



통합 구현(Spring, Django)

Spring Framework의 개념 파악하기



한국기술교육대학교
온라인평생교육원

학습내용

- Spring Framework
- Spring의 핵심기술
- 프로젝트 구조 파악

학습목표

- Framework란 무엇인지 파악할 수 있다.
- Spring만의 핵심 기술에 대해 설명할 수 있다.
- Spring 기반 프로젝트의 구조에 대해 파악할 수 있다.

Spring Framework

1 Framework란

Framework

Framework

소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 형태로 클래스들을 제공하는 것 - 랄프 존슨(Ralph Johnson)

소프트웨어 제작을 편리하게 할 수 있도록 미리 뼈대를 이루는 클래스와 인터페이스를 제작하여 제공

프레임워크는 애플리케이션의 틀과 구조를 결정할 뿐 아니라, 그 위에 개발된 개발자의 코드를 제어

2 Spring의 정의

Spring

- 자바 프로그램 개발을 위한 애플리케이션 프레임워크
- 애플리케이션 개발의 전 과정을 빠르고 편리하며 효율적으로 진행하게 해줌
- 애플리케이션 전 영역을 관통하는 일관된 프로그래밍 원칙과 핵심 기술 제공

Spring의 핵심 기술

1 POJO

1 POJO(Plain Old Java Objects)

Martin Fowler가 2000년도에 컨퍼런스에서 발표한 용어

발표 내용의 핵심 : 기존의 EJB는 너무 복잡하고 제한이 많은 기술임을 설명

POJO는 단순히 군더더기 없는 간단한 Java Object 사용을 지향함

2 POJO의 조건

1 특정 규약(Contract)에 종속되지 않음

- 개발자는 자유로운 객체지향 설계에 방해를 받으면 안됨

2 특정 환경이 종속되지 않음

- 특정 서버에서만 사용 가능한 API를 사용하는 빈
- 의존적인 빈

3 재사용성이 높아야 함

- 책임과 역할이 다른 코드를 모두 가지고 있는 덩치 큰 빈은 지양

Spring의 핵심 기술

1 POJO

3 POJO의 장점

깔끔한 코드
생산

자동화
테스트에
유리

객체지향
설계에
자유로움

4 POJO 프레임워크

Spring은 대표적인 POJO 프레임워크

POJO를 이용한 프로그래밍이 가능하도록 기술적인 기반을 제공

개발자들이 객체지향적인 설계와 원리에 집중할 수 있도록 도와줌

Spring의 핵심 기술

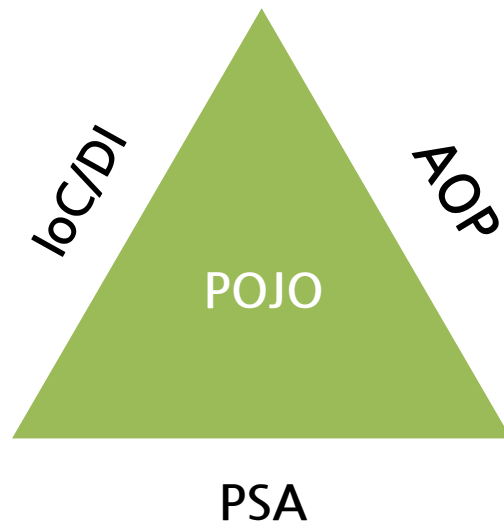
2 PSA (Portable Service Abstraction)

PSA

- 이식 가능한 서비스 추상화
- 코드들을 비슷한 것들끼리 모으고(강한 응집력) 서로 다른 코드들이 서로를 구속하지 않는(낮은 결합도) 프로그램 개발
- 환경과 세부 기술의 변화에 관계없이 일관된 방식으로 기술에 접근하게 해주는 기술

PSA를 실현하기 위해서는?

4장의 IoC/DI를 적극 활용해서 개발한다면 서비스 추상화는 자연스럽게 만들어 쓸 수 있음



프로젝트 구조 파악

1 기본 설정

1 파일 살펴보기

파일명	설명
/src/main/java	.java 파일이 위치
/src/main/webapp	.jsp 파일이 위치
/src/main/resources	각종 설정파일(.xml, .properties..)이 위치
/target	실행파일(.class 파일 및 빌드 패키지) 위치
pom.xml	Maven 설정 파일
External Libraries	외부 라이브러리 파일

2 파일 살펴보기(pom.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <name>Spring</name>
  <groupId>com.spring</groupId>
  <artifactId>framework</artifactId>
  <version>001</version>
  <packaging>war</packaging>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-
8</project.reporting.outputEncoding>
```

프로젝트 구조 파악

1 기본 설정

2 파일 살펴보기(pom.xml)

```
<spring.version>4.2.0.RELEASE</spring.version>  
</properties>
```

```
<dependencies>  
  <!-- Servlet -->  
  <dependency>  
    <groupId>javax.servlet</groupId>  
    <artifactId>javax.servlet-api</artifactId>  
    <version>3.0.1</version>  
    <scope>provided</scope>  
  </dependency>
```

```
<dependency>  
  <groupId>javax.servlet.jsp</groupId>  
  <artifactId>jsp-api</artifactId>  
  <version>2.2</version>  
  <scope>provided</scope>  
</dependency>  
<dependency>  
  <groupId>javax.servlet</groupId>  
  <artifactId>jstl</artifactId>  
  <version>1.2</version>  
</dependency>
```


프로젝트 구조 파악

1 기본 설정

2 파일 살펴보기(pom.xml)

```
<!-- Spring -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>${spring.version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${spring.version}</version>
</dependency>
</dependencies>
</project>
```

프로젝트 구조 파악

1 기본 설정

3 파일 살펴보기 (/src/main/webapp/WEB-INF/dispatcher-servlet.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

</beans>
```

4 파일 살펴보기 (/src/main/resources/services.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <mvc:annotation-driven />
    <context:component-scan base-package="com.spring" />
```

프로젝트 구조 파악

1 기본 설정

4 파일 살펴보기(/src/main/resources/services.xml)

```

<bean id="contentNegotiatingViewResolver"
class="org.springframework.web.servlet.view.ContentNegotiatingViewResolv
er">
    <property name="viewResolvers">
        <list>
            <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
>
                <property name="prefix" value="/WEB-INF/views/" />
                <property name="suffix" value=".jsp" />
            </bean>
        </list>
    </property>
</bean>
</beans>

```

5 파일 살펴보기(/src/main/webapp/WEB-INF/web.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
    <display-name>Spring Framework</display-name>
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>
            classpath:services.xml
        </param-value>
    </context-param>

```

프로젝트 구조 파악

1 기본 설정

《5》 파일 살펴보기(/src/main/webapp/WEB-INF/web.xml)

```
<listener>
  <listener-class>
    org.springframework.web.context.ContextLoaderListener
  </listener-class>
</listener>
<!--dispatcher-servlet.xml-->
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>
```

프로젝트 구조 파악

1 기본 설정

Dispatcher Servlet

- 1 Client의 요청(URL)
- 2 DispatcherServlet이 요청을 파악하고 요청 URL과 매칭되는 Controller 검색
- 3 해당 Controller에 처리 요청
- 4 Controller가 요청을 처리 후 ModelAndView 리턴
- 5 DispatcherServlet이 ViewResolver를 통해 Controller의 실행 결과를 보여줄 View 검색
- 6 해당 View를 응답으로 출력

1. Spring Framework

- Framework

- ‘GoF의 디자인 패턴’으로 유명한 랄프 존슨(Ralph Johnson) 교수는 프레임워크를 "소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 형태로 클래스들을 제공하는 것"이라고 정의
- 소프트웨어 제작을 편리하게 할 수 있도록 미리 뼈대를 이루는 클래스와 인터페이스를 제작하여 제공
- 프레임워크는 애플리케이션의 틀과 구조를 결정할 뿐 아니라, 그 위에 개발된 개발자의 코드를 제어

- Spring의 정의

- 자바 프로그램 개발을 위한 애플리케이션 프레임워크
- 애플리케이션 개발의 전 과정을 빠르고 편리하며 효율적으로 진행하게 해줌
- 애플리케이션 전 영역을 관통하는 일관된 프로그래밍 원칙과 핵심 기술 제공

2. Spring의 핵심기술

- POJO(Plain Old Java Objects)
 - Martin Fowler가 2000년도에 컨퍼런스 발표를 위해 만든 용어
 - 발표 내용의 핵심 : 기존의 EJB는 너무 복잡하고 제한이 많은 기술임을 설명
 - POJO는 단순히 군더더기 없는 간단한 Java Object를 사용을 지향함
- PSA(Portable Service Abstraction)
 - 이식 가능한 서비스 추상화
 - 코드들을 비슷한 것들끼리 모으고(강한 응집력) + 서로 다른 코드들이 서로를 구속하지 않는(낮은 결합도) 프로그램 개발
 - 환경과 세부 기술의 변화에 관계없이 일관된 방식으로 기술에 접근하게 해주는 기술

3. 프로젝트 구조 파악

- 기본 설정

- /src/main/java : .java 파일이 위치
- /src/main/webapp : .jsp 파일이 위치
- /src/main/resources : 각종 설정파일(.xml, .properties..)이 위치
- /target : 실행파일(.class 파일 및 빌드 패키지) 위치
- pom.xml : Maven 설정 파일
- External Libraries : 외부 라이브러리 파일

- Dispatcher Servlet

- Client의 요청(URL)
- DispatcherServlet이 요청을 파악하고 요청 URL과 매칭되는 Controller 검색
- 해당 Controller에 처리 요청
- Controller가 요청을 처리 후 ModelAndView 리턴
- DispatcherServlet이 ViewResolver를 통해 Controller의 실행 결과를 보여줄 View 검색
- 해당 View를 응답으로 출력