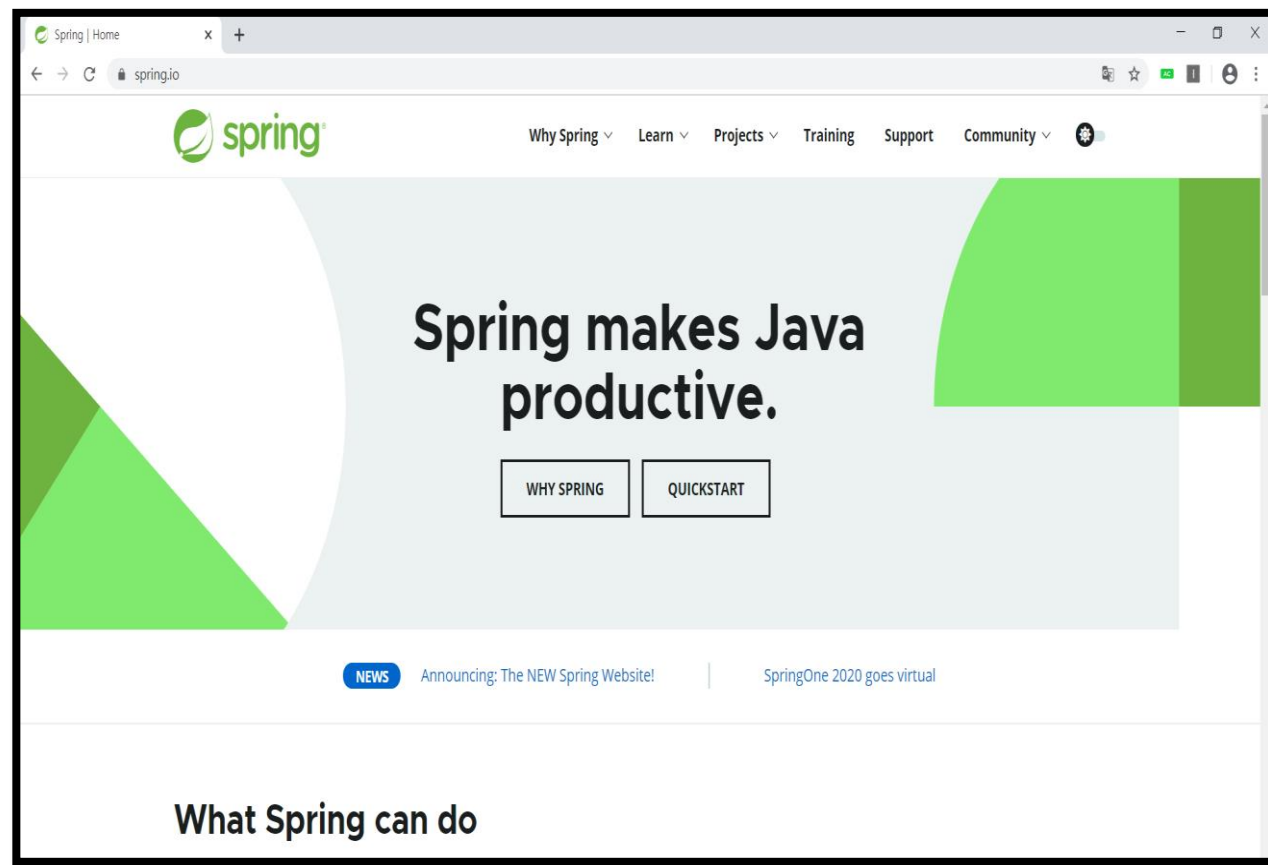
JAVA 온라인 라이브 강의

2003년 6월 아파치 2.0 라이선스로 공개

Confidential

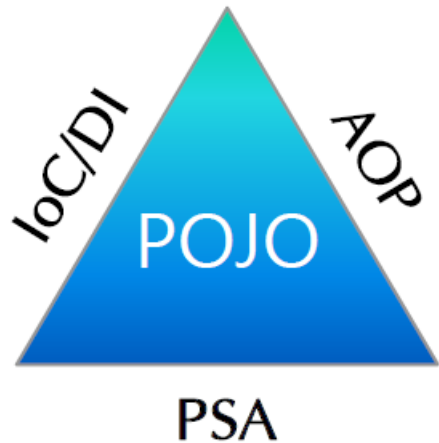
Release History

1.0 : 2004년 4월
2.0 : 2006년 10월
2.5 : 2007년 11월
3.0 : 2009년 12월
3.1 : 2011년 12월
3.1.4 : 2013년 1월
4.0.1 : 2014년 1월
4.0.5 : 2014년 5월
...
5.2.6 : 2020년 4월



스프링 삼각형

스프링 애플리케이션은 POJO를 이용해서 만든 애플리케이션 코드
POJO가 어떻게 관계를 맺고 동작하는지를 정의해놓은 설계정보로 구성된다.



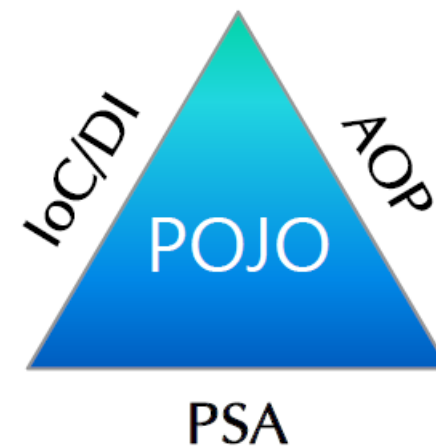
+



POJO (Plain Old Java Object)

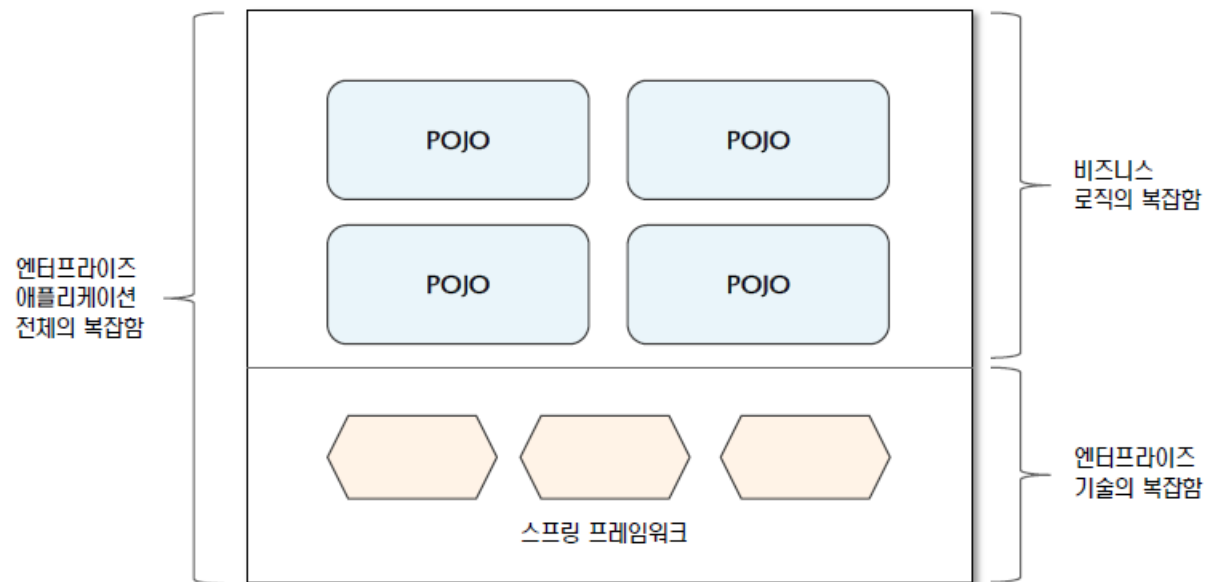
특별한 기술에 종속되지 않은 순수한 자바 객체
테스트하기 쉽고, 객체 지향 설계를 자유롭게 적용할 수 있음

- Martin Fowler, Rebecca Parson, Josh Mackenzie가
2000년에 만든 용어



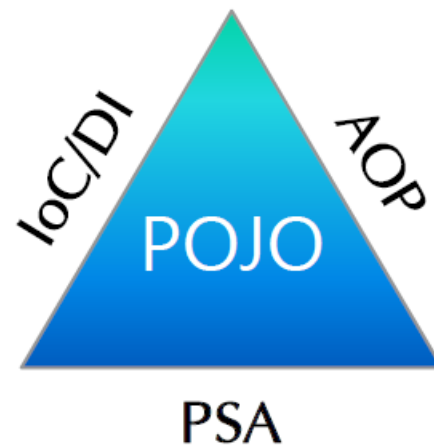
스프링이 엔터프라이즈 시스템의 복잡함을 다루는 방법

스프링은 비즈니스 로직의 복잡함과 엔터프라이즈 기술의 복잡함을 분리해서 구성할 수 있도록 도와준다.



Inversion Of Control

- IoC는 객체지향 언어에서 객체 간의 연결 관계를 런타임에 결정하게 하는 방법
- 객체 간의 관계가 느슨하게 연결됨(loose coupling)
- IoC의 구현 방법 중 하나가 DI(Dependency Injection)



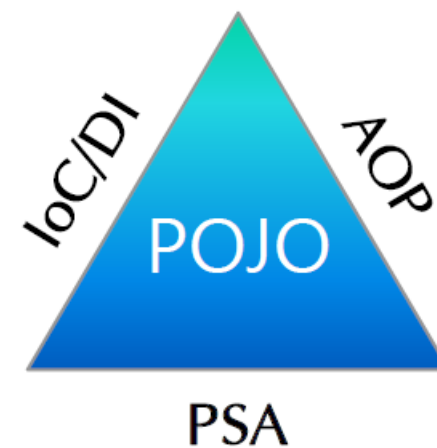
DI (Dependency Injection)

Confidential



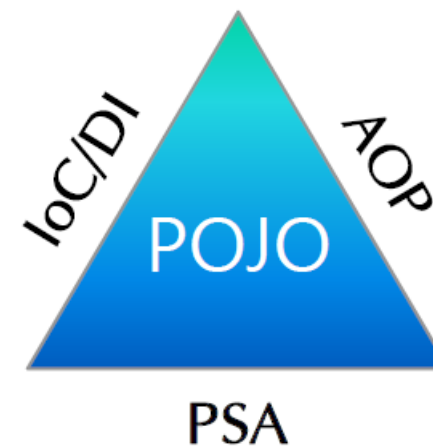
Aspect Oriented Programming

- AOP는 관심사의 분리를 통해서 소프트웨어의 모듈성을 향상시키고자 하는 프로그래밍 패러다임
- 소스코드 레벨에서 관심사의 모듈화를 지향하는 프로그래밍 방법이나 도구를 포함한다.



Portable Service Abstraction

PSA는 환경과 세부 기술의 변화에 관계없이 일관된 방식으로 기술에 접근할 수 있게 해주는 설계 원칙
(예를 들어, 데이터베이스에 관계 없이 동일하게 적용할 수 있는 트랜잭션 처리 방식)



경량 컨테이너

- **가벼움 (Lightweight)**

20 여개의 모듈로 세분화되는 수십만 코드 라인

- **불필요하게 무겁지 않음**

Spring이 필요로 하는 처리 능력(성능)은 무시할 정도

- **경량 컨테이너(Lightweight Container)**

애플리케이션 객체의 생명주기(Lifecycle)와 설정(Configuration)을 담고 관리한다.

애플리케이션 객체를 생성하고, 설정하고, 객체 간의 관계를 지정하는 방법을 제공한다.

내일 방송에서 만나요!

삼성 청년 SW 아카데미