

코드로 배우는 스프링 웹 프로젝트

• PART 3



- 웹 계층에서 가장 먼저 설계하는 것은 URI의 설계

Task	URL	Method	Parameter	From	URL 이동
전체 목록	/board/list	GET			
등록 처리	/board/register	POST	모든 항목	입력화면 필요	이동
조회	/board/read	GET	bno=123		
삭제 처리	/board/modify	POST	bno	입력화면 필요	이동
수정 처리	/board/remove	POST	모든 항목	입력화면 필요	이동

02 진행 작업의 순서

- 목록 페이지 - 모든 진입 경로인 동시에 입력을 가는 링크
- 등록 입력/처리 - 게시물 등록 및 처리, 처리후 이동
- 조회 - 목록 페이지에서 특정 게시물로 이동
- 수정/삭제 - 조회 페이지에서 수정/삭제 선택해 처리

02 BoardController 목록의 처리

- 게시물(BoardVO)의 목록을 Model에 담아서 전달

```
@Controller
@Log4j
@RequestMapping("/board/*")
@AllArgsConstructor
public class BoardController {

    private BoardService service;

    @GetMapping("/list")
    public void list(Model model) {

        Log.info("list");
        model.addAttribute("list", service.getList());

    }
}
```

02 BoardController의 테스트

```
@RunWith(SpringJUnit4ClassRunner.class)

//Test for Controller
@WebAppConfiguration

@ContextConfiguration({
    "file:src/main/webapp/WEB-INF/spring/root-context.xml",
    "file:src/main/webapp/WEB-INF/spring/appServlet/servlet-context.xml"})
// Java Config
// @ContextConfiguration(classes = {
//     org.zerock.config.RootConfig.class,
//     org.zerock.config.ServletConfig.class} )
@Log4j
public class BoardControllerTests {

    @Setter(onMethod_ = {@Autowired} )
    private WebApplicationContext ctx;

    private MockMvc mockMvc;
```

@Before

```
public void setup() {  
    this.mockMvc = MockMvcBuilders.webAppContextSetup(ctx).build();  
}
```

@Test

```
public void testList() throws Exception {  
  
    Log.info(  
        mockMvc.perform(  
            MockMvcRequestBuilders.get("/board/list"))  
            .andReturn()  
            .getModelAndView()  
            .getModelMap());  
    }  
}
```

02 BoardController의 등록 처리

```
@PostMapping("/register")
public String register(BoardVO board, RedirectAttributes rttr) {

    Log.info("register: " + board);

    service.register(board);

    rttr.addFlashAttribute("result", board.getBno());

    return "redirect:/board/list";
}
```

- POST방식으로 처리되는 데이터를 BoardVO 타입의 인스턴스로 바인딩해서 메서드에서 활용
- BoardService를 이용해서 등록 처리
- 'redirect: '를 이용해서 다시 목록으로 이동

02 등록처리의 테스트

```
@Test
public void testRegister() throws Exception{

    String resultPage = mockMvc.perform(MockMvcRequestBuilders.post("/board/register")
        .param("title", "테스트 새글 제목")
        .param("content", "테스트 새글 내용")
        .param("writer", "user00")
    ).andReturn().getModelAndView().getViewName();

    Log.info(resultPage);

}
```


02 BoardController의 조회/테스트

```
@GetMapping("/get")
public void get(@RequestParam("bno") Long bno, Model model) {

    Log.info("/get");
    model.addAttribute("board", service.get(bno));
}
```

```
@Test
public void tetGet() throws Exception {

    Log.info(mockMvc.perform(MockMvcRequestBuilders
        .get("/board/get")
        .param("bno", "2"))
        .andReturn()
        .getModelAndView().getModelMap());
}
```

02 BoardController의 수정/테스트

```
@PostMapping("/modify")
public String modify(BoardVO board, RedirectAttributes rttr) {
    Log.info("modify:" + board);

    if (service.modify(board)) {
        rttr.addFlashAttribute("result", "success");
    }
    return "redirect:/board/list";
}
```

```
@Test
public void testModify() throws Exception {

    String resultPage = mockMvc
        .perform(MockMvcRequestBuilders.post("/board/modify")
            .param("bno", "1")
            .param("title", "수정된 테스트 새글 제목")
            .param("content", "수정된 테스트 새글 내용")
            .param("writer", "user00"))
        .andReturn().getModelAndView().getViewName();

    Log.info(resultPage);
}
```

02 BoardController의 삭제/테스트

```
@PostMapping("/remove")
public String remove(@RequestParam("bno") Long bno, RedirectAttributes rttr) {

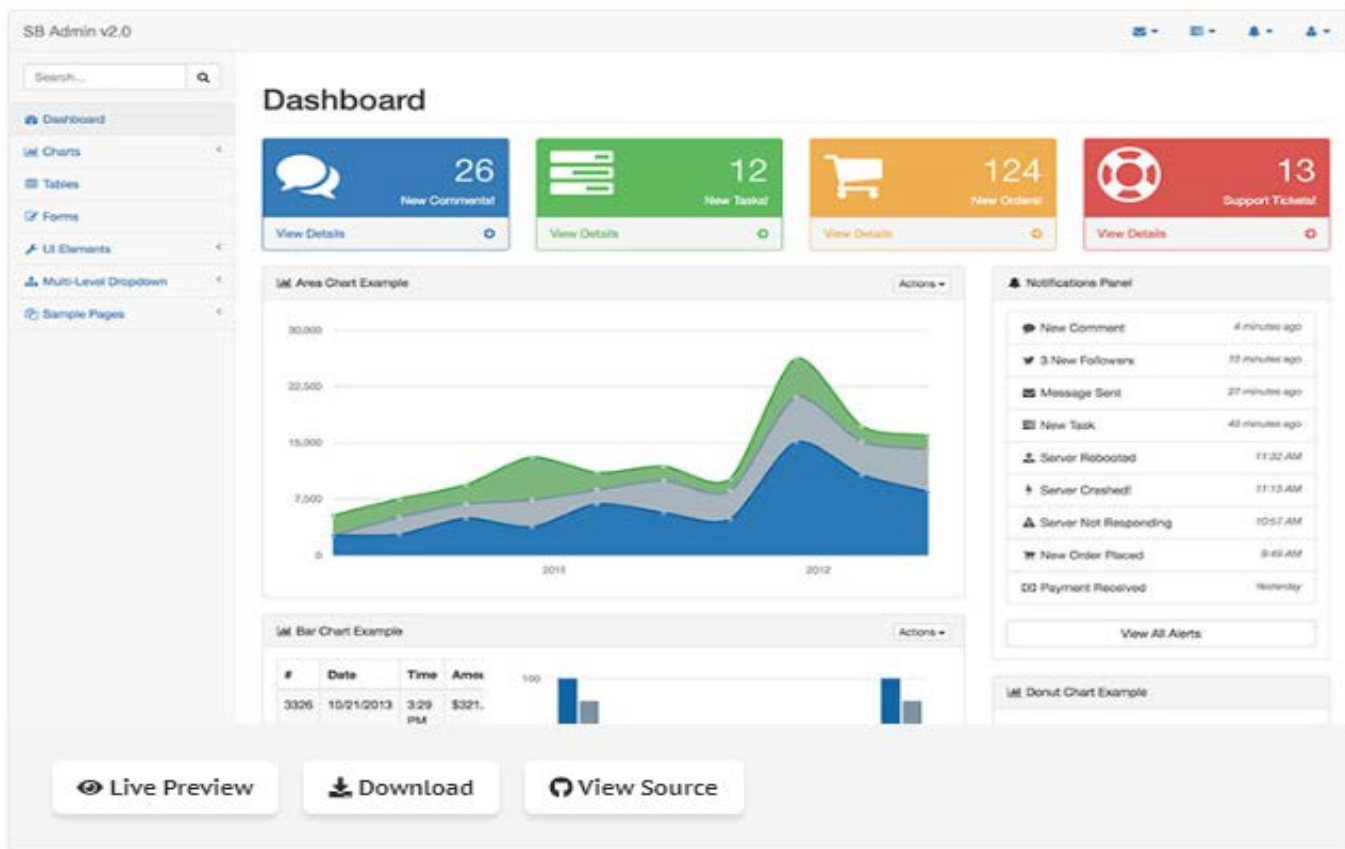
    Log.info("remove..." + bno);
    if (service.remove(bno)) {
        rttr.addFlashAttribute("result", "success");
    }
    return "redirect:/board/list";
}
```

```
@Test
public void testRemove() throws Exception{
    //삭제전 데이터베이스에 게시물 번호 확인할 것
    String resultPage = mockMvc.perform(MockMvcRequestBuilders.post("/board/remove")
        .param("bno", "25")
        ).andReturn().getModelAndView().getViewName();

    Log.info(resultPage);
}
```

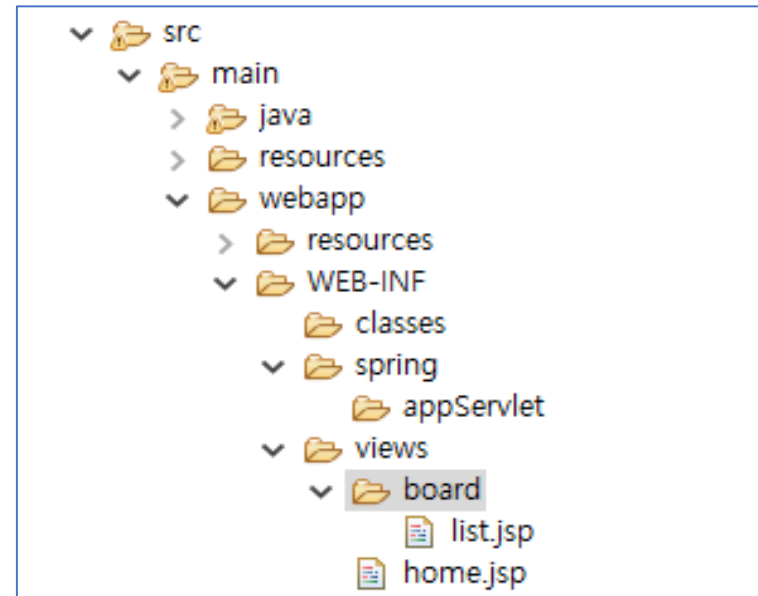
02 화면의 처리

- SB-Admin 2(MIT 라이선스)를 이용해서 페이지 디자인
- <https://startbootstrap.com/template-overviews/sb-admin-2/>



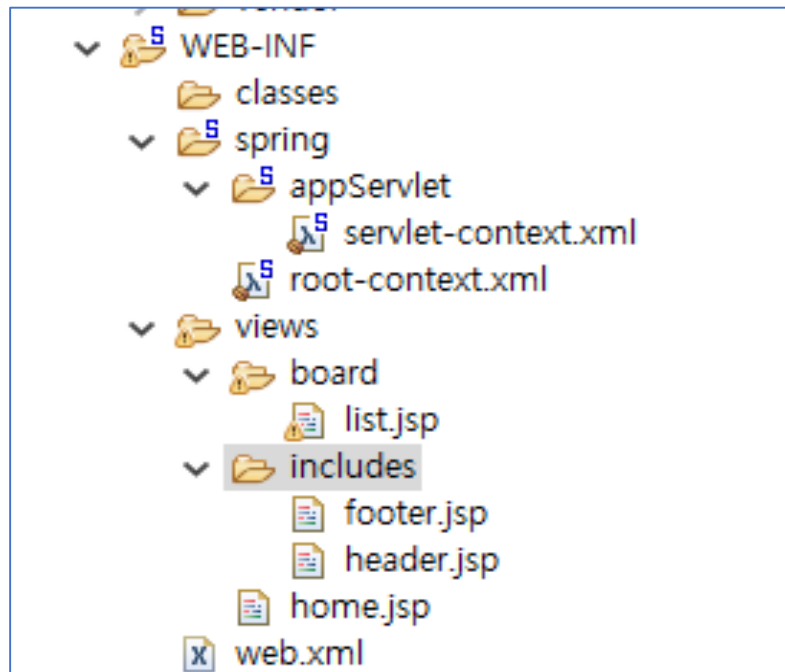
02 목록 페이지 작성

- 프로젝트의 경로는 '/'를 이용하도록 수정
- Tomcat등을 이용해서 실제로 JSP 처리에 문제 없는지 확인 후 진행



02 includes 적용

- JSP페이지의 공통으로 사용되는 페이지의 일부를 header.jsp와 footer.jsp로 분리해서 각 페이지에서 include 하는 방식으로 사용



02 list.jsp의 적용

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<%@include file="../../includes/header.jsp" %>
```

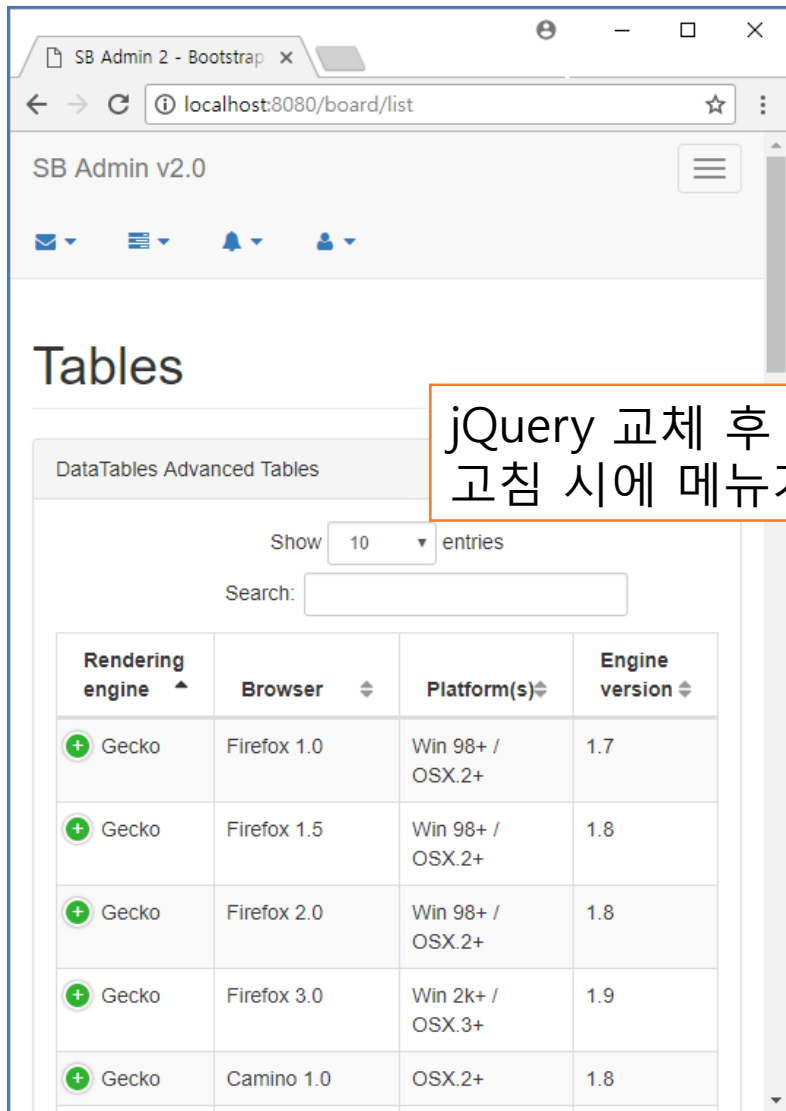
```
    <div class="row">
```

```
        <div class="col-lg-12">
```

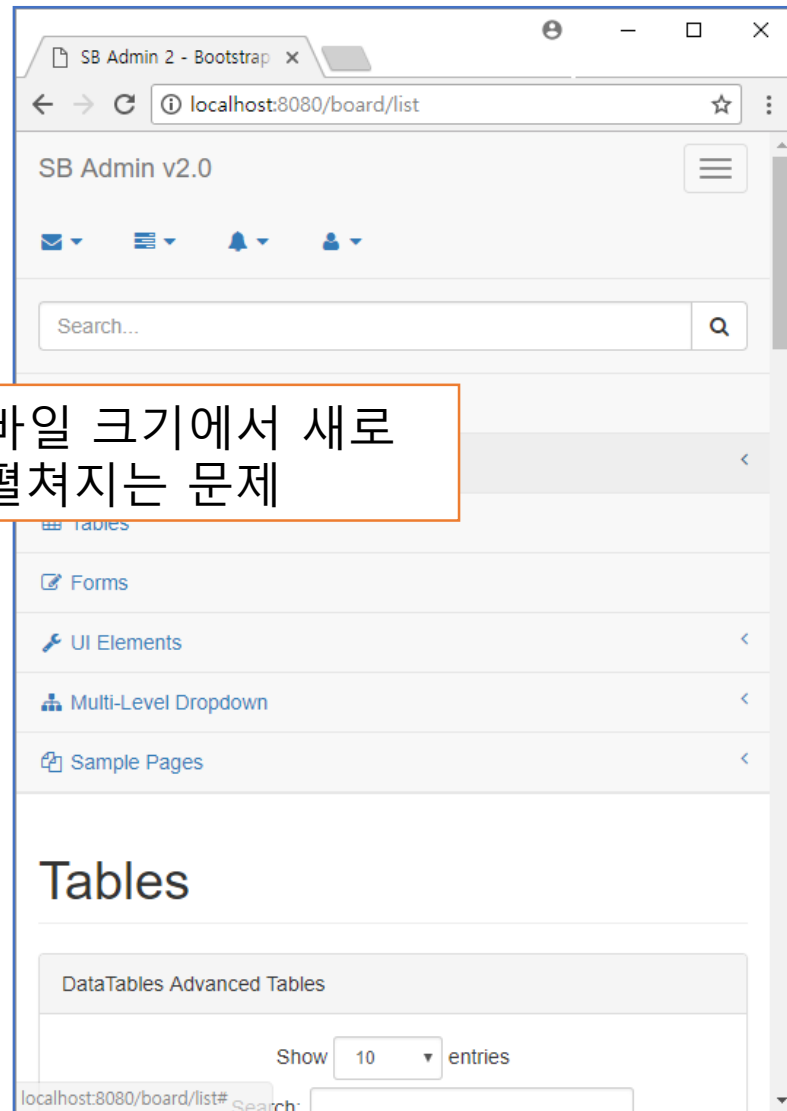
```
            <h1 class="page-header">Tables</h1>
```

...생략...

02 jQuery 버전 변경과 문제



jQuery 교체 후 모바일 크기에서 새로 고침 시에 메뉴가 펼쳐지는 문제




```
<script>
$(document).ready(function() {
  $('#dataTables-example').DataTable({
    responsive: true
  });
  $(".sidebar-nav")
    .attr("class","sidebar-nav navbar-collapse collapse")
    .attr("aria-expanded",'false')
    .attr("style","height:1px");
});
</script>
```

02 목록 화면의 처리

```
<thead>
  <tr>
    <th>#번호</th>
    <th>제목</th>
    <th>작성자</th>
    <th>작성일</th>
    <th>수정일</th>
  </tr>
</thead>

<c:forEach items="${list}" var="board">
  <tr>
    <td><c:out value="${board.bno}" /></td>
    <td><c:out value="${board.title}" /></td>
    <td><c:out value="${board.writer}" /></td>
    <td><fmt:formatDate pattern="yyyy-MM-dd"
      value="${board.regdate}" /></td>
    <td><fmt:formatDate pattern="yyyy-MM-dd"
      value="${board.updateDate}" /></td>
  </tr>
</c:forEach>
</table>
```

SB Admin 2 - Bootstrap x

localhost:8080/board/list

Tables

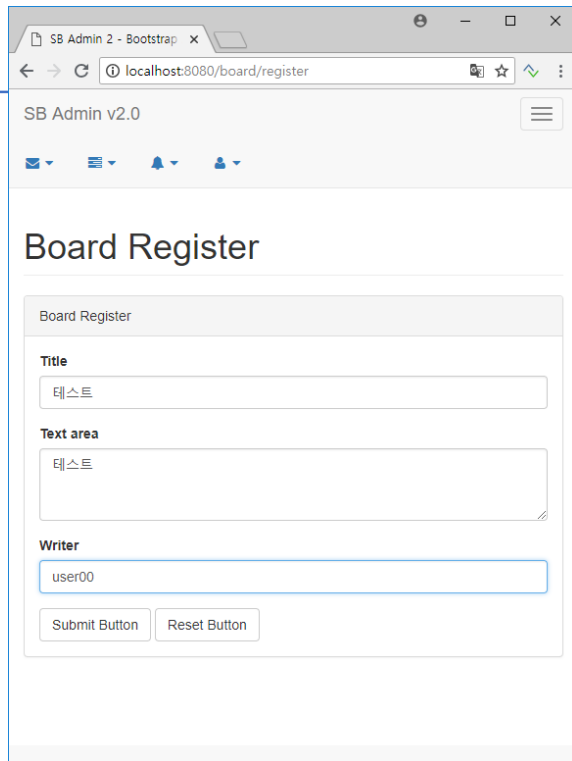
Board List Page

#번호	제목	작성자	작성일	수정일
1	테스트 제목	user00	2018-06-22	2018-06-22
2	테스트 제목	user00	2018-06-22	2018-06-22
3	테스트 제목	user00	2018-06-22	2018-06-22
4	테스트 제목	user00	2018-06-22	2018-06-22
5	테스트 제목	user00	2018-06-22	2018-06-22
21	새로 작성하는 글	newbie	2018-06-23	2018-06-23
22	새로 작성하는 글 select key	newbie	2018-06-23	2018-06-23
23	새로 작성하는 글 select key	newbie	2018-06-23	2018-06-23
24	새로 작성하는 글	newbie	2018-06-25	2018-06-25
25	테스트 새글 제목	user00	2018-06-25	2018-06-25

02 등록 입력 페이지와 등록

- GET방식으로 게시물 등록 화면 제공
- POST방식으로 실제 게시물 등록 처리
- 이후 목록 페이지로 이동

```
@GetMapping("/register")  
public void register() {  
  
}
```



SB Admin v2.0

Board Register

Board Register

Title

Text area

Writer

```
views  
└─ board  
    ├── list.jsp  
    ├── register.jsp  
    └─ includes  
        └─ home.jsp
```

02 한글 깨짐과 필터 설정

SB Admin 2 - Bootstrap x

localhost:8080/board/register

SB Admin v2.0

Board Register

Board Register

Title

테스트

Text area

테스트

Writer

user00

Submit Button Reset Button

게시글이 등록되지만 한글이 깨지는
문제가 발생하는 경우

Board List Page

#번호	제목	작성자	작성일	수정일
1	테스트 제목	user00	2018-06-22	2018-06-22
2	테스트 제목	user00	2018-06-22	2018-06-22
3	테스트 제목	user00	2018-06-22	2018-06-22
4	테스트 제목	user00	2018-06-22	2018-06-22
5	테스트 제목	user00	2018-06-22	2018-06-22
21	새로 작성하는 글	newbie	2018-06-23	2018-06-23
22	새로 작성하는 글 select key	newbie	2018-06-23	2018-06-23
23	새로 작성하는 글 select key	newbie	2018-06-23	2018-06-23
24	새로 작성하는 글	newbie	2018-06-25	2018-06-25
25	테스트 새글 제목	user00	2018-06-25	2018-06-25
26	이이이이이이	user00	2018-06-25	2018-06-25

02 UTF-8 필터 처리

- web.xml을 이용한 필터 설정

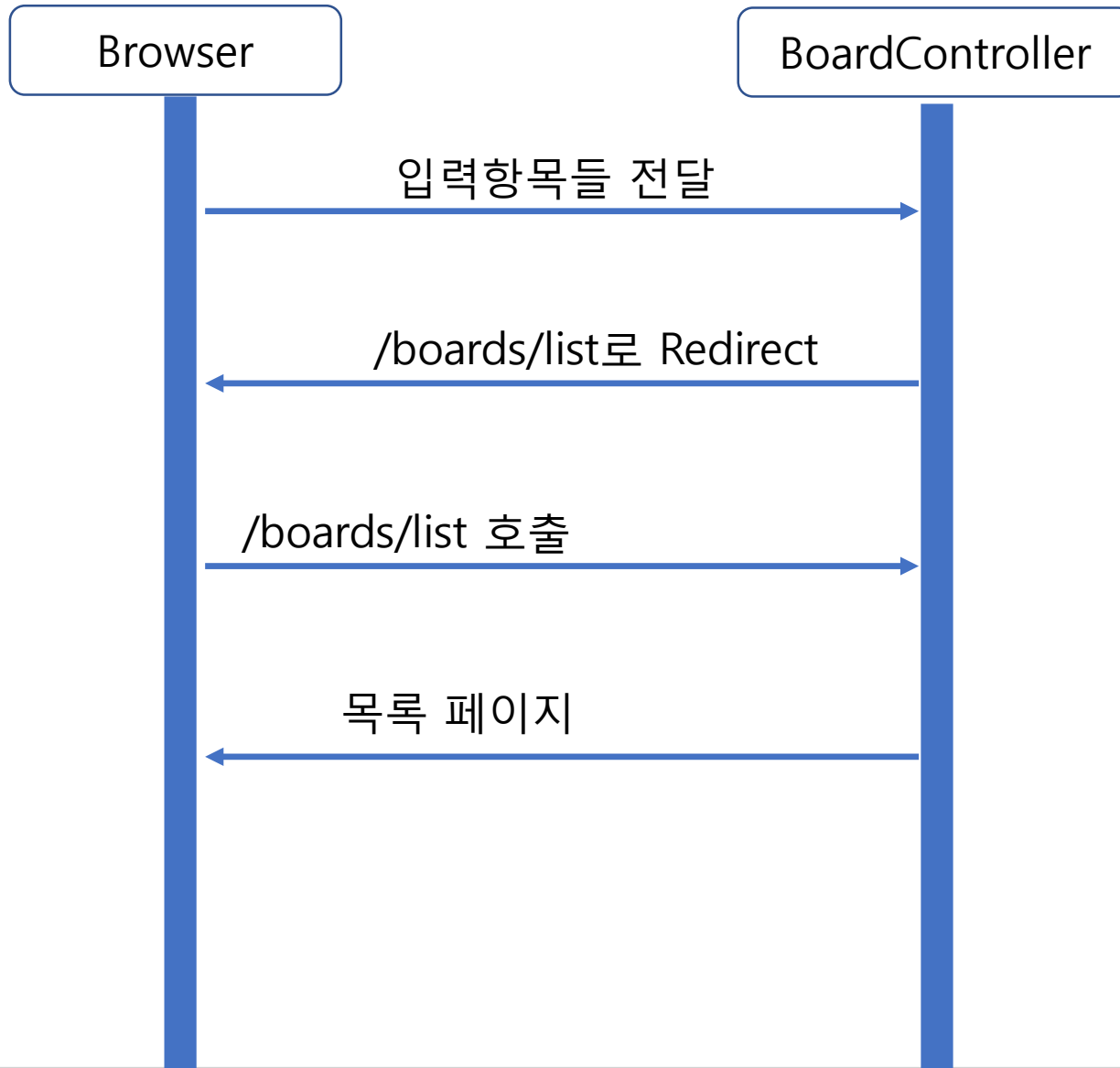
```
<filter>
  <filter-name>encoding</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>encoding</filter-name>
  <servlet-name>appServlet</servlet-name>
</filter-mapping>
```

```
@Override
protected Filter[] getServletFilters() {
    CharacterEncodingFilter characterEncodingFilter = new CharacterEncodingFilter();
    characterEncodingFilter.setEncoding("UTF-8");
    characterEncodingFilter.setForceEncoding(true);

    return new Filter[] { characterEncodingFilter };
}
```

02 재전송(redirect) 처리



- 게시물의 등록, 수정, 삭제의 경우에 해당
- 작업이 완료된 후에는 리스트 페이지로 다시 이동
- BoardController에서는 RedirectAttributes의 addFlashAttribute()를 이용해서 단 한번만 전송되는 데이터 저장 후 전송
- JSP등의 화면에서는 JavaScript를 이용해서 경고창이나 모달창 등을 보여주는 형식

```
@PostMapping("/register")
public String register(BoardVO board, RedirectAttributes rttr) {

    Log.info("register: " + board);

    service.register(board);

    rttr.addFlashAttribute("result", board.getBno());

    return "redirect:/board/list";
}
```

```
<script type="text/javascript">

$(document).ready(function(){

    var result = '<c:out value="${result}"/>';

});
</script>
```

작업후 생성되는 결과

```
<script type="text/javascript">

$(document).ready(function(){

    var result = '15';

});
</script>
```

일반 호출의 결과

```
<script type="text/javascript">

$(document).ready(function(){

    var result = '';

});
</script>
```


02 모달창 보여주기

- BoardController에서 특정한 데이터가 RedirectAttribute에 포함된 경우에 모달창을 보여주기

```
<script type="text/javascript">
$(document).ready(function() {

    var result = '<c:out value="${result}"/>';

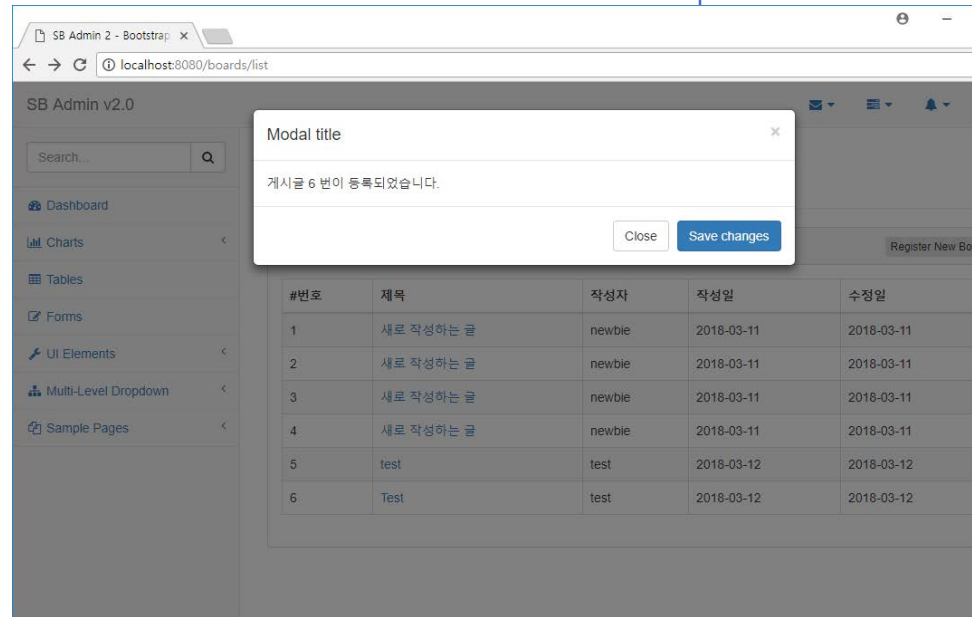
    checkModal(result);

    function checkModal(result) {

        if (result === '') {
            return;
        }

        if (parseInt(result) > 0) {
            $(".modal-body").html(
                "게시글 " + parseInt(result) + " 번이 등록되었습니다.");
        }
        $("#myModal").modal("show");
    }

});
</script>
```



- 목록에서 특정 게시물 선택후 이동 처리
- 조회의 경우는 반드시 GET방식으로만 처리

```
<tr>
  <td> <c:out value= "${board.bno}"/> </td>
  <td> <a href= '/board/get?bno=<c:out value= "${board.bno}"/>' > <c:out
value= "${board.title}"/> </a> </td>
  <td> <c:out value= "${board.writer}"/> </td>
  <td> <fmt:formatDate pattern = "yyyy-MM-dd" value = "${board.regdate}"/> </td>
  <td> <fmt:formatDate pattern = "yyyy-MM-dd" value = "${board.updateDate}"/> </td>
</tr>
```

02 뒤로 가기와 window의 history객체



1

/boards/list ★

2

/boards/register

/boards/list

3

/boards/list ★

/boards/register

/boards/list ★

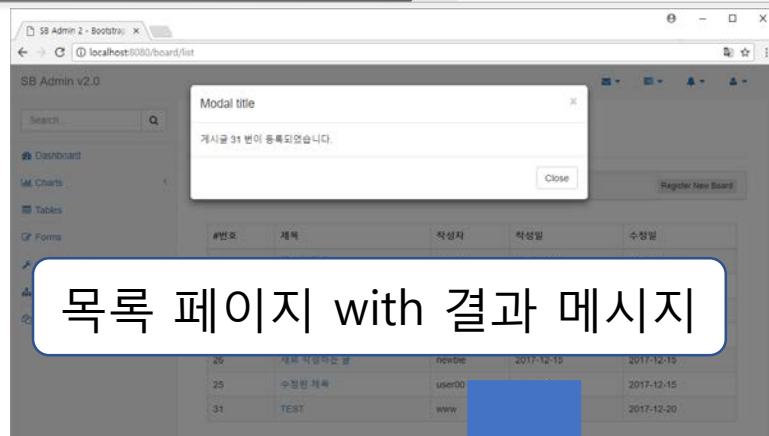
4

/boards/register

/boards/list ★

/boards/register

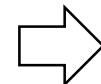
/boards/list



목록 페이지 with 결과 메시지



정상적인 이동

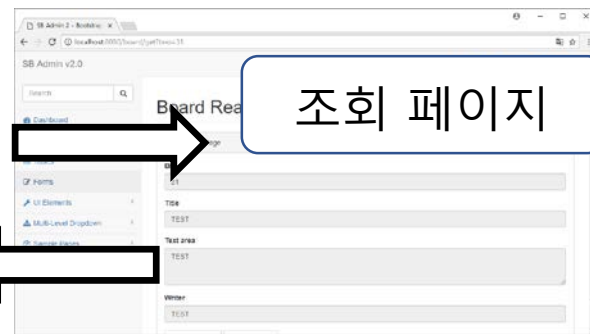
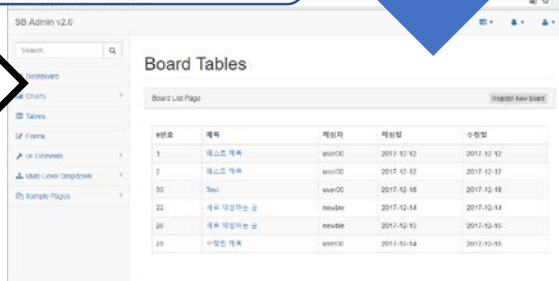
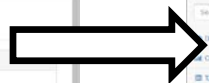


브라우저를 이용한 이동

목록 페이지

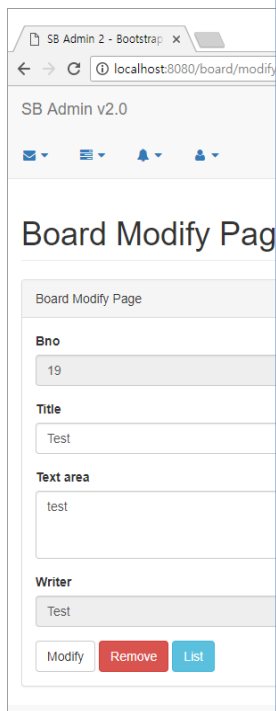
등록 페이지

조회 페이지



02 게시물의 수정/삭제 처리

- 게시물 조회이후 수정/삭제 페이지로 이동
- 사용자의 선택에 따라서 작업을 처리
- 수정/삭제는 POST방식으로 처리



SB Admin v2.0

Board Modify Page

Board Modify Page

Bno
19

Title
Test

Text area
test

Writer
Test

Modify Remove List

```
<script type="text/javascript">
$(document).ready(function() {

    var formObj = $("form");

    $('button').on("click", function(e){

        e.preventDefault();

        var operation = $(this).data("oper");

        console.log(operation);

        if(operation === 'remove'){
            formObj.attr("action", "/board/remove");
        }else if(operation === 'list'){
            //move to list
            self.location= "/board/list";
            return;
        }
        formObj.submit();

    });
});
```

열공합시다!!

