

Part 3

Chapter 03 등록 구현

- 컨트롤러와 프레젠테이션 계층

3.1. 컨트롤러의 구현

- 웹에서 가장 중요한 단계는 컨트롤러와 뷰를 제작하는 것이다.

3.1.1. 컨트롤러 관련 고민들

- 공통적인 URI 경로와 각 기능별 URI
- 각 URI에 대한 호출 방식(GET, POST)
- 결과 처리와 리다이렉트 방식의 페이지 결정
- 예외 페이지

3.1.1.1. 공통 경로의 처리와 호출 방식

방식	URI	설명
GET	/board/register	게시물의 등록 페이지를 보여준다.
POST	/board/register	게시물을 실제로 등록한다.
GET	/board/read?bno=xxx	특정 번호의 게시물을 조회한다.
GET	/board/mod?bno=xxx	게시물의 수정 화면으로 이동한다.
POST	/board/mod	게시물을 수정한다.
POST	/board/remove	게시물을 삭제한다.
GET	/board/list	게시물의 목록을 확인한다.

3.1.1.2 리다이렉트의 처리 방식

- Ajax방식을 사용하여 화면 전환, 슬라이드 이동 기능을 사용하기 위한 조건
 - 등록 작업이 끝나고 리스트가 보여지기 전에 사용자에게 어떤 식으로 등록을 성공했는지 알려줄 것인가?
 - 수정 작업이 끝나고 어떤 식으로 결과를 알려주고 리스트로 이동하는가?
 - 삭제 작업이 끝나면 어떻게 결과를 알려줘야 하는가?

3.1.2. 컨트롤러의 선언

```

@Controller
@RequestMapping("/board/*")
public class BoardController {
    private static final Logger logger = LoggerFactory.getLogger(BoardController.class);

    @Inject
    private BoardService service;
}

```

3.1.2.1. 등록 작업과 파라미터 결정

- 파라미터 수집은 스프링 MVC에서 자동으로 이루어지므로, 파라미터의 수집이 필요하다면 원하는 객체를 파라미터로 선언
- 특별한 경우가 아니라면 VO 클래스 혹은 DTO클래스를 파라미터로 사용하는 것이 편하다.
- 브라우저에서 들어오는 요청(request)이 자동으로 파라미터로 지정한 클래스의 객체 속성값으로 처리되는데 이를 바인딩(binding)이라고 한다.
- 스프링 MVC의 Model 객체는 해당 메서드에서 뷰(jsp 등)에 필요한 데이터를 전달하는 용도로 사용된다. 그러므로 만일 메서드 내에서 뷰로 전달할 데이터가 있다면, Model을 파라미터로 선언해 주는 것이 편리하다.

3.1.2.2. 등록 작업의 메소드

- 등록을 위한 입력 페이지를 보는 경우(GET) - 1) 입력 페이지, 2) 조회 페이지
- 실제로 데이터를 처리하는 부분(POST) - 외부에서 많은 정보를 입력하는 경우, 보안

```

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

    private static final Logger logger = LoggerFactory.getLogger(BoardController.class);

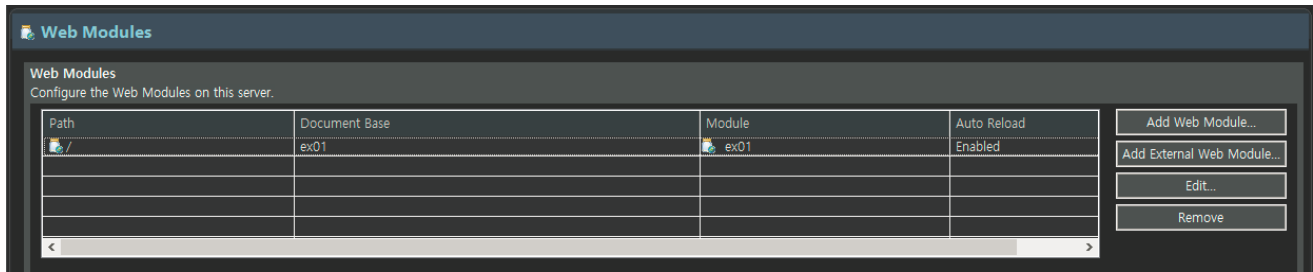
    @Inject
    private BoardService service;

    @RequestMapping(value = "/register", method = RequestMethod.GET)
    public void registerGET(BoardVO board, Model model) throws Exception {
        logger.info("register get .....");
    }

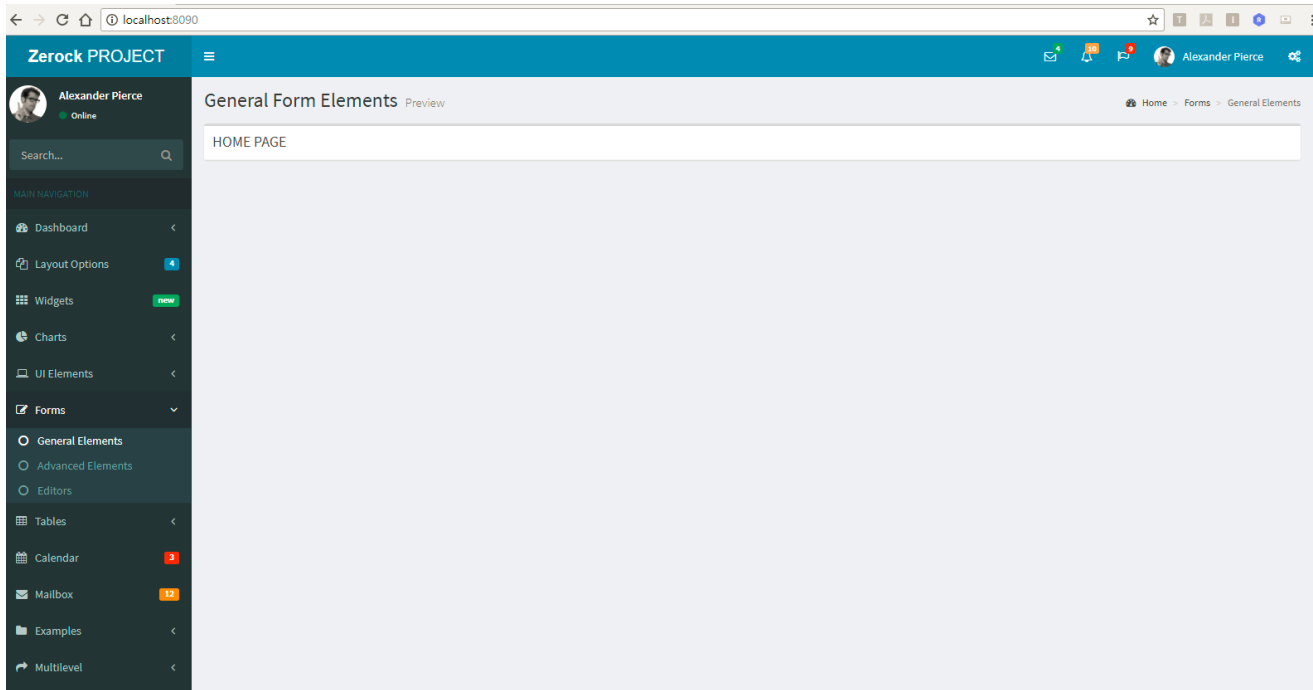
    @RequestMapping(value = "/regiser", method = RequestMethod.POST)
    public String registPOST(BoardVO board, Model model) throws Exception {
        logger.info("regist post");
        logger.info(board.toString());
        service.regist(board);
        model.addAttribute("result", "success");
        return "/board/success";
    }
}

```

3.2. 컨트롤러의 동작 확인과 루트 경로 지정



- rootPage



3.3. 뷰(View)의 구현 - 등록

- 필요한 jsp 페이지 작성
- jsp 페이지 내에 필요한 데이터 전달 확인
- jsp 페이지 내에서의 출력
- register.jsp

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

<!-- Main content -->
<section class="content">
    <div class="row">
        <!-- left column -->
        <div class="col-md-12">
            <!-- general form elements -->
            <div class="box box-primary">
                <div class="box-header">

                    <h3 class="box-title">REGISTER BOARD</h3>
```

```

        </div>
        <!-- /.box-header -->

        <form role="form" method="post">
            <div class="box-body">
                <div class="form-group">
                    <label for="exampleInputEmail1">Title</label> <input
type="text"
                                name='title' class="form-control" placeholder="Enter
Title">

                    </div>
                    <div class="form-group">
                        <label for="exampleInputPassword1">Content</label>
                        <textarea class="form-control" name="content" rows="3"
placeholder="Enter ..."></textarea>
                    </div>
                    <div class="form-group">
                        <label for="exampleInputEmail1">Writer</label> <input
type="text"
                                name="writer" class="form-control" placeholder="Enter
Writer">

                        </div>
                    </div>
                <!-- /.box-body -->

                <div class="box-footer">
                    <button type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>

        </div>
        <!-- /.box -->
    </div>
    <!--/.col (left) -->

</div>
<!-- /.row -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->

<%@include file="../include/footer.jsp"%>

```

- success.jsp

```

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

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

    <!-- Main contents -->
    <section class="contents">

```

```

<div class="row">
  <!-- left column -->
  <div class="col-md-12">
    <!-- general form elements -->
    <div class="box">
      <div class="box-header with-border">
        <h3 class="box-title">SUCCESS PAGE</h3>
      </div>
      <div class="box-body">SUCCESS!!!</div>
      <!-- /.box-body -->
      <div class="box-footer">Footer</div>
      <!-- /.box-footer -->
    </div>
  </div>
  <!-- /.col (left) -->
</div>
<!-- /.row -->
</section>
<!-- /.contents -->
</div>
<!-- /.contents-wrapper -->

<%@include file="../include/footer.jsp"%>

```

3.3.1. 컨트롤러에서의 데이터 전달

- 게시물의 등록 작업
 - /board/register (GET) -> BoardController -> /WEB-INF/views/board/register.jsp
 - /board/register (POST) -> BoardController -> /WEB-INF/views/board/success.jsp
- 사용자가 GET 방식으로 BoardController를 호출, BoardController는 입력할 수 있는 페이지를 호출
- 사용자의 입력 작업이 완료시 POST 방식으로 다시 BoardController를 처리, 등록 작업이 성공 후에 success.jsp 페이지를 보여준다.

3.3.2. 결과 페이지의 문제점 - 새로 고침

- 현재 코드로는 새로 고침시에 중복 등록을 하게 되기 때문에 redirect를 사용하여 흐름을 바꿔준다.

3.3.3. 리다이렉트(redirect)와 결과 데이터

```

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

    private static final Logger logger = LoggerFactory.getLogger(BoardController.class);

    @Inject
    private BoardService service;

    @RequestMapping(value = "/register", method = RequestMethod.GET)
    public void registerGET(BoardVO board, Model model) throws Exception {

```

```

        logger.info("register get .....");
    }

    @RequestMapping(value = "/register", method = RequestMethod.POST)
    public String registPOST(BoardVO board, Model model) throws Exception {
        logger.info("regist post.....");
        logger.info(board.toString());
        service.regist(board);
        model.addAttribute("result", "success");
        // return "/board/success";
        return "redirect:/board/listAll";
    }

    @RequestMapping(value = "/listAll", method = RequestMethod.GET)
    public void listAll(Model model) throws Exception {
        logger.info("show all list .....");
    }
}

```

- listAll.jsp

```

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

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

    <!-- Main contents -->
    <section class="content">
        <div class="row">
            <!-- left column -->
            <div class="col-md-12">
                <!-- general form elements -->

                <div class="box">
                    <div class="box-header with-border">
                        <h3 class="box-title">LIST ALL PAGE</h3>
                    </div>
                    <div class="box-body">
                        <!-- 208 page -->
                        <table class="table table-bordered">
                            <tr>
                                <th style="width: 10px">BNO</th>
                                <th>TITLE</th>
                                <th>WRITER</th>
                                <th>REGDATE</th>
                                <th style="width: 40px">VIEWCNT</th>
                            </tr>

                            </table>
                        </div>

                        <!-- /.box-body -->

```

```

        <div class="box-footer">Footer</div>
        <!-- /.box-footer-->
    </div>
</div>
<!--/.col (left) -->

</div>
<!-- /.row -->
</section>
<!-- /.contents -->
</div>
<!-- /.contents-wrapper -->

<%@include file="../../include/footer.jsp"%>

```

- 브라우저에서 서버로의 데이터 흐름
 - 사용자가 POST 방식으로 BoardController에 데이터를 전송
 - 서버에서는 응답 메시지에 브라우저가 어떤 경로로 이동해야 하는지를 알려준다.
 - 브라우저는 다시 BoardController를 호출, 이때 호출하는 경로는 '/board/listAll?result=success'가 된다.
 - BoardController는 '/WEB-INF/views/board/listAll.jsp'의 결과를 브라우저로 전송

3.3.4. RedirectAttributes를 이용한 숨김 데이터의 전송

- RedirectAttributes 객체의 addFlashAttribute() 메서드 사용

```

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

    private static final Logger logger = LoggerFactory.getLogger(BoardController.class);

    @Inject
    private BoardService service;

    @RequestMapping(value = "/register", method = RequestMethod.GET)
    public void registerGET(BoardVO board, Model model) throws Exception {
        logger.info("register get .....");
    }

    @RequestMapping(value = "/register", method = RequestMethod.POST)
    public String registPOST(BoardVO board, RedirectAttributes rttr) throws Exception {
        logger.info("regist post.....");
        logger.info(board.toString());
        service.regist(board);
        rttr.addFlashAttribute("msg", "SUCCESS");
        // return "/board/success";
        return "redirect:/board/listAll";
    }
}

```

```

@RequestMapping(value = "/listAll", method = RequestMethod.GET)
public void listAll(Model model) throws Exception {
    logger.info("show all list .....");
}
}

```

- listAll.jsp - 스크립트 추가

```

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

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

<!-- Main contents -->
<section class="content">
    <div class="row">
        <!-- left column -->
        <div class="col-md-12">
            <!-- general form elements -->

            <div class="box">
                <div class="box-header with-border">
                    <h3 class="box-title">LIST ALL PAGE</h3>
                </div>
                <div class="box-body">
                    <!-- 208 page -->
                    <table class="table table-bordered">
                        <tr>
                            <th style="width: 10px">BNO</th>
                            <th>TITLE</th>
                            <th>WRITER</th>
                            <th>REGDATE</th>
                            <th style="width: 40px">VIEWCNT</th>
                        </tr>

                        </table>
                    </div>
                    <!-- /.box-body -->
                    <div class="box-footer">Footer</div>
                    <!-- /.box-footer-->
                </div>
            </div>
            <!-- /.col (left) -->

        </div>
        <!-- /.row -->
    </section>
    <!-- /.contents -->
</div>
<!-- /.contents-wrapper -->

<script>

```



```
var result = '${msg}';
if(result === 'SUCCESS'){
    alert("처리가 완료되었습니다.");
}
</script>
<%@include file="../../include/footer.jsp"%>
```