

Spring Framework

- 엔터프라이즈 급 애플리케이션을 만들기 위한 모든 기능을 종합적으로 제공하는 경량화 된 솔루션이다.
- JEE(Java Enterprise Edition)가 제공하는 다수의 기능을 지원하고 있기 때문에, JEE를 대체하는 Framework로 자리잡고 있다.
- SpringFramework는 JEE가 제공하는 다양한 기능을 제공하는 것 뿐만 아니라, DI(Dependency Injection)나 AOP (Aspect Oriented Programming)와 같은 기능도 지원 한다.

Spring Framework

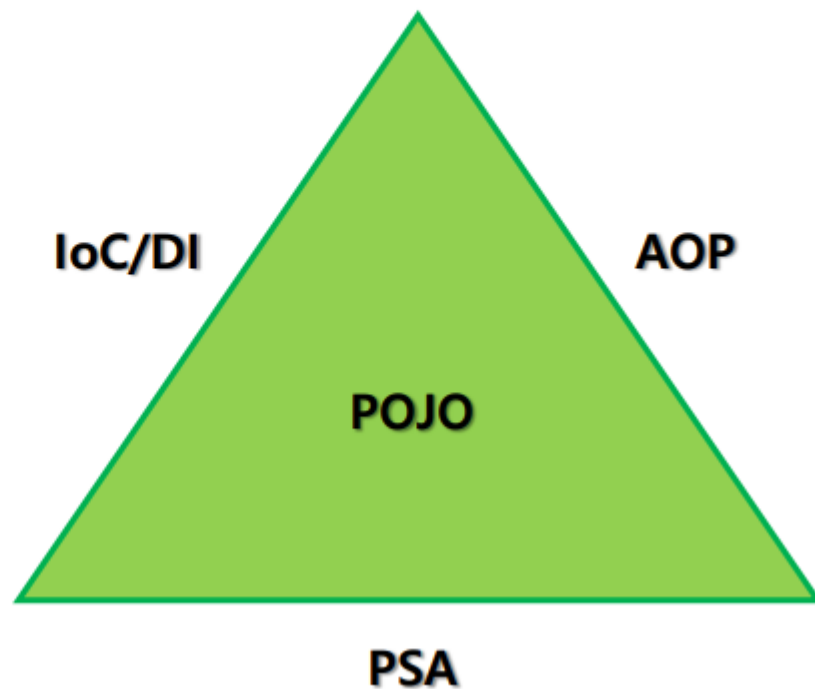
- Spring Framework는 자바로 Enterprise Application을 만들 때 포괄적으로 사용하는 Programming 및 Configuration Model을 제공해 주는 Framework로 Application 수준의 인프라 스트럭처를 제공.
- 즉, 개발자가 복잡하고 실수하기 쉬운 Low Level에 신경 쓰지 않고 Business Logic개발에 전념할 수 있도록 해준다.



Enterprise System이란 서버에서 동작하며
기업의 업무를 처리해주는 System

Spring Framework-구조

- Spring 삼각형.
 - Enterprise Application 개발 시 복잡함을 해결하는 Spring의 핵심.
 - ① POJO
 - ② PSA
 - ③ IoC/DI
 - ④ AOP



Spring Framework-구조

- Spring 삼각형.

- Enterprise Application 개발 시 복잡함을 해결하는 Spring의 핵심.

① **POJO**

② PSA

③ IoC/DI

④ AOP



✓ **POJO (Plain Old Java Object)**

- 특정 환경이나 기술에 종속적이지 않은 객체지향 원리에 충실한 자바객체.
- 테스트하기 용이하며, 객체지향 설계를 자유롭게 적용할 수 있다.

Spring Framework-구조

▪ Spring 삼각형.

- Enterprise Application 개발 시 복잡함을 해결하는 Spring의 핵심.

① POJO

② **PSA**

③ IoC/DI

④ AOP



✓ **PSA (Portable Service Abstraction)**

- 환경과 세부기술의 변경과 관계없이 일관된 방식으로 기술에 접근할 수 있게 해 주는 설계 원칙.
- 트랜잭션 추상화, OXM 추상화, 데이터 액세스의 Exception 변환기능..등 기술적인 복잡함은 추상화를 통해 Low Level의 기술 구현 부분과 기술을 사용하는 인터페이스로 분리.
- 예를 들어 데이터베이스에 관계없이 동일하게 적용 할 수 있는 트랜잭션 처리방식.

Spring Framework-구조

- Spring 삼각형.
 - Enterprise Application 개발 시 복잡함을 해결하는 Spring의 핵심.

① POJO

② PSA

③ IoC/DI

④ AOP



✓ IoC/DI (Dependency Injection)

- DI는 유연하게 확장 가능한 객체를 만들어 두고 객체 간의 의존관계는 외부에서 다이나믹하게 설정.

Spring Framework-구조

- Spring 삼각형.

- Enterprise Application 개발 시 복잡함을 해결하는 Spring의 핵심.

- ① POJO

- ② PSA

- ③ IoC/DI

- ④ **AOP**



✓ **AOP (Aspect Oriented Programming)**

- 관심사의 분리를 통해서 소프트웨어의 모듈성을 향상.
- 공통 모듈을 여러 코드에 쉽게 적용가능.

Spring Framework-특징

▪ 경량컨테이너

- 스프링은 자바객체를 담고 있는 컨테이너이다.
- 스프링 컨테이너는 이들 자바 객체의 생성과 소멸과 같은 라이프사이클을 관리.
- 언제든지 스프링 컨테이너로부터 필요한 객체를 가져와 사용할 수 있다.

▪ DI(Dependency Injection - 의존성 지원) 패턴 지원

- 스프링은 설정 파일이나, 어노테이션을 통해서 객체 간의 의존 관계를 설정할 수 있다.
- 따라서, 객체는 의존하고 있는 객체를 직접 생성하거나 검색할 필요가 없다.

Spring Framework-특징

▪ AOP(Aspect Oriented Programming - 관점 지향 프로그래밍) 지원

- AOP는 문제를 바라보는 관점을 기준으로 프로그래밍하는 기법이다.
- 이는 문제를 해결하기 위한 핵심관심 사항과 전체에 적용되는 공통관심 사항을 기준으로 프로그래밍 함으로서 공통 모듈을 여러 코드에 쉽게 적용할 수 있도록 한다.
- 스프링은 자체적으로 프록시 기반의 AOP를 지원하므로 트랜잭션이나, 로깅, 보안과 같이 여러 모듈에서 공통으로 필요로 하지만 실제 모듈의 핵심이 아닌 기능들을 분리하여 각 모듈에 적용이 가능하다.

▪ POJO(Plain Old Java Object) 지원

- 특정 인터페이스를 구현하거나 또는 클래스를 상속하지 않는 일반 자바 객체 지원.
- 스프링 컨테이너에 저장되는 자바객체는 특정한 인터페이스를 구현 하거나, 클래스 상속 없이도 사용이 가능하다.
- 일반적인 자바 객체를 칭하기 위한 별칭 개념이다.

Spring Framework-특징

▪ IoC(Inversion of Control - 제어의 반전)

- IoC는 스프링이 갖고 있는 핵심적인 기능이다.
- 자바의 객체 생성 및 의존관계에 있어 모든 제어권은 개발자에게 있었다.
- Servlet과 EJB가 나타나면서 기존의 제어권이 Servlet Container 및 EJB Container에게 넘어가게 됐다
- 단, 모든 객체의 제어권이 넘어간 것은 아니고 Servlet, EJB에 대한 제어권을 제외한 나머지 객체 제어권은 개발자들이 담당하고 있다.
- 스프링에서도 객체에 대한 생성과 생명주기를 관리할 수 있는 기능을 제공하고 있는데 이런 이유로 [Spring Container] 또는 [IoC Container] 라고 부르기도 한다.

▪ 스프링은 트랜잭션 처리를 위한 일관된 방법을 제공

- JDBC, JTA 또는 컨테이너가 제공하는 트랜잭션을 사용하든, 설정파일을 통해 트랜잭션 관련정보를 입력하기 때문에 트랜잭션 구현에 상관 없이 동일한 코드를 여러 환경에서 사용이 가능하다.

Spring Framework-특징

- **스프링은 영속성과 관련된 다양한 API를 지원**

- 스프링은 JDBC를 비롯하여 iBatis, myBatis, Hibernate, JPA등 DB처리를 위해 널리 사용되는 라이브러리와 연동을 지원하고 있다.

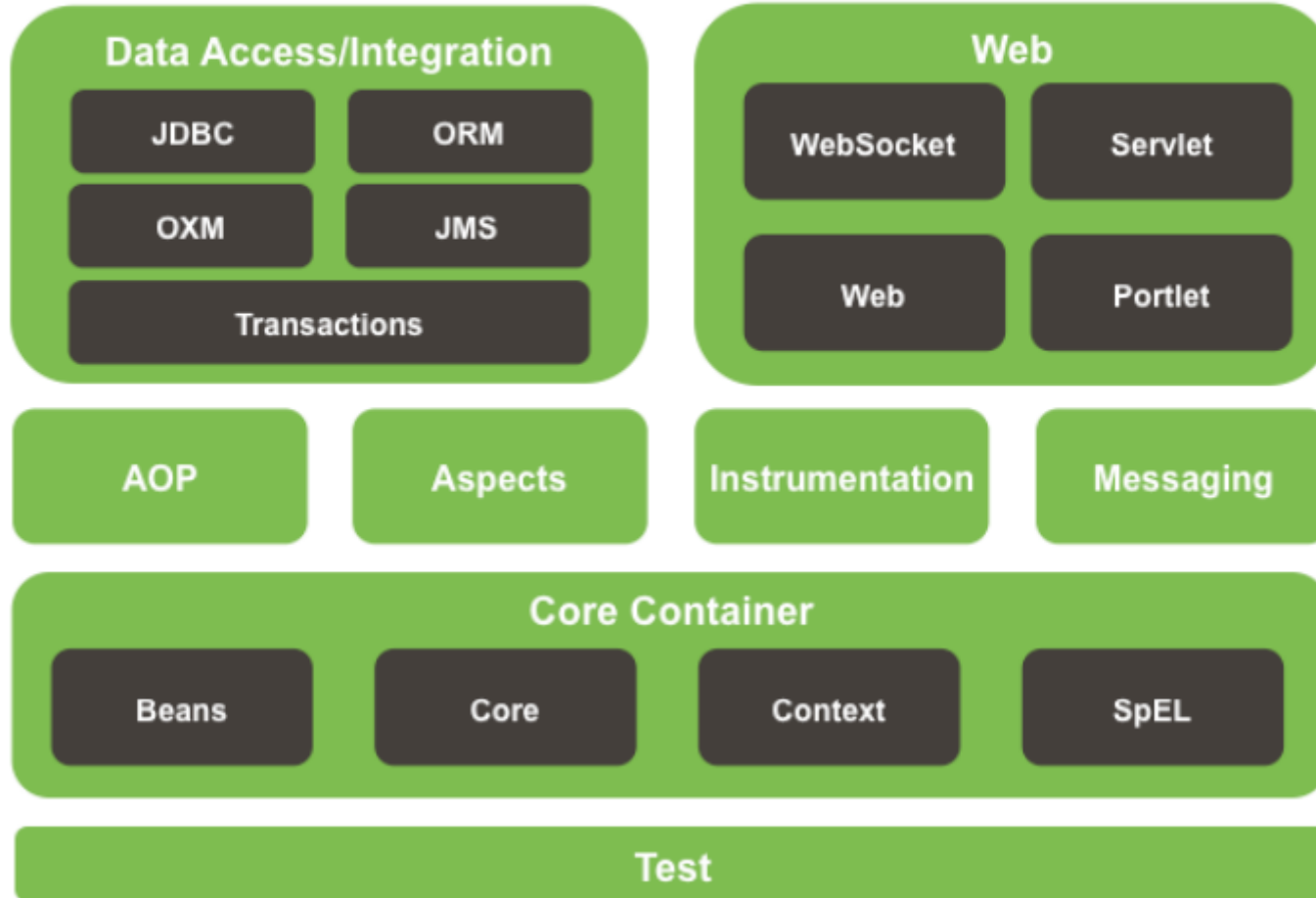
- **스프링은 다양한 API에 대한 연동을 지원**

- 스프링은 JMS, 메일, 스케줄링 등 엔터프라이즈 어플리케이션 개발에 필요한 다양한 API를 설정 파일과 어노테이션을 통해서 손쉽게 사용할 수 있도록 지원하고 있다.

Spring Framework-Module



Spring Framework Runtime



Spring Framework-Module

1. Spring Core	Spring Framework의 핵심 기능을 제공 하며, Core 컨테이너의 주요 컴포넌트는 Bean Factory이다.
2. Spring Context	- Spring을 컨테이너로 만든 것이 Spring core의 BeanFactory라면, Spring을 Framework로 만든 것은 Context module이며, 이 module은 국제화된 메시지, Application 생명주기 이벤트, 유효성 검증 등을 지원함으로써 BeanFactory의 개념을 확장을 한다.
3. Spring AOP	설정 관리 기능을 통해 AOP 기능을 Spring Framework와 직접 통합 시킨다.
4. Spring DAO	Spring JDBC DAO 추상 레이어는 다른 데이터베이스 벤더들의 예외 핸들링과 오류 메시지를 관리하는 중요한 예외계층을 제공합니다.
5. Spring ORM	Spring Framework는 여러 ORM(Object Relational Mapping) Framework에 플러그인 되어, Object Relational 툴(JDO, Hibernate, iBatis)을 제공한다.
6. Spring Web	Web Context module은 Application Context module 상위에 구현되어, Web 기반 Application에 context를 제공한다.
7. Spring Web MVC	Spring Framework는 자체적으로 MVC 프레임워크를 제공하고 있으며, 스프링만 사용 해도 MVC기반의 웹 어플리케이션을 어렵지 않게 개발이 가능하다.