

유효성 검증 (Validator)

프로젝트명 : B04Validator

준비사항

- 의존설정 : Spring Web, Lombok
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.

Validator 인터페이스

home : webapp/WEB-INF/views/home.jsp

```
9<body>
10    <h2>스프링 부트 프로젝트</h2>
11    <ul>
12        <li><a href= "/">루트</a></li>
13    </ul>
14
15    <h2>Validator 인터페이스</h2>
16    <ul>
17        <li><a href= "./write.do">글쓰기폼</a></li>
18    </ul>
19 </body>
```

컨트롤러 : com.edu.springboot.MainController.java

```
1 package com.edu.springboot;
2
3 import org.springframework.stereotype.Controller;
4
5
6
7
8
9 @Controller
10 public class MainController {
11
12     @RequestMapping("/")
13     public String home() {
14         return "home";
15     }
16
17     @RequestMapping("/write.do")
18     public String insert1() {
19         return "write";
20     }
21 }
```

뷰 : webapp/WEB-INF/views/write.jsp

```
9<body>
10<script>
11function sending(f, gubun){
12    if(gubun==1)
13        f.action="./writeAction1.do"
14    else
15        f.action="./writeAction2.do"
16    f.submit();
17}
18</script>
19<h2>Validator 인터페이스를 통한 유효성검증</h2>
20<form method="post">
21    일련번호 : <input type="number" name="idx" value="1">
22    <br />
23    아이디 : <input type="text" name="userid" value="${dto.userid}">
24    <br />
25    제목 : <input type="text" name="title" value="${dto.title}">
26    <br />
27    내용 : <input type="text" name="content" value="${dto.content}">
28    <br />
29    <input type="button" value="전송1" onclick="sending(this.form, 1);">
30    <input type="button" value="전송2" onclick="sending(this.form, 2);">
31</form>
32</body>
```

DTO : com.edu.springboot.BoardDTO.java

롬복으로 설정한다.

```
1 package com.edu.springboot;
2
3 public class BoardDTO {
4
5     private int idx;
6     private String userid;
7     private String title;
8     private String content;
9 }
```

여기까지 작성하세요.

검증클래스 : com.edu.springboot.BoardValidator.java

```
1 package com.edu.springboot;
2
3 import org.springframework.validation.Errors;
4
5
6
7 public class BoardValidator implements Validator {
8
9     @Override
10    public boolean supports(Class<?> clazz) {
11        return BoardDTO.class.isAssignableFrom(clazz);
12    }
13
14    @Override
15    public void validate(Object target, Errors errors) {
16
17        BoardDTO boardDTO = (BoardDTO) target;
18
19        //1
20        String userid = boardDTO.getUserid();
21        if(userid==null || userid.trim().isEmpty()){
22            System.out.println("아이디를 입력해주세요.");
23            errors.rejectValue("userid", "idError", "아이디 검증 실패");
24        }
25
26        //2
27        ValidationUtils.rejectIfEmptyOrWhitespace(errors, "title", "titleError",
28            "제목 검증 실패");
29
30        //3
31        boolean contentValidate = myEmptyOrWhitespace(boardDTO.getContent());
32        if(contentValidate==false){
33            System.out.println("내용을 입력해주세요.");
34            errors.rejectValue("content", "contentError", "내용 검증 실패");
35        }
36    }
37
38    public boolean myEmptyOrWhitespace(String value){
39        if(value==null || value.trim().length()==0){
40            return false;
41        }
42        else{
43            return true;
44        }
45    }
46 }
```

컨트롤러 : com.edu.springboot.MainController.java

```
22 @RequestMapping("/writeAction1.do")
23 public String writeAction1(@ModelAttribute("dto") BoardDTO boardDTO,
24     BindingResult result) {
25     String page = "result";
26     System.out.println(boardDTO);
27
28     BoardValidator validator = new BoardValidator();
29     validator.validate(boardDTO, result);
30
31     if (result.hasErrors()) {
32         page = "write";
33         System.out.println("검증 실패 반환값1:"+result.toString());
34
35         if(result.getFieldError("title")!=null) {
36             System.out.println("제목 검증1(에러코드):"
37                 +result.getFieldError("title").getCode());
38         }
39
40         if(result.getFieldError("content")!=null) {
41             System.out.println("내용 검증1(디폴트메세지):"
42                 +result.getFieldError("content").getDefaultMessage());
43         }
44     }
45
46     return page;
47 }
```

뷰 : webapp/WEB-INF/views/result.jsp

```
<body>
    <h2>글쓰기 결과</h2>
    <p>
        일련번호 : ${dto.idx} <br />
        아이디 : ${dto.userid} <br />
        제목 : ${dto.title} <br />
        내용 : ${dto.content}
    </p>
</body>
```

여기까지 작성하세요.

어노테이션

Gradle : build.gradle

```
//validation 어노테이션을 위한 의존설정  
implementation 'org.springframework.boot:spring-boot-starter-validation'
```

VO : com.edu.springboot.BoardVO.java

```
1 package com.edu.springboot;  
2  
3+ import javax.validation.constraints.NotEmpty;[]  
8  
9 @Data  
10 public class BoardVO {  
11  
12     private int idx;  
13  
14     @NotNull(message = "아이디가 null입니다.")  
15     @NotEmpty(message = "아이디를 입력해주세요.")  
16     @Size(min = 5, max = 12, message = "아이디는 5~12자로 입력해주세요.")  
17     private String userid;  
18  
19     @NotNull(message = "제목이 null입니다.")  
20     @NotEmpty(message = "제목을 입력해주세요.")  
21     private String title;  
22  
23     @NotNull(message = "내용이 null입니다.")  
24     @NotEmpty(message = "내용을 입력해주세요.")  
25     private String content;  
26 }
```

컨트롤러 : com.edu.springboot.MainController.java

```
49 @RequestMapping("/writeAction2.do")  
50 public String writeAction2(@ModelAttribute("dto") @Validated BoardVO boardVO,  
51     BindingResult result) {  
52     String page = "result";  
53     System.out.println(boardVO);  
54  
55     // BoardValidator validator = new BoardValidator();  
56     // validator.validate(boardVO, result);  
57  
58     if (result.hasErrors()) {  
59         page = "write";  
60         System.out.println("검증 실패 반환값2:"+result.toString());  
61  
62         if(result.getFieldError("title")!=null) {  
63             System.out.println("제목 검증2(에러코드):"+  
64                 result.getFieldError("title").getCode());  
65         }  
66     }
```

```
67         if(result.getFieldError("content")!=null) {
68             System.out.println("내용 검증2(디폴트메세지):"+
69                 result.getFieldError("content").getDefaultMessage());
70         }
71     }
72
73     return page;
74 }
75 }
```