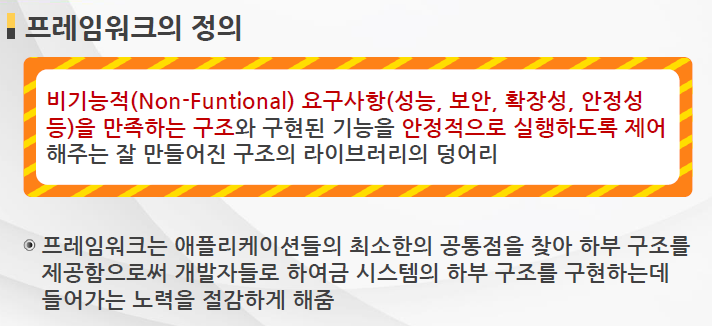
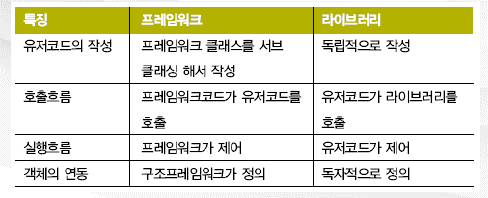
교육 결과 보고서

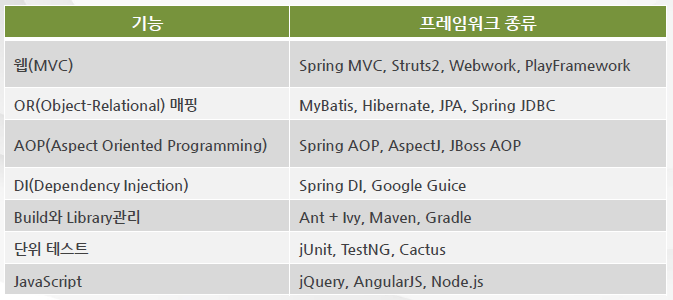
[Online] Spring Framework



# 프레임워크 와 라이브러리 비교



# 프레임 워크 종류



**Spring Framework**

Java 엔터프라이즈 개발을 편하게 해주는 오픈소스 경량급 애플리케이션 프레임워크

1. 특징

# 컨테이너 역할 수행

Java 객체의 LifeCycle 관리 및 Spring 컨테이너로부터 필요한 객체를 가져와 사용 할 수 있음.

# DI (Dependency Injection) 지원

설정 파일이나 어노테이션을 통해 객체 간의 의존관계를 설정 할 수 있음.

# AOP (Aspect Oriented Programming) 지원

트랜잭션, 로깅, 보안과 같은 공통 모듈을 실제 핵심 모듈에서 분리해서 적용 할 수 있음.

# POJO (Plain Old Java Object) 지원

Spring 컨테이너에 저장되는 Java 객체는 특정한 인터페이스를 구현하거나, 특정 클래스를

상속 받지 않아도 됨.

# 트랜잭션 처리를 위한 일관된 방법을 지원

JDBC, JTA 등 어떤 트랜잭션을 사용하던 설정을 통해 정보를 관리 하므로 트랜잭션 구현에

상관 없이 동일한 코드 사용 가능

# 영속성 (Persistence) 과 관련된 다양한 API 지원

MyBatis, Hibernate 등 데이터베이스 처리를 위한 ORM (Object Relational Mapping) 프레임워크와

연동 지원

1. 기능 요소



# Spring Core

Spring 프레임워크의 기본 기능 제공.

Spring Core 에 있는 BeanFactory 는 Spring 의 기본 컨테이너이면서 스프링 DI 의 기반.

# Spring AOP

Aspect 지향 프로그래밍 지원

# Spring ORM

MyBatis, Hibernate, JPA 등 ORM 프레임워크와의 연결 고리 제공.

ORM 제품들을 Spring 기능과 조합하여 사용 가능 하게 함.

# Spring DAO

JDBC 에 대한 추상화 계층으로 JDBC 코딩이나 예외 처리 부분 간소화.

AOP 모듈을 이용해 트랜잭션 관리 서비스도 제공.

# Spring Web

웹 애플리케이션 개발에 필요한 기본 기능 제공.

Webwork 나 Struts 와 같은 다른 웹 애플리케이션 프레임워크와의 통합 지원.

# Spring Context

BeanFactory 개념을 확장, 국제화(I18N) 메시지, 애플리케이션 생명 주기 이벤트 및

유효성 검증 등 지원.

# Spring Web MVC

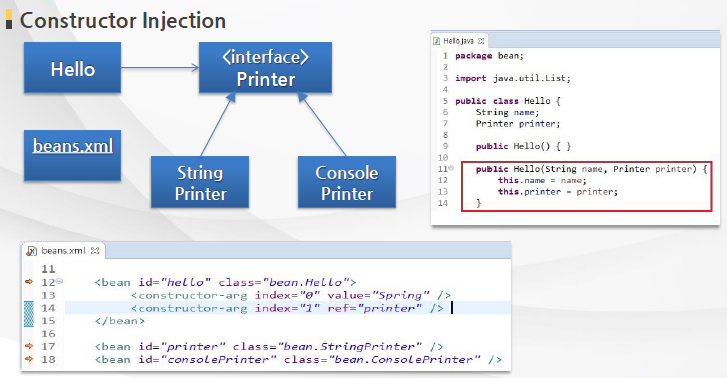
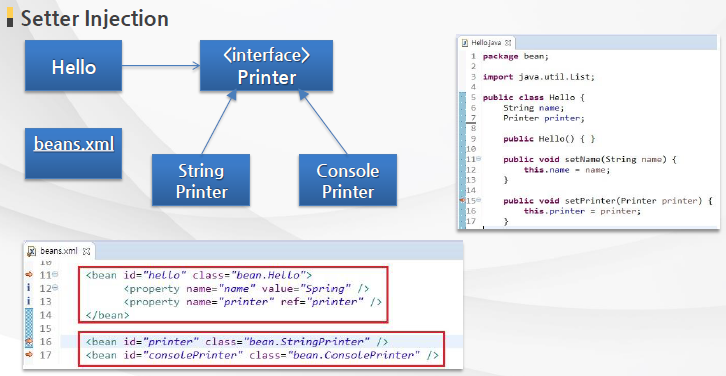
Model, View, Controller 모델을 이용하는 웹 애플리케이션 지원

1. IoC & DI
   1. IoC (Inversion of Control)

객체의 생성, 생명 주기 관리 등 모든 객체에 대한 제어권을 프레임워크가 제공함.

* 1. DI (Dependency Injection)

각 클래스(Bean) 간 의존 관계를 빈 설정 (Bean Definition) 정보를 바탕으로 프레임워크의 컨테이너가 자동으로 연결함.

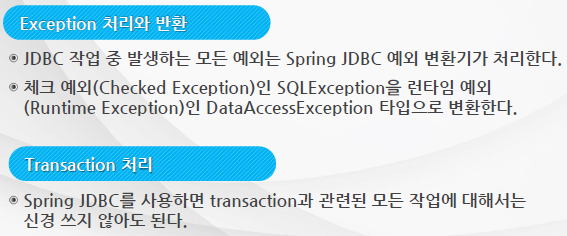
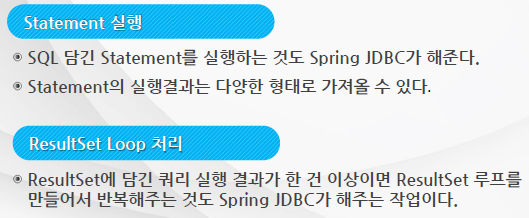


Spring DI 컨테이너 : BeanFactory, Application Context

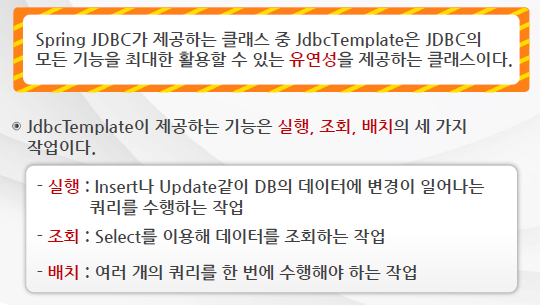
1. Spring JDBC

JDBC 의 장점과 단순성을 유지 하며 기존 JDBC 의 단점 극복.

간결한 형태의 API 사용법을 제공하며, JDBC API 에서 지원되지 않는 편리한 기능 제공.



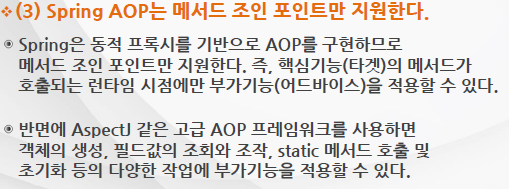
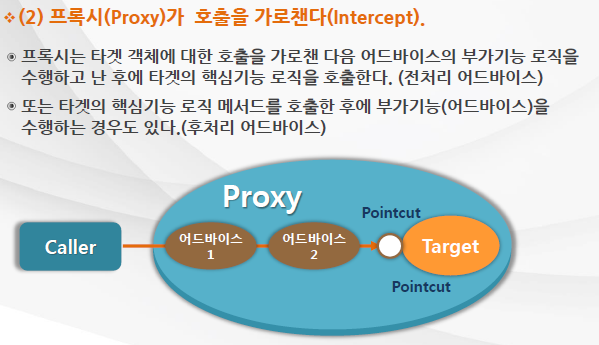
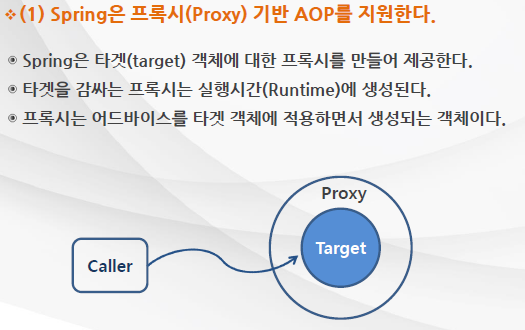
# JDBC Template 클래스



1. Spring AOP (Aspect Oriented Programming)

AOP 는 애플리케이션에서의 관심사의 분리(기능의 분리) 즉, 핵심적인 기능에서 부가기능을 분리한다. 분리한 부가기능을 애스팩트(Aspect)라는 형태의 독특한 모듈 형태로 만들어서 설계하고 개발하는 방법.

# 특징

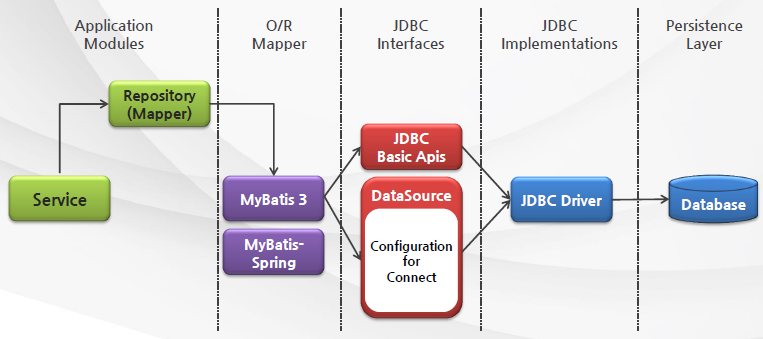


# 구현 방식

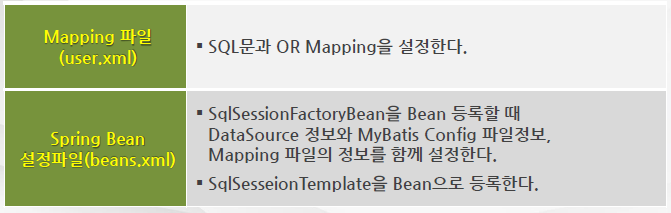
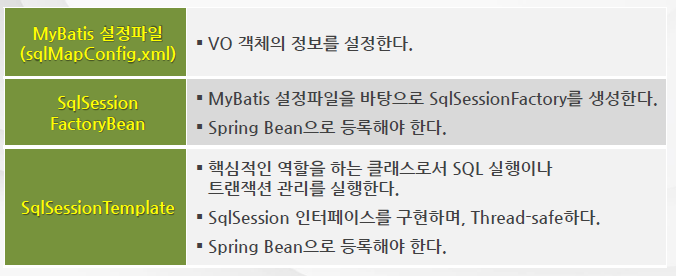
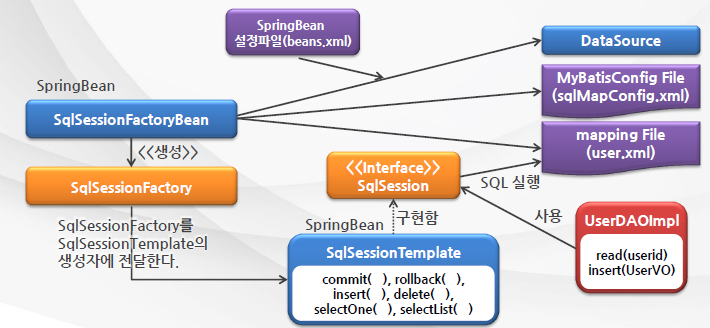
XML 기반이 POJO 클래스를 이용한 구현과 @Apsect 어노테이션을 이용한 AOP 구현 가능.

1. MyBatis-Spring 컴포넌트

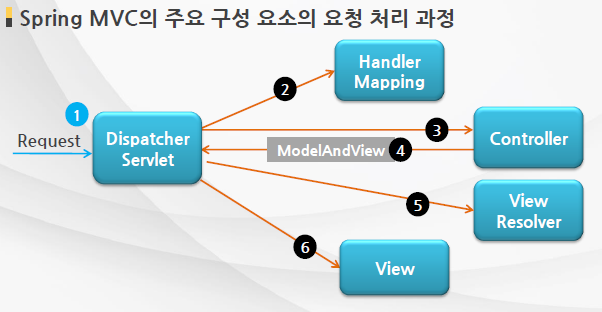
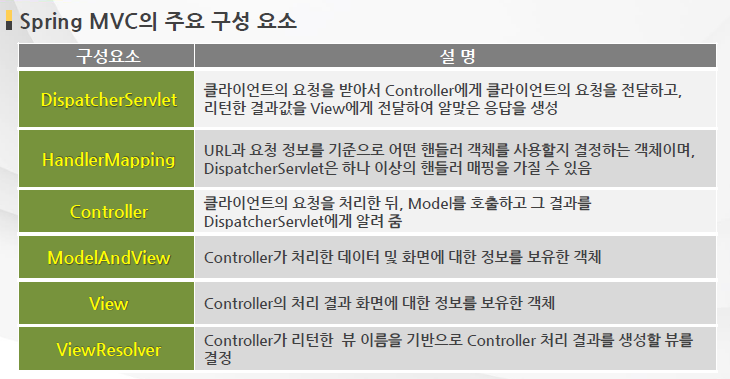
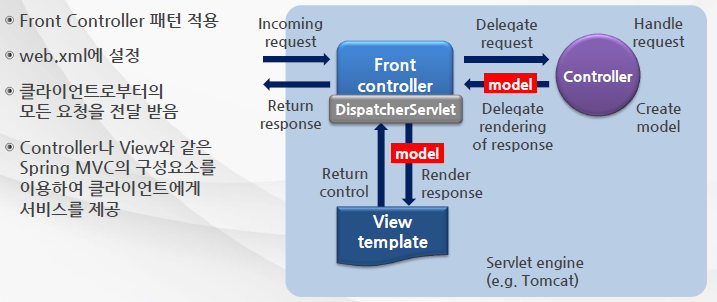
MyBatis는 자바 오브젝트와 SQL 문 사이의 자동 Mapping 기능을 지원하는 ORM(Object Relation Mapping) 프레임 워크.



# 주요 컴포넌트



1. Spring MVC



1. 클라이언트의 요청이 DispatcherServlet 에게 전달
2. DispatcherServlet 은 HandlerMapping을 사용하여 클라이언트의 요청을 처리할 Controller 를 획득
3. DispatcherServlet은 Controller 객체를 이용하여 클라이언트의 요청을 처리
4. Controller 는 클라이언트 요청 처리 결과와 View 페이지 정보를 담은 ModleAndView 객체 반환
5. DispatcherServlet은 ViewReolver로부터 응답 결과를 생성할 View 객체 선택
6. View는 클라이언트에게 전송할 응답을 생성
7. spring RESTful 웹 서비스

