

SPRING BOOT 장별 학습 내용

동의대학교 컴퓨터소프트웨어공학과

이종민 교수

목차

- 2장
- 3장
- 4장
- 5장
- 6장

2장

(참고) Spring 주요 애노테이션

- **@Component**
 - Spring에게 커스텀 빈을 자동으로 탐지할 수 있게 하는 애노테이션
- **@Controller**
 - MVC 패턴의 제어기로 사용하는 클래스를 지정
- **@Service**
 - @Component의 특수한 경우로, 비즈니스 로직을 가지고 있음을 알려 줌.
- **@Repository**
 - @Component의 특수한 경우로, 지정된 클래스가 객체에 대한 CRUD 오퍼레이션을 제공함.
- **@Configuration**
 - 지정된 클래스가 Spring 설정 정보를 가지고 있음을 알려 줌.
- **@Bean**
 - 메서드에 사용하여 반환되는 객체를 Spring Bean으로 사용할 수 있도록 함.
- 참고 URL
 - <https://www.baeldung.com/spring-component-annotation>
 - <https://www.baeldung.com/spring-component-repository-service>

@Controller

- 패키지: org.springframework.web.servlet.mvc

Base Controller interface, representing a component that receives HttpServletRequest and HttpServletResponse instances just like a HttpServlet but is able to participate in an MVC workflow

- 사용법

@Controller

```
public class Ch02Controller {  
  
    @GetMapping("/ch02")  
    public String ch02() {  
        return "ch02/index"; // (참고) spring.mvc.view.prefix=/WEB-INF/views/  
                             //          spring.mvc.view.suffix=.jsp  
    }  
}
```

@GetMapping

- 패키지: org.springframework.web.service.annotation
 - URL: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/GetMapping.html>
 - @RequestMapping(method="GET")과 동일

- 사용법

```
// @GetMapping(value="/ch02"), @GetMapping(path="/ch02")와 동일
@GetMapping("/ch02")
public String ch02() {
    return "ch02/index";
}
```

- (참고) value 또는 path는 String[] 자료형이어서 아래와 같이 두 개 이상의 URL 사용 가능
 - @GetMapping({"/url1", "url2"})

3장

@PathVariable

- 패키지: org.springframework.web.bind.annotation
 - 메서드 매개변수가 URI 템플릿 변수에 국한되는 것을 알려주는 애노테이션
- @PathVariable
 - URL: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/PathVariable.html>
 - @PathVariable(**name**="projectName") 과 동일
 - @PathVariable(**value**="projectName") 과 동일
 - @PathVariable(name="projectName", **required**=true)에서처럼 필수 여부 지정 가능
 - @PathVariable(name="projectName", **required**=false) 지정 시 **Optional**로 매개변수 지정 (5장 참고)
- 사용법

```
@GetMapping("/ch03/project/{projectName}")
public String ch03ProjectName(@PathVariable String projectName) {
    if ("simple_page".equals(projectName)) {
        jspPage = "ch03/simple_page/index";
    } else if ("error1".equals(projectName)) {
        // 이하 생략
    }
}
```

4장

컨텍스트 경로

- 사용 예

- `${pageContext.request.contextPath}` → “ /ood ” , “ /spring_jsp ” , ...

```
<link rel="stylesheet" type="text/css"
      href="${pageContext.request.contextPath}/css/my_style.css">
```

```
<form action="${pageContext.request.contextPath}/ch04/beans/show_sum1" method="POST">
    n을 입력하시오 : <input type="text" name="n" value="" size="10" /> <br />
    <input type="submit" value="계산" /> <input type="reset" value="초기화" />
</form>
```

Model (1/2)

- 패키지: org.springframework.ui

Interface that defines a holder for model attributes.
Primarily designed for adding attributes to the model.

Allows for accessing the overall model as a java.util.Map.

- 주요 메서드

```
* Model addAttribute(String attributeName, @Nullable Object attributeValue)
* @Nullable
  Object getAttribute(String attributeName)
* boolean containsAttribute(String attributeName)
* Map<String,Object> asMap()
```

Model (2/2)

- 사용법 예

```
@GetMapping("/ch04/beans")
public String ch04Beans(Model m) {
    log.debug("Real path of \"/\" = {}", ctx.getRealPath("/"));
    m.addAttribute("ctx_path", ctx.getRealPath("/"));

    return "/ch04/beans/index";
}
```

@Autowired

- 생성자, 필드, setter 메서드, config 메서드를 Spring의 의존성 삽입 기능(dependency injection facilities)에 의해 자동 연결

```
@Autowired
private ServletContext ctx;    // jakarta.servlet.ServletContext 객체 연결
@Autowired
private SumSpringBean sumBean; // 사용자 정의 SumSpringBean 객체 연결
```

@PostMapping

- 패키지: org.springframework.web.bind.annotation

Annotation for mapping HTTP POST requests onto specific handler methods.

- 사용법

```
@PostMapping("/ch04/beans/show_sum2")
public String ch04BeansShowSum2(@RequestParam Integer n, Model model) {
    // sumBean.setN(Integer.parseInt(n));
    sumBean.setN(n);
    sumBean.calculate();
}
```

- (참고)
 - @PostMapping(value="url1")과 @PostMapping(path="ur1")처럼 사용 가능
 - @PostMapping({"ur1", "ur2"})처럼 두 개 이상의 URL 명시 가능
 - @RequestMapping(method="POST")와 동일

@RequestParam

- 패키지: org.springframework.web.bind.annotation

Annotation which indicates that a method parameter should be bound to a web request parameter.

- 사용법

- @RequestParam String n : request 객체에 변수 이름 n과 동일한 애트리뷰트가 있으면 생략 가능
- @RequestParam Integer n: <input name="n" type="number">로 인한 필드 값은 Integer로 받을 수 있음
- @RequestParam(name="n") String n : name에서 지정하는 이름과 매개변수 이름이 다를 경우에 주로 사용
- @RequestParam(value="n") String n
- @RequestParam(name="n", required=false, defaultValue="1") String n

- 사례

```
@PostMapping("/ch04/beans/show_sum2")
public String ch04BeansShowSum2(@RequestParam String n, Model model) {
    sumBean.setN(Integer.parseInt(n));
}
```

5장

@PathVariable Optional<Integer>

- 사용법

```
@GetMapping("/{ch05/tagfiletest", "/ch05/tagfiletest/{number}")
public String tagFileTest(@PathVariable Optional<Integer> number) {
    String execUrl = "ch05/tagfiletest/index";
    if (number.isPresent()) {
        execUrl += String.format("%d", number.get());
    }
    return execUrl;
}
```

- (참고)

- @PathVariable(name="number", required=false) Optional<Integer> n
- @PathVariable(value="number", required=false) Optional<Integer> n

- @PathVariable 사용 시 name 또는 value 사용 예

```
@GetMapping({"/ch05/tagfiletest ", " /ch05/tagfiletest/{number} " })  
// Public String tagFileTest(@PathVariable(name="number") Optional<Integer> n) {  
Public String tagFileTest(@PathVariable(value="number") Optional<Integer> n) {  
    String execUrl = "ch05/tagfiletest/index";  
    if (n.isPresent()) {  
        execUrl += String.format("%d", n.get());  
    }  
    log.debug("execUrl = {}", execUrl);  
    return execUrl;  
}
```

6장

Spring JSP Tag Library

- 참고 URL: <https://docs.spring.io/spring-framework/docs/4.3.5.RELEASE/spring-framework-reference/html/spring-tld.html>

- Taglib 지시어 사용법

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
```

- <spring:eval> 사용법

```
<spring:eval expression="@environment.getProperty('spring.datasource.username')"  
    var="db_username" />    또는
```

```
<spring:eval expression="@environment['spring.datasource.username']"  
    var="db_username" />
```

- <spring:eval>에 의해 정의된 프로퍼티 사용법

```
${db_username}    또는  
<% final String USER_NAME = pageContext.getAttribute("db_username") %>
```

(참고)

- <https://docs.spring.io/spring-framework/docs/4.2.x/spring-framework-reference/html/spring-tld.html>
- <https://docs.spring.io/spring-framework/reference/web/webmvc-view/mvc-jsp.html#mvc-view-jsp-formtaglib-formtag>
- <https://docs.spring.io/spring-framework/reference/web/webmvc-view/mvc-jsp.html>
- <https://docs.spring.io/spring-security/reference/servlet/integrations/jsp-taglibs.html>

@PropertySource

- 패키지: org.springframework.context.annotation

Annotation providing a convenient and declarative **mechanism for adding a PropertySource to Spring's Environment**. To be used in conjunction with @Configuration classes.

- 사용법

```
@Controller
@Slf4j
@PropertySource("classpath:/config.properties")
public class Ch06Controller {

    @Autowired
    private Environment env;

    @Value("${mysql.server.ip}")
    private String mysqlServerIp;
```

- 사용법 (계속)

```
@GetMapping("/ch06/showtable1")
public String showTable1(Model model) {
    model.addAttribute("mysql_server_ip", this.mysqlServerIp);

    log.info("env: mysql.server.ip = {}", env.getProperty("mysql.server.ip"));
    return "ch06/showtable1/index";
}
```

PropertiesFactoryBean

- Allows for making a properties file from a classpath location available as Properties instance in a bean factory

[SpringJspApplication.java (main 메서드 아래에 추가)]

```
@Bean
public PropertiesFactoryBean configProperties() {
    log.debug("configProperties() called...");
    PropertiesFactoryBean bean = new PropertiesFactoryBean();
    bean.setLocation(new ClassPathResource("/config.properties"));
    return bean;
}
```

[index.jsp]

```
<%@taglib prefix="spring" uri="http://www.springframework.org/tags" %>
```

```
<spring:eval expression="@configProperties['mysql.server.ip']" var="mysqlServerIp"/>
```

Spring's form tag library

- 참고 URL: <https://docs.spring.io/spring-framework/reference/web/webmvc-view/mvc-jsp.html#mvc-view-jsp-formtaglib-formtag>
- Taglib 지시어 사용법

```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
```

- <form:form> 사용법

```
<form:form>
  <table>
    <tr>
      <td>First Name:</td>
      <td> <form:input path="firstName"/> </td>
    </tr>
    <tr>
      <td>Last Name:</td>
      <td> <form:input path="lastName"/> </td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="Save Changes"/>
      </td>
    </tr>
  </table>
</form:form>
```

• 사용 가능한 form 태그

- `<form:input path="name"/> == <input name="name" type="text" value=""/>`
- `<form:checkbox path="chk" value="test1"/>`
- `<form:checkboxes path="chks" items="${my_list}"/>`
- `<form:radio button path="surname" value="Kim"/>`
- `<form:radio buttons path="surname" value="${surname_list}"/>`
- `<form:password path="password"/>`
- `<form:select path="skills" items="${skills}"/>`
- `<form:textarea path="notes" rows="3" cols="20"/>`
- `<form:hidden path="house"/>`
- `<form:errors>`

```
<tr>
  <td>First Name:</td>
  <td><form:input path="firstName"/></td>
  <%-- Show errors for firstName field --%>
  <td><form:errors path="firstName"/></td>
</tr>
```

<form:form> 사용 예

[index.jsp 일부]

```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>

<form:form method="POST" modelAttribute="formData"
            action="${pageContext.request.contextPath}/ch04/beans/show_sum3">
    n을 입력하시오 : <form:input path="n"/> <br />
    <input type="submit" value="계산" /> <input type="reset" value="초기화" />
</form:form>
```

[Ch04Controller.java 일부]

```
@GetMapping("/ch04/beans2")
public String ch04Beans2(Model m) {
    m.addAttribute("formData", new FormData());
    return "/ch04/beans_with_form/index2";
}

@PostMapping("/ch04/beans/show_sum3")
public String ch04BeansShowSum3(@ModelAttribute("formData") FormData formData,
                                Model model) {
    log.debug("show_sum3: formData.n = {}", formData.getN());
    sumBean.setN(formData.getN());
    sumBean.calculate();
    model.addAttribute("result", sumBean.getResult());
    return "ch04/beans/show_sum3";
}
```

[FormData.java 일부]

```
@Slf4j
public class FormData {
    @Getter
    @Setter
    private Integer n;
}
```