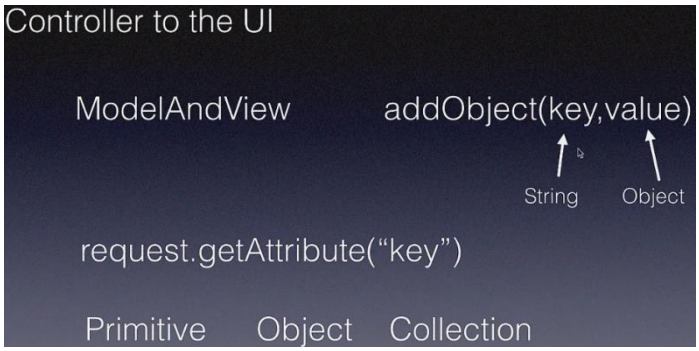
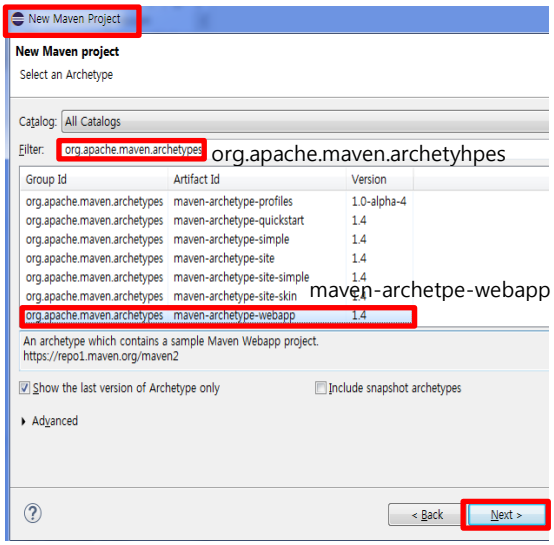
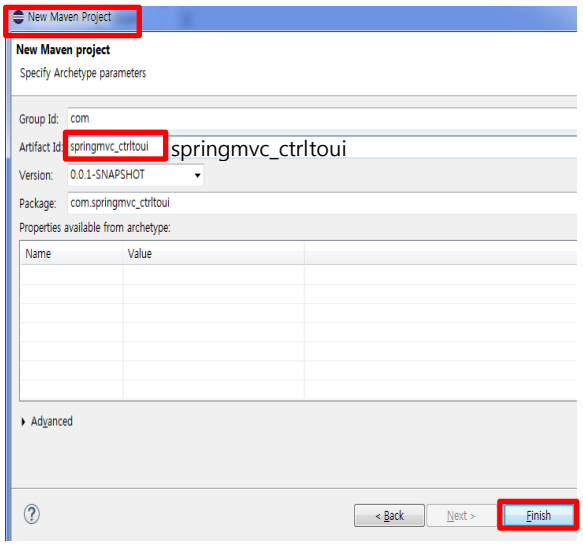
1. 소개

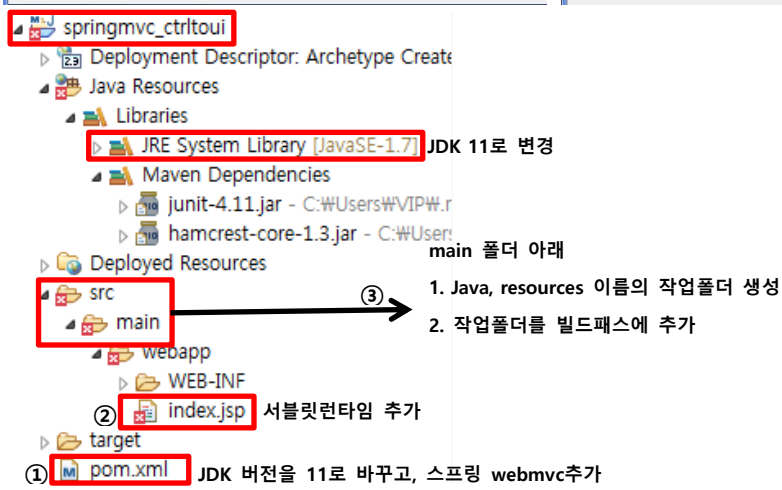


2. 메이븐 프로젝트 만들기

가. 프로젝트 생성



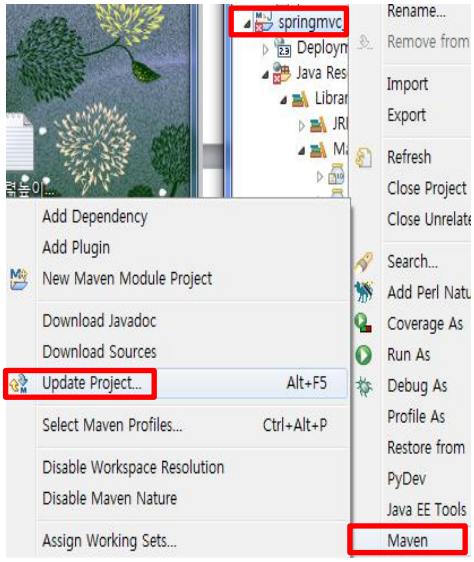


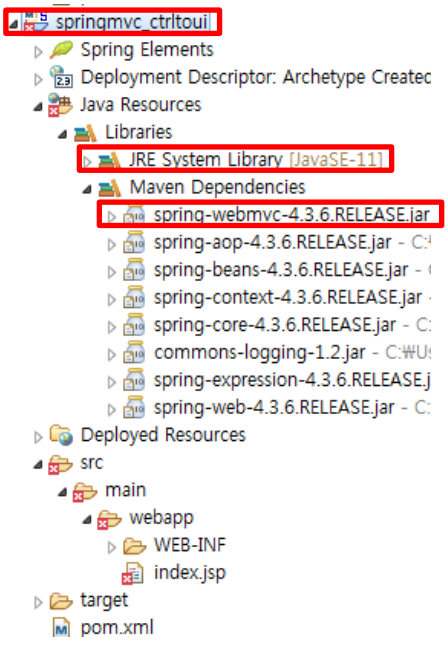


나. pom.xml 수정(1)

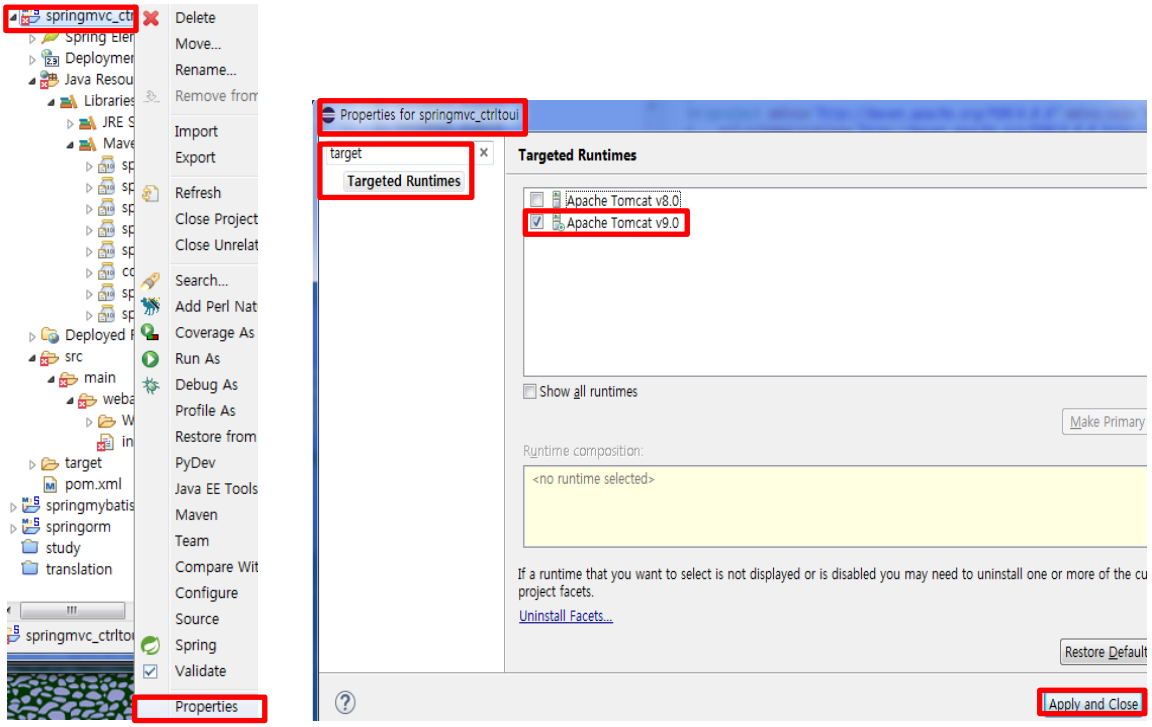
springmvc_ctrltoui/pom.xml

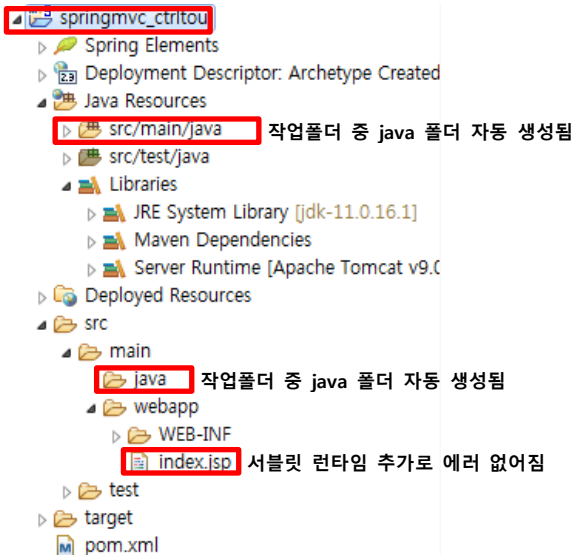
```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http.
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://mave
  <modelVersion>4.0.0</modelVersion>

  <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.ver
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>${springframework.version}</version>
        </dependency>
    </dependencies>
    <build>
        <pluginManagement>
            <plugins>
                <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-compiler-plugin</artifactId>
                    <version>3.8.0</version>
                    <configuration>
                        <source>11</source>
                        <target>11</target>
                    </configuration>
                </plugin>
            </plugins>
        </pluginManagement>
    </build>
</project>
```



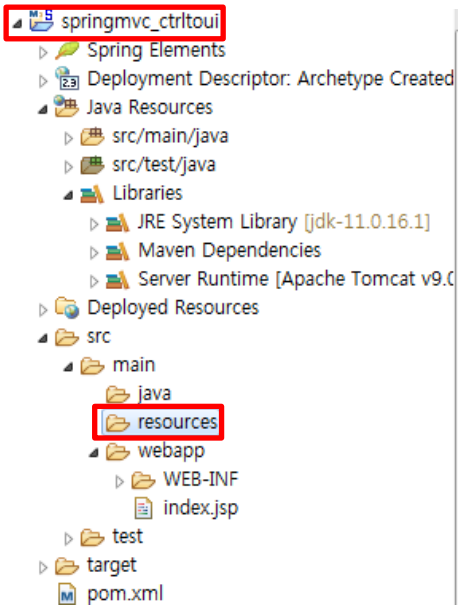


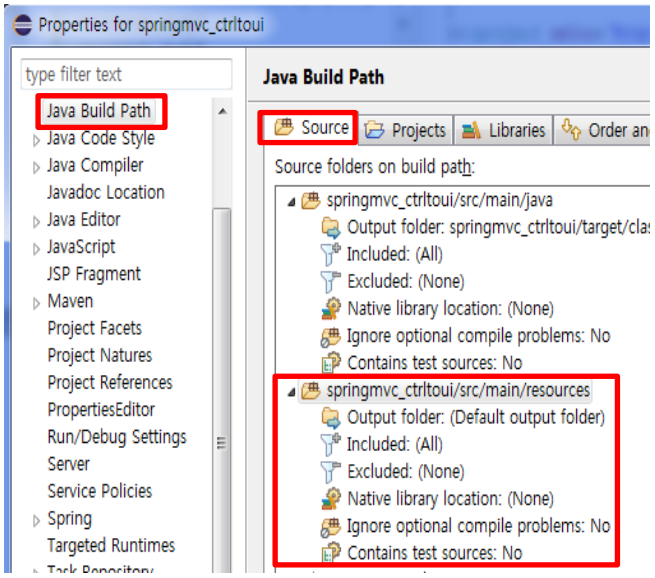
다. 서블릿 런타임 추가(2)

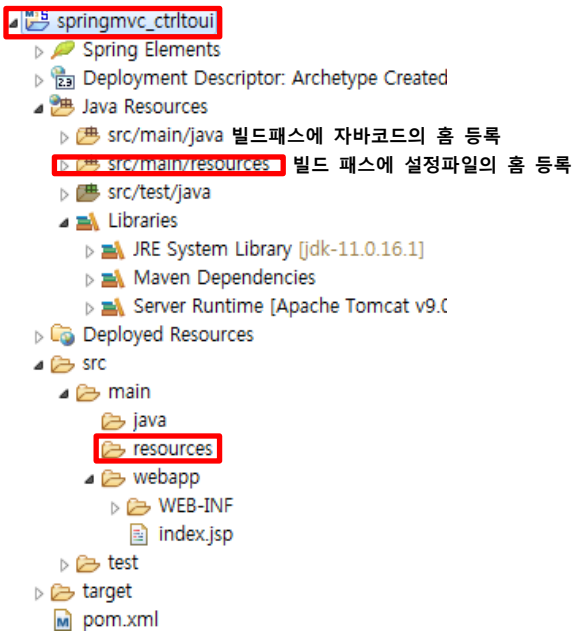

Delete
Move...
Rename...
Import
Export
Refresh
Close Project
Search...
Run As
Debug As
Profile As
Maven
Team
Configure
Source
Spring
Validate
Properties
Properties for springmvc_ctrltoui
target
Targeted Runtimes
Apache Tomcat v8.0
Apache Tomcat v9.0
Show all runtimes
Make Primary
Runtime composition:
<no runtime selected>
Uninstall Facets...
Restore Default
Apply and Close


springmvc_ctrltoui
Spring Elements
Deployment Descriptor: Archetype Created
Java Resources
src/main/java
작업폴더 중 java 폴더 자동 생성됨
src/test/java
Libraries
JRE System Library [jdk-11.0.16.1]
Maven Dependencies
Server Runtime [Apache Tomcat v9.0
Deployed Resources
src
main
java
작업폴더 중 java 폴더 자동 생성됨
webapp
WEB-INF
index.jsp
서블릿 런타임 추가로 에러 없어짐
test
target
pom.xml

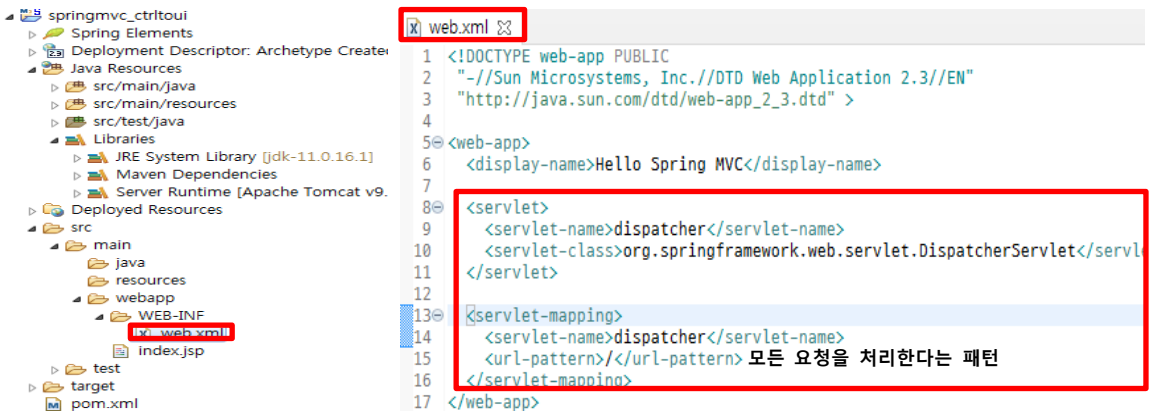
라. 리소스 폴더 생성하고 빌드 패스에 추가(3): resources







마. 디스패처 서블릿 설정: web.xml



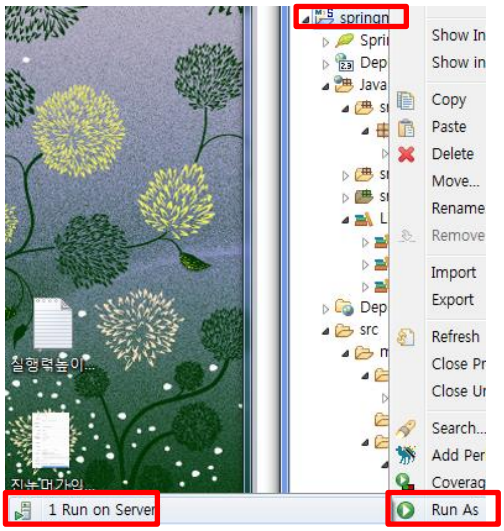
바. 스프링 설정 파일 만들기: WEB-INF 폴더 아래

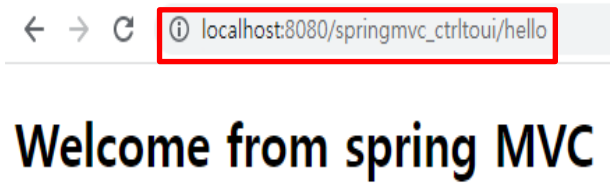
사. 뷰 리졸버 설정

아. 컨트롤러 클래스 만들기

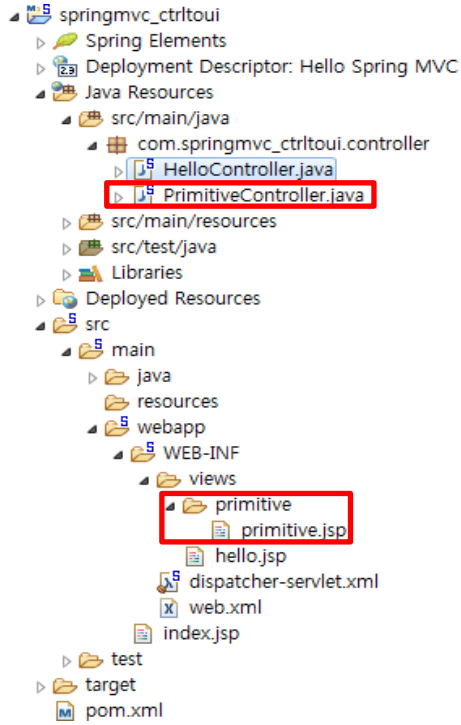
자. WEB-INF 아래 폴더 및 JSP 페이지 만들기

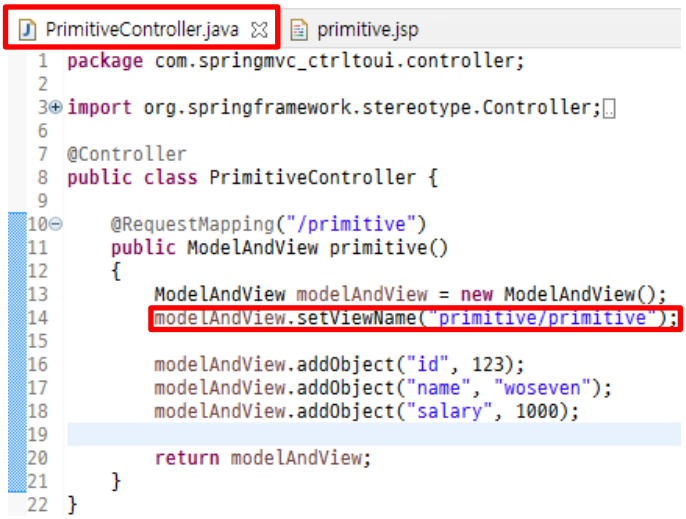
차. 실행





3. 원시타입(Primitive Type)을 UI에 보내기

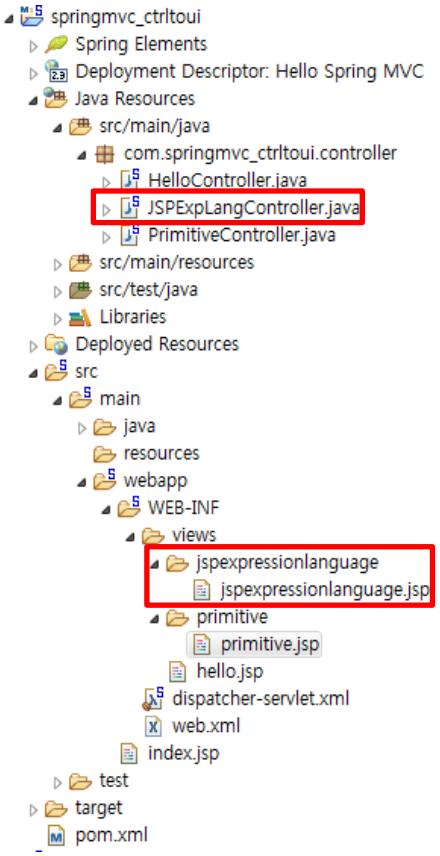




PrimitiveController.java | primitive.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Primitive</title>
    </head>
    <body>
        <%
            Integer id = (Integer)request.getAttribute("id");
            String name = (String)request.getAttribute("name");
            Integer salary = (Integer)request.getAttribute("salary");
            out.println("ID: " + id);
            out.println("Name: " + name);
            out.println("Salary: " + salary);
        %>
    </body>
</html>
```

4. JSP Expression Language 사용

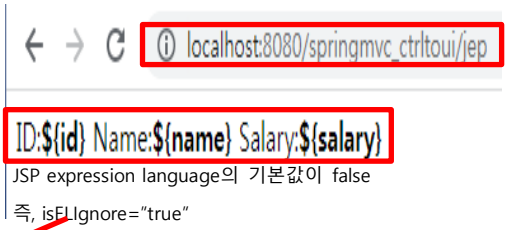


jspexpressionlanguage.jsp | JSPExpLangController.java

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JSP Expression Language</title>
    </head>
    <body>
        ID:<b>${id}</b>
        Name:<b>${name}</b>
        Salary:<b>${salary}</b>
    </body>
</html>
```

jspexpressionlanguage.jsp | JSPExpLangController.java

```java
package com.springmvc_ctrltoui.controller;

import org.springframework.stereotype.Controller;

@Controller
public class JSPExpLangController {

    @RequestMapping("/jep")
    public ModelAndView primitive()
    {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("jspexpressionlanguage/jspexpressionlanguage");

        modelAndView.addObject("id", 123);
        modelAndView.addObject("name", "woseven");
        modelAndView.addObject("salary", 1000);

        return modelAndView;
    }
}
```



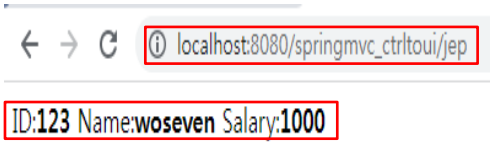
JSP expression language의 기본값이 false

즉, isELIgnore="true"

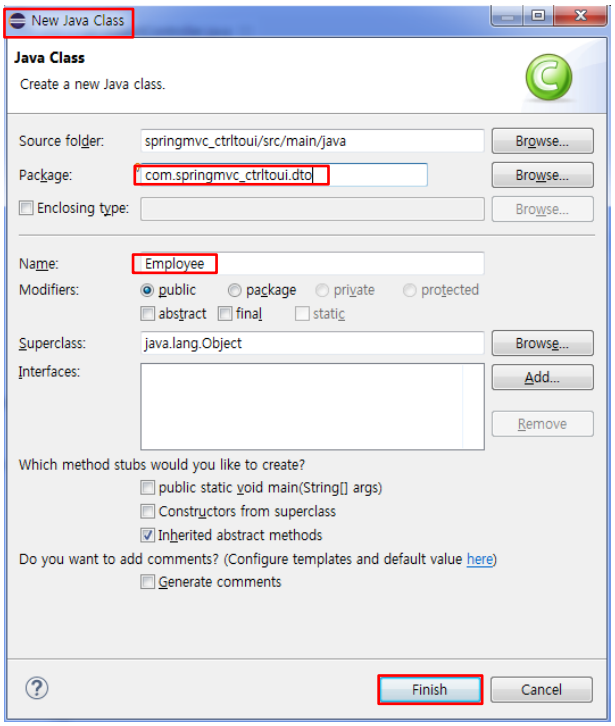
jspexpressionlanguage.jsp | JSPExpLangController.java | PrimitiveController.java

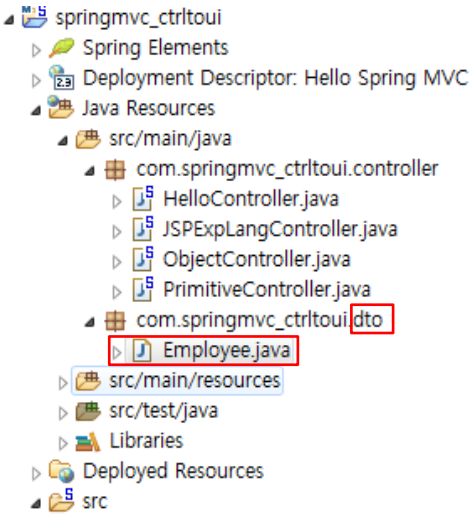
isELIgnore="false" 추가

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isELIgnored="false" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JSP Expression Language</title>
    </head>
    <body>
        ID:<b>${id}</b>
        Name:<b>${name}</b>
        Salary:<b>${salary}</b>
    </body>
</html>
```



5. 오브젝트(Object) 데이터를 UI에 보내기

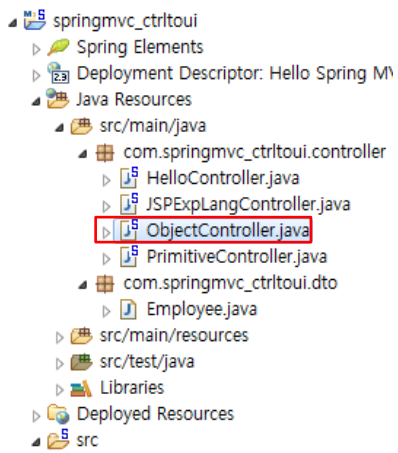




Employee.java

```java
package com.springmvc_ctrltoui.dto;

public class Employee {

    private int id;
    private String name;
    private double salary;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";
    }
}
```

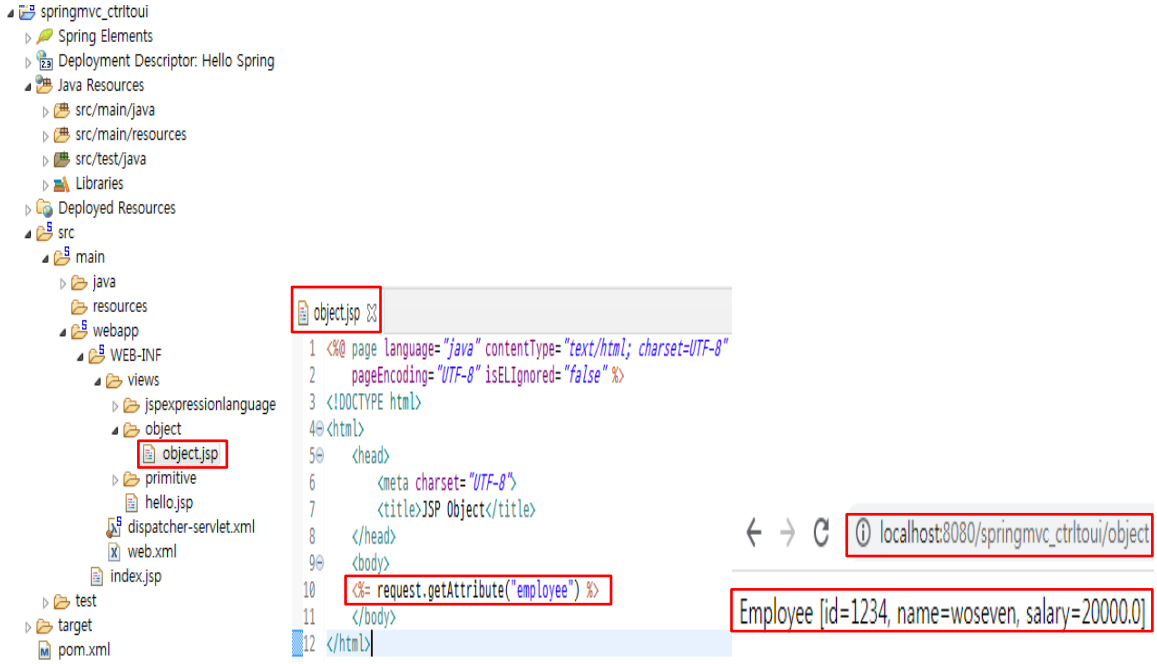
객체 내 필드를 getter로 하나씩 값을 가져오지 않고,
전체를 한번에 가져오려면 **Object의 toString()메소드**를
오버라이드 해야 함.

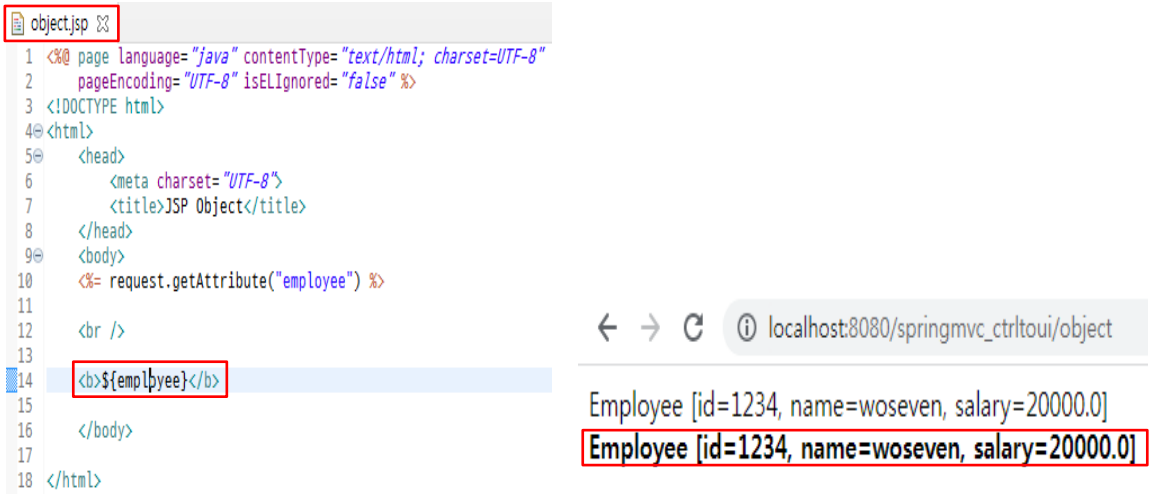


ObjectController.java

```java
package com.springmvc_ctrltoui.controller;

import org.springframework.stereotype.Controller;

@Controller
public class ObjectController {

    @RequestMapping("/object")
    public ModelAndView object()
    {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.setViewName("object/object");

        Employee employee = new Employee();
        employee.setId(1234);
        employee.setName("woseven");
        employee.setSalary(20000);

        modelAndView.addObject("employee", employee);

        return modelAndView;
    }
}
```

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
object.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Object</title>
</head>
<body>
<%= request.getAttribute("employee") %>
</body>
</html>
localhost:8080/springmvc_ctrltoui/object
Employee [id=1234, name=woseven, salary=20000.0]


object.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Object</title>
</head>
<body>
<%= request.getAttribute("employee") %>

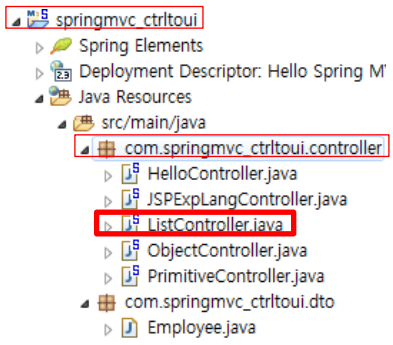
${employee}
</body>
</html>
localhost:8080/springmvc_ctrltoui/object
Employee [id=1234, name=woseven, salary=20000.0]
Employee [id=1234, name=woseven, salary=20000.0]


object.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Object</title>
</head>
<body>
<%= request.getAttribute("employee") %>

${employee}

ID: ${employee.id}
Name: ${employee.name}
Salary: ${employee.salary}
</body>
</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

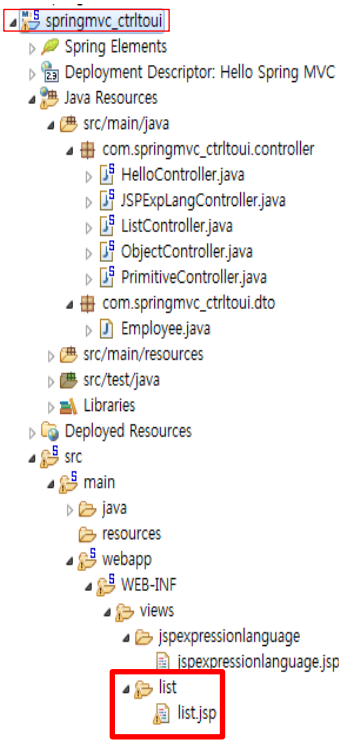


ListController.java

```java
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import com.springmvc_ctrltoui.dto.Employee;

@Controller
public class ListController {

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

        Employee employee1 = new Employee();
        employee1.setId(1234);
        employee1.setName("woseven");
        employee1.setSalary(20000);

        Employee employee2 = new Employee();
        employee2.setId(1122);
        employee2.setName("windy72");
        employee2.setSalary(10000);

        Employee employee3 = new Employee();
        employee3.setId(2233);
        employee3.setName("keunbae");
        employee3.setSalary(5000);

        List<Employee> employees = new ArrayList<Employee>();
        employees.add(employee1);
        employees.add(employee2);
        employees.add(employee3);

        modelAndView.addObject("employees", employees);
        return modelAndView;
    }
}
```

8. 리스트(List) 뷰 만들기

가. JSTL 미적용



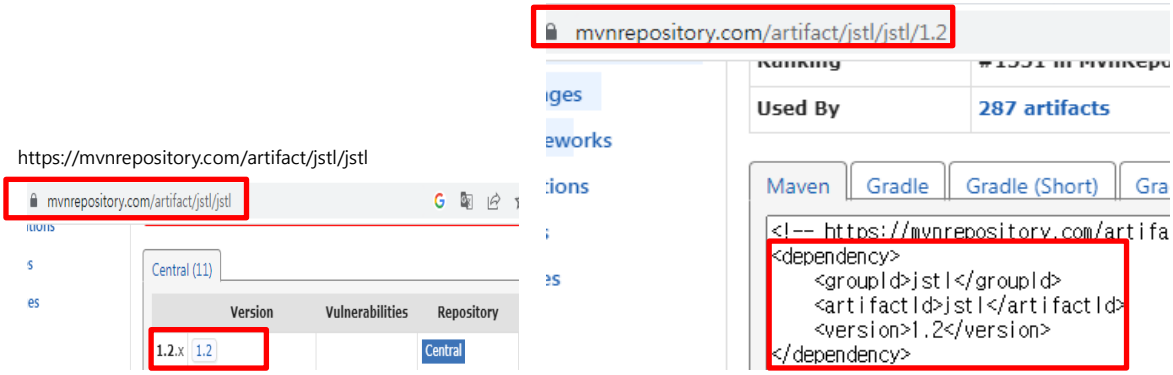
list.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isELIgnored="false"
    import="com.springmvc_ctrltoui.dto.Employee, java.util.List" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>JSP List</title>
    </head>
    <body>
        <h3>첫번째</h3>
        <%
            List<Employee> employees = (List<Employee>)request.getAttribute("employees");
            for(Employee e : employees)
            {
                out.println(e.getId());
                out.println(e.getName());
                out.println(e.getSalary());
                out.println("<br />");
            }
        %>
        <br />

        <h3>두번째</h3>
        <b>${employees}</b>
        <br />

        <h3>세번째</h3>
        <%
            for(int i = 0; i < employees.size(); i++)
            {
                out.println(employees.get(i).getId());
                out.println(employees.get(i).getName());
                out.println(employees.get(i).getSalary());
                out.println("<br />");
            }
        %>
        <br />
    </body>

</html>
```

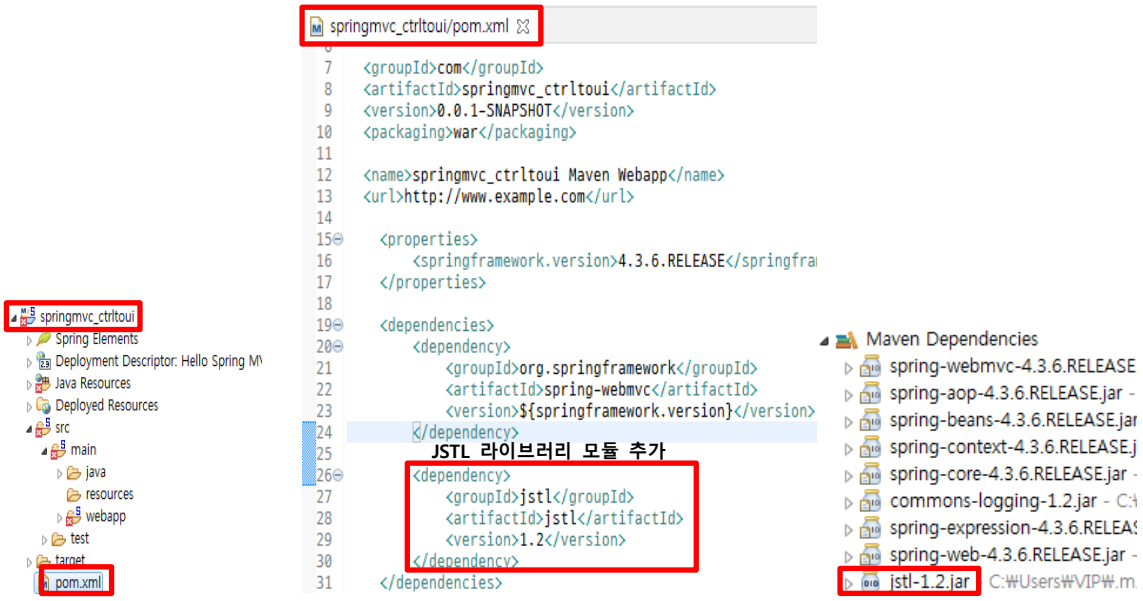
나. JSTL 적용

- 의존성 정보 가져오기

https://mvnrepository.com/artifact/jstl/jstl

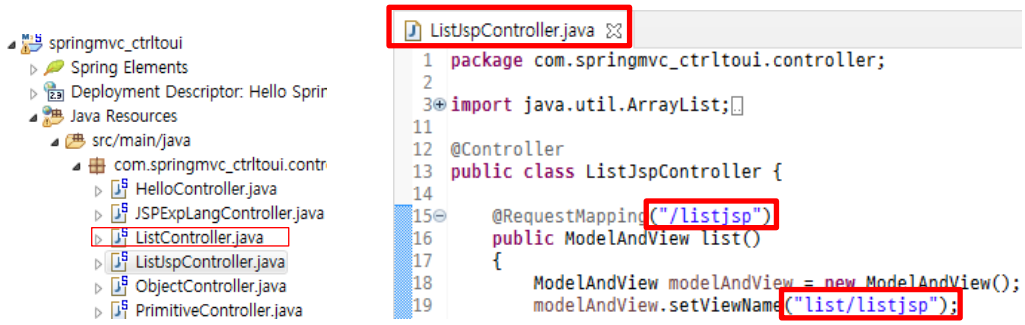


- pom.xml 수정

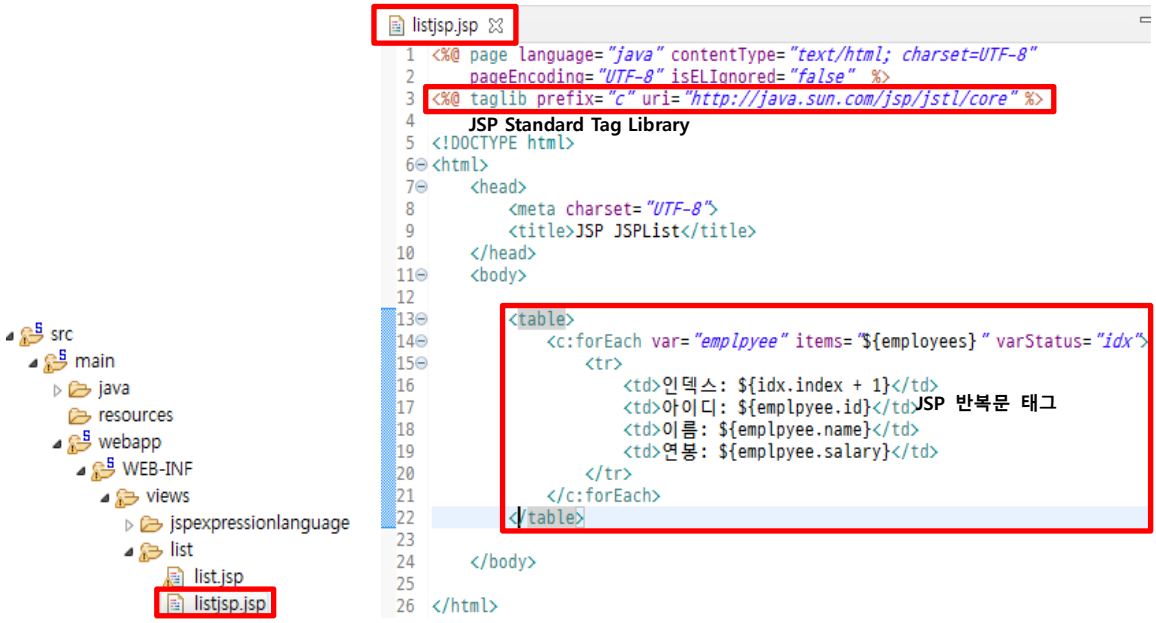


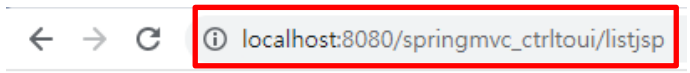
- Update Maven 실행

- JSPEL용 컨트롤러 만들기



- JSTL 용 뷰 페이지 만들기





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