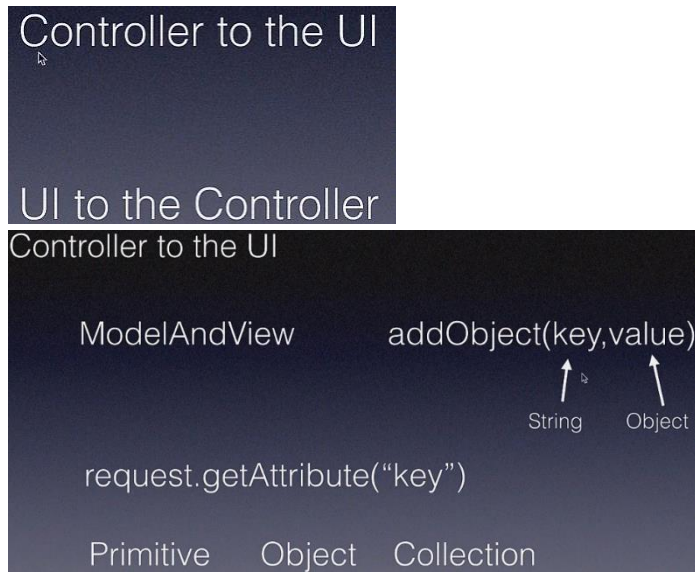


## 1. 소개



## 2. 메이븐 프로젝트 만들기

### 가. 프로젝트 생성

**New Maven Project**

Select an Archetype

Catalog: All Catalogs

Filter: org.apache.maven.archetypes

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.4
org.apache.maven.archetypes	maven-archetype-simple	1.4
org.apache.maven.archetypes	maven-archetype-site	1.4
org.apache.maven.archetypes	maven-archetype-site-simple	1.4
org.apache.maven.archetypes	maven-archetype-site-skin	1.4
org.apache.maven.archetypes	maven-archetype-webapp	1.4

An archetype which contains a sample Maven Webapp project.  
https://repo1.maven.org/maven2

☒ Show the last version of Archetype only      ☐ Include snapshot archetypes

Advanced

< Back      Next >      Finish

**springmvc\_ctrltoui**

Specify Archetype parameters

Group Id: com

Artifact Id: springmvc\_ctrltoui

Version: 0.0.1-SNAPSHOT

Package: com.springmvc\_ctrltoui

Properties available from archetype:

Name	Value

Advanced

< Back      Next >      Finish

**springmvc\_ctrltoui**

Deployment Descriptor: Archetype Create

Java Resources

Libraries

JRE System Library [JavaSE-1.7]      JDK 11로 변경

Maven Dependencies

junit-4.11.jar - C:\Users\VIPW\...  
hamcrest-core-1.3.jar - C:\Users\VIPW\...

Deployed Resources

src

main      ③ 1. Java, resources 이름의 작업폴더 생성  
2. 작업폴더를 빌드패스에 추가

webapp

WEB-INF

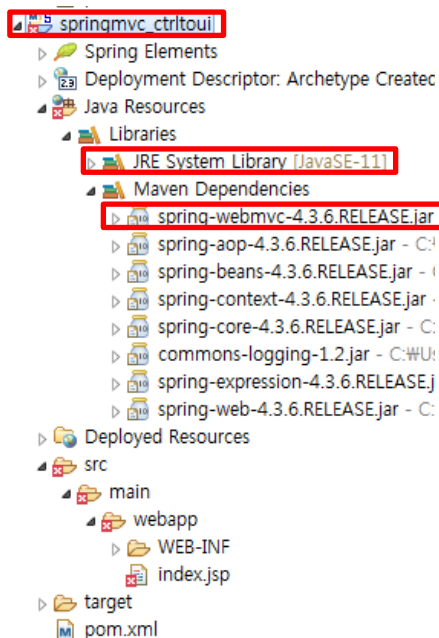
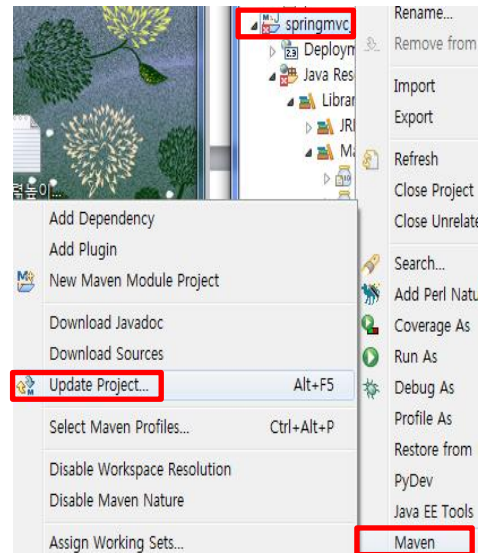
index.jsp      ② 서버릿런타임 추가

target

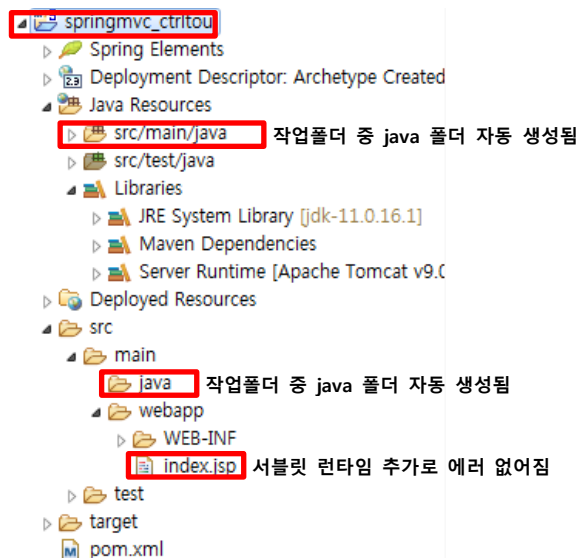
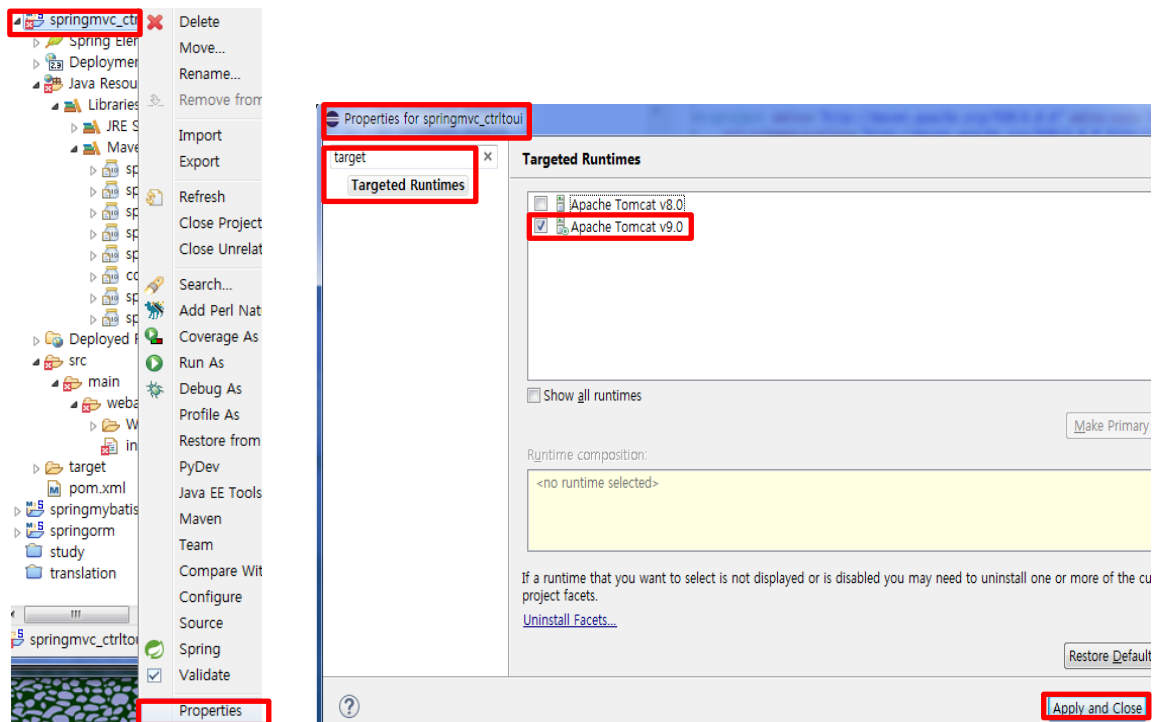
pom.xml      ① JDK 버전을 11로 바꾸고, 스프링 webmvc추가

## 나. pom.xml 수정(1)

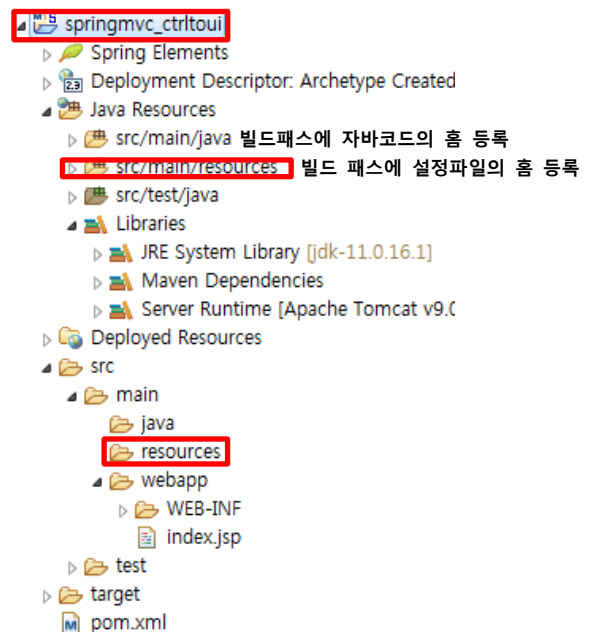
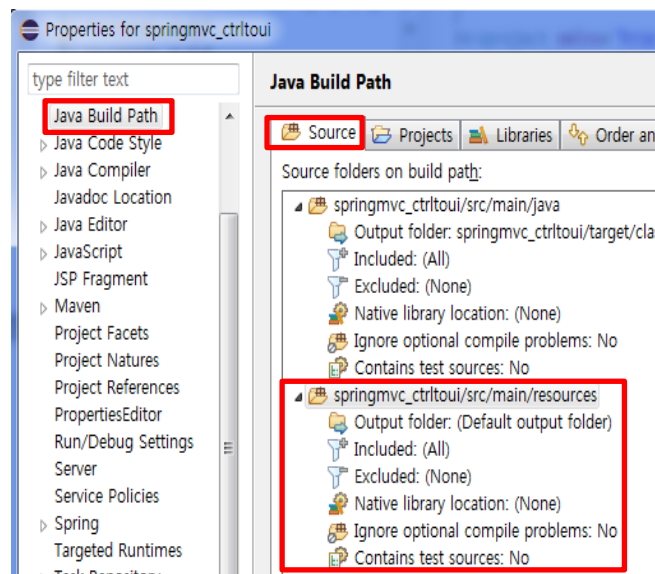
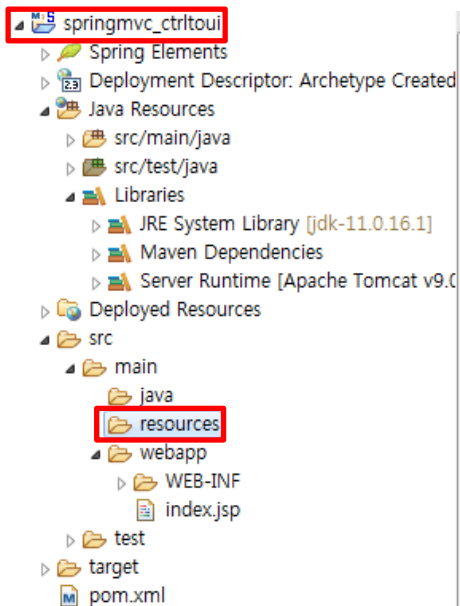
```
springmvc_ctrltoui/pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://mav
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com</groupId>
8   <artifactId>springmvc_ctrltoui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



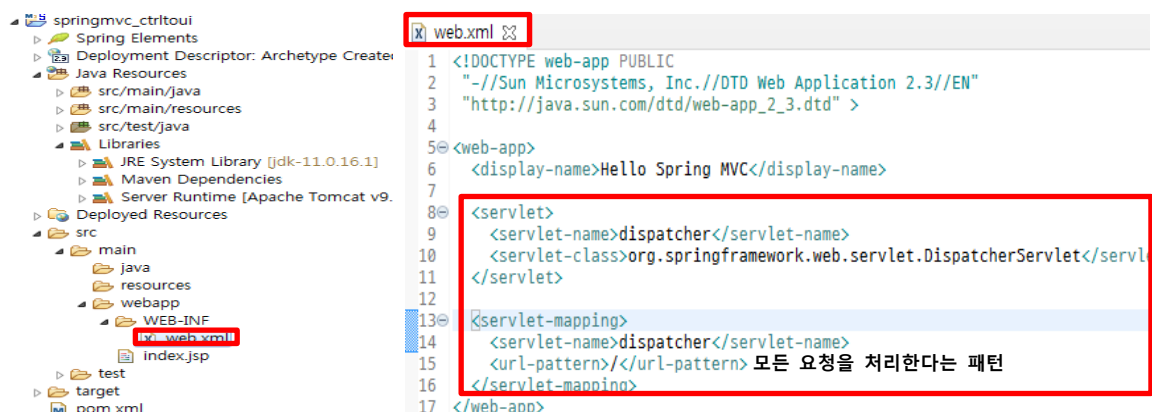
## 다. 서블릿 런타임 추가(2)



라. 리소스 폴더 생성하고 빌드 패스에 추가(3): resources



마. 디스패처 서블릿 설정: web.xml



바. 스프링 설정 파일 만들기: WEB-INF 폴더 아래

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
15     <context:component-scan base-package="com.springmvc_ctrltoui.controller" />
16
17 </beans>

```

컨트롤러 클래스 상단에 @Controller 붙여 스프링이 관리하는 빈으로 등록

사. 뷰 리졸버 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
15     <context:component-scan base-package="com.springmvc_ctrltoui.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         <property name="suffix">
24             <value>.jsp</value>
25         </property>
26     </bean>
27
28 </beans>

```

뷰 파일 저장 위치

아. 컨트롤러 클래스 만들기

dispatcher-servlet.xml

```

1 package com.springmvc_ctrltoui.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.servlet.ModelAndView;
6
7 @Controller <context:component-scan base-package="com.springmvc_ctrltoui.controller" />
8 public class HelloController {
9
10     @RequestMapping("/hello") <url-pattern>/</url-pattern> "/>
11     public ModelAndView hello()
12     {
13         ModelAndView modelAndView = new ModelAndView();
14         modelAndView.setViewName("hello");
15         return modelAndView;
16     }
17 }

```

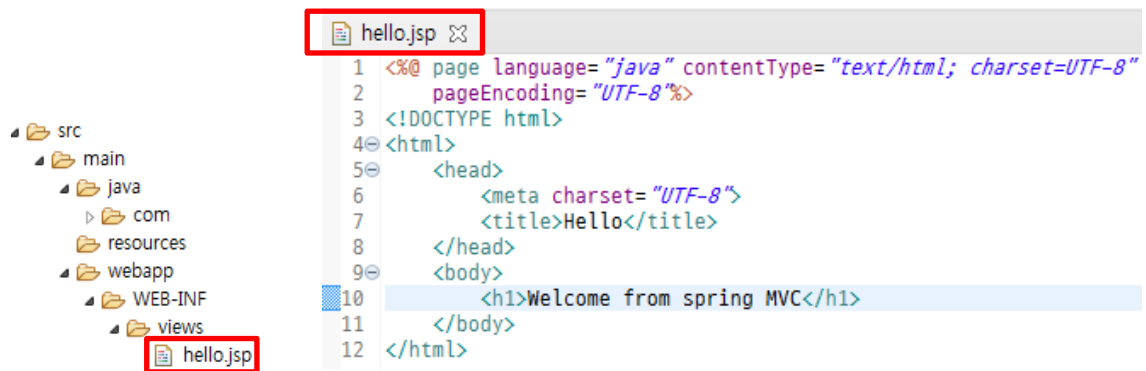
web.xml

dispatcher-servlet.xml

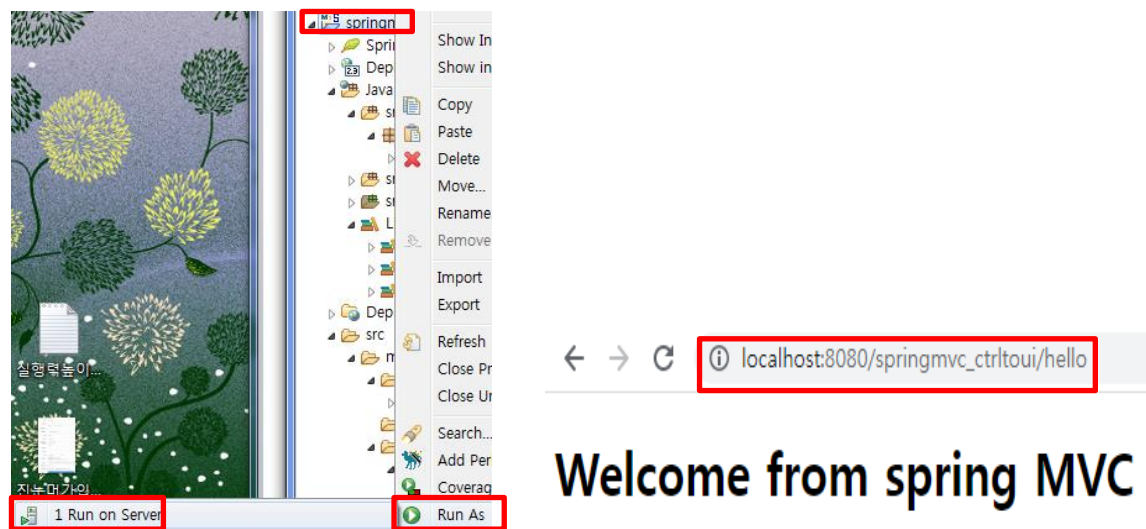
name="viewResolver"

/WEB-INF/views/hello.jsp 호출

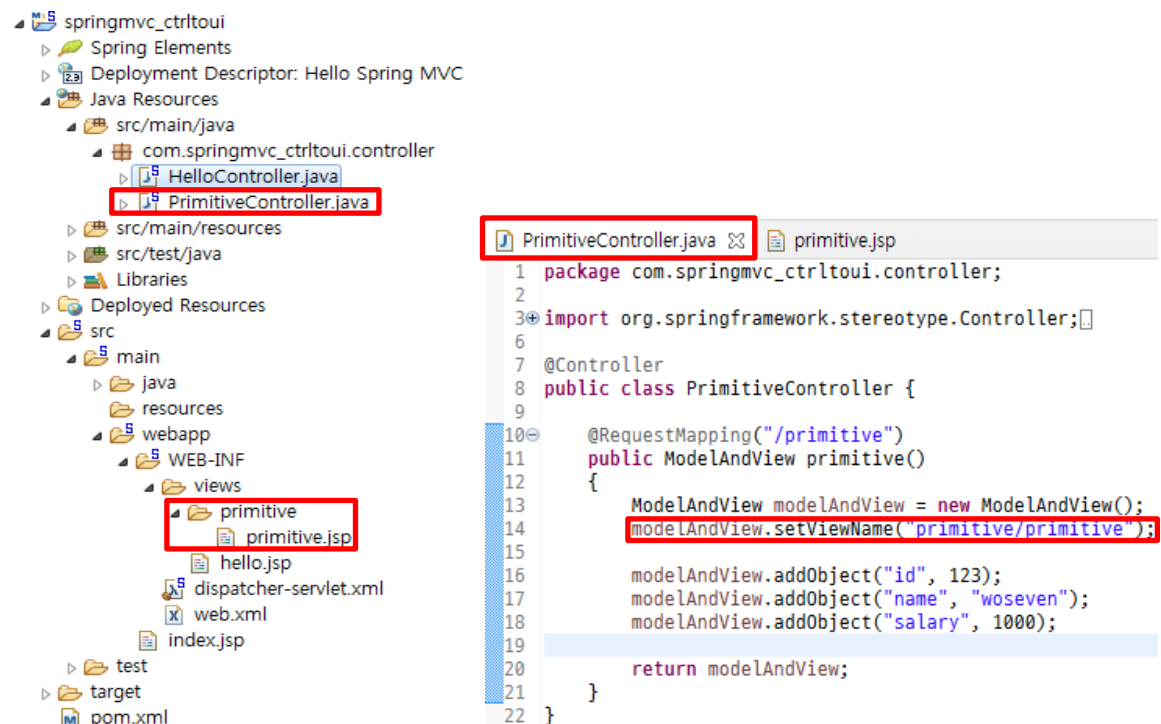
자. WEB-INF 아래 폴더 및 JSP 페이지 만들기



차. 실행



3. 원시타입(Primitive Type)을 UI에 보내기





PrimitiveController.java

primitive.jsp

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>Primitive</title>
8   </head>
9   <body>
10    <%
11      Integer id = (Integer)request.getAttribute("id");
12      String name = (String)request.getAttribute("name");
13      Integer salary = (Integer)request.getAttribute("salary");
14      out.println("ID: " + id);
15      out.println("Name: " + name);
16      out.println("Salary: " + salary);
17    %>
18  </body>
19 </html>

```

← → ↺

localhost:8080/springmvc\_ctrltoui/primitive

ID: 123 Name: woseven Salary: 1000

#### 4. JSP Expression Language 사용

springmvc\_ctrltoui

Spring Elements

Deployment Descriptor: Hello Spring MVC

Java Resources

src/main/java

com.springmvc\_ctrltoui.controller

HelloController.java

JSPExpLangController.java

PrimitiveController.java

src/main/resources

src/test/java

Libraries

Deployed Resources

src

main

java

resources

webapp

WEB-INF

views

jspexpressionlanguage

jspexpressionlanguage.jsp

primitive

primitive.jsp

hello.jsp

dispatcher-servlet.xml

web.xml

index.jsp

test

target

pom.xml

jspexpressionlanguage.jsp

JSPExpLangController.java

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>JSP Expression Language</title>
8   </head>
9   <body>
10    ID:<b>${id}</b>
11    Name:<b>${name}</b>
12    Salary:<b>${salary}</b>
13  </body>
14 </html>

```

```

jspexpressionlanguage.jsp JSPEXPController.java
1 package com.springmvc_ctrltoui.controller;
2
3 import org.springframework.stereotype.Controller;
4
5
6
7 @Controller
8 public class JSPEXPController {
9
10 @RequestMapping("/jep")
11 public ModelAndView primitive()
12 {
13     ModelAndView modelAndView = new ModelAndView();
14     modelAndView.setViewName("jspexpressionlanguage/jspexpressionlanguage");
15
16     modelAndView.addObject("id", 123);
17     modelAndView.addObject("name", "woseven");
18     modelAndView.addObject("salary", 1000);
19
20     return modelAndView;
21 }
22 }

```

localhost:8080/springmvc\_ctrltoui/jep

ID:\${id} Name:\${name} Salary:\${salary}

JSP expression language의 기본값이 false

즉, isELIgnore="true"

```

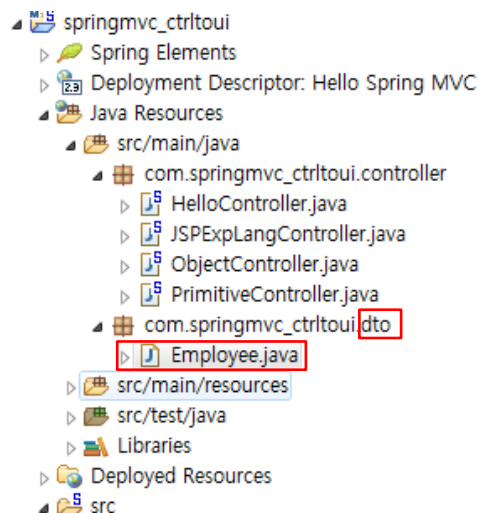
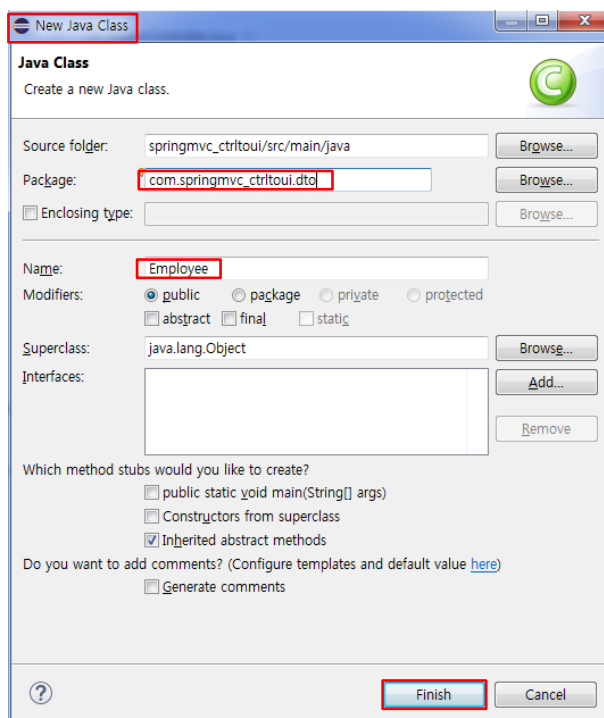
jspexpressionlanguage.jsp JSPEXPController.java PrimitiveController.java
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isELIgnore="false"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-4">
7   <title>JSP Expression Language</title>
8 </head>
9 <body>
10   ID:<b>${id}</b>
11   Name:<b>${name}</b>
12   Salary:<b>${salary}</b>
13 </body>
14 </html>

```

localhost:8080/springmvc\_ctrltoui/jep

ID:123 Name:woseven Salary:1000

## 5. 오브젝트(Object) 데이터를 UI에 보내기





## Employee.java

```

1 package com.springmvc_ctrltoui.dto;
2
3 public class Employee {
4
5     private int id;
6     private String name;
7     private double salary;
8
9     public int getId() {
10         return id;
11     }
12
13     public void setId(int id) {
14         this.id = id;
15     }
16
17     public String getName() {
18         return name;
19     }
20
21     public void setName(String name) {
22         this.name = name;
23     }
24
25     public double getSalary() {
26         return salary;
27     }
28
29     public void setSalary(double salary) {
30         this.salary = salary;
31     }
32
33     @Override
34     public String toString() {
35         return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";
36     }
37 }

```

객체 내 필드를 getter로 하나씩 값을 가져오지 않고,  
전체를 한번에 가져오려면 **Object**의 **toString()**메소드를  
오버라이드 해야 함.

## ObjectController.java

```

1 package com.springmvc_ctrltoui.controller;
2
3 import org.springframework.stereotype.Controller;
4
5 @Controller
6 public class ObjectController {
7
8     @RequestMapping("/object")
9     public ModelAndView object()
10     {
11         ModelAndView modelAndView = new ModelAndView();
12         modelAndView.setViewName("object/object");
13
14         Employee employee = new Employee();
15         employee.setId(1234);
16         employee.setName("woseven");
17         employee.setSalary(20000);
18
19         modelAndView.addObject("employee", employee);
20
21         return modelAndView;
22     }
23 }

```

springmvc\_ctrltoui

- Spring Elements
- Deployment Descriptor: Hello Spring M
- Java Resources
  - src/main/java
    - com.springmvc\_ctrltoui.controller
      - HelloController.java
      - JSPExpLangController.java
      - ObjectController.java**
      - PrimitiveController.java
    - com.springmvc\_ctrltoui.dto
      - Employee.java
  - src/main/resources
  - src/test/java
  - Libraries
  - Deployed Resources
  - src

## 6. 오브젝트 뷰 만들기



```
springmvc_ctrltoui
├── Spring Elements
├── Deployment Descriptor: Hello Spring
├── Java Resources
│   ├── src/main/java
│   ├── src/main/resources
│   ├── src/test/java
│   └── Libraries
├── Deployed Resources
├── src
│   ├── main
│   │   ├── java
│   │   ├── resources
│   │   └── webapp
│   │       ├── WEB-INF
│   │       │   ├── views
│   │       │   │   ├── jspexpressionlanguage
│   │       │   │   └── object
│   │       │   │       └── object.jsp
│   │       ├── primitive
│   │       ├── hello.jsp
│   │       ├── dispatcher-servlet.xml
│   │       ├── web.xml
│   │       └── index.jsp
│   ├── test
│   └── target
└── pom.xml
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isELIgnored="false" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>JSP Object</title>
8   </head>
9   <body>
10    <%= request.getAttribute("employee") %>
11  </body>
12 </html>
```

localhost:8080/springmvc\_ctrltoui/object

Employee [id=1234, name=woseven, salary=20000.0]

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isELIgnored="false" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>JSP Object</title>
8   </head>
9   <body>
10    <%= request.getAttribute("employee") %>
11
12    <br />
13
14    <b>${employee}</b>
15
16    </body>
17
18 </html>
```

localhost:8080/springmvc\_ctrltoui/object

Employee [id=1234, name=woseven, salary=20000.0]

**Employee [id=1234, name=woseven, salary=20000.0]**

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isELIgnored="false" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>JSP Object</title>
8   </head>
9   <body>
10    <%= request.getAttribute("employee") %>
11
12    <br />
13
14    <b>${employee}</b>
15
16    <br />
17
18    <b>ID: ${employee.id}</b>
19    <b>Name: ${employee.name}</b>
20    <b>Salary: ${employee.salary}</b>
21
22    </body>
23
24 </html>
```

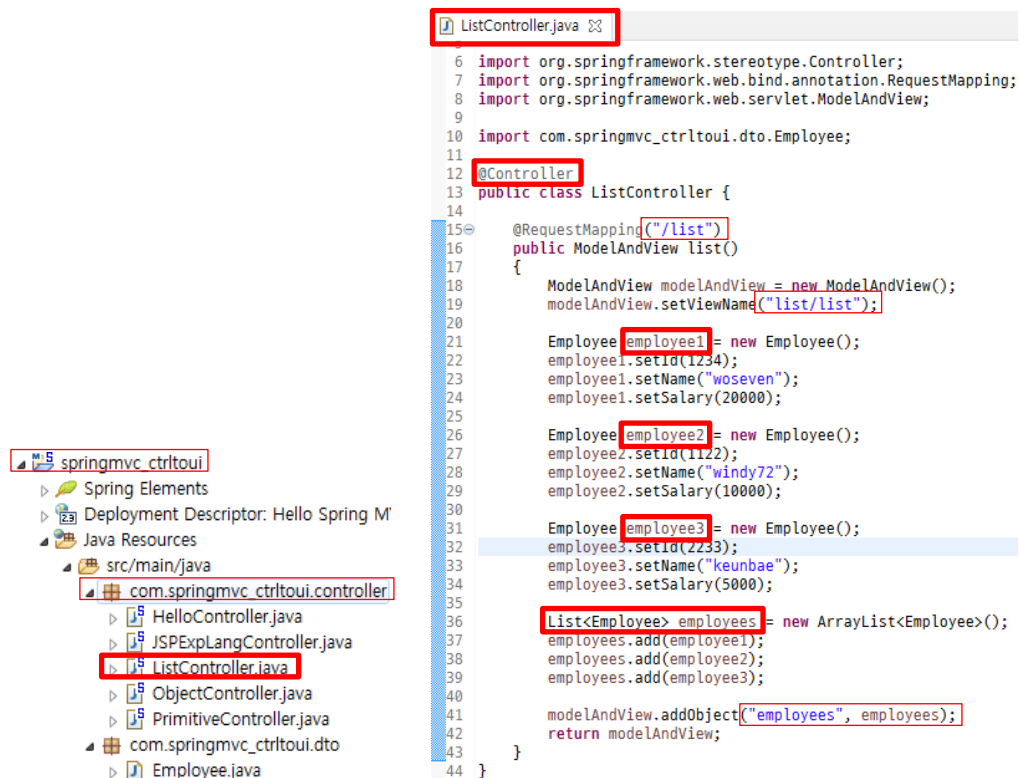
localhost:8080/springmvc\_ctrltoui/object

Employee [id=1234, name=woseven, salary=20000.0]

**Employee [id=1234, name=woseven, salary=20000.0]**

**ID: 1234 Name: woseven Salary: 20000.0**

## 7. 리스트(List) 컨트롤러 만들기

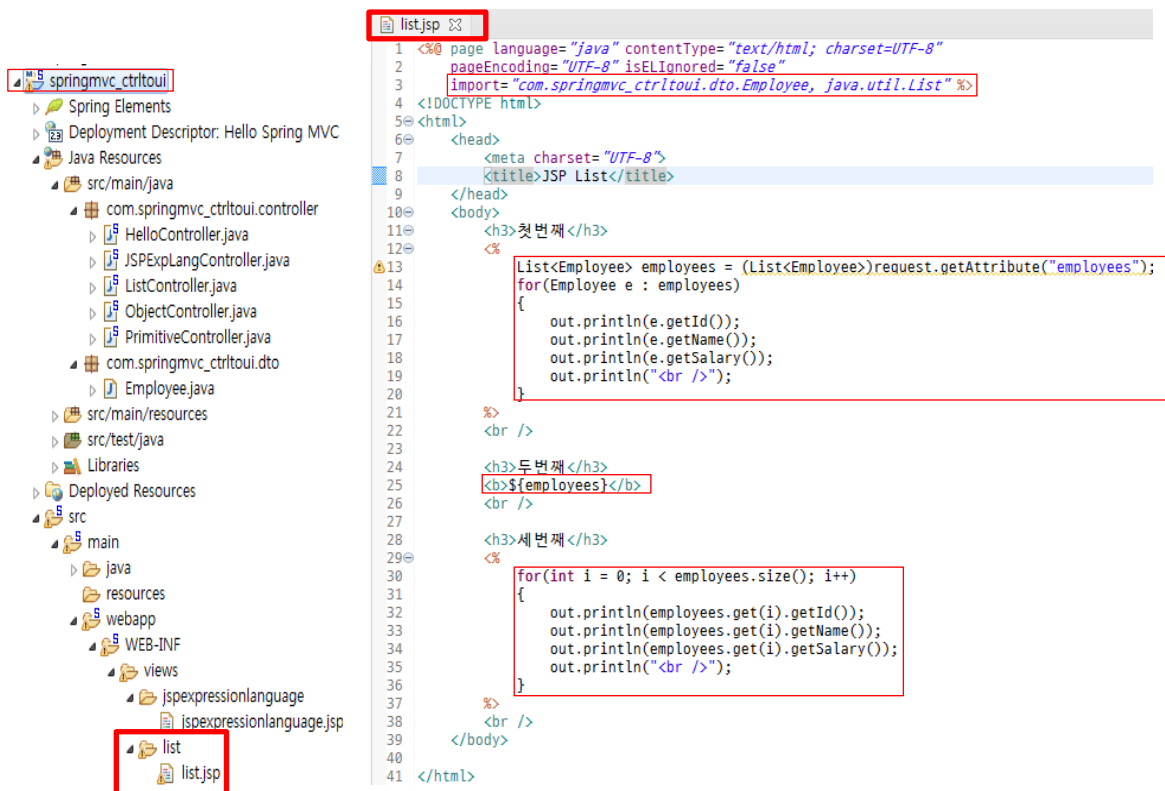


The screenshot shows the project structure on the left and the code for ListController.java on the right. The project structure includes a package named `com.springmvc_ctrltoui` with a sub-package `controller` containing `ListController.java`. The code for `ListController.java` is as follows:

```
1 ListController.java
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.servlet.ModelAndView;
6
7 import com.springmvc_ctrltoui.dto.Employee;
8
9 @Controller
10 public class ListController {
11
12     @RequestMapping("/list")
13     public ModelAndView list()
14     {
15         ModelAndView modelAndView = new ModelAndView();
16         modelAndView.setViewName("list/list");
17
18         Employee employee1 = new Employee();
19         employee1.setId(1234);
20         employee1.setName("woseven");
21         employee1.setSalary(20000);
22
23         Employee employee2 = new Employee();
24         employee2.setId(1122);
25         employee2.setName("windy72");
26         employee2.setSalary(10000);
27
28         Employee employee3 = new Employee();
29         employee3.setId(2233);
30         employee3.setName("keunbae");
31         employee3.setSalary(5000);
32
33         List<Employee> employees = new ArrayList<Employee>();
34         employees.add(employee1);
35         employees.add(employee2);
36         employees.add(employee3);
37
38         modelAndView.addObject("employees", employees);
39         return modelAndView;
40     }
41 }
```

## 8. 리스트(List) 뷰 만들기

### 가. JSTL 미적용



The screenshot shows the project structure on the left and the code for list.jsp on the right. The project structure includes a package named `com.springmvc_ctrltoui` with a sub-package `controller` containing `ListController.java`. The code for `list.jsp` is as follows:

```
1 list.jsp
2
3 <%@ page language="java" contentType="text/html; charset=UTF-8"
4   pageEncoding="UTF-8" isELIgnored="false"
5   import="com.springmvc_ctrltoui.dto.Employee, java.util.List" %>
6
7 <html>
8   <head>
9     <meta charset="UTF-8">
10    <title>JSP List</title>
11  </head>
12  <body>
13    <h3>첫 번째</h3>
14
15    <pre>
16      List<Employee> employees = (List<Employee>)request.getAttribute("employees");
17      for(Employee e : employees)
18      {
19          out.println(e.getId());
20          out.println(e.getName());
21          out.println(e.getSalary());
22          out.println("<br />");
23      }
24    </pre>
25
26    <h3>두 번째</h3>
27    <b>${employees}</b>
28
29    <h3>세 번째</h3>
30
31    <pre>
32      for(int i = 0; i < employees.size(); i++)
33      {
34          out.println(employees.get(i).getId());
35          out.println(employees.get(i).getName());
36          out.println(employees.get(i).getSalary());
37          out.println("<br />");
38      }
39    </pre>
40  </body>
41 </html>
```

## 나. JSTL 적용

- 의존성 정보 가져오기

The screenshot shows the Maven repository page for `jstl/jstl/1.2`. The URL `https://mvnrepository.com/artifact/jstl/jstl` is visible. The page displays the version `1.2.x | 1.2` and the repository `Central`. On the right, the 'Used By' section shows `287 artifacts`. Below, the 'Maven' tab shows the dependency XML snippet:

```
<!-- https://mvnrepository.com/artifact/jstl/jstl/1.2 -->
<dependency>
  <groupId>jstl</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

- pom.xml 수정

The screenshot shows an IDE with the `springmvc_ctrltoui/pom.xml` file open. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<groupId>com</groupId>
<artifactId>springmvc_ctrltoui</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>

<name>springmvc_ctrltoui Maven Webapp</name>
<url>http://www.example.com</url>

<properties>
  <springframework.version>4.3.6.RELEASE</springframework.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${springframework.version}</version>
  </dependency>
  <!-- JSTL 라이브러리 모듈 추가 -->
  <dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>
</dependencies>
```

The 'Maven Dependencies' section on the right shows the resolved dependencies, including `jstl-1.2.jar`.

- Update Maven 실행
- JSPEL용 컨트롤러 만들기

The screenshot shows an IDE with the `ListJspController.java` file open. The code is as follows:

```
package com.springmvc_ctrltoui.controller;

import java.util.ArrayList;

@Controller
@RequestMapping("/")
public class ListJspController {

    @RequestMapping("/listjsp")
    public ModelAndView list() {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("list/listjsp");
    }
}
```

- JSTL 용 뷰 페이지 만들기

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes `src/main/resources/webapp/WEB-INF/views/list`. The code editor displays the content of `list.jsp`, which includes JSP directives for page encoding and the JSTL core taglib, followed by an HTML table that iterates over an `employees` collection. The browser preview at the bottom shows the rendered output of the JSP page.

```
list.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8" isELIgnored="false" %>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 JSP Standard Tag Library
5 <!DOCTYPE html>
6 <html>
7   <head>
8     <meta charset="UTF-8">
9     <title>JSP JSPList</title>
10  </head>
11  <body>
12
13    <table>
14      <c:forEach var="employee" items="${employees}" varStatus="idx">
15        <tr>
16          <td>인덱스: ${idx.index + 1}</td>
17          <td>아이디: ${employee.id}</td>JSP 반복문 태그
18          <td>이름: ${employee.name}</td>
19          <td>연봉: ${employee.salary}</td>
20        </tr>
21      </c:forEach>
22    </table>
23
24  </body>
25
26 </html>
```

localhost:8080/springmvc\_ctrltoui/listjsp

인덱스: 1 아이디: 1234 이름: woseven 연봉: 20000.0

인덱스: 2 아이디: 1122 이름: windy72 연봉: 10000.0

인덱스: 3 아이디: 2233 이름: keunbae 연봉: 5000.0