Day11; 20220919

날짜	@2022년 9월 19일
유형	@2022년 9월 19일
태그	

spring/src at main · u8yes/spring

Contribute to u8yes/spring development by creating an account on GitHub.

https://github.com/u8yes/spring/tree/main/src

u8yes/spring

1 Contributor 0 Issues 1 Star 0 Forks

UserRowMapper 자체를 DAO 내부클래스로 넣어줌.

UserDAO.java

```
package com.springproj.biz.dao;

import java.sql.ResultSet;
import java.sql.SQLException;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.stereotype.Repository;

import com.springproj.biz.domain.BoardVO;
import com.springproj.biz.domain.UserVO;

@Repository("userDAO")
public class UserDAO {
  @Autowired
  JdbcTemplate jdbcTemplate;

  private final String USER_GET = "select * from users where id=? and password=?";

  // CRUD 기능의 메소드 구현
  // 회원 상세 정보
```

```
  public UserVO getUser(UserVO user) {
    //System.out.println("getUser() 호출...");
    Object[] args = {user.getId(), user.getPassword()};

    return jdbcTemplate.queryForObject(USER_GET, args, new UserRowMapper());  // 프리젠테이션 레이어

  }

  class UserRowMapper implements RowMapper<UserVO>{   // UserRowMapper를 내부 클래스로 옮겨서 가독성을 높여줌.

    @Override
    public UserVO mapRow(ResultSet rs, int rowNum) throws SQLException {

      UserVO user = new UserVO();

      user.setId(rs.getString("id"));
      user.setPassword(rs.getString("password"));
      user.setName(rs.getString("name"));
      user.setRole(rs.getString("role"));

      return user;
    }
  }



}
```

LogInOutController.java

```
package com.springproj.controller;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import com.springproj.biz.domain.UserVO;
import com.springproj.biz.service.UserService;

@Controller
public class LogInOutController {

  @Autowired
  UserService userService;

  @GetMapping("/login.me")
  public String loginForm() {
    //System.out.println("loginForm() 호출.");

    return "login";
  }

  @PostMapping("/login.me")
  public String loginProc(UserVO user, HttpSession session) {
    //System.out.println("loginProc() 호출.");
    String retVal = null;


    // 1. user의 id 존재 여부를 db에서 가져오기.
    UserVO vo = userService.getService(user);

    if((vo != null) &&
        vo.getPassword().equals(user.getPassword())) {
      session.setAttribute("userName", vo.getName());
```

```
      retVal = "redirect:getBoardList.do";
    }else {
      retVal = "redirect:login.me";
    }
    return retVal;
  }

  @RequestMapping("logout.me")  // logout을 클릭(요청)이 있을 때 호출함.
  public String logoutProc(HttpSession session) {
    System.out.println("로그아웃 처리");

    session.invalidate();

    return "redirect:login.me";
  }
}
```

UserServiceImpl.java

```
package com.springproj.biz.serviceImpl;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.springproj.biz.dao.UserDAO;
import com.springproj.biz.domain.UserVO;
import com.springproj.biz.service.UserService;

@Service("userService")
public class UserServiceImpl implements UserService {
  @Autowired
  UserDAO userDAO;

  @Override
  public UserVO getService(UserVO user) {
    return userDAO.getUser(user);

  }

}
```

---

getBoardList.jsp 안에서 검색 모양만 추가

```
<!-- 검색 시작 -->
    <div class="container w-50" >
    <form action="getBoardList.do" method="post">
      <table border="1" cellpadding="0" cellspacing="0" width="700" class="table table-striped table-sm text-center">
        <tr>
          <td align="right">
            <select name="searchCondition">
              <!-- conditionMap데이터를 option 변수에 저장 -->
              <c:forEach items="${conditionMap}" var="option">
                <option value="${option.value}">${option.key}</option>
              </c:forEach>
            </select>
            <input type="search" name="searchKeyword">  <!-- text를 search로 바꾼 상태 -->
            <input type="submit" value="검색">
          </td>
        </tr>
      </table>

    </form>
    </div>
```

JSP Scope영역

1. page
2. request
3. session
4. application

page, request는 session의 남긴 정보를 모두 지워준다.

데이터를 모든 영역이 다 저장할 수 있다. setAttribute에 담아주면 getAttribute 출력 가능함.

page는 그 페이지에서만 저장.

request는 여러 페이지에서도 저장 가능해짐.

그래서 대부분의 데이터는 request영역에 담는다.

session은 아예 인터넷 창을 (강제로) 닫거나, invalidate로 종료하기 전까지만 그 데이터를 유지하고 있는 영역. <로그인>

application은 서버를 끊기 전까지 데이터를 유지하는 영역.

session영역에 전부다 저장되게 해줌.

상단에

```
@Controller     //  implements Controller 삭제하면서 Controller 사용가능해짐.
@SessionAttributes("board") // board라는 이름으로 모델에 저장하는 순간 session에 같이 저장해줌.
public class BoardController {
```

```
@RequestMapping("/updateBoard.do")  // /insertBoard.do를 부르면 바로 Con
public String updateBoardProc(@ModelAttribute("board") BoardVO vo) {
    //System.out.println("글 수정 처리");
```

담아져있는 out 확인

```
@RequestMapping("/updateBoard.do")  // /insertBoard.do를 부르면 바로 Controller 작업 들어가라
public String updateBoardProc(@ModelAttribute("board") BoardVO vo) {
    //System.out.println("글 수정 처리");

    boardService.updateService(vo);

    System.out.println(vo);

    return "redirect:getBoardList.do";
}
```

```
Markers  Properties  Servers  Data Source Explorer  Snippets  Console  SQL Results
Tomcat v9.0 Server at localhost [Apache Tomcat] C:₩Program Files₩Java₩jdk-11.0.16.1₩bin₩javaw.exe  (2022. 9. 19. 오후 12:46:42) [pid: 13068]
[공통로그] updateService(Before) 비즈니스 로직 수행 전 동작
updateService() 메서드 수행에 걸린 시간 : 2(ms)초

BoardVO[seq=6, title=첫 게시글-수정, writer=민용기, content=드디어 나는 개발자-수정, date=2022-09-13, cnt=0]
글 목록 검색 처리
[공통로그] getServiceList(Before) 비즈니스 로직 수행 전 동작
getServiceList() 메서드 수행에 걸린 시간 : 1(ms)초

[정상 종료] getServiceList(AfterReturning) 비즈니스 로직 수행 후 동작
```

컨트롤러에서 서비스로 보냄

```
List<BoardVO> list = boardService.getServiceList();
```

서비스에서 받음.

서비스에서 dao로 보냄

```
@Override
public List<BoardVO> getServiceList() {

    return dao.getBoardList();
}
```

Apache Commons FileUpload

https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload

pom.xml

```
<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc6 --> <!-- DB와 연결할 수 있게 Java11버전에 맞춘 -->

<dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.4</version>
</dependency>
```

---

servlet-context.xml

```
<!-- File 업로드 설정, 식별자 multipartResolver이름은 절대적이다. -->
<beans:bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
<beans:property name="maxUploadSize" value="100000" />  <!-- value="-1"은 제한없이 올리겠다. -->
</beans:bean>

</beans:beans>
```

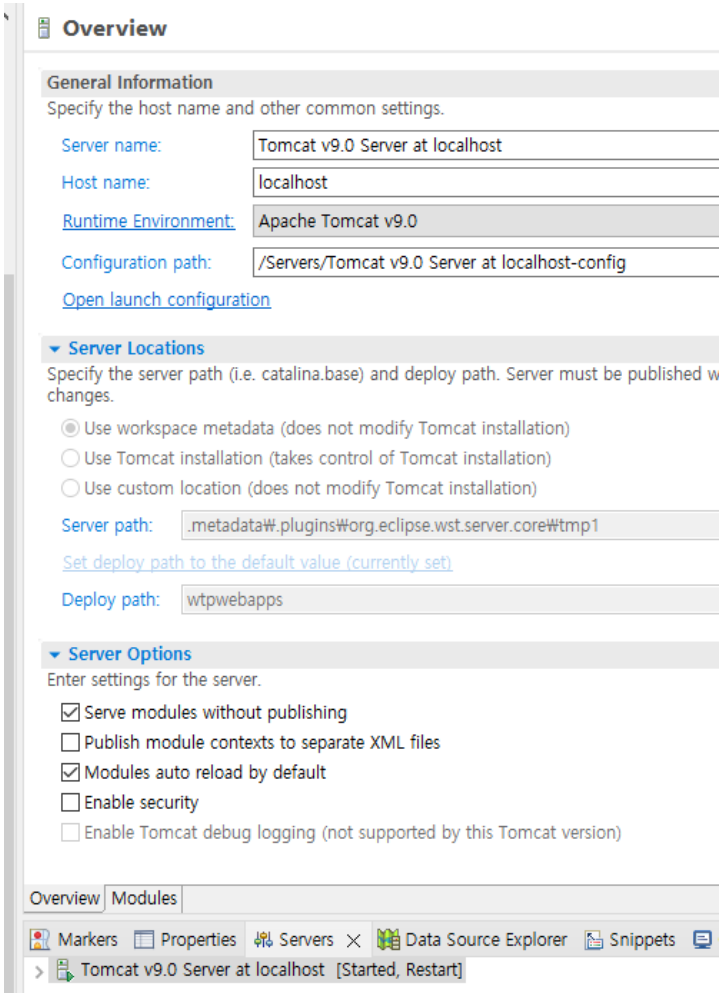
BoardVO.java

```
// 파일 업로드 추가
private MultipartFile uploadFile;
```

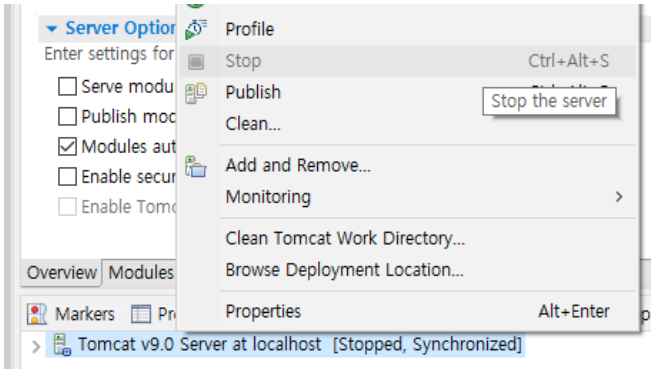
insertBoard.jsp

```
<center>
    <h1>글 등록</h1>
    <a href="logout.do">Log-out</a>
    <hr>
    <form action="insertBoard.do" method="post" enctype="multipart/form-data">  <!-- method는 디폴트가 get방식이다. -->
        <table border="1" cellpadding="0" cellspacing="0">
            <tr>
                <td bgcolor="orange" width="70">제목</td>
                <td align="left"><input type="text" name="title" /></td>
            </tr>
            <tr>
                <td bgcolor="orange">파일첨부</td>
                <td><input type="file" name="uploadFile"></td>
            </tr>
```

Serve modules without publishing 체크 해제

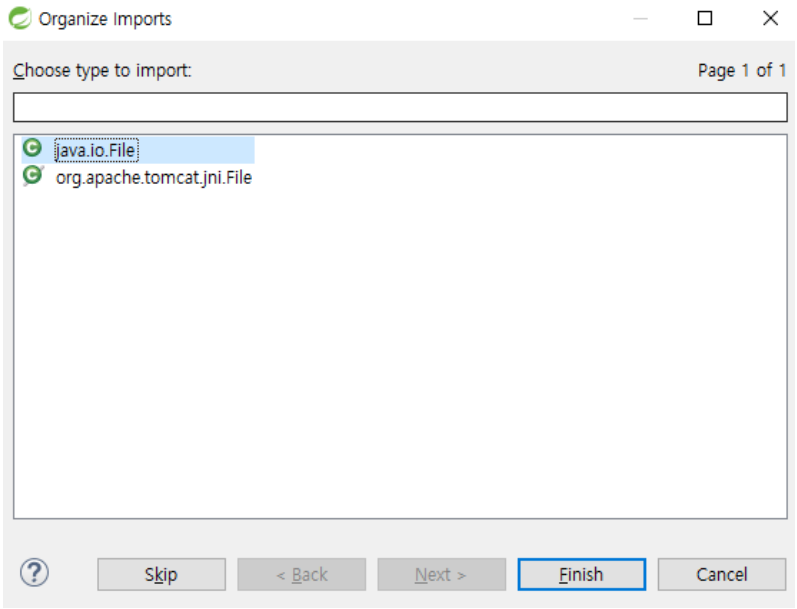


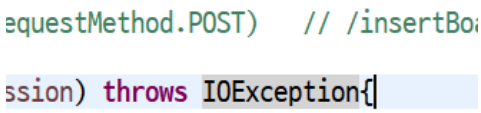
톰캣 정지시킴.



---

java.io.File로 추가해줌.





BoardController.java

```
//@RequestMapping(value = "/insertBoard.do", method = RequestMethod.POST) // /insertBoard.do를 부르면 바로 Controller 작업 들어가라
  @PostMapping("/insertBoard.do")
  public String insertBoard(BoardVO board, HttpSession session) throws IOException{
    //System.out.println("글 등록 처리");

    //파일 업로드 처리
    String fileSaveFolder = session.getServletContext().getRealPath("/boardUpload");
    System.out.println("=>" + fileSaveFolder);

    MultipartFile uploadFile = board.getUploadFile();
```

```
    if(!uploadFile.isEmpty()) {
      String fileName = uploadFile.getOriginalFilename();
      uploadFile.transferTo(new File(fileSaveFolder + fileName));
    }

    boardService.insertService(board);

    return "redirect:getBoardList.do";
  }
```

insertBoard.jsp

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

<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>새글등록</title>
</head>
<body>
  <center>
    <h1>글 등록</h1>
    <a href="logout.do">Log-out</a>
    <hr>
    <form action="insertBoard.do" method="post" enctype="multipart/form-data">  <!-- method는 디폴트가 get방식이다. -->
      <table border="1" cellpadding="0" cellspacing="0">
        <tr>
          <td bgcolor="orange" width="70">제목</td>
          <td align="left"><input type="text" name="title" /></td>
        </tr>
        <tr>
          <td bgcolor="orange">작성자</td>
          <td align="left"><input type="text" name="writer" size="10" /></td>
        </tr>
        <tr>
          <td bgcolor="orange">내용</td>
          <td align="left"><textarea name="content" cols="40" rows="10"></textarea></td>
        </tr>
        <tr>
          <td bgcolor="orange">파일첨부</td>
          <td><input type="file" name="uploadFile"></td>
        </tr>
        <tr>
          <td colspan="2" align="center">
            <input type="submit" value=" 새글 등록 " />
          </td>
        </tr>
      </table>
    </form>
    <hr>
    <a href="getBoardList.do">글 목록 가기</a>
  </center>
</body>
</html>
```

예외

LogInOutController.java

```
@PostMapping("/login.me")
public String loginProc(UserVO user, HttpSession session) {
    //System.out.println("loginProc() 호출.");
    if(user.getId() == null || user.getId().equals("")) {
        throw new IllegalArgumentException("아이디는 반드시 입력해야 합니다."); // 문법적인 오류가 있었을 때 발생시킴.
    }

    String retVal = null;


    // 1. user의 id 존재 여부를 db에서 가져오기.
    UserVO vo = userService.getService(user);

    if((vo != null) &&
            vo.getPassword().equals(user.getPassword())) {
        session.setAttribute("userName", vo.getName());
```