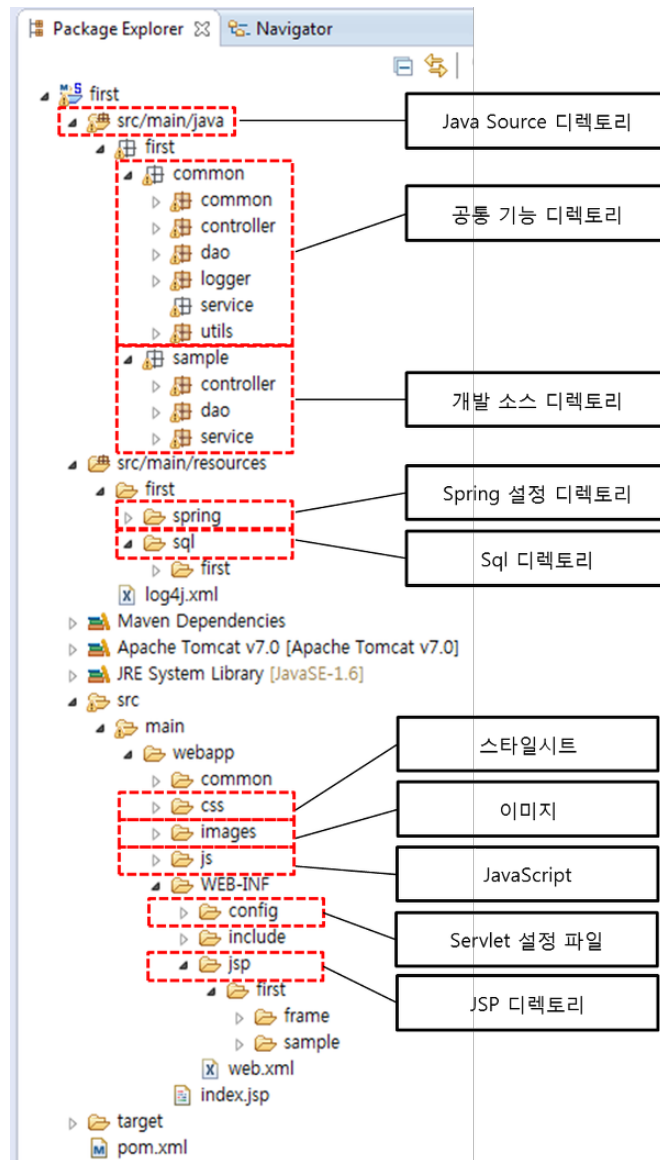


스프링(Spring) 개발

(6) 프로젝트 생성 및 개발 시작

1. 프로젝트의 구조



이 first 프로젝트가 우리가 최종적으로 완성하려고 하는 프로젝트의 구조이다. (여기에 기능들이 추가 될 수 있다.) 맨 처음 만들었던 first 프로젝트와 비교하면 그 구조가 상당히 복잡해지고, 여러가지 설정등이 추가된 것을 알 수 있다.

2. index.jsp 추가

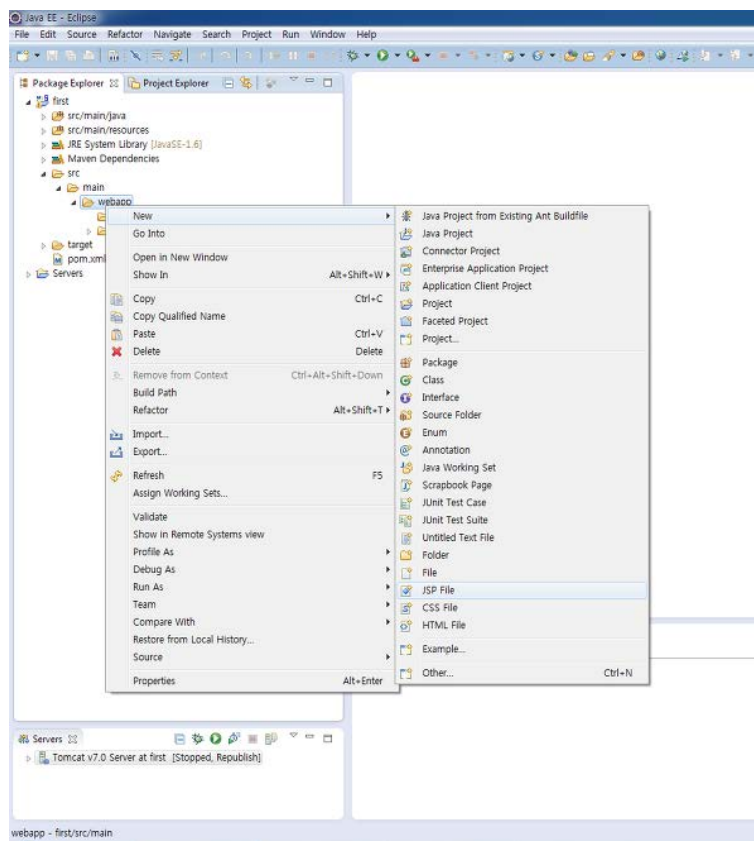
기존에는 프로그램을 실행하면 기본적으로 views 폴더에 있는 home.jsp가 실행되었다.

그렇지만 우리가 어떠한 웹사이트를 들어가면 보통 index.jsp (또는 php, html 등 종류는 여러가지가 있을 수 있다.)가 실행된다.

home.jsp가 실행되는것은 우리가 제어하기도 힘들고, 앞으로 개발할 구조에도 맞지않기 때문에, 웹사이트가 최초 보여주는 index.jsp를 추가하려고 한다.

src/main/webapp 디렉토리 밑에 index.jsp를 만들자.

1. webapp 우클릭 > new > JSP File을 선택



New에서 보여지는 목록이 위 사진과 다를 수 있다. 이러한 경우 이클립스 설정을 바꿔준다.

<http://addio3305.tistory.com/40> 를 참고하여 구성을 바꾸면 된다.

2. index.jsp 라는 이름으로 jsp 파일 생성.

index.jsp가 호출되었다는 것을 보기 위해, index.jsp의 body 안에 "This is index.jsp!"를 추가한다.

```
1 <body>
2   <h1>This is index.jsp.</h1>
3 </body>
```

3. web.xml 설정 변경

이제 서블릿 설정을 통해서, 이 프로젝트가 시작되면 index.jsp를 호출하도록 바꾸겠다.

web.xml에 다음의 내용을 추가한다.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" >
5
6   <welcome-file-list>
7     <welcome-file>index.jsp</welcome-file>
8   </welcome-file-list>
```

이 위치는 web.xml의 <web-app></web-app> 태그 사이에 존재하면 된다. 즉, 이 태그는 web.xml의 아무곳에서나 선언하면된다. 여기서는 web.xml 맨 위에 위치하도록 하였다.

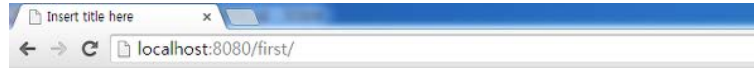
4. 실행시켜 보자.

서버를 실행시키고, 브라우저의 주소창에 **http://localhost:포트번호/first/** 를 입력한다.

다음과 같이 나오면 정상적으로 설정이 완료된 것이다.

[illegible]

- 이클립스의 콘솔창에서 아무런 에러 없이, 서버가 실행되는 모습-



This is index.jsp.

- index.jsp가 정상적으로 호출된 모습 -

모두 정상적으로 동작하는 것을 확인하자.

3. 서블릿 설정 변경

서블릿(Servlet)이란 간단히 말해서 자바에서 동적 웹 프로젝트를 개발할 때, 사용자의 요청과 응답을 처리해 주는 역할을 한다.

보통 스프링에서는 servlet 설정이 .do로 되어 있는데, 현재 기본 프로젝트에서는 .do로 되어있지 않다. 따라서, 서블릿 설정을 간단히 바꿔주려고 한다.

web.xml을 열어보면 다음과 같은 부분이 있다.

```
1 <servlet>
2   <servlet-name>appServlet</servlet-name>
3   <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
4   <init-param>
5     <param-name>contextConfigLocation</param-name>
6     <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
7   </init-param>
8   <load-on-startup>1</load-on-startup>
9 </servlet>
10
11<servlet-mapping>
12  <servlet-name>appServlet</servlet-name>
13  <url-pattern>/</url-pattern>
14</servlet-mapping>
```

이를 다음과 같이 바꾼다.

```
1   <servlet>
2     <servlet-name>action</servlet-name>
```

```

3      <servlet-class>
4          org.springframework.web.servlet.DispatcherServlet
5      </servlet-class>
6      <init-param>
7          <param-name>contextConfigLocation</param-name>
8          <param-value>
9              /WEB-INF/config/*-servlet.xml
10         </param-value>
11     </init-param>
12     <load-on-startup>1</load-on-startup>
13 </servlet>
14 <servlet-mapping>
15     <servlet-name>action</servlet-name>
16     <url-pattern>*.do</url-pattern>
17 </servlet-mapping>

```

여기서 볼 것은 **<url-pattern>*.do</url-pattern>** 부분이다.

이는 앞으로 서블릿에 어떠한 요청을 할 때, .do를 통해서만 요청을 전달하고, 다른 방식의 요청, 예를 들어 .html의 직접적인 호출등은 이제 허락되지 않는다.

그 다음 확인할 부분은 contextConfigLocation의 설정이다.

기존에는 dispatcher 의 설정인 contextConfigLocation이 /WEB-INF/spring/appServlet/servlet-context.xml에 존재하였는데, 이를 /WEB-INF/config/action-servlet.xml로 변경하고, 인터셉터 (Interceptor)도 추가하려는 목적이다.

마지막으로, <context-param>을 수정한다.

<context-param>에서 설정되어있던 root-context.xml은 모든 서블릿과 필터에서 사용되는 루트 스프링 컨테이너에 대한 설정이다(The definition of the Root Spring Container shared by all Servlets and Filters).

이 root-context.xml은 추후 다른곳에서 설정될 예정이므로 <param-value></param-value>안에 있던 /WEB-INF/spring/root-context.xml 부분은 다음과 같이 지운다.

```

1      <context-param>
2          <param-name>contextConfigLocation</param-name>
3          <param-value> </param-value>
4      </context-param>

```

여기까지 완료한 web.xml은 다음과 같다.

```

1 <?xml version="1.0" encoding="UTF-8"?>

```

```
2 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5 http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd" >
6
7   <welcome-file-list>
8     <welcome-file>index.jsp</welcome-file>
9   </welcome-file-list>
10
11   <!-- Creates the Spring Container shared by all Servlets and Filters -->
12   <listener>
13     <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
14class>
15   </listener>
16
17   <!-- Processes application requests -->
18   <servlet>
19     <servlet-name>action</servlet-name>
20     <servlet-class>
21       org.springframework.web.servlet.DispatcherServlet
22     </servlet-class>
23     <init-param>
24       <param-name>contextConfigLocation</param-name>
25       <param-value>
26         <!-- /WEB-INF/config/action-servlet.xml,
27         /WEB-INF/config/interceptor-servlet.xml -->
28       </param-value>
29     </init-param>
30     <load-on-startup>1</load-on-startup>
31   </servlet>
32   <servlet-mapping>
33     <servlet-name>action</servlet-name>
34     <url-pattern>*.do</url-pattern>
35   </servlet-mapping>
```

```

36
37 <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
38 <context-param>
39     <param-name>contextConfigLocation</param-name>
40     <param-value></param-value>
    </context-param>
</web-app>

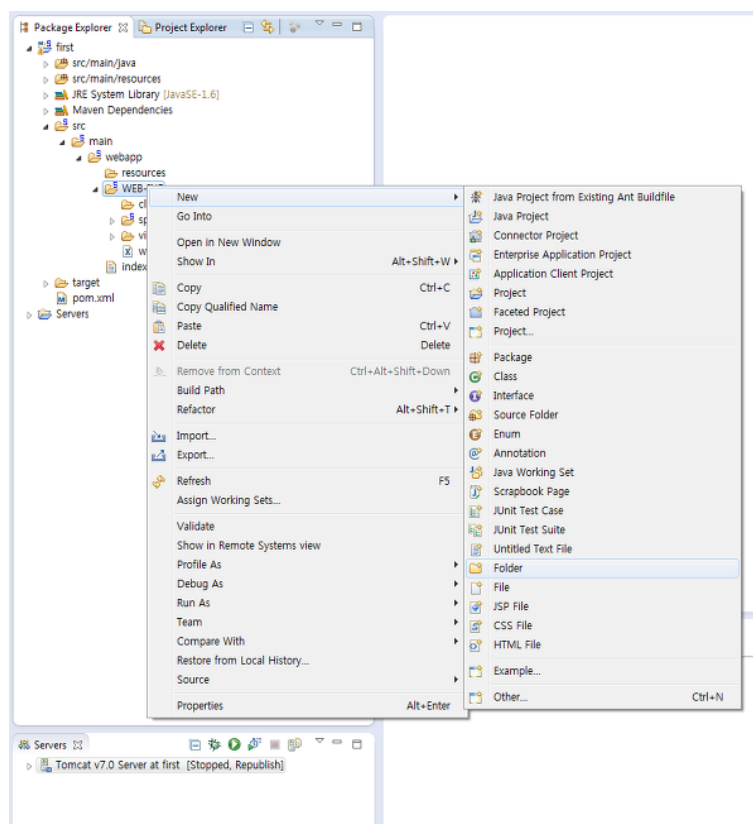
```

4. contextConfigLocation 변경

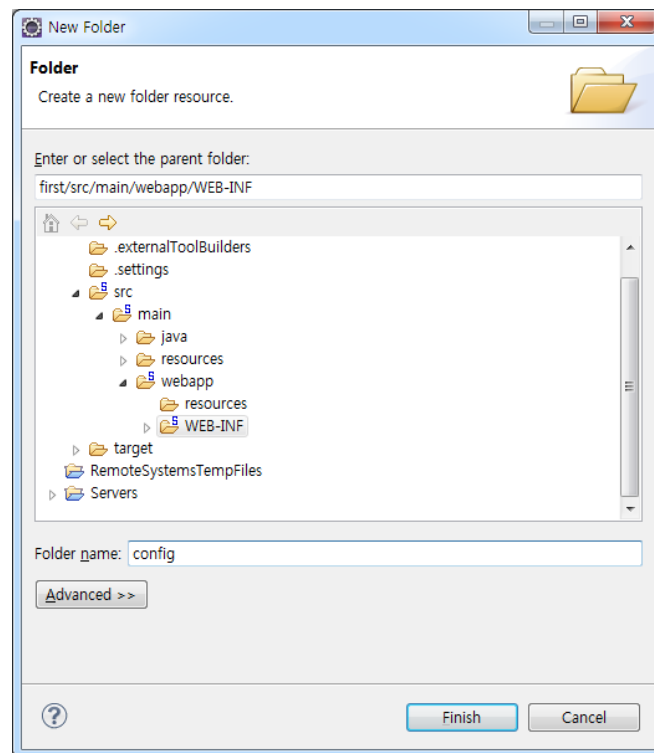
3. 서블릿 설정 변경에서 기존의 servlet-context.xml의 이름을 action-servlet.xml로 바꾸고, 위치도 바꾸었다. 설정파일은 변경되었으니, 이제 실제 소스를 바꾸려고 한다.

1. 위에서 설정한것처럼 /WEB-INF 디렉토리 밑에 config 라는 폴더를 만든다.

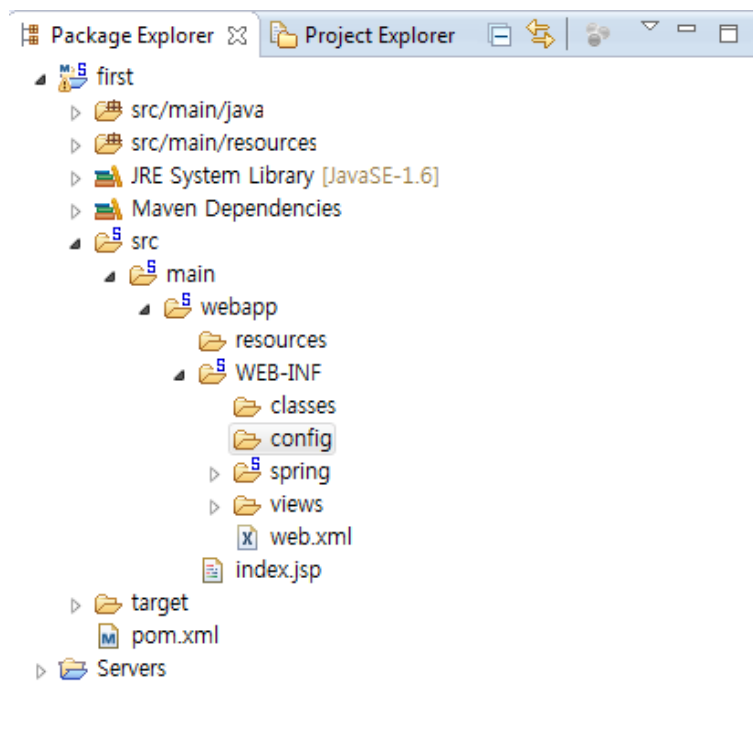
/WEB-INF 디렉토리 우클릭 -> new -> Folder 선택



2. config 폴더를 생성한다.



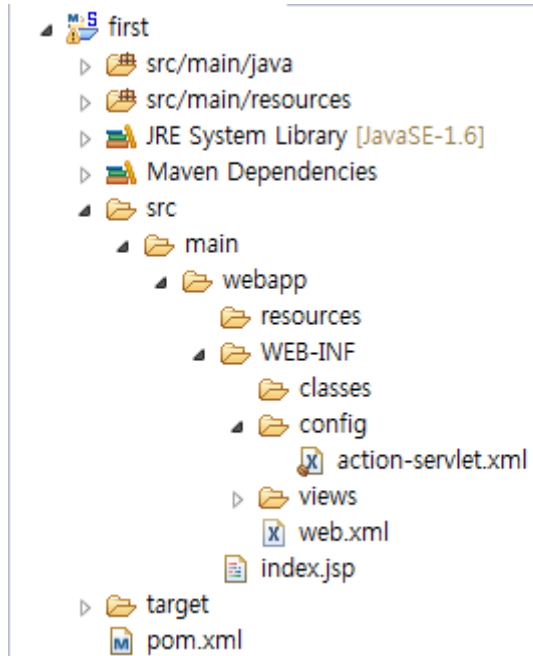
그럼 다음과 같은 폴더구조를 가진다.



3. /WEB-INF/spring/appServlet 디렉토리에 있는 servlet-context.xml을 복사해서, /WEB-

INF/config 폴더에 붙여넣고, 이름을 action-servlet.xml로 변경한다.

4. /WEB-INF/에 있는 spring이라는 디렉토리를 삭제한다.



(7) Spring MVC 구조 및 설정파일

MVC란 Model-View-Controller의 약자로, 사용자 인터페이스와 비즈니스 로직을 분리하여 웹 개발을 하는것을 가장 큰 장점으로 한다.

MVC 패턴도 MVC 모델 1과 MVC 모델 2로 나뉘어져 있는데, 요즘에는 MVC라고 하면 당연히 MVC 모델 2를 의미한다. 따라서 여기서는 MVC 모델 2 (이하 MVC)를 기준으로 진행한다.

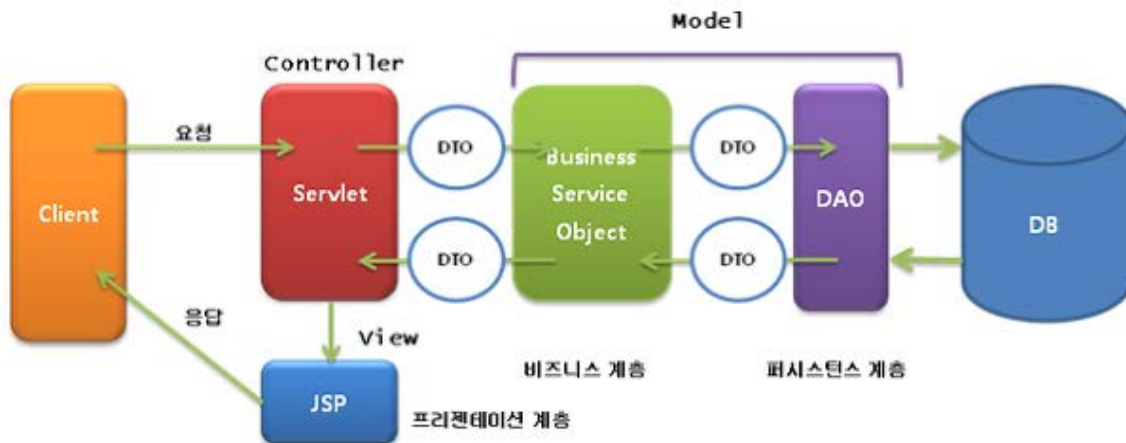
Model : 모델은 애플리케이션의 정보, 즉 데이터를 나타낸다.

View : 뷰는 사용자에게 보여주는 인터페이스, 즉 화면을 이야기한다.

자바 웹 애플리케이션에서는 JSP를 의미한다.

Controller : 컨트롤러는 비즈니스 로직과 모델의 상호동작의 조정 역할을 한다.

MVC2에서는 서블릿이 흐름을 제어하는 컨트롤러 역할을 수행한다.



1. Spring 라이브러리 추가

pom.xml는 간단히 말해서 프로젝트에서 필요한 모든 라이브러리들을 관리할 수 있도록 구성되어 있다. 예전에는 어떠한 라이브러리를 사용하려고 하면, 라이브러리를 다운받아서 특정 폴더에 복사하여 사용했었다. 라이브러리가 파일 하나로 구성되어 있는 경우라면 몰라도, 하나의 라이브러리에 다양한 파일로 구성되어 있다. 스프링도 마찬가지로, 스프링 프레임워크에는 상당히 많은 파일이 존재한다.

메이븐에서는 <dependency></dependency>라는 태그를 통해서 각 라이브러리를 추가할 수 있다.

여기서 추가해야할 라이브러리가 굉장히 많은 관계로 각 라이브러리에 대한 자세한 설명은 생략하고 바로 pom.xml만 먼저 확인하고, 복사해서 붙여넣는 것을 권장한다.

pom.xml 소스보기

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project                                xmlns="http://maven.apache.org/POM/4.0.0"
3  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-
5  v4_0_0.xsd">
6      <modelVersion>4.0.0</modelVersion>
7      <groupId>com.company</groupId>
8      <artifactId>first</artifactId>
9      <name>first</name>
10     <packaging>war</packaging>
11     <version>1.0.0-BUILD-SNAPSHOT</version>
12     <properties>
13         <java-version>1.7</java-version>
14         <org.springframework-version>3.2.4.RELEASE</org.springframework-version>
15         <org.aspectj-version>1.7.3</org.aspectj-version>
16         <org.slf4j-version>1.6.6</org.slf4j-version>
17     </properties>
18
19     <repositories>
20         <repository>
21             <id>mvn2</id>
22             <url>http://repo1.maven.org/maven2/</url>
23             <releases>
24                 <enabled>true</enabled>
25             </releases>
26             <snapshots>
27                 <enabled>true</enabled>
28             </snapshots>
29         </repository>
30
31         <repository>
32             <id>egovframe</id>
33             <url>http://www.egovframe.go.kr/maven/</url>
34             <releases>
35                 <enabled>true</enabled>
```

```
36         </releases>
37         <snapshots>
38             <enabled>false</enabled>
39         </snapshots>
40     </repository>
41
42     <repository>
43         <id>oracle</id>
44         <name>ORACLE JDBC Repository</name>
45         <url>http://mesir.googlecode.com/svn/trunk/mavenrepo</url>
46     </repository>
47 </repositories>
48
49 <dependencies>
50     <!-- Spring -->
51     <dependency>
52         <groupId>org.springframework</groupId>
53         <artifactId>spring-context</artifactId>
54         <version>${org.springframework-version}</version>
55         <exclusions>
56             <!-- Exclude Commons Logging in favor of SLF4j -->
57             <exclusion>
58                 <groupId>commons-logging</groupId>
59                 <artifactId>commons-logging</artifactId>
60             </exclusion>
61         </exclusions>
62     </dependency>
63
64     <dependency>
65         <groupId>org.springframework</groupId>
66         <artifactId>spring-web</artifactId>
67         <version>${org.springframework-version}</version>
68     </dependency>
69
70     <dependency>
71         <groupId>org.springframework</groupId>
72         <artifactId>spring-webmvc</artifactId>
73         <version>${org.springframework-version}</version>
74     </dependency>
```

```
75
76     <dependency>
77         <groupId>org.springframework</groupId>
78         <artifactId>spring-aop</artifactId>
79         <version>${org.springframework-version}</version>
80     </dependency>
81
82     <dependency>
83         <groupId>org.springframework</groupId>
84         <artifactId>spring-beans</artifactId>
85         <version>${org.springframework-version}</version>
86     </dependency>
87
88     <dependency>
89         <groupId>org.springframework</groupId>
90         <artifactId>spring-context-support</artifactId>
91         <version>${org.springframework-version}</version>
92     </dependency>
93
94     <dependency>
95         <groupId>org.springframework</groupId>
96         <artifactId>spring-core</artifactId>
97         <version>${org.springframework-version}</version>
98     </dependency>
99
100    <dependency>
101        <groupId>org.springframework</groupId>
102        <artifactId>spring-jdbc</artifactId>
103        <version>${org.springframework-version}</version>
104    </dependency>
105
106    <dependency>
107        <groupId>org.springframework</groupId>
108        <artifactId>spring-tx</artifactId>
109        <version>${org.springframework-version}</version>
110    </dependency>
111
112    <!-- AOP Alliance -->
113    <dependency>
```

```
114     <groupId>aopalliance</groupId>
115     <artifactId>aopalliance</artifactId>
116     <version>1.0</version>
117 </dependency>
118
119 <!-- MyBatis -->
120 <dependency>
121     <groupId>org.mybatis</groupId>
122     <artifactId>mybatis</artifactId>
123     <version>3.2.2</version>
124 </dependency>
125
126 <dependency>
127     <groupId>org.mybatis</groupId>
128     <artifactId>mybatis-spring</artifactId>
129     <version>1.2.0</version>
130 </dependency>
131
132 <!-- DBCP -->
133 <dependency>
134     <groupId>org.apache.tomcat</groupId>
135     <artifactId>tomcat-dbcp</artifactId>
136     <scope>provided</scope>
137     <version>7.0.53</version>
138 </dependency>
139
140 <dependency>
141     <groupId>commons-dbcp</groupId>
142     <artifactId>commons-dbcp</artifactId>
143     <version>1.4</version>
144 </dependency>
145
146 <!-- MySQL -->
147 <dependency>
148     <groupId>mysql</groupId>
149     <artifactId>mysql-connector-java</artifactId>
150     <version>5.1.31</version>
151 </dependency>
152
```

```
153     <!-- Oracle -->
154     <dependency>
155         <groupId>com.oracle</groupId>
156         <artifactId>ojdbc14</artifactId>
157         <version>10.2.0.4.0</version>
158     </dependency>
159
160     <!-- AspectJ -->
161     <dependency>
162         <groupId>org.aspectj</groupId>
163         <artifactId>aspectjrt</artifactId>
164         <version>${org.aspectj-version}</version>
165     </dependency>
166
167     <dependency>
168         <groupId>org.aspectj</groupId>
169         <artifactId>aspectjweaver</artifactId>
170         <version>${org.aspectj-version}</version>
171     </dependency>
172
173     <dependency>
174         <groupId>org.aspectj</groupId>
175         <artifactId>aspectjtools</artifactId>
176         <version>${org.aspectj-version}</version>
177     </dependency>
178
179     <!-- Web -->
180     <dependency>
181         <groupId>javax.servlet</groupId>
182         <artifactId>servlet-api</artifactId>
183         <scope>provided</scope>
184         <version>2.5</version>
185     </dependency>
186
187     <dependency>
188         <groupId>javax.servlet.jsp</groupId>
189         <artifactId>jsp-api</artifactId>
190         <scope>provided</scope>
191         <version>2.1</version>
```

```
192     </dependency>
193
194     <dependency>
195         <groupId>javax.servlet</groupId>
196         <artifactId>jstl</artifactId>
197         <version>1.2</version>
198     </dependency>
199
200     <dependency>
201         <groupId>javax.annotation</groupId>
202         <artifactId>jsr250-api</artifactId>
203         <version>1.0</version>
204     </dependency>
205
206     <!-- Logging -->
207     <dependency>
208         <groupId>org.slf4j</groupId>
209         <artifactId>slf4j-api</artifactId>
210         <version>${org.slf4j-version}</version>
211     </dependency>
212     <dependency>
213         <groupId>org.slf4j</groupId>
214         <artifactId>jcl-over-slf4j</artifactId>
215         <version>${org.slf4j-version}</version>
216     </dependency>
217     <dependency>
218         <groupId>org.slf4j</groupId>
219         <artifactId>slf4j-log4j12</artifactId>
220         <version>${org.slf4j-version}</version>
221     </dependency>
222     <dependency>
223         <groupId>log4j</groupId>
224         <artifactId>log4j</artifactId>
225         <version>1.3alpha-8</version>
226     </dependency>
227     <dependency>
228         <groupId>org.lazyluke</groupId>
229         <artifactId>log4jdbc-remix</artifactId>
230         <version>0.2.7</version>
```



```
231     </dependency>
232
233     <!-- MappingJacksonJsonView -->
234     <dependency>
235         <groupId>org.codehaus.jackson</groupId>
236         <artifactId>jackson-mapper-asl</artifactId>
237         <version>1.9.13</version>
238     </dependency>
239
240     <dependency>
241         <groupId>org.codehaus.jackson</groupId>
242         <artifactId>jackson-core-asl</artifactId>
243         <version>1.9.13</version>
244     </dependency>
245
246     <!-- MultipartHttpServletRequest -->
247     <dependency>
248         <groupId>commons-io</groupId>
249         <artifactId>commons-io</artifactId>
250         <version>2.0.1</version>
251     </dependency>
252
253     <dependency>
254         <groupId>commons-fileupload</groupId>
255         <artifactId>commons-fileupload</artifactId>
256         <version>1.2.2</version>
257     </dependency>
258
259     <!-- Apache Codec -->
260     <dependency>
261         <groupId>commons-codec</groupId>
262         <artifactId>commons-codec</artifactId>
263         <version>1.10</version>
264     </dependency>
265
266     <!-- EgovProperty -->
267     <dependency>
268         <groupId>egovframework.rte</groupId>
269         <artifactId>egovframework.rte.fdl.property</artifactId>
```

```
270     <version>2.7.0</version>
271 </dependency>
272
273 <!-- Log4j -->
274 <dependency>
275     <groupId>log4j</groupId>
276     <artifactId>log4j</artifactId>
277     <version>1.2.15</version>
278     <exclusions>
279         <exclusion>
280             <groupId>javax.mail</groupId>
281             <artifactId>mail</artifactId>
282         </exclusion>
283         <exclusion>
284             <groupId>javax.jms</groupId>
285             <artifactId>jms</artifactId>
286         </exclusion>
287         <exclusion>
288             <groupId>com.sun.jdmk</groupId>
289             <artifactId>jmxtools</artifactId>
290         </exclusion>
291         <exclusion>
292             <groupId>com.sun.jmx</groupId>
293             <artifactId>jmxri</artifactId>
294         </exclusion>
295     </exclusions>
296 </dependency>
297
298 <!-- @Inject -->
299 <dependency>
300     <groupId>javax.inject</groupId>
301     <artifactId>javax.inject</artifactId>
302     <version>1</version>
303 </dependency>
304
305 <dependency>
306     <groupId>org.apache.maven.plugins</groupId>
307     <artifactId>maven-war-plugin</artifactId>
308     <version>2.3</version>
```

```
309     </dependency>
310
311     <!-- Test -->
312     <dependency>
313         <groupId>junit</groupId>
314         <artifactId>junit</artifactId>
315         <version>4.7</version>
316         <scope>test</scope>
317     </dependency>
318 </dependencies>
319
320
321 <build>
322     <defaultGoal>install</defaultGoal>
323     <directory>${basedir}/target</directory>
324     <finalName>first</finalName>
325     <plugins>
326         <plugin>
327             <artifactId>maven-eclipse-plugin</artifactId>
328             <version>2.9</version>
329             <configuration>
330                 <additionalProjectnatures>
331                     <projectnature>org.springframework.ide.eclipse.core.springnature</project
332 nature>
333                 </additionalProjectnatures>
334                 <additionalBuildcommands>
335                     <buildcommand>org.springframework.ide.eclipse.core.springbuilder</build
336 command>
337                 </additionalBuildcommands>
338                 <downloadSources>true</downloadSources>
339                 <downloadJavadocs>true</downloadJavadocs>
340             </configuration>
341         </plugin>
342         <plugin>
343             <groupId>org.apache.maven.plugins</groupId>
344             <artifactId>maven-compiler-plugin</artifactId>
345             <version>2.5.1</version>
346             <configuration>
347                 <source>${java-version}</source>
```

```

348         <target>${java-version}</target>
349         <encoding>UTF-8</encoding>
350         <compilerArgument>-Xlint:all</compilerArgument>
351         <showWarnings>true</showWarnings>
352         <showDeprecation>true</showDeprecation>
353     </configuration>
354 </plugin>
355 <plugin>
356     <groupId>org.codehaus.mojo</groupId>
357     <artifactId>exec-maven-plugin</artifactId>
358     <version>1.2.1</version>
359     <configuration>
360         <mainClass>org.test.int1.Main</mainClass>
361     </configuration>
362 </plugin>
363
364 <!-- JavaDoc -->
365 <plugin>
366     <groupId>org.apache.maven.plugins</groupId>
367     <artifactId>maven-javadoc-plugin</artifactId>
368     <version>2.5</version>
369 </plugin>
370 </plugins>
371 </build>
372 </project>

```

사실 지금 여기에 있는 것들이 지금 당장 필요한 라이브러리들은 아니다. 그렇지만 앞으로 프로젝트를 하는데 있어서 필요한 대부분의 라이브러리들을 미리 추가해 놓았다. (실제로 진행하는 프로젝트에서도 이 정도는 기본으로 들어간다. 여기서 각 프로젝트의 특성에 맞는 라이브러리들이 추가된다.)

일단 pom.xml을 간단히 설명한다.

첫번째로 <properties></properties> 태그는 변수의 개념으로 생각하면 된다. 우리가 프로그래밍을 하면서 어떠한 값은 상수 (java에서 final static 으로 선언)로 선언하는 경우가 많은데, 이는 그 변수는 여기저기서 많이 사용되는 변수이기 때문이다. 이러한 변수를 직접 소스 여기저기에 일반적인 값으로 써 놓으면, 그 값을 변경해야 할 경우, 모든 소스에서 일일이 변경해야 하는 경우가 발생한다. 이럴때, 상수로 선언해 놓고, 그 하나만 변경하면 쉽게 값을 변경하면서도 다른

소스는 수정할 필요가 없다.

여기서도 마찬가지로 많이 사용되는 값은 변수로 지정하여, 추후 수정이 용이하도록 되어 있다.

잠시 살펴보면 `<org.springframework-version>4.2.0.RELEASE</org.springframework-version>` 라는 것을 볼 수 있는데, 이 글을 시작할 때 Spring 4.2.0 버전을 사용하기 때문에, 이 값을 이렇게 선언하였다. 만약 스프링 버전을 변경하고 싶으면 이 변수만 바꿔주면 된다.

다음은 `<repositories></repositories>` 태그다.

이는 실제 라이브러리를 다운받을 저장소를 의미한다. 보통은 따로 설정할 필요가 없다. 하지만 프로젝트를 진행하다 보면 인터넷에 연결할 수 없는 프로젝트도 상당히 많은데, 이럴때 내부 저장소를 만들어 놓고, 개발자들은 내부 저장소에서 라이브러리를 다운받도록 되어 있다.

여기서는 인터넷에서 라이브러리를 받으면서, 추가로 전자정부 프레임워크의 기능 중 하나를 사용하기 위해서 전자정부 프레임워크 저장소도 같이 추가했다.

그 다음으로 `<dependencies></dependencies>` 태그는 실제 라이브러리를 지정한다.

라이브러리는 다음과 같은 형태로 구성된다.

```
1      <dependency>
2          <groupId>org.springframework</groupId>
3          <artifactId>spring-web</artifactId>
4          <version>${org.springframework-version}</version>
5      </dependency>
```

여기서 `<version>` 태그 안에 아까 위에서 지정한 프로퍼티를 사용한 것을 볼 수 있다.

여기서는 pom.xml에 대해서 간단히 설명했는데, 좀 더 자세히 알고싶은 사람들은 인터넷에서 메이븐에 관련된 글을 읽어보는것을 추천한다.

2. 설정파일 변경 (web.xml, action-servlet.xml 등)

이제 하나씩 설정을 하도록 하자.

첫번째로 **web.xml** 파일을 열어보자.

몇가지 설정을 덧붙이려고 한다.

1. UTF-8 설정

다음은 web.xml에 추가한다.

```
1      <filter>
2          <filter-name>encodingFilter</filter-name>
3      <filter-class>
4          org.springframework.web.filter.CharacterEncodingFilter
5      </filter-class>
6      <init-param>
7          <param-name>encoding</param-name>
8          <param-value>utf-8</param-value>
9      </init-param>
10     </filter>
11     <filter-mapping>
12         <filter-name>encodingFilter</filter-name>
13         <url-pattern>*.do</url-pattern>
14     </filter-mapping>
```

요즘에는 기본적으로 UTF-8로 인코딩한다.

2. 그 외 설정파일 경로 설정

기존 설정파일을 보면 <servlet>설정 중에 <param-value> 라는 것이 있다.

기존에는 servlet 설정 파일이 action-servlet.xml 하나만 있었는데, 이제 또 추가되기 때문에, 확장성을 생각하여 특정 폴더에 있는 설정파일을 모두 읽어오는 방식으로 변경한다.

<param-value> 태그를 다음과 같이 수정한다.

<param-value>/WEB-INF/config/*-servlet.xml</param-value>

이는 /WEB-INF/config/ 폴더안에 있는 -servlet.xml로 끝나는 모든 파일을 읽어오는 것을 의미한다. 따라서 앞으로 설정파일을 추가할때는 XXXXXX-servlet.xml로 만들게 되면, 자동적으로 설정파일을 읽어 들인다.

다음으로는 Spring 설정파일을 추가한다. 기존에 <context-param> 태그의 <param-value>태그안에는 아마 아무것도 작성되지 않았을 것이다. 이제 본격적인 스프링 설정파일을 읽어오기 위해서 다음과 같이 바꾼다.

<param-value>classpath*:config/spring/context-*.xml</param-value>

이는 앞에서 설명한것과 비슷하게 context-로 시작하는 모든 .xml 을 읽어오는 것을 뜻한다.

여기서 지금 이렇게 추가하면 context-XXXXXX.xml 파일이 없기 때문에 에러가 날 것이다. 일단 작성만 해 놓고 주석처리 하도록 한다.

여기까지 작성해서 완성된 web.xml은 다음과 같다.

web.xml 보기

```
1 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
4   http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
5
6   <welcome-file-list>
7     <welcome-file>index.jsp</welcome-file>
8   </welcome-file-list>
9
10  <filter>
11    <filter-name>encodingFilter</filter-name>
12    <filter-class>
13      org.springframework.web.filter.CharacterEncodingFilter
14    </filter-class>
15    <init-param>
16      <param-name>encoding</param-name>
17      <param-value>utf-8</param-value>
18    </init-param>
19  </filter>
20  <filter-mapping>
21    <filter-name>encodingFilter</filter-name>
22    <url-pattern>*.do</url-pattern>
23  </filter-mapping>
24
25  <listener>
26    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
27class>
28  </listener>
29
30  <servlet>
31    <servlet-name>action</servlet-name>
32    <servlet-class>
```

```
33     org.springframework.web.servlet.DispatcherServlet
34     </servlet-class>
35     <init-param>
36         <param-name>contextConfigLocation</param-name>
37         <param-value>
38             /WEB-INF/config/*-servlet.xml
39         </param-value>
40     </init-param>
41     <load-on-startup>1</load-on-startup>
42 </servlet>
43 <servlet-mapping>
44     <servlet-name>action</servlet-name>
45     <url-pattern>*.do</url-pattern>
46 </servlet-mapping>
47
48 <context-param>
49     <param-name>contextConfigLocation</param-name>
50     <param-value>classpath*:config/spring/context-*.xml</param-value>
    </context-param>
</web-app>
```