

Spring Framework

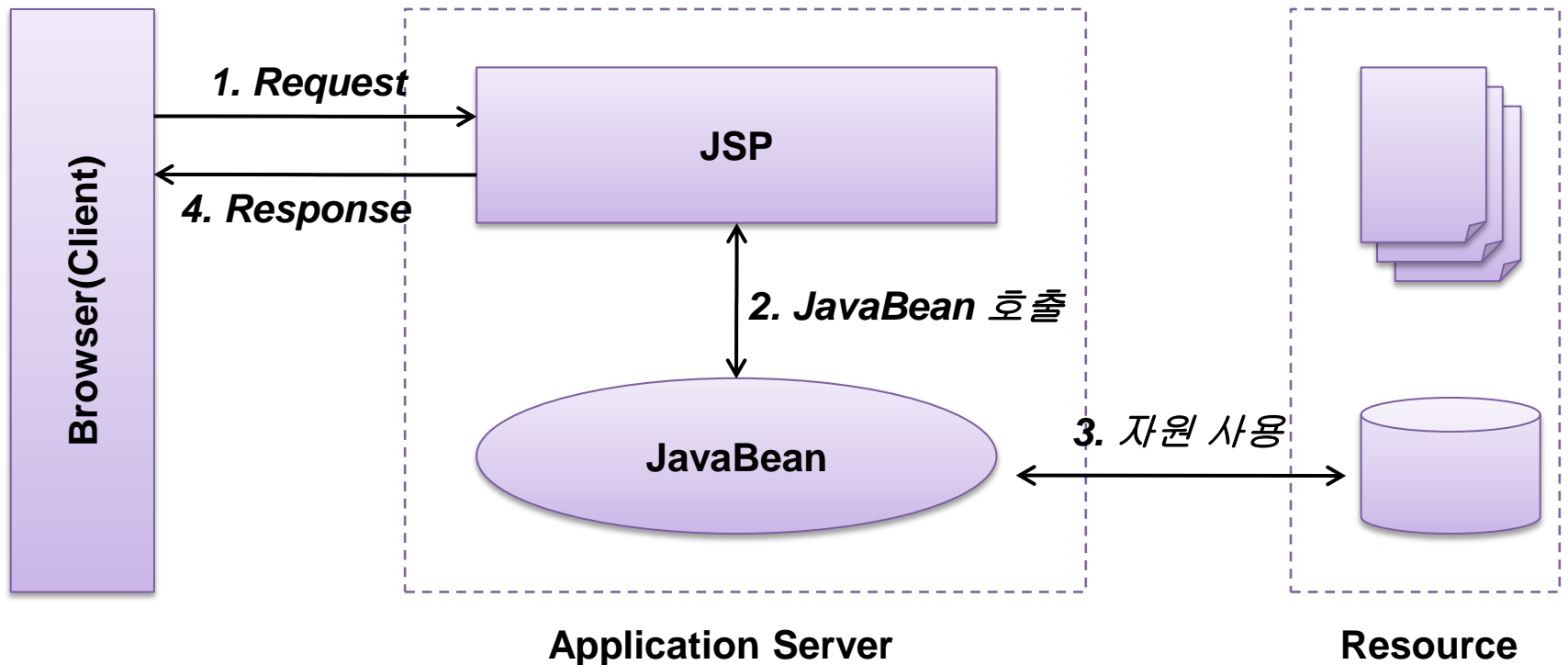
- 스프링 프레임워크 소개

■ CONTENTS

- Web Application 설계방식
- 프레임워크란
- 스프링 프레임워크란
- 스프링프레임워크 설치와 모듈 구성
- DepeDency Injection과 스프링프레임워크
- AOP와 스프링

■ Web Application 설계방식

- 모델1 개요
 - JSP 만 이용하여 개발하는 경우
 - JSP + Java Bean을 이용하여 개발하는 경우
 - Model2의 Controller 개념이 모호



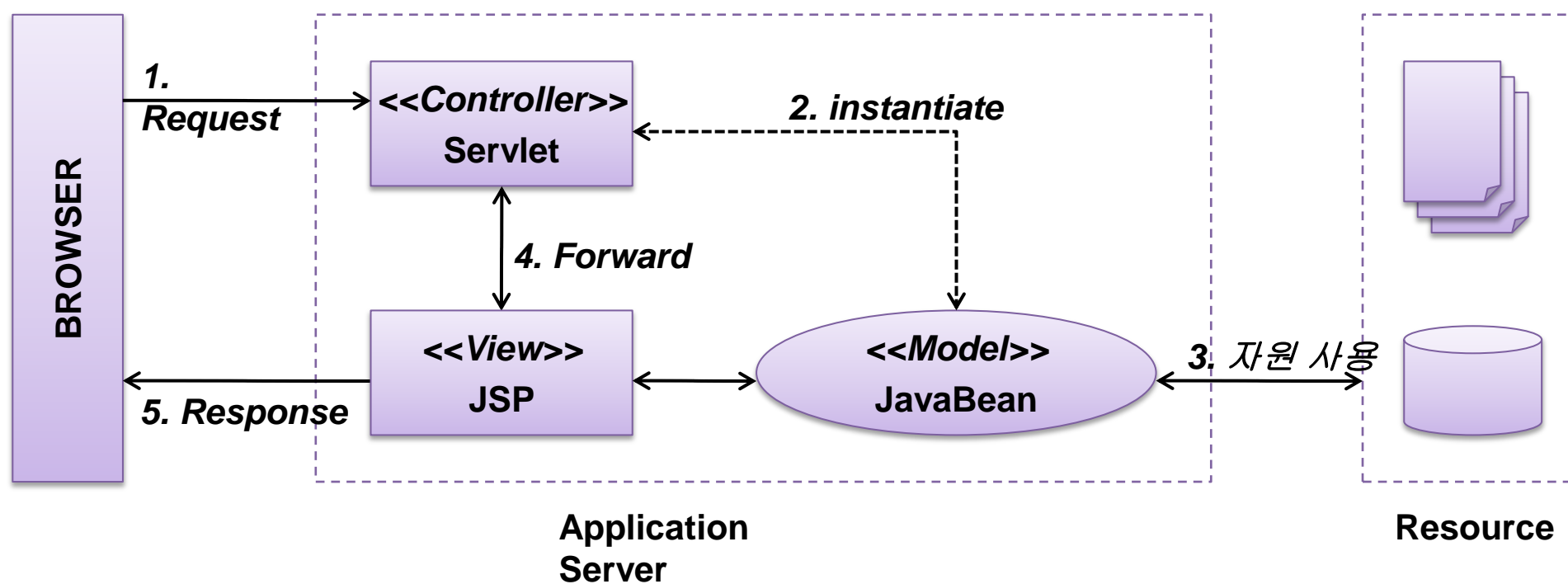
■ Web Application 설계방식

- 모델1의 장단점
 - 장점
 - 개발속도가 빠름
 - 개발자의 기술적인 숙련도가 낮아도 배우기 쉬워 빠르게 적용 가능
 - 단점
 - JSP 페이지에서 프레젠테이션 로직과 비즈니스 로직이 혼재되어 복잡
 - 로직의 혼재로 인해 개발자와 디자이너의 작업 분리가 어려움
 - JSP 코드의 복잡도로 인해 유지보수가 어려워짐
- 웹 애플리케이션이 복잡해지고 사용자 요구가 증가함에 따라 새로운 개발 방식을 요구

• 모델2 개요

- GUI 개발모델인 MVC를 웹 애플리케이션에 적용한 방식
- Application의 역할을 Model - View - Controller로 분리
- **Model:** Business Logic 담당
 - Java Bean으로 구현
 - Business Service(Manager)
 - Business Logic의 workflow 관리
 - DAO (Data Access Object) : DB 연동해 Business Logic 처리
- **View:** Client에게 응답을 처리한다.
 - JSP로 구현
- **Controller:** 클라이언트의 요청을 받아 Model과 View사이에서 이벤트 흐름 제어
 - Servlet으로 구현
 - Client의 요청을 받아 Client가 보낸 Data를 읽고 검사
 - Model에게 Business Logic을 요청
 - Model의 처리 결과에 맞는 View에게 응답 요청

■ Web Application 설계방식



- **모델2의 장단점**

- **장점**

- 비즈니스 로직과 프리젠테이션의 분리로 인해 어플리케이션이 명료해지며 유지보수와 확장이 용이함
 - 디자이너와 개발자의 작업을 분리해 줌

- **단점**

- 개발 초기에 아키텍처 디자인을 위한 시간의 소요로 개발 기간이 늘어남
 - MVC 구조에 대한 개발자들의 이해가 필요함

■ 프레임워크

- 프레임워크

- 뼈대 혹은 틀
- 소프트웨어 관점 : 아키텍처에 해당하는 골격 코드

- '아키텍처' , '골격'

- 애플리케이션을 개발할 때 가장 중요한 것이 애플리케이션의 구조를 결정하는 것이 아키텍처인데 이 아키텍처에 해당하는 골격 코드를 프레임워크가 제공한다.

예) 컵을 만든다.

A와 B는 자유롭게 컵을 만든다.

A의 퇴사, A가 만든 컵을 고쳐야 한다....

■ 프레임워크

- **기존 애플리케이션 개발 과정의 문제점**

- 시스템 개발과정에서 대부분의 개발자들은 산출물에 입각해서 개발을 하므로 아키텍처의 일관성이 잘 유지 되지만 유지보수 과정에서는 개발자의 경험에 의존하는 경우가 많다.

- **프레임워크는 이러한 문제를 근본적으로 해결**

- 애플리케이션 개발에 기본이 되는 뼈대나 틀을 제공

: 개발자에게 모든 것을 위임하는 것이 아니라 애플리케이션의 기본 아키텍처는 프레임워크가 제공하고, 그 뼈대에 살을 붙이는 작업만 개발자가 하는 것이다.

■ 프레임워크

- **프레임워크의 장점**

- 1. 빠른 구현 시간**

개발자는 비즈니스 영역만 구현하면 됨. 제한된 시간에 많은 기능을 구현할 수 있다.

- 2. 쉬운 관리**

유지보수에 들어가는 인력과 기간을 줄일 수 있다.

- 3. 개발자의 역량 획일화**

숙련된 개발자와 초급개발자의 코드가 비슷해진다.

관리자 입장에서 개발 인력을 더 효율적으로 구성할 수 있다.

- 4. 검증된 아키텍처의 재사용과 일관성 유지**

아키텍처에 관한 고민이나 검증 없이 애플리케이션을 개발한다.

유지보수 과정에서 아키텍처가 왜곡되거나 변형되지 않는다.

■ 스프링 프레임워크란

- 로드존슨이 2004년에 만든 오픈 소스 프레임워크.
- 스프링 프레임워크가 등장하기 전까지는 자바 기반의 엔터프라이즈 애플리케이션은 대부분 EJB(Enterprise Java Beans)로 개발되었다.
- EJB의 문제점
 - 복잡하고, 고가의 비용발생, 많은 시간과 노력이 필요.
스팩의 복잡함, 학습에 많은 시간이 필요, 유지보수 역시 복잡함,
설치를 위해 WAS(Web Application Sever)가 필요
: JEUS, Weblogic, WebSpere 등, 수천만원의 고가 장비.
 - 다양한 디자인 패턴을 이해하고 적용해야 함.
- 스프링 프레임워크는 이미 많은 디자인 패턴이 적용되어 배포되는 프레임워크이기 때문에 많은 디자인 패턴을 사용하는 것과 같다.

■ 스프링 프레임워크의 주요 특징

- 스프링은 경량의 프레임워크이다.
 - 자바 객체를 담고 있는 컨테이너다
 - 자바객체의 생성, 소멸과 같은 라이프 사이클을 관리한다.
- 스프링은 **DI(dependency Injection)패턴**을 지원한다.
 - 설정파일을 통해서 의존관계를 설정할 수 있다.
- 스프링은 **AOP(Aspect Oriented Programming)**를 지원한다.
 - 트랙잭션이나 로깅, 보안과 같은 공통으로 필요로 하는 기능을 분리해서 각각의 모듈에 적용할 수 있다.

■ 스프링 프레임워크의 주요 특징

- 트랜잭션 처리를 위한 일관된 방법을 제공한다.
- 영속성(데이터베이스)과 관련된 다양한 API를 제공한다.
 - JDBC, iBatis(myBatis), Hibernate, JPA 등과 같은 프레임워크와의 연동을 지원한다.
- 자체적으로 **MVC 프레임워크**를 제공
 - 스프링만 가지고 MVC 기반의 웹 어플리케이션을 개발

* 영속성(persistence)은

데이터를 생성한 프로그램의 실행이 종료되더라도 사라지지 않는 데이터의 특성을 의미한다. 영속성은 파일 시스템, **관계형 데이터베이스** 혹은 객체 데이터베이스 등을 활용하여 구현한다. 영속성을 갖지 않는 데이터는 단지 메모리에서만 존재하기 때문에 프로그램을 종료하면 모두 잃어버리게 된다. 결국 영속성은 특정 데이터 구조를 이전 상태로 복원할 수 있게 해주어 프로그램의 종료와 재개를 자유롭게 해준다.

■ 스프링 프레임워크의 주요 특징

HW / OS

행함 x 중속속 x

• 경량

- 스프링은 크기측면에서 가볍다.
- 여러 개의 모듈로 구성, 모듈은 하나 이상의 JAR 파일로 구성
- JAR 파일만 있으면 개발과 실행이 가능
- 스프링을 이용해서 만든 애플리케이션의 배포 역시 매우 빠르고 쉽다.
- POJO 특별한 규칙 없이 단순한 객체 (상속이나 포함의 조건이 없음.)





















(1.6
외국
~ ~ ~)

반드나 x

■ 스프링 프레임워크의 주요 특징

- 웹 어플리케이션에 필요한 라이브러리

▼ Maven Dependencies

- >  spring-context-3.1.1.RELEASE.jar - C:\Users\...
- >  spring-aop-3.1.1.RELEASE.jar - C:\Users\Wji...
- >  aopalliance-1.0.jar - C:\Users\Wjin\m2\wre...
- >  spring-beans-3.1.1.RELEASE.jar - C:\Users\W...
- >  spring-core-3.1.1.RELEASE.jar - C:\Users\Wj...
- >  spring-expression-3.1.1.RELEASE.jar - C:\WU...
- >  spring-asm-3.1.1.RELEASE.jar - C:\Users\Wji...
- >  spring-webmvc-3.1.1.RELEASE.jar - C:\WUse...
- >  spring-context-support-3.1.1.RELEASE.jar -
- >  spring-web-3.1.1.RELEASE.jar - C:\Users\Wji...
- >  aspectjrt-1.6.10.jar - C:\Users\Wjin\m2\wre...
- >  slf4j-api-1.6.6.jar - C:\Users\Wjin\m2\wrep...
- >  jcl-over-slf4j-1.6.6.jar - C:\Users\Wjin\m2\...
- >  slf4j-log4j12-1.6.6.jar - C:\Users\Wjin\m2\...
- >  log4j-1.2.15.jar - C:\Users\Wjin\m2\wrepos...
- >  javax.inject-1.jar - C:\Users\Wjin\m2\wrepo...
- >  servlet-api-2.5.jar - C:\Users\Wjin\m2\wrep...
- >  jsp-api-2.1.jar - C:\Users\Wjin\m2\wreposit...
- >  jstl-1.2.jar - C:\Users\Wjin\m2\wrepository\...
- >  junit-4.7.jar - C:\Users\Wjin\m2\wrepositor...

■ 스프링 프레임워크의 주요 특징

- Plain Old Java Object[POJO] 지원
 - 특정 규약 및 환경에 종속적이지 않은 평범한 일반 자바 클래스 의미
 - 상속이나 구현이 필요하지 않은 클래스
 - Spring 개발에 POJO 클래스를 활용할 수 있다는 건 특정 구현 기술에 종속적이지 않다는 의미
 - 개발 후의 테스트 시에도 DB와 특정 서버 없이도 테스트를 할 수 있기 때문에 개발이 빨라진다는 장점이 있음

- **특정 기술에 종속적인 클래스의 예**

- **Servlet 클래스**

- 서블릿 컨테이너에서 실행 가능한 클래스를 만들기 위해서는 반드시 `javax.servlet.http.HttpServlet` 클래스를 구현해 함.
 - 기본 생성자가 있어야 한다.
 - 기능 메서드를 재 정의 해야한다.

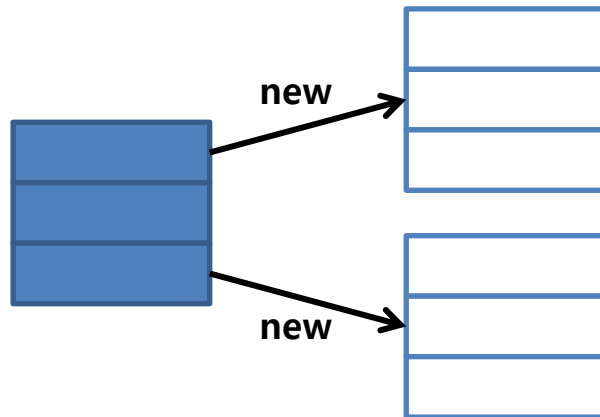
스프링 프레임워크의 주요 특징

■ 스프링 프레임워크의 주요 특징

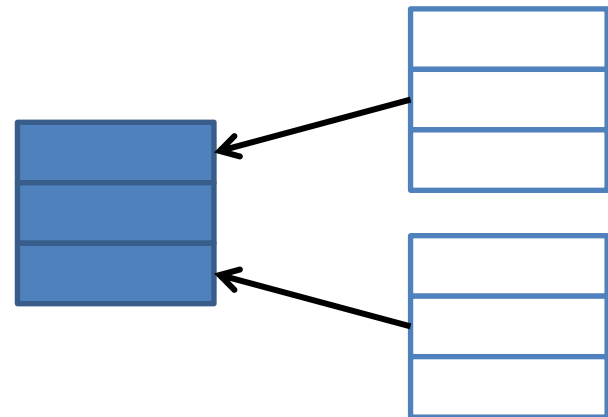
- Dependency Injection[DI] 지원

- Spring은 설정 파일(xml)이나 애노테이션을 통해서 객체간의 의존 관계를 설정 할 수 있도록 함
- 객체는 의존하고 있는 객체를 코드 상에서 직접 생성하거나 검색할 필요가 없음

[Not IoC]



[IoC]



■ 스프링 프레임워크의 주요 특징

- 제어의 역행

- 프로그램 개발에서 낮은 결합도(의존관계)와 높은 응집도를 고려해서 개발.
- 제어의 역행(Inversion of Control, IoC)을 통해 애플리케이션을 구성하는 객체간의 결합, 즉 낮은 결합도를 유지한다.
- IOC가 적용되면 객체 생성을 자바 코드에서 직접 처리하는 것이 아니라 스프링 컨테이너가 대신 처리, 객체와 객체 사이의 의존관계 역시 컨테이너가 처리한다.
- 따라서 자바 코드에서 의존관계가 명시적으로 처리되지 않기 때문에 결합도가 떨어져서 유지보수가 편리해진다.

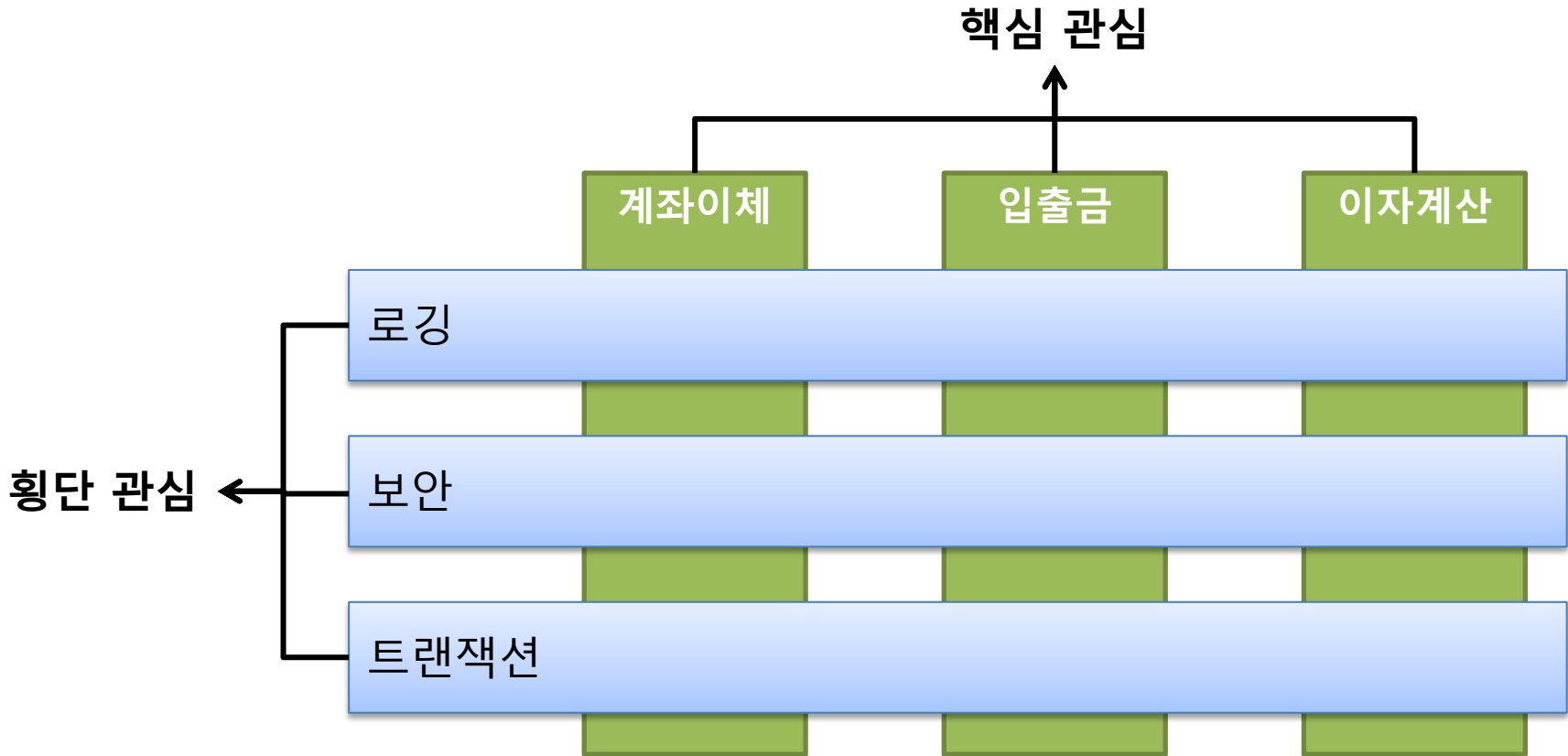
■ 스프링 프레임워크의 주요 특징

핵심 처리

- **관점지향적인 프로그래밍(Aspect Oriented Programming, AOP)**
 - 비즈니스 메서드를 개발할 때, 핵심 비즈니스 로직과 각 비즈니스 메서드마다 반복해서 처리되는 공통 로직을 분리해서 분리함으로써 응집도가 높게 개발이 되도록 지원
 - 관점지향적인 프로그래밍은 **공통 기능들을 외부 독립된 클래스로 분리, 해당 기능을 코드 형태로 직접 명시하지 않고 선언적(설정)으로 처리하여 적용하는 것이 기본 개념**
 - 응집도를 높이고, 유지보수를 혁신적으로 향상 시킬 수 있다. *x m l*
 - AOP 는 공통의 관심 사항을 적용해서 발생하는 **의존 관계의 복잡성과 코드의 중복을 해소시켜주는 프로그래밍 기법**이다.
 - **AOP 는 각 클래스에서 공통 관심사항을 구현한 모듈에 대한 의존 관계를 각기 보다는 , Aspect를 이용하여 핵심 로직을 구현한 각 클래스에 공통 기능을 적용한다.**
 - AOP에서 중요한 점은 Aspect가 핵심 로직 구현 클래스에 의존하지 않는다는 것.
 - 일반적으로 AOP에서는 설정 파일이나 설정 클래스 등을 이용해서 Aspect를 개발하게 되면, Aspect를 수정하지 않고 여러 클래스에 적용할 수 있다.

■ 스프링 프레임워크의 주요 특징

- 관점지향적인 프로그래밍 (Aspect Oriented Programming, AOP)



■ 스프링 프레임워크의 주요 특징

- 트랜잭션 처리를 위한 일관된 방법 제공
 - JDBC API 및 JTA를 사용하거나 컨테이너가 제공하는 트랜잭션을 사용, 설정 파일을 통해 트랜잭션 관련 정보를 관리하기 때문에 특정 트랜잭션 구현 방법에 상관없이 동일한 코드를 여러 환경에서 사용 가능
 - 선언적인 트랜잭션을 지원하여 코드를 수정하지 않고도 트랜잭션을 적용 및 변경 가능

■ 스프링 프레임워크의 주요 특징

- 컨테이너

- 컨테이너는 특정 객체의 생성과 관리를 담당하고 운용에 필요한 다양한 기능을 제공
- 컨테이너는 일반적으로 서버 안에 포함되어 배포 및 구동된다.
- 대표 컨테이너 'Servlet 컨테이너', 'EJB 컨테이너'가 있다.
- 'Servlet 컨테이너' - TOMCAT
- 스프링도 컨테이너.

■ 스프링 프레임워크의 주요 특징

- **Enterprise Application에서 필요로 하는 기능 제공**
 - Spring은 단순한 툴과 기본적인 개발 환경만으로도 Enterprise 개발에서 필요로 하는 주요 기능을 갖춘 애플리케이션 개발에 적합
 - 예 : 고비용을 요하는 WAS(Web Application Server)를 사용하지 않고도, WAS에서 지원해주는 트랜잭션 관리 및 보안, 객체 pooling과 같은 Enterprise 개발의 고급 기술들도 Spring Framework를 통해서 단순한 작업 및 설정만으로 동일한 수준의 기능들을 사용할 수 있게 됨

스프링 프레임워크의 장점

■ 스프링 프레임워크의 장점

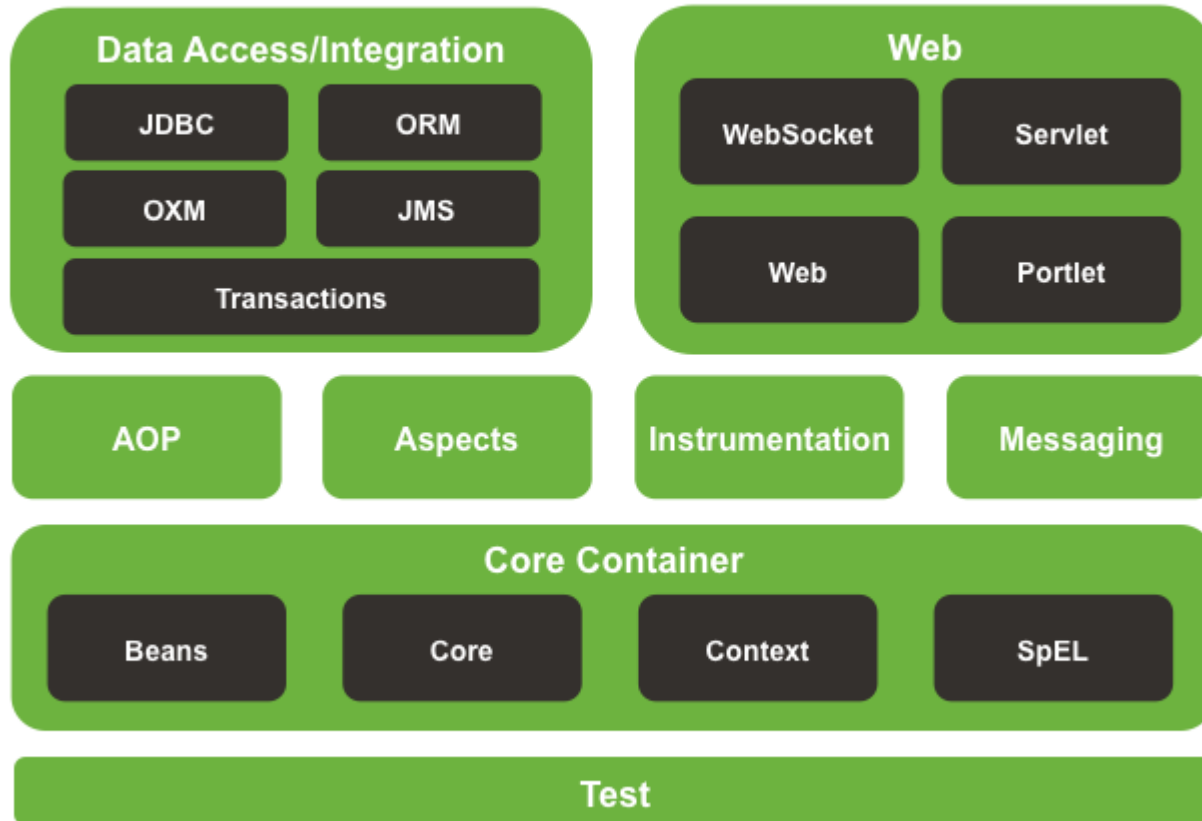
- **개발자들이 개발하고자 하는 애플리케이션 로직 개발에만 집중할 수 있음**
 - 기술에 대한 접근 방식이 일관성이 없거나, 특정 환경에 종속적이지 않음
 - 따라서 실행 로직의 기능이 변경되는 것이 아니라 서버 등의 실행 환경이 바뀌고 적용되는 조건이 바뀐다 해도 코드까지 수정할 필요가 없음
- **개발이 단순해짐**
 - Spring의 의존 관계, 트랜잭션등의 설정 방법에 대한 지식을 습득한 후에는 설정 적용 기술만으로도 Enterprise 개발의 기술적인 복잡함과 그에 따른 수고를 제거 가능
- **POJO 방식의 기술 사용이 가능**
 - 특정 규약 및 환경에 종속되지 않은 일반 자바 클래스를 지원하므로 컨테이너에 의존적인 코드를 추가하지 않아도 애플리케이션을 개발 할 수 있음
 - 개발후의 테스트도 쉽고 빠르게 할 수 있음

■ 스프링 프레임워크의 모듈

- Spring4.0은 20여개의 모듈로 구성



Spring Framework Runtime



■ 스프링 프레임워크의 모듈

모듈 명	설 명
Beans	BeanFactory 인터페이스를 통해 구현됨
Core	프레임워크의 가장 기본적인 부분 컨테이너 기능을 수행하기 위해 의존성 주입[DI] 기능을 제공
Context	spring-core, spring-beans 모듈을 확장해서 국제화, 이벤트 처리, 리소스 로딩, 서블릿 컨테이너를 위한 컨텍스트 생성 등의 기능을 추가로 제공. ApplicationContext 인터페이스를 통해 구현됨
Expression Language	객체에 접근하고 객체를 조작하기 위한 표현 언어를 제공
AOP	AOP Alliance에 호환되는 AOP 구현을 제공
Aspects	AspectJ와의 통합을 제공
Web(MVC/Remoting)	Spring MVC를 제공하며 struts와도 연동 기능 제공등 웹 관련 기능 지원
Data Access/Integration	JDBC를 위한 템플릿 제공. 따라서 간결한 코드로 JDBC 프로그램 가능, iBatis 및 하이버네이트 등의 ORM api를 위한 통합 레이어 제공. Spring이 제공하는 트랜잭션과의 연동 지원

■ 스프링 프레임워크의 설치와 모듈구성

- **MAVEN** 이용하기
 - 의존 관계의 library들도 자동 다운로드

<http://repo.springsource.org/release>

<https://mvnrepository.com/>

■ STS 다운로드

- <http://spring.io/tools>

- Spring Tool Suite는 Spring 응용 프로그램 개발을 위해 사용자 정의 된 Eclipse 기반 개발 환경입니다.

Pivotal tc Server, Pivotal Cloud Foundry, Git, **Maven**, AspectJ 등의 통합을 포함하여 Spring 애플리케이션을 구현, 디버깅, 실행 및 배포 할 수 있는 환경을 제공합니다.

■ 스프링 프레임워크의 설치와 모듈구성

- Spring Project 를 위한 구성

- Spring Framework library

관계 등록
의존

- 설정 메타정보 파일 설정 정보

pom.xml

- Spring은 설정 메타데이터 정보를 필요로 함

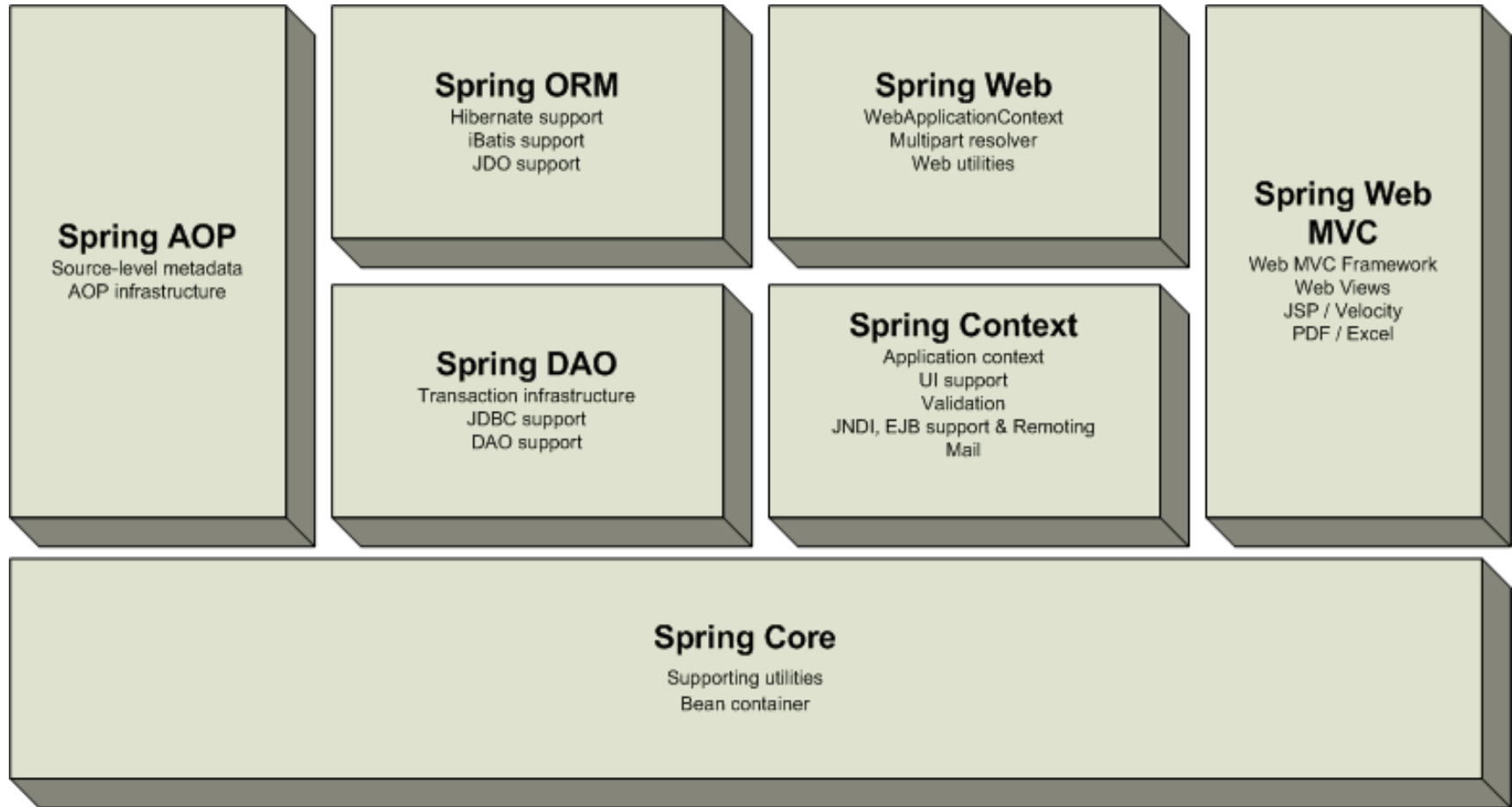
- 이 설정 메타데이터는 Spring 컨테이너에 객체 생성 및 관계 설정 내용을 XML 또는 properties[프로퍼티] 파일, 소스코드 애노테이션과 같은 외부 리소스로 작성

- 자바 소스들

- Spring 빈 class 객체들
 - Spring 빈 사용 클래스들
 - 이외의 자바 클래스들

+ View Disp

■ What's Spring?

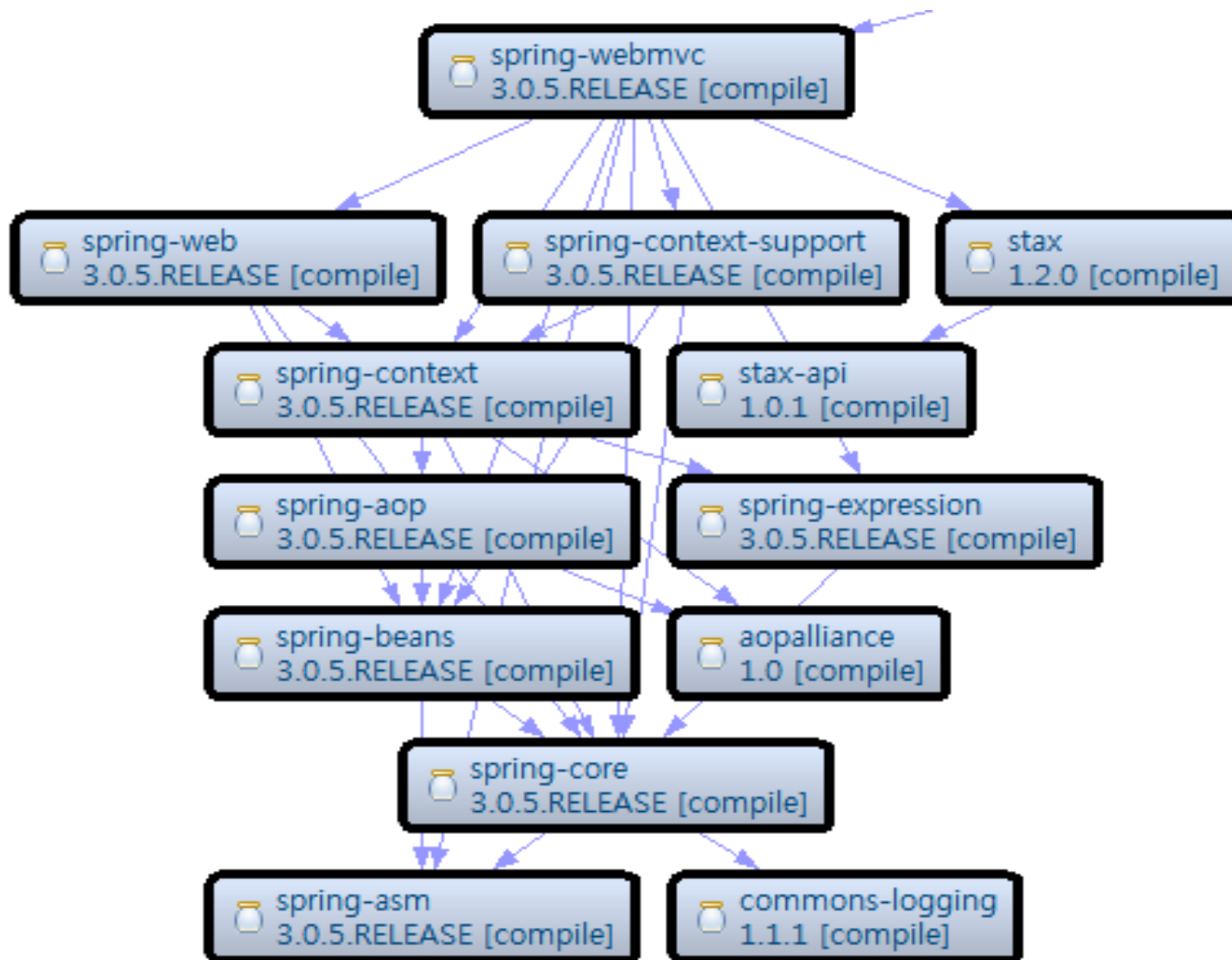


■ What's Spring?

- **Spring AOP** (Aspect Oriented Programming)
- **Spring Web**
 - Web Application 개발에 필요한 Web Application Context와 Multipart Request등의 기능 지원
 - Struts, Webwork와 같은 프레임워크의 통합 지원
- **Spring Web MVC**
 - Spring 프레임워크에서 독립적으로 MVC 기능 지원
 - Struts, Webwork 기능들을 Spring Web MVC로 대체 가능
 - Excel, PDF등의 다양한 UI 를 위한 API 제공

■ What's Spring?

필수 Lib : /WEB-INF/lib 폴더에 jar 파일 추가



- Spring을 이용한 웹 어플리케이션 작성

1. Spring 기본 환경 설정

2. 컨트롤러 작성

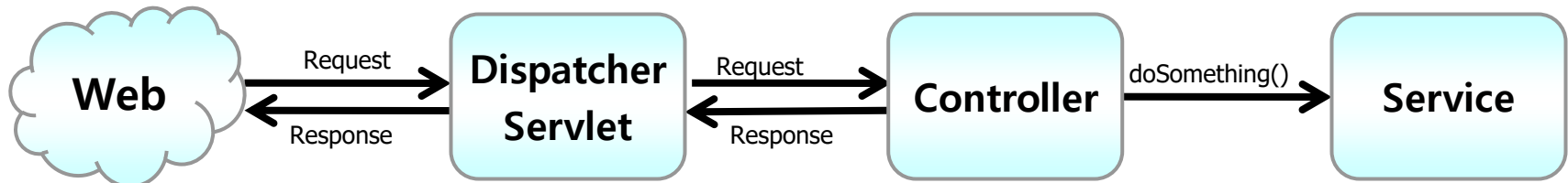
3. 컨텍스트 설정 파일에 컨트롤러 설정

([ServletName]-servlet.xml)

4. 컨트롤러와 JSP의 연결 위해 View Resolver 설정

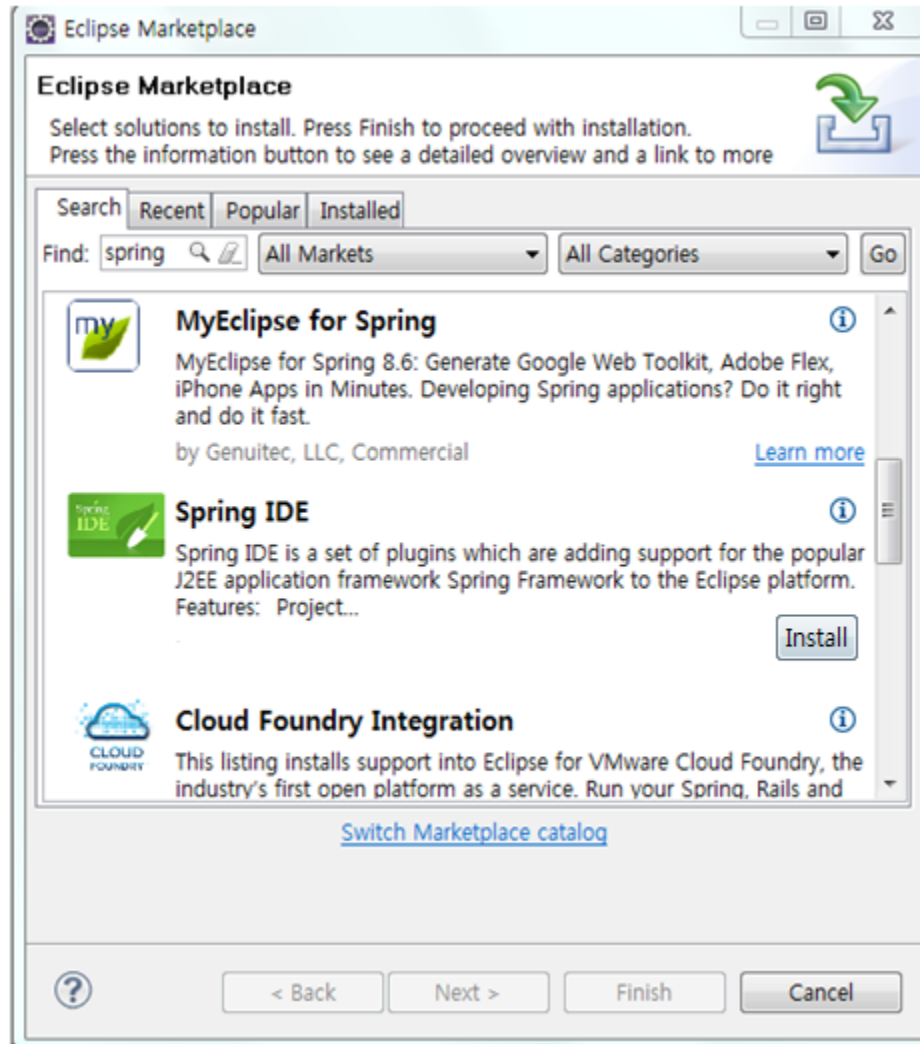
5. JSP 코드 작성

6. 실행



■ 스프링 프레임워크의 설치와 모듈구성

- Spring IDE 설치



■ 스프링 프레임워크의 설치와 모듈구성

- Spring IDE 설치

