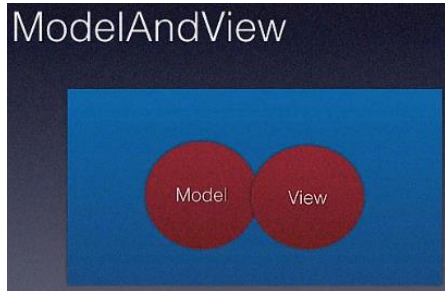


1. 소개

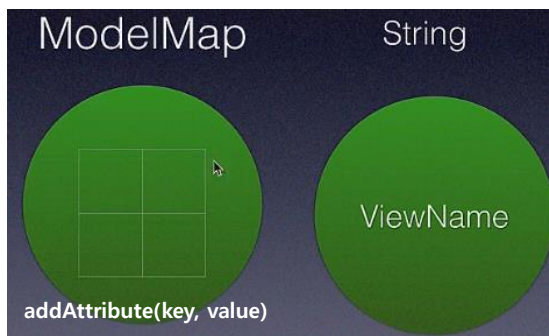
가. ModelAndView

- 컨트롤러 내에서 사용자 요청을 처리하는 메소드의 리턴 타입으로 사용
- 디자인 측면: 모델과 뷰가 밀접하게 커플 되어 있음
- 단점: 모델 또는 데이터를 사용하지 않더라도 ModelAndView 오브젝트를 생성해야 함



나. ModelMap

- 새롭게 도입
- 뷰 이름(ViewName): 간단한 문자열로
- 모델 또는 데이터가 필요 없을 때: ModelAndView 만들 필요없이 문자열로 뷰이름을 생성
- 모델 또는 데이터가 필요할 때: ModelMap 사용
- ModelMap: 키(문자열)와 값(오브젝트)으로 이루어진 맵(map)과 유사, addAttribute 메소드



```
UserController.java
1 package com.springmvc_uitoctrl.controller;
2
3 import org.springframework.stereotype.Controller;
10
11 @Controller
12 public class UserController {
13
14     @RequestMapping("registrationPage")
15     public ModelAndView showRegistrationPage()
16     {
17         ModelAndView modelAndView = new ModelAndView();
18
19         modelAndView.setViewName("userReg");
20
21         return modelAndView;
22     }
23
24     @RequestMapping(value = "registerUser", method = RequestMethod.POST)
25     public ModelAndView registerUser(@ModelAttribute("user") User user)
26     {
27         System.out.println(user);
28         // ModelAndMap으로 변경
29         ModelAndView modelAndView = new ModelAndView();
30         modelAndView.setViewName("userReg");
31         modelAndView.addObject("user", user);
32         modelAndView.setViewName("regResult");
33         return modelAndView;
34     }
35 }
36
37 }
```

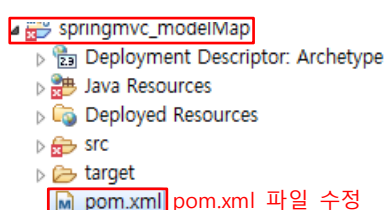
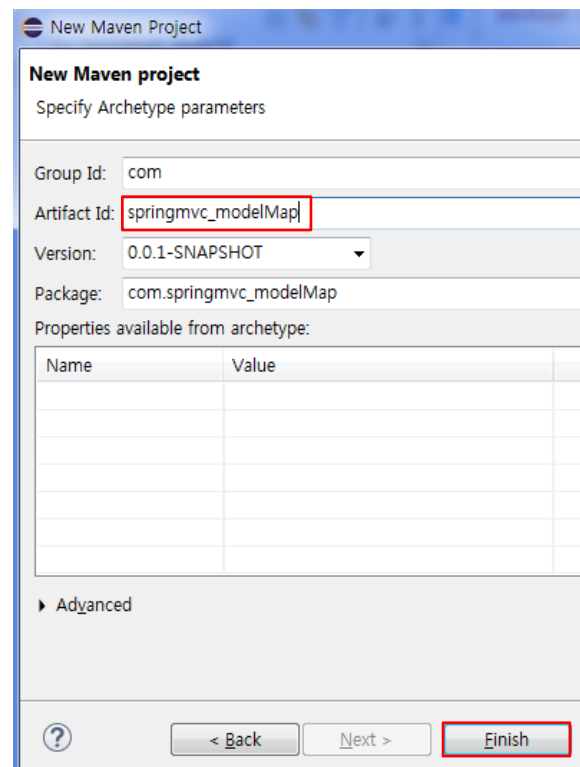
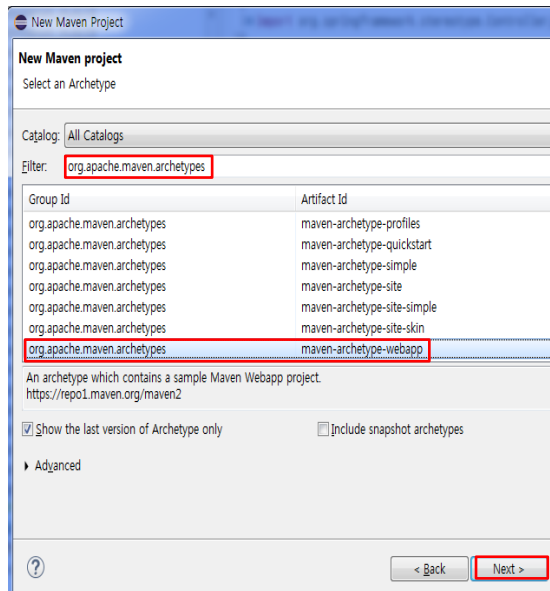
모델이나 데이터가 필요 없어

ModelAndView 객체 생성 없이 뷰 이름만 지정해주면 됨.

모델이나 데이터가 필요해

ModelMap 객체 생성해 데이터 설정하고
뷰 이름은 문자열로 지정.

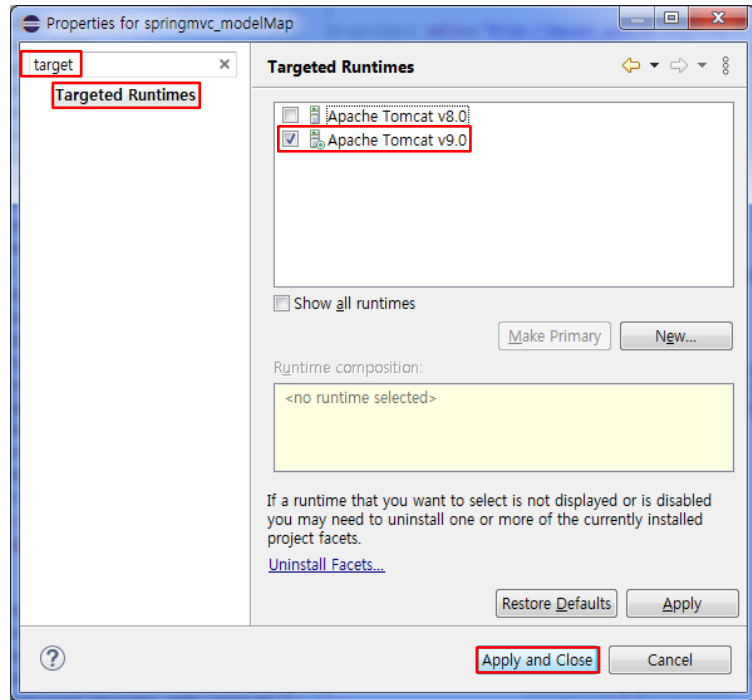
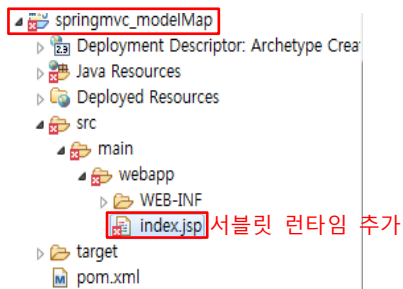
2. 메이븐 프로젝트 만들기: 기존 springmvc_uitoctrl 요소를 가져와 복사
- 가. 기본 프로젝트 생성



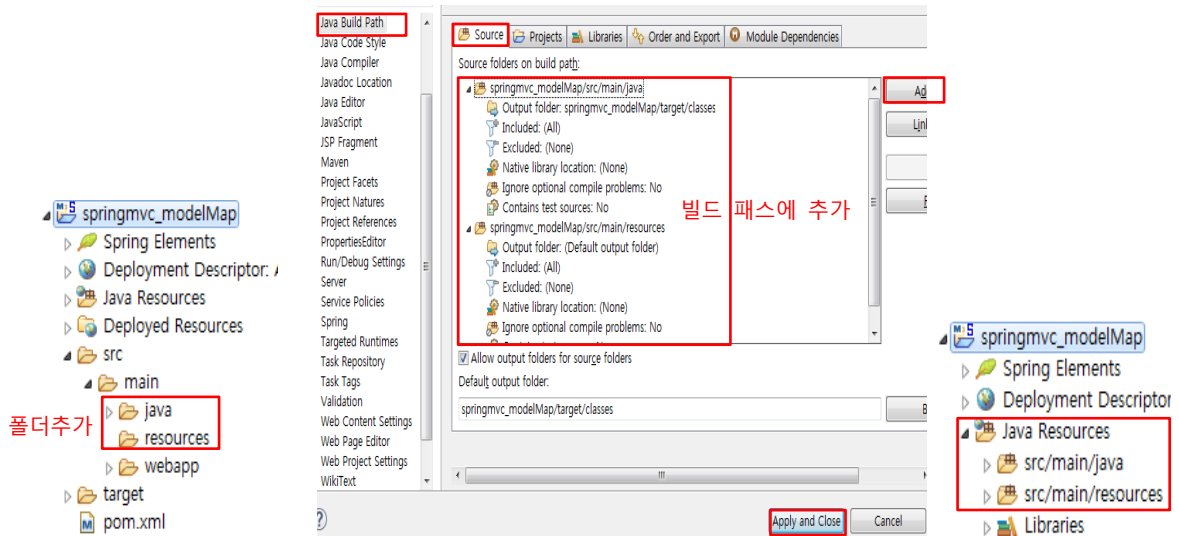
```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/POM/4.0.0" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0/modelVersion" modelVersion="4.0.0">
4
5      <groupId>com</groupId>
6      <artifactId>springmvc_modelMap</artifactId>
7      <version>0.0.1-SNAPSHOT</version>
8      <packaging>war</packaging>
9
10     <name>springmvc_modelMap Maven Webapp</name>
11     <url>http://www.example.com</url>
12
13     <properties>
14         <springframework.version>4.3.6.RELEASE</springframework.version>
15     </properties>
16
17     <dependencies>
18         <dependency>
19             <groupId>org.springframework</groupId>
20             <artifactId>spring-webmvc</artifactId>
21             <version>${springframework.version}</version>
22         </dependency>
23         <dependency>
24             <groupId>jstl</groupId>
25             <artifactId>jstl</artifactId>
26             <version>1.2</version>
27         </dependency>
28     </dependencies>
29
30     <build>
31         <pluginManagement>
32             <plugins>
33                 <plugin>
34                     <groupId>org.apache.maven.plugins</groupId>
35                     <artifactId>maven-compiler-plugin</artifactId>
36                     <version>3.8.0</version>
37                     <configuration>
38                         <source>11</source>
39                         <target>11</target>
40                     </configuration>
41                 </plugin>
42             </plugins>
43         </pluginManagement>
44     </build>
45 </project>

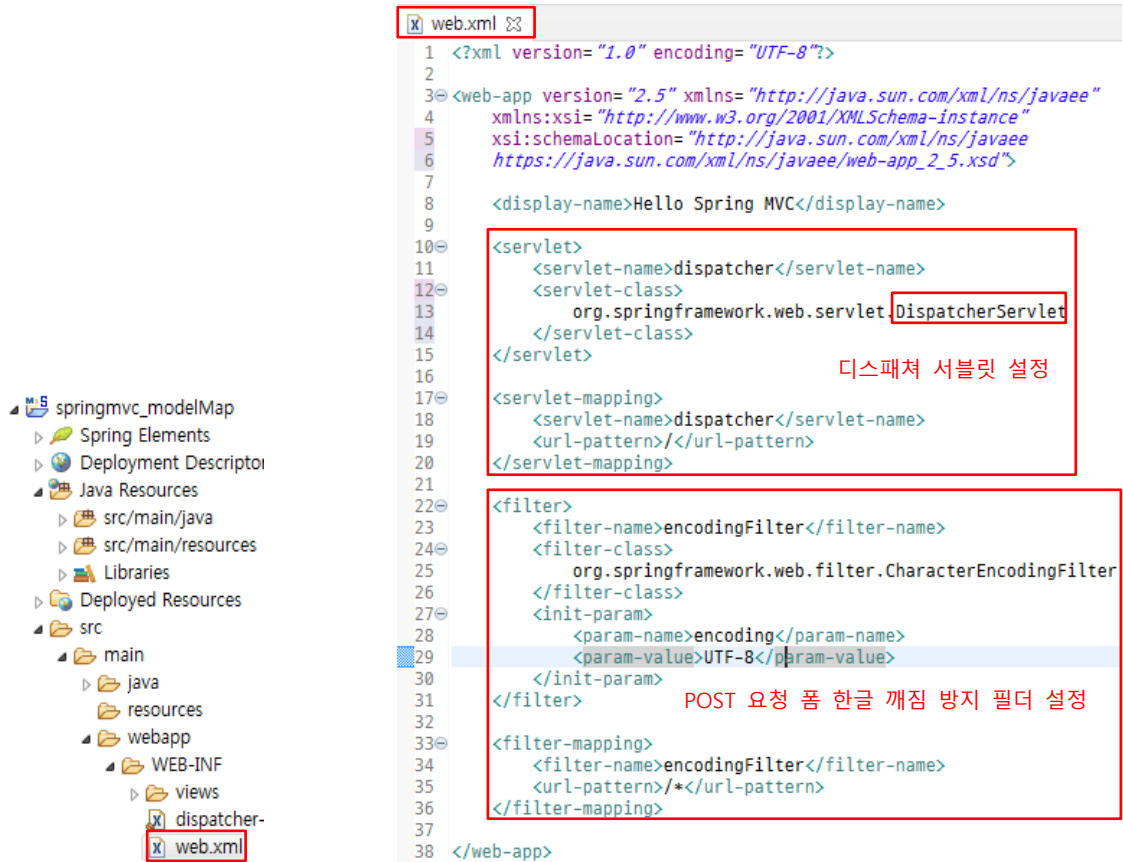
```



나. 개발 및 리소스 폴더 추가



다. 디스패처 서블릿 설정: web.xml



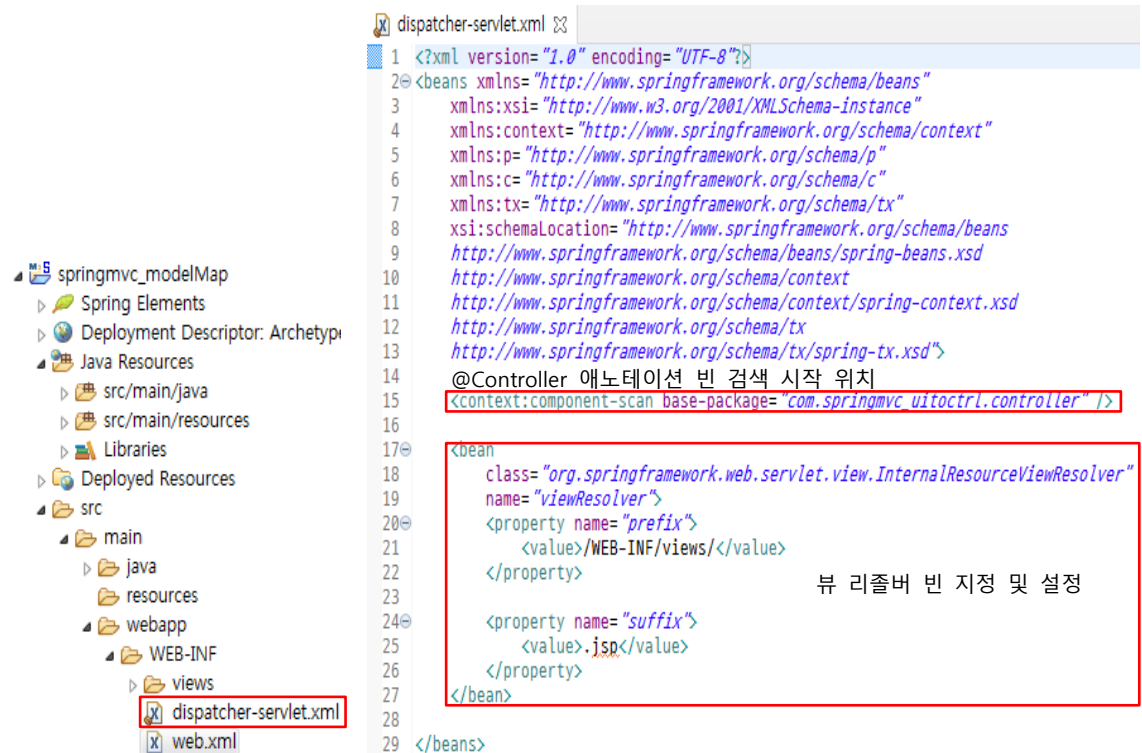
```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6     https://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
7
8   <display-name>Hello Spring MVC</display-name>
9
10  <servlet>
11    <servlet-name>dispatcher</servlet-name>
12    <servlet-class>
13      org.springframework.web.servlet.DispatcherServlet
14    </servlet-class>
15  </servlet>
16
17  <servlet-mapping>
18    <servlet-name>dispatcher</servlet-name>
19    <url-pattern>/</url-pattern>
20  </servlet-mapping>
21
22  <filter>
23    <filter-name>encodingFilter</filter-name>
24    <filter-class>
25      org.springframework.web.filter.CharacterEncodingFilter
26    </filter-class>
27    <init-param>
28      <param-name>encoding</param-name>
29      <param-value>UTF-8</param-value>
30    </init-param>
31  </filter>
32
33  <filter-mapping>
34    <filter-name>encodingFilter</filter-name>
35    <url-pattern>/*</url-pattern>
36  </filter-mapping>
37
38 </web-app>
  
```

디스패처 서블릿 설정

POST 요청 폼 한글 깨짐 방지 필터 설정

라. 스프링 설정: dispatcher-servlet.xml

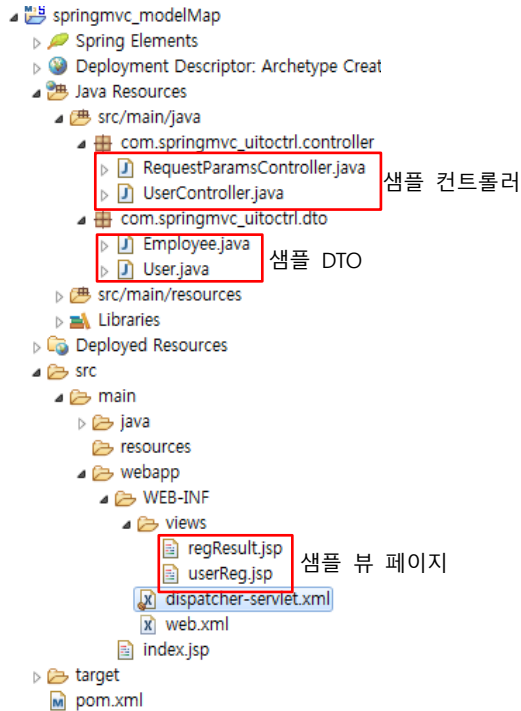


```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:context="http://www.springframework.org/schema/context"
5   xmlns:p="http://www.springframework.org/schema/p"
6   xmlns:c="http://www.springframework.org/schema/c"
7   xmlns:tx="http://www.springframework.org/schema/tx"
8   xsi:schemaLocation="http://www.springframework.org/schema/beans
9     http://www.springframework.org/schema/beans/spring-beans.xsd
10    http://www.springframework.org/schema/context
11    http://www.springframework.org/schema/context/spring-context.xsd
12    http://www.springframework.org/schema/tx
13    http://www.springframework.org/schema/tx/spring-tx.xsd">
14   <!-- @Controller 애노테이션 빈 검색 시작 위치 -->
15   <context:component-scan base-package="com.springmvc.utictrol.controller"/>
16
17   <bean
18     class="org.springframework.web.servlet.view.InternalResourceViewResolver"
19     name="viewResolver">
20     <property name="prefix">
21       <value>/WEB-INF/views/</value>
22     </property>
23
24     <property name="suffix">
25       <value>.jsp</value>
26     </property>
27   </bean>
28
29 </beans>
  
```

뷰 리졸버 빈 지정 및 설정

마. 샘플 컨트롤러, DTO, 뷰 페이지 생성



3. 어플리케이션 마이그레이션

```

1 package com.springmvc_uitoctrl.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.ModelMap;
5 import org.springframework.web.bind.annotation.ModelAttribute;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestMethod;
8
9 import com.springmvc_uitoctrl.dto.User;
10
11 @Controller
12 public class UserController {
13
14     @RequestMapping("registrationPage")
15     public String showRegistrationPage()
16     {
17         return "userReg";
18     }
19
20     @RequestMapping(value = "registerUser", method = RequestMethod.POST)
21     public String registerUser(@ModelAttribute("user") User user,
22                               @ModelAttribute("model") ModelMap model)
23     {
24         System.out.println(user);
25         model.addAttribute("user", user);
26         return "regResult";
27     }
28 }
                
```

localhost:8080/springmvc_modelMap/registrationPage

Id:

Name:

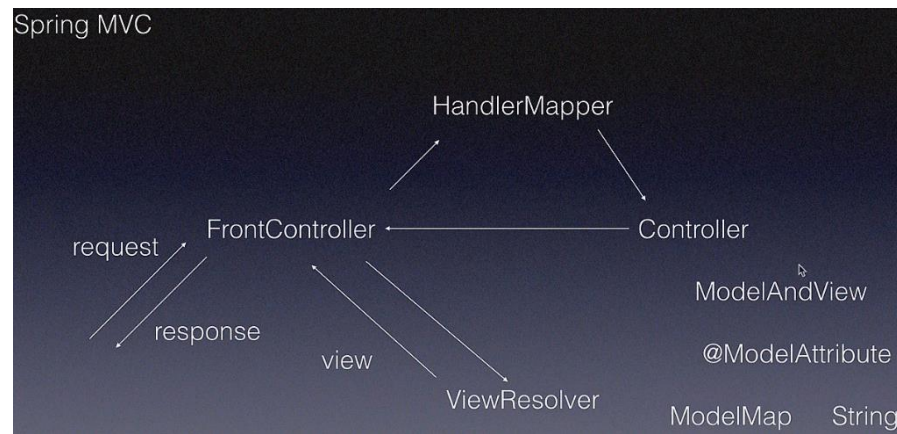
Email:

localhost:8080/springmvc_modelMap/registerUser

성공적으로 사용자를 등록했습니다. 사용자의 자세한 정보는:
아이디: 123 이름: 김형태 이메일: www@daum.net

성공적으로 사용자를 등록했습니다. 사용자의 자세한 정보는:
123 김형태 www@daum.net

4. 스프링 요약



5.