

Spring Framework

22. Spring MVC 어플리케이션 작성(2)

CONTENTS

1

특정 사용자 조회 기능 구현

2

사용자 등록화면 기능 구현

3

사용자 등록 기능 구현

학습 목표

- 특정 사용자 조회 기능 구현에 대하여 이해할 수 있습니다.

- 사용자 등록화면 기능 구현에 대하여 이해할 수 있습니다.

- 사용자 등록 기능 구현에 대해 이해할 수 있습니다.

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, warm-toned bokeh lights. A semi-transparent dark banner is at the bottom, containing a yellow graphic element and text.

1. 특정 사용자 조회 기능 구현

■ ViewResolver 설정

- ◉ ViewResolver는 Controller의 실행 결과를 어떤 view에서 보여줄 것인지 결정하는 기능을 제공
- ◉ InternalResourceViewResolver는 JSP를 사용하여 view를 생성

```
*beans-web.xml
<!-- 아래 주석은 Controller에서 포워딩 되는 .jsp 확장자를 생략할 수 있다. -->
<bean id="viewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/" />
  <property name="suffix" value=".jsp" />
</bean>
```

- ◉ **prefix** - Controller가 리턴한 view 이름 앞에 붙을 접두어
- ◉ **suffix** - Controller가 리턴한 view 이름 뒤에 붙을 확장자
- ◉ Controller가 처리 결과를 보여줄 view의 이름으로 "hello"를 리턴 했다면 InternalResourceViewResolver에 의해 사용되는 view는 "/hello.jsp"가 됨

I 특정 사용자 조회 : Controller와 JSP 구현 절차

1 사용자 목록을 조회하는 `getUser(String id)` 메서드를 작성하고 `@RequestMapping`과 `@RequestParam` 어노테이션을 선언

2 `userList.jsp` 페이지를 수정

3 `userInfo.jsp` 페이지에 View 영역의 코드를 작성

4 Browser 상에서 JSP를 실행

■ Controller 를 위한 핵심 어노테이션(Annotation)

구성요소	설명
@RequestParam	HTTP 요청에 포함된 파라미터 참조 시 사용

```
userList.jsp
<c:forEach var="user" items="${userList}">
  <tr>
    <td>
      <a href="getUser.do?id=${user.userId}">
        ${user.userId}
      </a>
    </td>
    <td>${user.name}</td>
    <td>${user.gender}</td>
    <td>${user.city}</td>
  </tr>
</c:forEach>
```

```
UserController.java
package myspring.user.controller;

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

    @RequestMapping("/getUser.do")
    public ModelAndView getUser(@RequestParam String id) {
        UserVO user = userService.getUser(id);
        return new ModelAndView("userInfo", "user", user);
    }
}
```

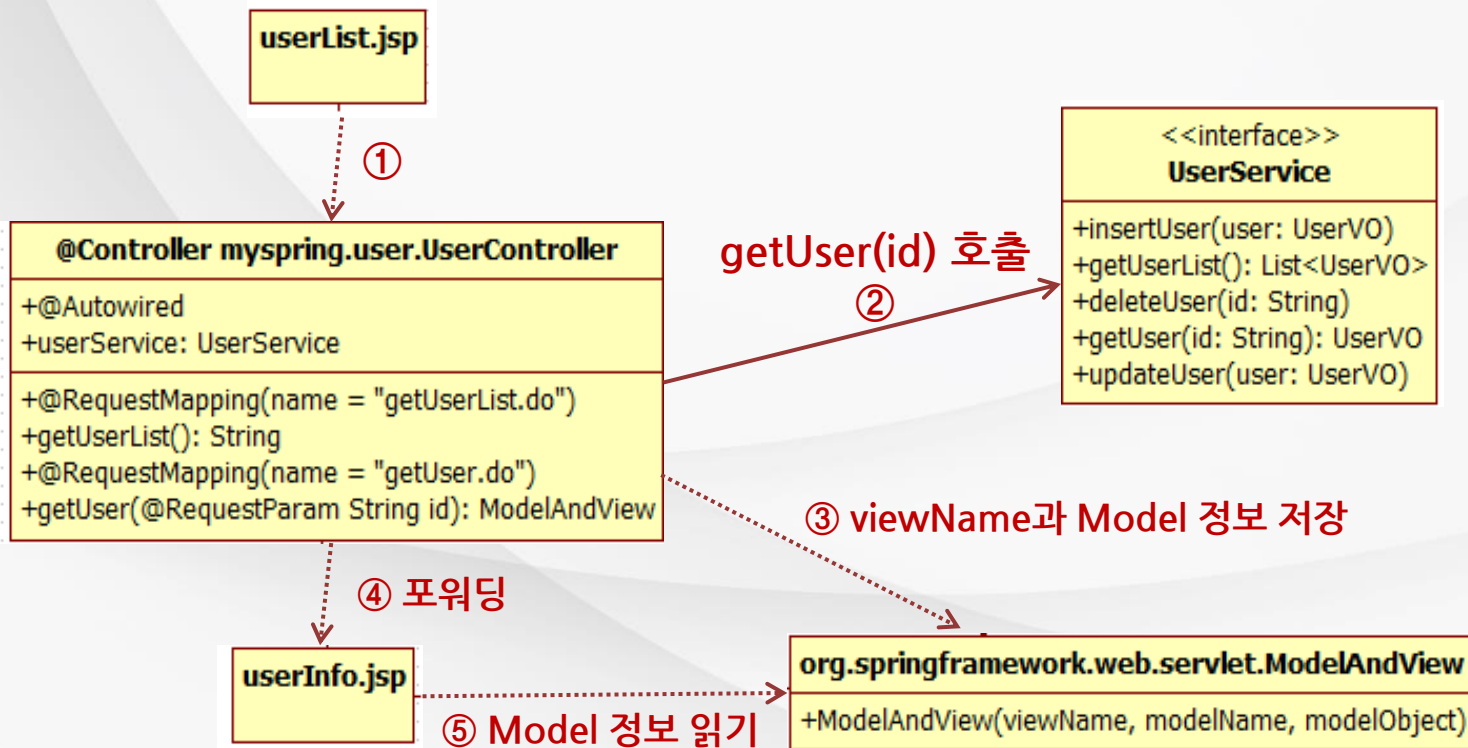
■ View에 데이터와 화면정보를 전달하는 ModelAndView 클래스

org.springframework.web.servlet.ModelAndView

```
+ModelAndView(viewName: String, modelName: String, modelObject: Object)
+ModelAndView(viewName: String)
+addObject(attrName: String, attrValue: Object)
+getModel(): Map<String, Object>
+getView(): View
+setView(view: View)
+setViewName(viewName: String)
```

- ◉ Controller에서 Service를 호출한 결과를 받아서 View에게 전달하기 위해, 전달받은 데이터와 화면정보를 ModelAndView 객체에 저장
- ◉ ModelAndView 클래스의 생성자나, setViewName() 메서드를 이용해서 View 이름을 지정할 수 있음
- ◉ addObject(String name, Object value) 메서드를 이용해 View에 전달할 데이터를 저장할 수 있음

■ 특정 사용자 조회 : Controller 와 JSP 호출 순서



■ 특정 사용자 조회 : Controller와 JSP 구현

1. userList.jsp

```
userList.jsp
<c:forEach var="user" items="${userList}">
  <tr>
    <td>
      <a href="getUser.do?id=${user.userId}">
        ${user.userId}
      </a>
    </td>
    <td>${user.name}</td>
    <td>${user.gender}</td>
    <td>${user.city}</td>
  </tr>
</c:forEach>
```

사용자 목록

아이디	이름	성별	거주지		
gildong	홍길동2	남2	경기2	수정	삭제
polar	연아2	여2	경기2	수정	삭제

[사용자 등록](#)

■ 특정 사용자 조회 : Controller와 JSP 구현

2. UserController.java

```
UserController.java ✕  
  
package myspring.user.controller;  
  
@Controller  
public class UserController {  
    @Autowired  
    private UserService userService;  
  
    @RequestMapping("/getUser.do")  
    public ModelAndView getUser(@RequestParam String id) {  
        UserVO user = userService.getUser(id);  
        return new ModelAndView("userInfo","user",user);  
    }  
}
```

■ 특정 사용자 조회 : Controller와 JSP 구현

3. userInfo.jsp

```
*userInfo.jsp
<body>
  <div class="container">
    <h2 class="text-center">사용자 상세정보</h2>
    <table class="table table-bordered table table-hover">
      <tr><td>아이디 :</td><td>${user.userId}</td></tr>
      <tr><td>이름 :</td><td>${user.name}</td></tr>
      <tr><td>성별 :</td><td>${user.gender}</td></tr>
      <tr><td>거주지 :</td><td>${user.city}</td></tr>
    </table>
  </div>
</body>
```

■ 특정 사용자 조회 : 결과 화면

사용자 목록

아이디	이름	성별	거주지		
gildong	홍길동2	남2	경기2	수정	삭제
polar	연아2	여2	경기2	수정	삭제

[사용자 등록](#)



사용자 상세정보

아이디 :	gildong
이름 :	홍길동2
성별 :	남2
거주지 :	경기2

AOP는

2. 사용자 등록화면 기능 구현

■ 사용자 정보 등록 화면 : Controller 와 JSP 구현 절차

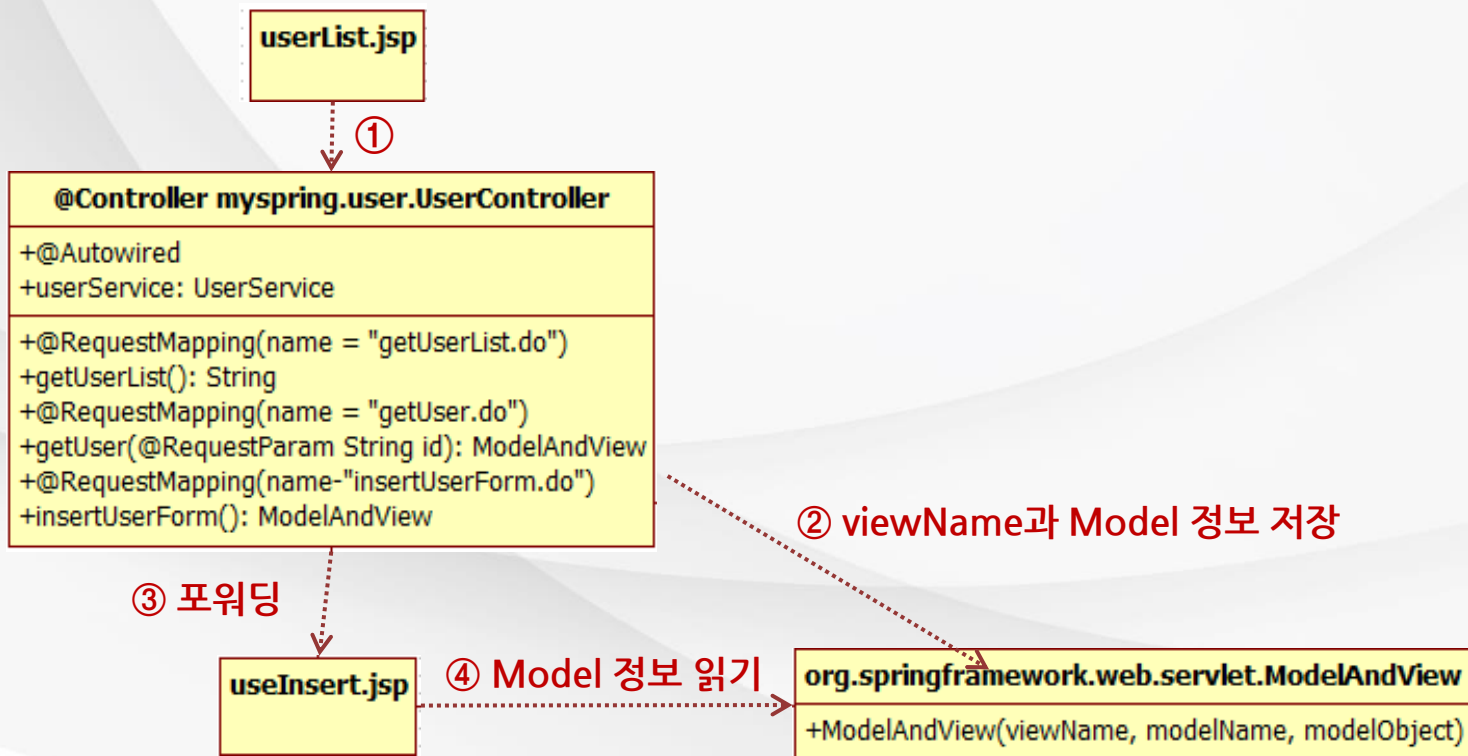
1 사용자 정보를 등록하는 화면을 포워딩 해주는 insertUserForm() 메서드를 작성하고 @RequestMapping 어노테이션을 선언

2 userList.jsp 페이지를 수정

3 userInsert.jsp 페이지에 View 영역의 코드를 작성

4 Browser 상에서 JSP를 실행

■ 사용자 정보 등록 화면 : Controller 와 JSP 호출 순서



■ 사용자 정보 등록 화면 : Controller와 JSP 구현

1. userList.jsp

```
*userList.jsp
<c:forEach var="user" items=${userList} ">
    .....
</c:forEach>
<tr>
    <td colspan="7">
        <a href="insertUserForm.do">사용자 등록</a>
    </td>
</tr>
```

사용자 목록

아이디	이름	성별	거주지		
gildong	홍길동2	남2	경기2	수정	삭제
polar	연아2	여2	경기2	수정	삭제
사용자 등록					

■ 사용자 정보 등록 화면 : Controller와 JSP 구현

2. UserController.java

```
UserController.java ✕  
@Controller  
public class UserController {  
    @RequestMapping("/insertUserForm.do")  
    public ModelAndView insertUserForm() {  
        List<String> genderList = new ArrayList<String>();  
        genderList.add("남");  
        genderList.add("여");  
        List<String> cityList = new ArrayList<String>();  
        cityList.add("서울");  
        cityList.add("부산");  
        cityList.add("대구");  
        cityList.add("제주");  
        Map<String, List<String>> map = new HashMap<>();  
        map.put("genderList", genderList);  
        map.put("cityList", cityList);  
        return new ModelAndView("userInsert", "map", map);  
    }  
}
```

성별 :	<input type="radio"/> 남 <input type="radio"/> 여
거주지 :	<div>서울 ▾ 서울 부산 대구 제주</div> <div>등록</div>
	사용자 목록보기

■ 사용자 정보 등록 화면 : Controller와 JSP 구현

3. userInsert.jsp

userInsert.jsp

```
<div class="container">
  <h2 class="text-center">사용자 정보 등록</h2>
  .....
  <tr>
    <td>성별 :</td>
    <td><c:forEach var="genderName" items="${map.genderList}">
      <input type="radio" name="gender" value="${genderName}">${genderName}
    </c:forEach></td>
  </tr>
  <tr>
    <td>거주지 :</td>
    <td><select name="city">
      <c:forEach var="cityName" items="${map.cityList}">
        <option value="${cityName}">${cityName}</option>
      </c:forEach>
    </select></td>
  </tr>
</div>
```

UserController.java

```
Map<String, List<String>> map = new HashMap<>();
map.put("genderList", genderList);
map.put("cityList", cityList);
return new ModelAndView("userInsert.jsp", "map", map);
```

2. 사용자 등록화면 기능 구현

■ 사용자 정보 등록 화면 : 결과 화면

사용자 목록					
아이디	이름	성별	거주지		
gildong	홍길동2	남2	경기2	수정	삭제
polar	연아2	여2	경기2	수정	삭제
사용자 등록					



사용자 정보 등록	
아이디 :	<input type="text"/>
이름 :	<input type="text"/>
성별 :	<input type="radio"/> 남 <input type="radio"/> 여
거주지 :	<div>서울 ▼ 서울 부산 대구 제주</div>
	<input type="button" value="등록"/>
	사용자 목록보기



3. 사용자 등록 기능 구현

■ 사용자 정보 등록 : Controller 와 JSP 구현 절차

1 사용자 정보를 등록하는 insertUser(@ModelAttribute UserVO userVO) 메서드를 작성하고 @RequestMapping과 @ModelAttribute 어노테이션을 선언

: 등록 후에 목록 조회가 redirect 되도록 하여, 등록된 사용자 정보를 확인할 수 있도록 해야 함

2 userInsert.jsp 페이지에 View 영역의 코드를 작성

3 Browser 상에서 JSP를 실행

Controller 를 위한 핵심 어노테이션(Annotation)

구성요소	설명
@ModelAttribute	HTTP 요청에 포함된 파라미터를 모델 객체로 바인딩

userInsert.jsp

```
아이디 :<input type="text" name="userId" />
이름 :<input type="text" name="name" />
성별 :<c:forEach var="genderName" items="${map.genderList}">
    <input type="radio" name="gender" value="${genderName}">${genderName}
</c:forEach>
거주지 :<select name="city">
    <c:forEach var="cityName" items="${map.cityList}">
        <option value="${cityName}">${cityName}</option>
    </c:forEach>
</select>
```

UserController.java

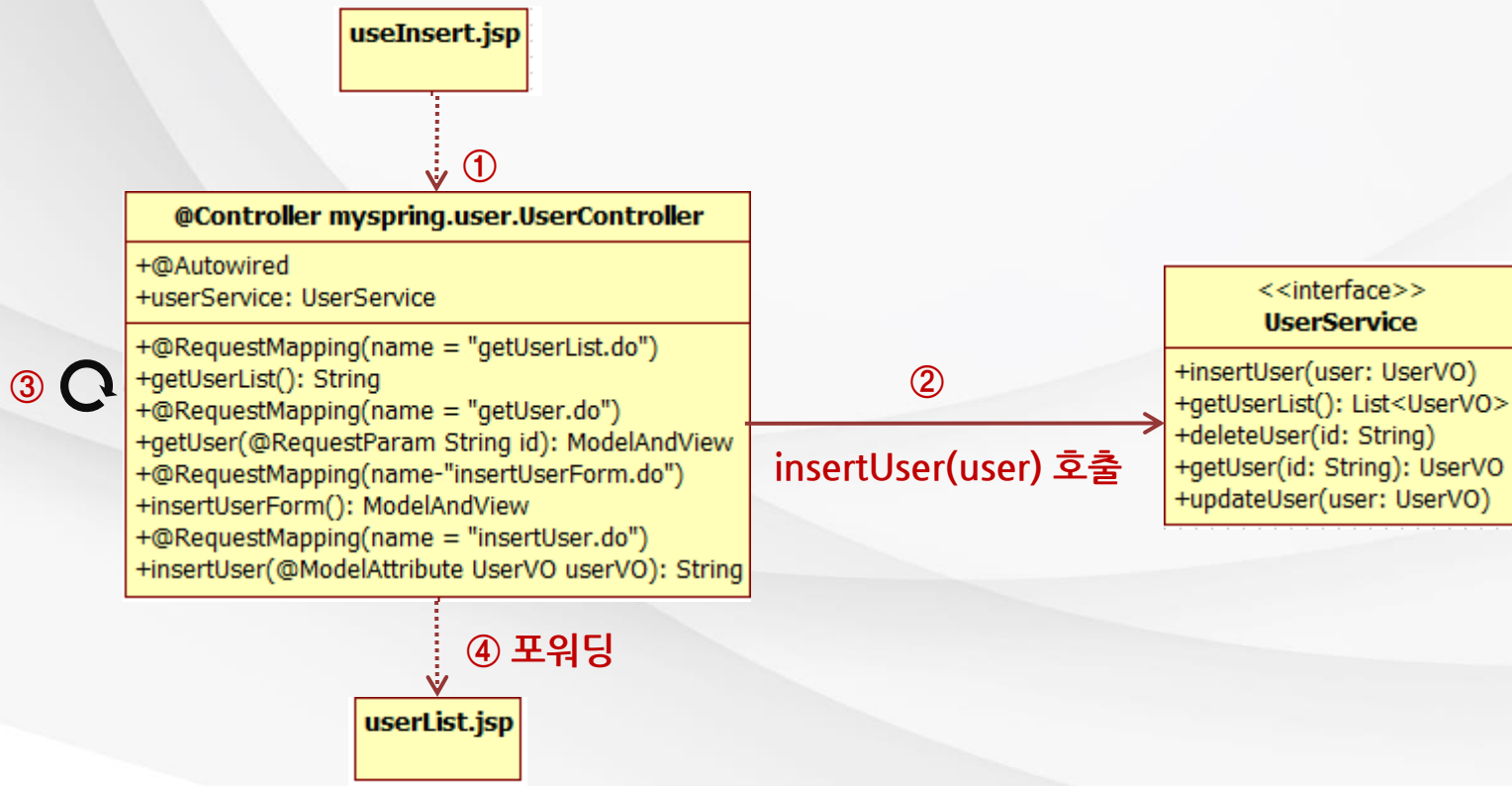
```
@RequestMapping("/insertUser.do")
public String insertUser(@ModelAttribute UserVO user) {
```

UserVO.java

```
public class UserVO {

    private String userId;
    private String name;
    private String gender;
    private String city;
```

■ 사용자 정보 등록 : Controller와 JSP 호출 순서



■ 사용자 정보 등록 : Controller와 JSP 구현

1. userInsert.jsp

```
*userInsert.jsp
<form method="post" action="insertUser.do" >
  <table class="table table-bordered table table-hover">
    <tr>
      <td>아이디 :</td>
      <td><input type="text" name="userId" /></td>
    </tr>
    <tr>
      <td>이름 :</td>
      <td><input type="text" name="name" /></td>
    </tr>
    .....
    <tr>
      <td colspan="2" class="text-center">
        <input type="submit" value="등록" /></td>
      </tr>
    <tr>
      <td colspan="2" style="text-align: center;">
        <input type="button" value="취소" />
      </td>
    </tr>
  </table>
</form>
```

■ 사용자 정보 등록 : Controller와 JSP 구현

2. UserController.java

```
UserController.java
package myspring.user.controller;

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

    @RequestMapping("/insertUser.do")
    public String insertUser(@ModelAttribute UserVO user) {
        if (user != null)
            userService.insertUser(user);
        return "redirect:/getUserList.do";
    }
}
```

■ web.xml : CharacterEncodingFilter 클래스 설정

- ◉ 요청(request) 데이터를 인코딩 해주는 Filter 클래스 설정

```
*web.xml
<!-- CharacterEncoding Filter -->
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>EUC-KR</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
```

3. 사용자 등록 기능 구현

■ 사용자 정보 등록 : 결과 화면

사용자 정보 등록	
아이디 :	<input type="text" value="vega2k"/>
이름 :	<input type="text" value="박소울"/>
성별 :	<input type="radio"/> 남 <input checked="" type="radio"/> 여
거주지 :	<input type="text" value="서울"/>
<input type="button" value="등록"/>	
사용자 목록보기	

사용자 목록

아이디	이름	성별	거주지		
gildong	홍길동2	남2	경기2	수정	삭제
polar	연아2	여2	경기2	수정	삭제
vega2k	박소울	여	서울	수정	삭제
사용자 등록					

지금까지 [Spring MVC 어플리케이션 작성(2)]에 대해서 살펴보았습니다.

특정 사용자 조회 기능 구현

- ◉ UserService Bean을 호출하여 특정 사용자 정보를 조회하는 기능 구현
- ◉ @RequestParam 어노테이션, ViewResolver 설정

사용자 등록화면 기능 구현

- ◉ 사용자 등록하는 화면을 처리하는 기능 구현

사용자 등록 기능 구현

- ◉ UserService Bean을 호출하여 사용자 등록하는 기능 구현
- ◉ @ModelAttribute 어노테이션, CharacterEncodingFilter 설정