

2장 스프링 프레임워크

강사 김영석

A top-down view of a wooden desk. On the desk, there is a silver laptop with a black keyboard, a pair of black-rimmed glasses, a white coffee cup with a yellow handle, and a small green succulent in a pot. The word "CONTENT" is written in large, white, sans-serif capital letters across the middle of the image.

CONTENT

1

프레임워크란?

2

STS 구조와 구성 요소

1. 프레임워크란?

✓ 프레임워크(Framework) 정의

- 사전적 의미는 ‘어떤 것을 구성하는 구조 또는 뼈대’라는 의미가 있다.
- 소프트웨어적 의미로는 ‘기능을 미리 클래스나 인터페이스 등으로 만들어 제공하는 반제품’으로 기본적인 구조가 만들어져 있고 개발자는 그 위에 살을 붙여 사용하면 된다.

✓ 프레임워크(Framework) 장점

- 일정한 기준에 따라 개발이 이루어지므로 개발 생산성과 품질이 보장된 애플리케이션을 개발할 수 있다.
- 개발 후 유지 보수 및 기능의 확장성에서도 고품질이 보장된다.



✓ 스프링 프레임워크

- 스프링 프레임워크는 자바 웹 애플리케이션 개발을 위한 오픈 소스 프레임워크다.
- EJB(Enterprise Java Bean) 보다 가벼운 경량 프레임워크다.
- EJB는 기능성면에서는 좋지만 구조가 복잡하고 물리적인 WAS 서버가 필요하다. 비용과 프레임자체의 복잡성 때문에 POJO 방식의 프레임워크가 부각 되었으며 그 방식의 프레임워크에 포함된 것이 스프링 프레임워크이다.



✓ 컨테이너

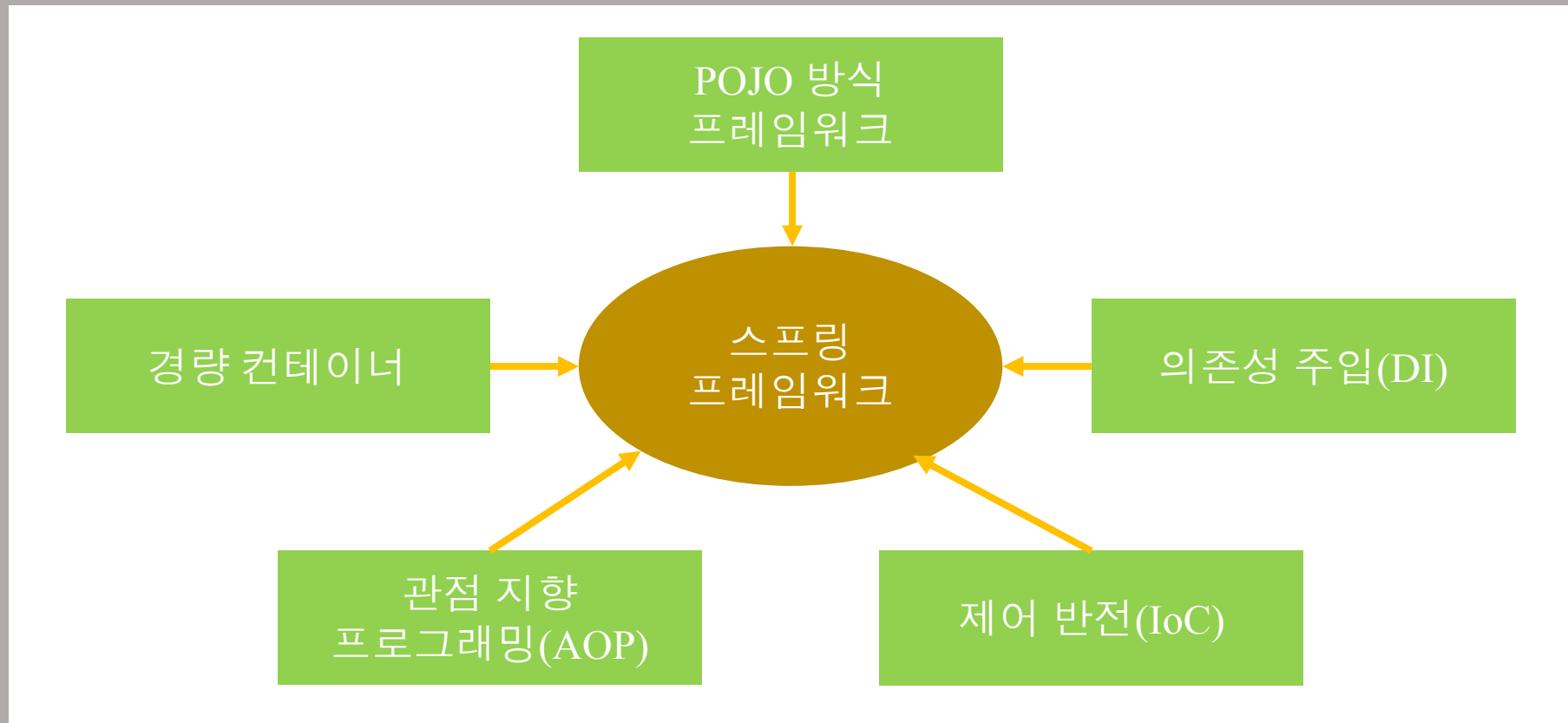
- 톰캣은 서블릿 컨테이너라고 부르는데 톰캣은 서블릿의 생성, 초기화, 서비스 실행, 소멸에 관한 모든 권한을 가지고 서블릿을 관리한다.
- 스프링에서의 컨테이너는 애플리케이션에서 사용되는 여러 가지 빈을 개발자가 아닌 스프링이 권한을 가지고 직접 관리한다는 점에서 스프링도 일종의 컨테이너라고 할 수 있다.



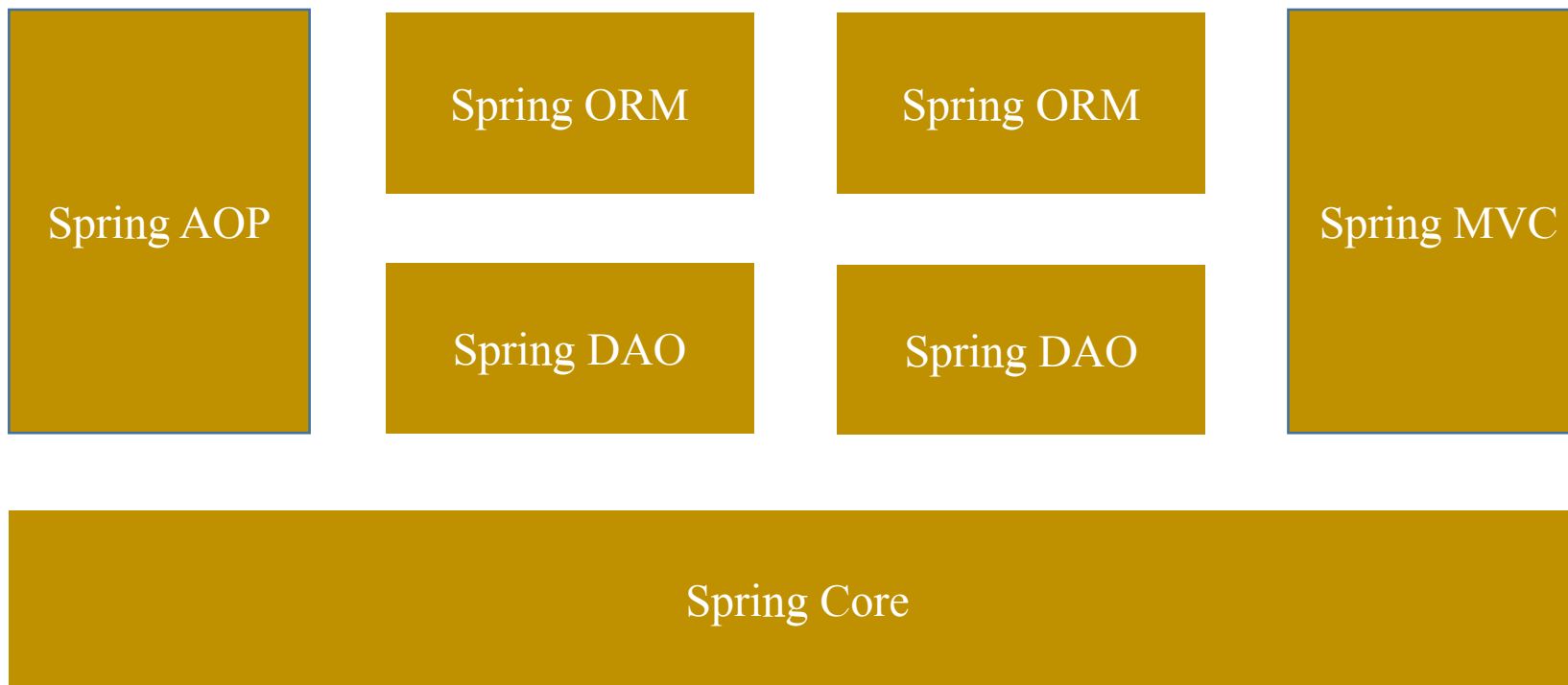
✓ 스프링의 특징


- EJB보다 가볍고 배우기도 쉬우며 경량 컨테이너의 기능을 수행한다.
- 제어 역행(IoC, Inversion of Control) 기술을 이용해 애플리케이션 간의 느슨한 결합을 제어한다.
- 의존성 주입(DI, Dependency Injection) 기능을 지원한다.
- 관점 지향(AOP, Aspect-Oriented Programming) 기능을 이용해 자원을 관리한다.
- 영속성과 관련된 다양한 서비스를 지원한다.
- 수많은 라이브러리와의 연동 기능을 지원한다.

✓ 스프링 프레임워크의 특징



✓ 스프링 프레임워크의 주요 기능

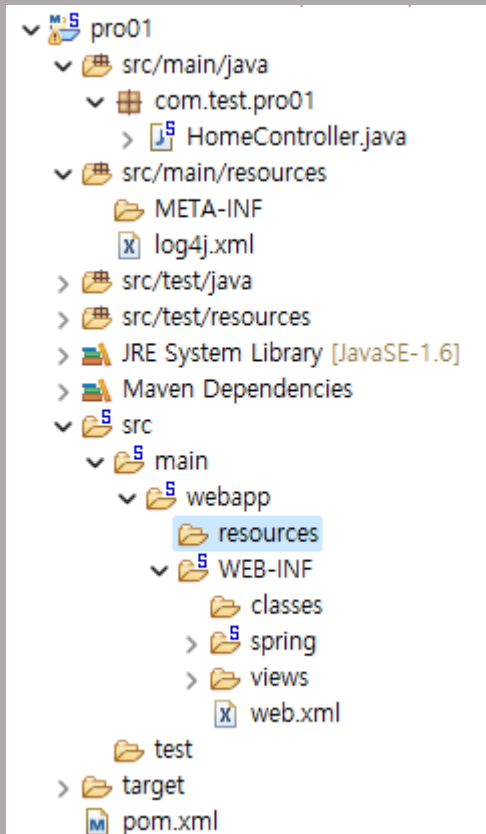




스프링 기능	설명
Core	다른 기능과 설정을 분리하기 위한 IoC 기능을 제공한다.
Context	스프링의 기본 기능으로 애플리케이션의 각 기능을 하는 빈에 대한 접근 방법을 제공한다.
DAO	JDBC 기능을 좀더 편리하게 사용할 수 있도록 한다.
ORM	하이버네이트나 마이바티스 같은 영속성 관련 프레임워크와 연동된 기능을 제공한다.
AOP	관점 지향 기능을 제공한다.
Web	웹 애플리케이션 개발에 필요한 기능을 제공한다.
WebMVC	스프링에서 MVC 구현에 관련된 기능을 제공한다.

2. STS 구조 및 구성 요소

✓ STS의 기본 디렉터리 구조



구성 요소	설명
pom.xml	프로젝트 정보가 표시되며 스프링에서 사용되는 여러 가지 라이브러리를 설정해 다운로드 할 수 있다.
src/main/java	자바 소스 파일이 위치한다.
src/main/resources	프로퍼티 파일이나 XML 파일 등 리소스 파일이 위치한다.
src/main/webapp/WEB-INF	WEB-INF 등 웹 애플리케이션 리소스가 위치한다.
src/test/java	JUnit 등 테스트 파일이 위치한다.
src/test/resources	테스트 할 때 필요한 resource 파일이 위치한다.
src/main/webapp/resources	이미지나 JavaScript, CSS 파일 등이 위치한다.



```
pro01/pom.xml  ✖
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/maven-v4_0_0.xsd">
4      <modelVersion>4.0.0</modelVersion>
5      <groupId>com.test</groupId>
6      <artifactId>pro01</artifactId>
7      <name>pro01</name>
8      <packaging>war</packaging>
9      <version>1.0.0-BUILD-SNAPSHOT</version>
10     <properties>
11         <java-version>1.6</java-version>
12         <org.springframework-version>3.1.1.RELEASE</org.springframework-version>
13         <org.aspectj-version>1.6.10</org.aspectj-version>
14         <org.slf4j-version>1.6.6</org.slf4j-version>
15     </properties>
16     <dependencies>
17         <!-- Spring -->
18         <dependency>
19             <groupId>org.springframework</groupId>
20             <artifactId>spring-context</artifactId>
21             <version>${org.springframework-version}</version>
22             <exclusions>
23                 <!-- Exclude Commons Logging in favor of SLF4j -->
24                 <exclusion>
25                     <groupId>commons-logging</groupId>
26                     <artifactId>commons-logging</artifactId>
27                 </exclusion>
28             </exclusions>
29         </dependency>
```

✓ pom.xml의 프로젝트 정보 설정 태그 구성 요소

속성	설명
groupId	프로젝트 그룹 id를 나타내며 일반적으로 도메인 이름을 사용해 설정한다.
artifactId	프로젝트 아티팩트 id를 설정한다. 대개는 패키지 이름으로 설정한다.
version	프로젝트의 버전을 설정한다.
packaging	애플리케이션 배포 할 때 패키징 타입을 설정한다. 이 경우는 war 파일로 패키징한다.

✓ pom.xml의 dependencies 정보 설정 태그 구성 요소

속성	설명
dependency	해당 프로젝트에서 의존하는 다른 라이브러리 정보를 기술한다.
groupId	의존하는 프로젝트의 그룹 ID를 의미한다.
artifactId	의존하는 프로젝트의 아티팩트 ID를 의미한다.
version	의존하는 프로젝트 버전 정보다.

✓ 다운로드 된 스프링 관련 라이브러리

```
> JRE System Library [JavaSE-1.6]
✓ Maven Dependencies
  > spring-context-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-context\3.1.1.RELEASE
  > spring-aop-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-aop\3.1.1.RELEASE
  > aopalliance-1.0.jar - C:\Users\kys05\m2\repository\org\springframework\spring-aop\3.1.1.RELEASE
  > spring-beans-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-beans\3.1.1.RELEASE
  > spring-core-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-core\3.1.1.RELEASE
  > spring-expression-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-expression\3.1.1.RELEASE
  > spring-asm-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-asm\3.1.1.RELEASE
  > spring-webmvc-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-webmvc\3.1.1.RELEASE
  > spring-context-support-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-context-support\3.1.1.RELEASE
  > spring-web-3.1.1.RELEASE.jar - C:\Users\kys05\m2\repository\org\springframework\spring-web\3.1.1.RELEASE
  > aspectjrt-1.6.10.jar - C:\Users\kys05\m2\repository\org\aspectj\aspectjrt\1.6.10
  > slf4j-api-1.6.6.jar - C:\Users\kys05\m2\repository\org\slf4j\slf4j-api\1.6.6
  > jcl-over-slf4j-1.6.6.jar - C:\Users\kys05\m2\repository\org\slf4j\jcl-over-slf4j\1.6.6
  > slf4j-log4j12-1.6.6.jar - C:\Users\kys05\m2\repository\org\slf4j\slf4j-log4j12\1.6.6
  > log4j-1.2.15.jar - C:\Users\kys05\m2\repository\log4j\log4j\1.2.15
  > javax.inject-1.jar - C:\Users\kys05\m2\repository\javax\inject\javax.inject\1
  > servlet-api-2.5.jar - C:\Users\kys05\m2\repository\javax\servlet\servlet-api\2.5
  > jsp-api-2.1.jar - C:\Users\kys05\m2\repository\javax\servlet\jsp\jsp-api\2.1
  > jstl-1.2.jar - C:\Users\kys05\m2\repository\javax\servlet\jstl\1.2
  > junit-4.7.jar - C:\Users\kys05\m2\repository\junit\junit\4.7
```


A photograph of a server room with rows of server racks on both sides of a central aisle. The racks have glass doors and are filled with server units, some of which have glowing blue indicator lights. The ceiling has several long, rectangular light fixtures. The overall atmosphere is dimly lit, with the primary light sources being the server lights and the ceiling fixtures.

수고하셨습니다.