Spring Framework

21. Spring MVC 어플리케이션 작성(1)

Tacademy

CONTENTS

학습 목표

- EL과 JSTL에 대하여 이해할 수 있습니다.
- Hello 컨트롤러 작성에 대하여 이해할 수 있습니다.
- 사용자 목록 조회 컨트롤러 작성에 대해 이해할 수 있습니다.

1. EL과 JSTL

EL(Expression Language)의 개요

EL과 JSTL(Java Standard Tag Library)을 사용하면 <% %>와 같은 스크립팅 태그를 JSP에서 없앨 수 있다.

EL 표현식은 중괄호({ })로 묶고 앞에 달러($)기호를 붙이며, 도트 연산자를 사용한다.

EL은 저장 객체의 출력을 단순화 하는 용도로 사용되므로,
저장 객체를 출력할 때도 스크립팅을 전혀 쓰지 않는다.

예를 들어, <%=request.getParameter("name")%>
대신에 ${param.name} 구문을 사용한다.

EL(Expression Language)의 개요

EL은 기본적으로 4가지 Scope(page, request, session,application)의 객체에 접근하여 출력을 처리한다.

EL에서는 해당값이 null이거나 공백일 경우에는 아무 내용도 표시하지 않고 에러도 발생하지 않는다.

EL은 JSP에서 기본으로 지원하고, JSTL은 따로 설치해야 한다.

EL(Expression Language)과 스크립팅 태그 비교

EL (Expression Language)	스크립팅 태그
${param.name}	<%=request.getParameter("name")%>
${greet}	<% String value = (String)request.getAttribute("greet"); out.println(value); %>
${user}	<% UserVO user = (UserVO)request.getAttribute("user"); out.println(user); %>

EL(Expression Language)과 스크립팅 태그 비교

EL (Expression Language)	스크립팅 태그
${user.name}	<% UserVO user = (UserVO)request.getAttribute("user"); out.println(user.getName()); %>
${sessionScope.user.name}	<% UserVO user = (UserVO)session.getAttribute("user"); out.println(user.getName()); %>

JSTL(Java Standard Tag Library) 개요

JSTL이란 JSP 표준 라이브러리(JSP Standard Tag Library)

JSTL은 JSP에서 스크립팅을 사용하지 않으면서,
루프를 돌리거나 조건문을 실행할 수 있도록 해줌

JSP에서 자주 사용하는 기능(반복과 조건, 데이터 관리 포맷, XML 조작, 데이터베이스 액세스)을 구현해 놓은 Custom Tag Library 모음

JSTL은 EL(Expression Language)를 사용하여 표현

JSTL은 request, response, pageContext, application과 같은
JSP 내장 객체(Implicit Object)에 쉽게 접근할 수 있음

JSTL(Java Standard Tag Library) 개요

◉ JSTL은 아래와 같이 5개의 Library가 있음

라이브러리	기능	URL 식별자	접두어
코어(core)	일반 프로그램 언어에서 제공하는 변수선언, 조건/제어/반복문 등의 기능을 제공한다.	http://java.sun.com/jsp/jstl/core	C
포맷팅 (formatting)	숫자, 날짜, 시간을 포맷팅 하는 기능과 국제화, 다국어 지원 기능을 제공한다.	http://java.sun.com/jsp/jstl/fmt	fmt
함수 (function)	문자열을 처리하는 함수를 제공한다	http://java.sun.com/jsp/jstl/functions	fn

JSTL(Java Standard Tag Library) 개요

◉ JSTL은 아래와 같이 5개의 Library가 있음

라이브러리	기능	URL 식별자	접두어
데이터 베이스 (database)	데이터베이스의 데이터를 입력/수정/삭제/조회하는 기능을 제공한다.	http://java.sun.com/jsp/jstl/sql	sql
XML처리 (xml)	XML 문서를 처리할 때 필요한 기능을 제공한다.	http://java.sun.com/jsp/jstl/xml	x

JSTL(Java Standard Tag Library) 개요

◉ JSTL의 사용

```
<!-- 선언부 -->
<%@ taglib prefix="c"    uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<!-- 사용-->
<c:set var="hello" value="Hello" />
${hello}
```

JSTL(Java Standard Tag Library) Core 태그

- JSTL 태그 라이브러리 중에 가장 많이 사용하는 태그이다.
- <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>로 선언해야 함

라이브러리	태그	설명
변수지원	set	JSP에서 사용될 변수를 설정한다.
	remove	설정할 변수를 제거한다.
흐름제어	if	조건에 따라 내부 코드를 수행한다.
	choose	다중 조건을 처리할 때 사용한다.
	forEach	컬렉션이나 Map의 각 항목을 처리할 때 사용한다.
	forTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용한다.

JSTL(Java Standard Tag Library) Core 태그

- JSTL 태그 라이브러리 중에 가장 많이 사용하는 태그이다.
- <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>로 선언해야 함

라이브러리	태그	설명
URL 처리	import	URL을 사용하여 다른 자원의 결과를 삽입한다.
	redirect	지정한 경로로 리다이렉트 한다.
	url	URL을 재작성한다.
기타태크	catch	익셉션 처리에 사용된다.
	out	JspWriter에 내용을 알맞게 처리한 후 출력한다.

JSTL 라이브러리 검색 및 설치


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

2. Hello 컨트롤러 작성

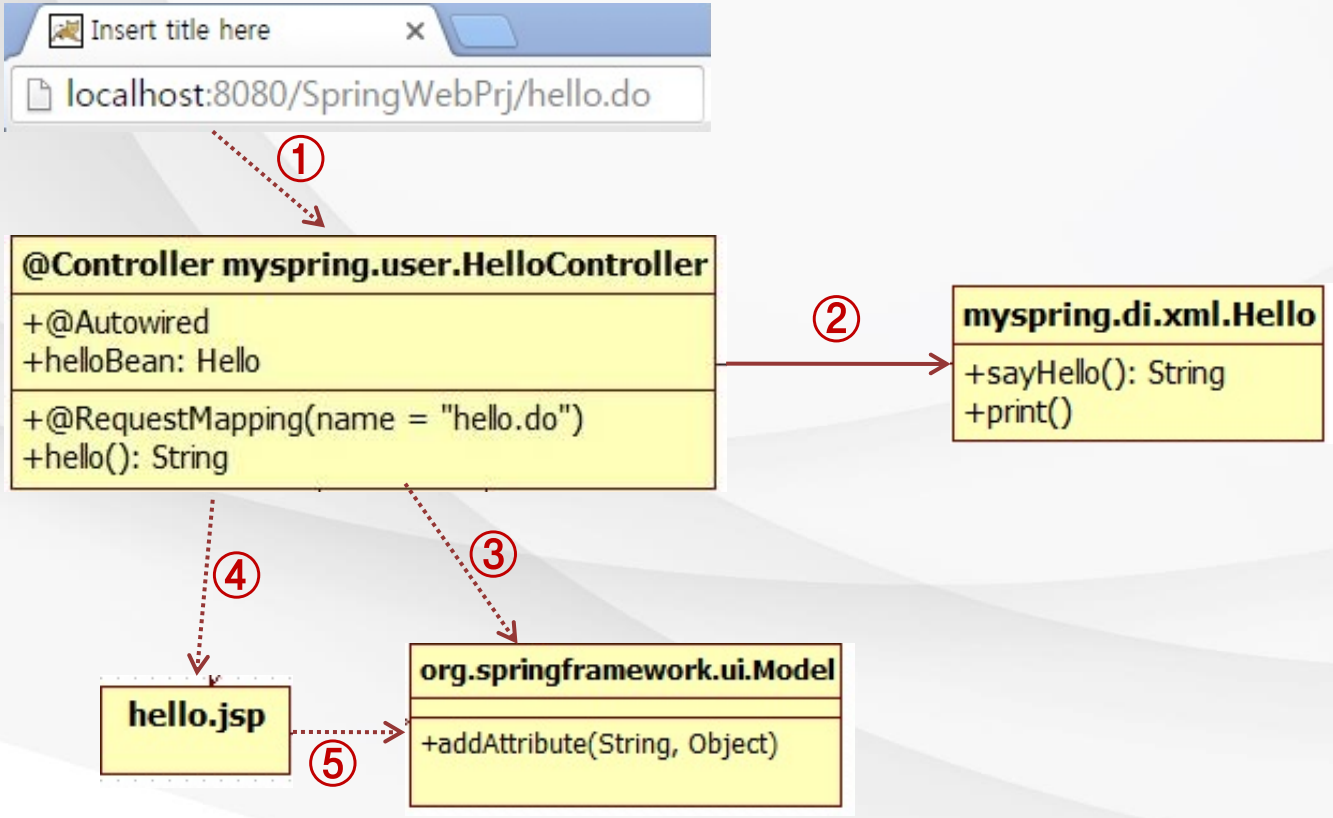
Controller 와 JSP 구현 절차

1. 클라이언트의 요청을 처리할 POJO 형태의 HelloController 클래스를 작성
2. Controller 클래스에 @Controller 어노테이션을 선언
3. 요청을 처리할 메서드를 작성하고 @RequestMapping 어노테이션을 선언
4. JSP를 이용한 View 영역의 코드를 작성
5. Browser 상에서 JSP를 실행

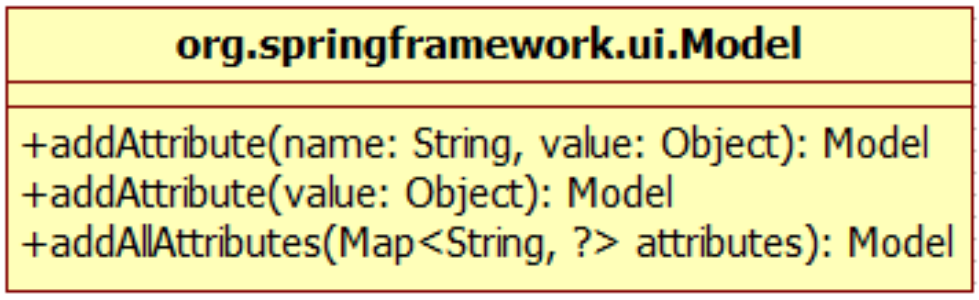
Controller를 위한 핵심 어노테이션(Annotation)

구성요소	설명
@Controller	Controller 클래스 정의
@RequestMapping	HTTP 요청 URL을 처리할 Controller 메소드 정의

Controller와JSP 호출 순서



View에 데이터를 전달하는 Model 클래스



- Controller에서 Service를 호출한 결과를 받아서
  view에게 전달하기 위해, 전달받은 결과를 Model 객체에 저장
- Model addAttribute(string name, Object value)
  : value 객체를 name 이름으로 저장하고,
  view 코드에서는 name으로 지정한 이름을 통해서 value를 사용

Controller 와 JSP 구현

*HelloController.java

```
package my.spring.controller;

@Controller
public class HelloController {
    @Autowired
    private Hello helloBean;

    @RequestMapping("/hello.do")
    public String hello(Model model) {
        String msg = helloBean.sayHello();
        model.addAttribute("greet", msg);
        return "hello.jsp";
    }
```

*Hello.java

```
package myspring.di.xml;
public class Hello {
    String name;
    Printer printer;

    public Hello() {}

    public String sayHello() {
        return "Hello " + name;
    }
```

2. Hello 컨트롤러 작성

Controller와 JSP 구현

hello.jsp

```
<%@ page language="java" co
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W
<html>
<head>
<meta http-equiv="Content-Ty
<title>Insert title here</title>
</head>
<body>
    ${greet}
</body>
</html>
```

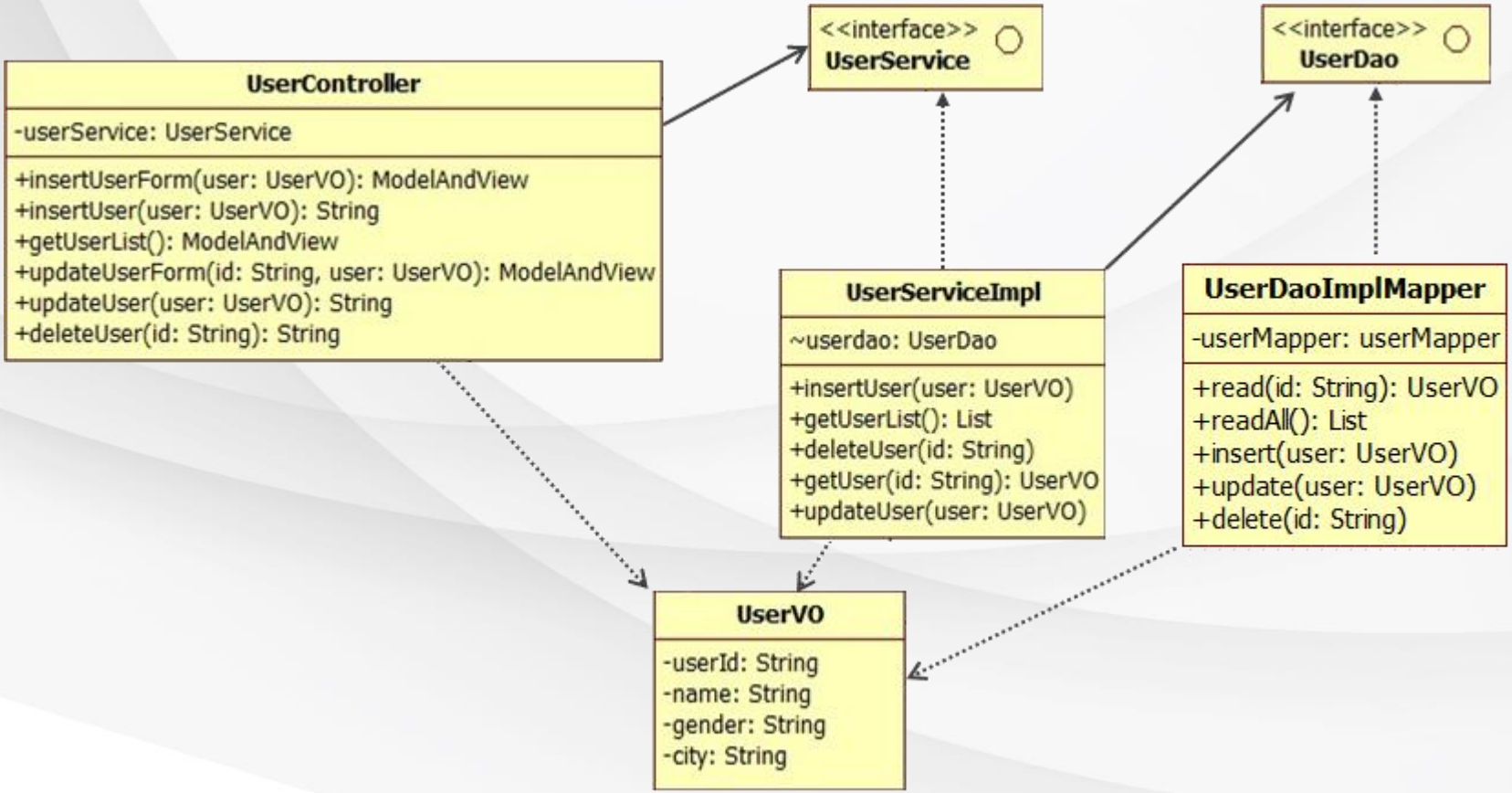
beans.xml

```
<context:component-scan
    base-package="myspring.user,myspring.aop.annot" />
```

3. 사용자 목록 조회 컨트롤러 작성

3. 사용자 목록 조회 컨트롤러 작성

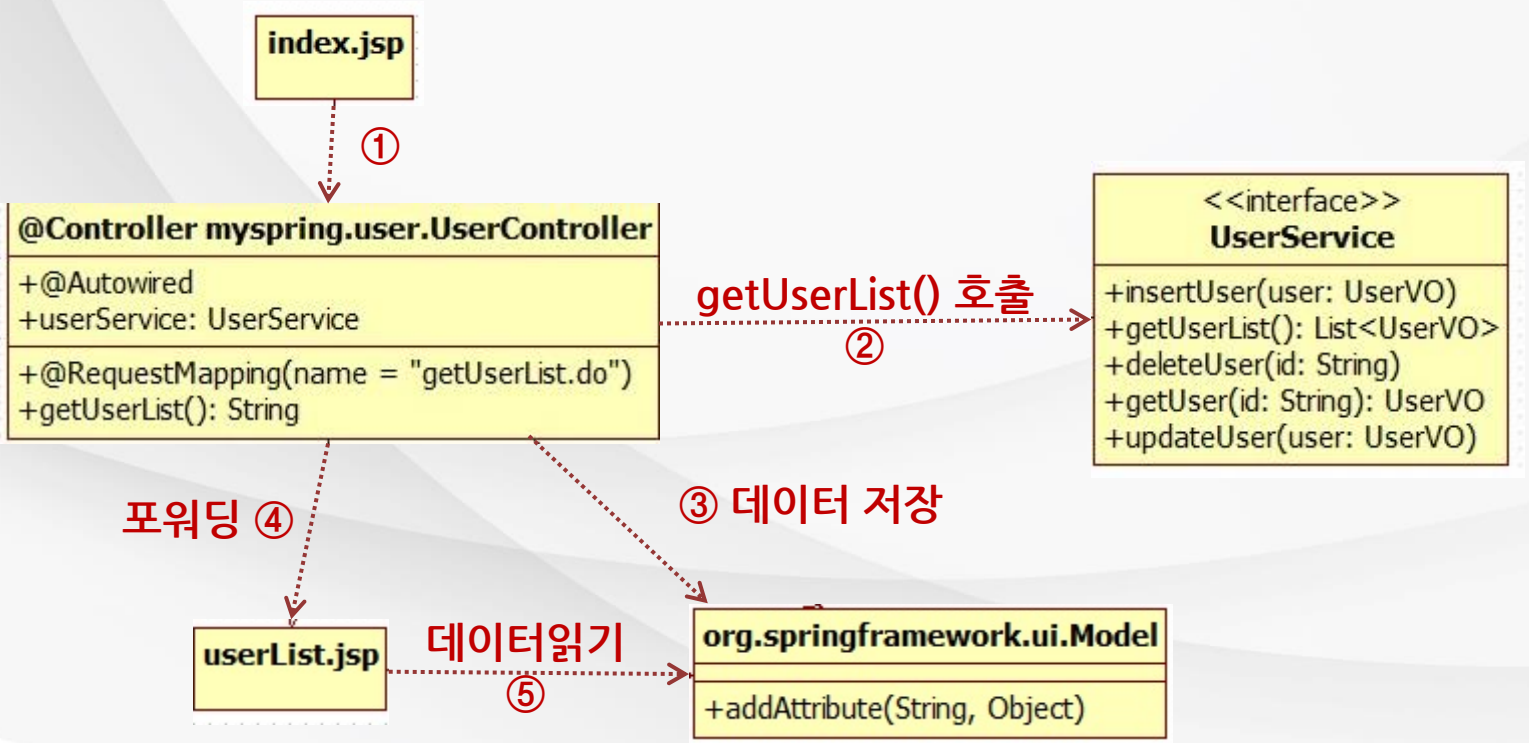
사용자 관리 프로젝트 클래스 설계



사용자 목록조회 : Controller와 JSP 구현 절차

1. 클라이언트의 요청을 처리할 POJO 형태의 UserController클래스를 작성
2. Controller 클래스에 @Controller 어노테이션을 선언
3. 사용자 목록을 조회하는 getUserList() 메서드를 작성하고 @RequestMapping 어노테이션을 선언
4. userList.jsp 페이지에 View 영역의 코드를 작성
5. Browser 상에서 JSP를 실행

사용자 목록조회 : Controller와JSP 호출 순서



3. 사용자 목록 조회 컨트롤러 작성

사용자 목록조회 : Controller와 JSP 구현

1. index.jsp

*index.jsp

```jsp
<%response.sendRedirect("getUserList.do");%>
```

2. UserController.java

*UserController.java

```java
package myspring.user.controller;

@Controller
public class UserController {
    @Autowired
    private UserService userService;

    @RequestMapping("/getUserList.do")
    public String getUserList(Model model) {
        List<UserVO> userList = userService.getUserList();
        model.addAttribute("userList", userList);
        return "userList.jsp";
    }
}
```

3. beans.xml

beans.xml

```xml
<context:component-scan
    base-package="myspring.user,myspring.aop.annot" />
```

사용자 목록조회 : Controller와 JSP 구현

4. userList.jsp

userList.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<tbody>
    <c:forEach var="user" items="${userList}">
        <tr>
            <td>${user.userId}</td>
            <td>${user.name}</td>
            <td>${user.gender}</td>
            <td>${user.city}</td>
            <td>수정</td>
            <td>삭제</td>
        </tr>
    </c:forEach>
</tbody>
```

사용자 목록조회 : 결과 화면

학습정리

지금까지 [Spring MVC 어플리케이션 작성(1)]에 대해서 살펴보았습니다.

EL과 JSTL

- EL(Expression Language)의 개요
- JSTL(Java Standard Tag Library)의 개요 및 설치

Hello 컨트롤러 작성

- 기존의 Hello Bean을 호출하는 HelloController 클래스 작성
- @Controller , @RequestMapping 어노테이션의 사용

사용자 목록 조회 컨트롤러 작성

- UserService Bean을 호출하여 사용자 목록을 조회하는 UserController 클래스 작성