

게시판 구현하기

(파일 업로드 - 설정)

게시판 구현하기 - 파일 업로드 설정

- pom.xml

```
<!-- 파일 업로드 -->  
<dependency>  
  <groupId>commons-fileupload</groupId>  
  <artifactId>commons-fileupload</artifactId>  
  <version>1.4</version>  
</dependency>
```

```
<!-- 파일 업로드 -->  
<dependency>  
  <groupId>commons-fileupload</groupId>  
  <artifactId>commons-fileupload</artifactId>  
  <version>1.4</version>  
</dependency>
```

게시판 구현하기 - 파일 업로드 설정

- **servlet-context.xml**

```
<beans:bean id="multipartResolver"  
class="org.springframework.web.multipart.support.Standard  
ServletMultipartResolver"> </beans:bean>
```

```
<context:component-scan base-package="kr.co.greenart" />  
  
<beans:bean id="authenticationInterceptor" class="kr.co.greenart.common.interceptor.LoginInterceptor"></beans:bean>  
<beans:bean id="accessInterceptor" class="kr.co.greenart.common.interceptor.AccessInterceptor"></beans:bean>  
<beans:bean id="multipartResolver" class="org.springframework.web.multipart.support.StandardServletMultipartResolver"> </beans:bean>  
  
<interceptors>
```

게시판 구현하기 - 파일 업로드 설정

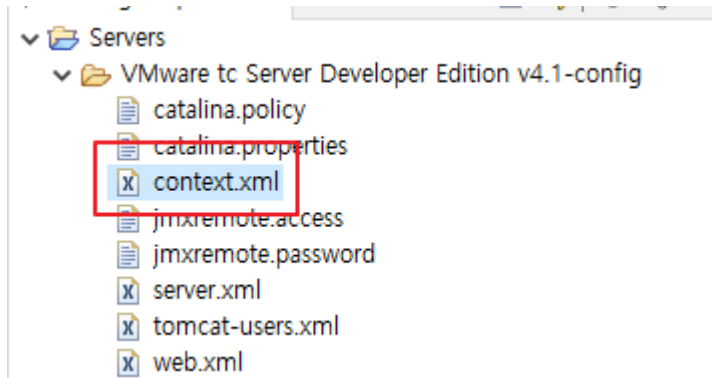
- web.xml

```
<multipart-config>
  <max-file-size>20971520</max-file-size>
  <!-- 업로드 되는 파일의 최대 크기 -->
  <max-request-size>41943040</max-request-size>
  <!-- 한번에 올릴 수 있는 최대 크기 -->
  <file-size-threshold>20971520</file-size-threshold>
  <!-- 특정 사이즈의 메모리 사용 -->
</multipart-config>
```

```
<multipart-config>
  <location>C:\\upload</location>
  <!-- 업로드 되는 파일 저장 공간 -->
  <max-file-size>20971520</max-file-size>
  <!-- 업로드 되는 파일의 최대 크기 -->
  <max-request-size>41943040</max-request-size>
  <!-- 한번에 올릴 수 있는 최대 크기 -->
  <file-size-threshold>20971520</file-size-threshold>
  <!-- 특정 사이즈의 메모리 사용 -->
</multipart-config>
```

게시판 구현하기 - 파일 업로드 설정

- Servers → context.xml (톰캣 설정)



게시판 구현하기 - 파일 업로드 설정

- Servers → context.xml (톰캣 설정)

```
<?xml version="1.0" encoding="UTF-8"?>
<Context allowCasualMultipartParsing="true" path="/">
  <!-- Default set of monitored resources -->
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <!-- Uncomment this to disable session persistence across Tomcat restarts -->
  <!--
    <Manager pathname="" />
  -->
  <!-- Uncomment this to enable Comet connection tacking (provides events
    on session expiration as well as webapp lifecycle) -->
  <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
  -->
  <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
  <Resources cachingAllowed="true" cacheMaxSize="100000" />
</Context>
```

게시판 구현하기 - 파일 업로드 설정

- Servers → context.xml (톰캣 설정)

```
<?xml version="1.0" encoding="UTF-8"?>
<Context allowCasualMultipartParsing="true" path="/">
  <!-- Default set of monitored resources -->
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <!-- Uncomment this to disable session persistence across Tomcat restarts -->
  <!--
    <Manager pathname="" />
  -->
  <!-- Uncomment this to enable Comet connection tacking (provides events
    on session expiration as well as webapp lifecycle) -->
  <!--
    <Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />
  -->
  <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
<Resources cachingAllowed="true" cacheMaxSize="100000" />
</Context>
```

게시판 구현하기 (파일 업로드)

게시판 구현하기 - 파일 업로드

• 참고

실습 시 아래와 같이 진행

1. DB에는 파일 경로, 변경된 파일명, 원본 파일명 저장
2. 파일 경로를 따로 저장하는 이유
→ 나중에 서버 쪽의 업로드 파일명이 변경될 경우 별도의 문자열에 대한 처리 없이 하나의 컬럼만 전체적으로 수정해주면 됨
3. 변경된 파일명으로 사용하는 이유
→ 업로드 시 파일명 중복(ex.제목 없음.png)을 방지
4. 원본 파일명을 따로 저장하는 이유
→ 사용자 요청 시 파일 찾기 및 상세보기에서 가져올 때 사용자가 업로드했던 원본 파일명을 보여줄 수 있음

게시판 구현하기 - 파일 업로드

- 데이터베이스 테이블 추가

```
ALTER TABLE board ADD upload_path VARCHAR(255);
```

```
ALTER TABLE board ADD upload_name VARCHAR(255);
```

```
ALTER TABLE board ADD upload_origin_name VARCHAR(255);
```

```
COMMIT;
```

게시판 구현하기 - 파일 업로드

- boardEnroll.jsp

파일 업로드를 하기 위한 input 작성

```
<form action="/board/insert.do" method="post" enctype="multipart/form-data">
  <div class="mb-3 justify-content-center">
    <input type="text" class="form-control" id="exampleFormControlInput1" name="title" placeholder="제목을 입력하세요.">
  </div>
  <div id="marteeditor mb-3 justify-content-center">
    <textarea name="content" id="editorTxt"
      class="form-control"
      rows="13" cols="10"
      placeholder="내용을 입력해주세요"></textarea>
  </div>
  <input type="file" name="upload"/>
  <div class="row">
    <div class="col text-center">
      <button type="button" class="btn btn-danger" onclick="history.back()">취소</button>
      <button type="submit" onclick="save();" class="btn btn-primary">작성</button>
    </div>
  </div>
</form>
```

게시판 구현하기 - 파일 업로드

- Board

파일 업로드에 필요한 변수 선언

```
@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Board {
    private int idx;
    private String title;
    private String content;
    private String writer;
    private String indate;
    private int count;

    // file Upload
    // UPLOAD_PATH, originalName, fileName
    private String uploadPath;
    private String originalName;
    private String fileName;
}
```

게시판 구현하기 - 파일 업로드

- BoardController

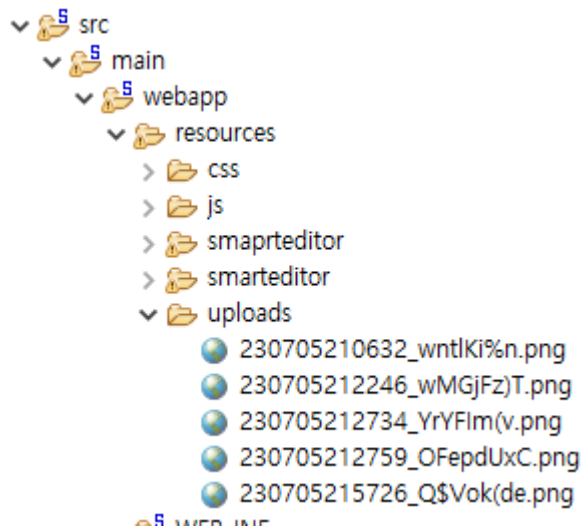
컨트롤러 하단에 업로드할 경로 상수 작성

```
@Controller
@RequestMapping("board")
public class BoardController {
    public static final String UPLOAD_PATH = "D:\\springWorkspace\\springProject\\src\\main\\webapp\\resources\\uploads\\";
```

게시판 구현하기 - 파일 업로드

• BoardController

파일이 실제 서버에 저장이 잘 되는지 확인



```
@PostMapping("insert.do")
public String insertBoard(MultipartFile upload, Board board, HttpSession session, Model model) {
    board.setWriter((String) session.getAttribute("memberName"));

    if(!upload.isEmpty()) {
        // 원본 파일명
        String originalName = upload.getOriginalFilename();

        // 확장자 구하기
        String extension = originalName.substring(originalName.lastIndexOf("."));

        // 현재 년-월-일-시-분-초
        LocalDateTime now = LocalDateTime.now();

        // 년월일시분초 형식으로 데이터 가공
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyMMddHHmmss");
        String output = now.format(formatter);

        // 랜덤 문자열 생성
        int length = 8; // 생성할 문자열의 길이
        String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!@#$$%^&*()"; // 사용 가능한 문자들

        Random random = new Random();
        String randomString = random.ints(length, 0, characters.length())
            .mapToObj(characters::charAt)
            .map(Object::toString)
            .collect(Collectors.joining());

        // DB에 저장할 파일명
        String fileName = (output+"_"+randomString+extension);
        String filePathName = UPLOAD_PATH+fileName;

        Path filePath = Paths.get(filePathName);

        try {
            upload.transferTo(filePath);
        } catch (IllegalStateException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

게시판 구현하기 - 파일 업로드

• BoardController

데이터베이스에 경로, 원본, 파일명, 수정된 파일명 저장

```
try {
    upload.transferTo(filePath);

    // 경로, 원본 파일명, 수정된 파일명 저장
    board.setUploadPath(UPLOAD_PATH);
    board.setOriginalName(originalName);
    board.setFileName(fileName);
    board.setIdx((Integer) session.getAttribute("memberIdx"));
} catch (IllegalStateException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
} else {
    System.out.println(upload);
}
```

저장된 값들은 여기서 쓰임

```
if(titleLength && titleNull) {
    int result = boardService.insertBoard(board);

    if(result>0) {
```

게시판 구현하기 - 파일 업로드

- board-mapper.xml

```
<insert id="insertBoard" parameterType="Board">
    INSERT INTO board(B_IDX,
                      B_TITLE,
                      B_CONTENT,
                      B_WRITER,
                      B_INDATE,
                      B_COUNT,
                      UPLOAD_PATH,
                      UPLOAD_NAME,
                      UPLOAD_ORIGIN_NAME)
    VALUES(seq_b_idx.NEXTVAL,
           #{title},
           #{content},
           #{writer},
           SYSDATE,
           0,
           #{uploadPath},
           #{fileName},
           #{originalName})
</insert>
```


게시판 구현하기 (파일 업로드 - 수정1)

게시판 구현하기 - 파일 업로드 - 수정1

• boardDetail.jsp

파일 업로드를 하기 위한 input 작성

```
<form id="form-submit" method="post" enctype="multipart/form-data" >
  <input type="hidden" name="boardIdx" value="${detail.idx}" />
  <div class="mb-3 justify-content-center">
    <input type="text" class="form-control" id="exampleFormControlInput1" name="title"
      value="${detail.title}" placeholder="제목을 입력하세요." ${detail.writer != user ? 'readonly' : ''}>
    </div>
  <div id="marteditor mb-3 justify-content-center">
    <c:choose>
      <c:when test="${detail.writer == user}">
        <textarea name="content" id="editorTxt" class="form-control" rows="13" cols="10"
          placeholder="내용을 입력해주세요">${detail.content}</textarea>
        </c:when>
      <c:otherwise>
        <div name="content" id="editorTxt" class="form-control" rows="13" cols="10"
          placeholder="내용을 입력해주세요" style="height: 300px;" readonly>${detail.content}</div>
        </c:otherwise>
      </c:choose>
    </div>
    <br>
    <input type="file" name="upload"/>
    <div class="row">
      <div class="col text-center">
        <button type="button" class="btn btn-light" onclick="/member/list.do">취소</button>
        <c:if test="${detail.writer == user}">
          <button type="submit" onclick="submitForm('/board/update.do');" class="btn btn-primary">수정</button>
          <button type="submit" onclick="submitForm('/board/delete.do');" class="btn btn-danger">삭제</button>
        </c:if>
      </div>
    </div>
  </form>
</div>
```

게시판 구현하기 - 파일 업로드 - 수정1

- **kr.co.greenart.common.controller.UploadFileController**

중복 코드 제거를 위한 common controller 생성 (다음 슬라이드)

게시판 구현하기 - 파일 업로드 - 수정1

- kr.co.greenart.common.controller.UploadFileController

```
@Controller
public class UploadFileController {
    public static final String UPLOAD_PATH = "D:\\springWorkspace\\springProject\\src\\main\\webapp\\resources\\uploads\\";

    public Board uploadMethod(MultipartFile upload, Board board, HttpSession session) {
        if(!upload.isEmpty()) {
            // 원본 파일명
            String originalName = upload.getOriginalFilename();
            // 확장자 구하기
            String extension = originalName.substring(originalName.lastIndexOf("."));
            // 현재 년-월-일-시-분-초
            LocalDateTime now = LocalDateTime.now();
            // 년월일시분초 형식으로 데이터 가공
            DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyMMddHHmmss");
            String output = now.format(formatter);
            // 랜덤 문자열 생성
            int length = 8; // 생성할 문자열의 길이
            String characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!@#%&*"; // 사용 가능한 문자들

            Random random = new Random();
            String randomString = random.ints(length, 0, characters.length())
                .mapToObj(characters::charAt)
                .map(Object::toString)
                .collect(Collectors.joining());

            // DB에 저장할 파일명
            String fileName = (output+"_"+randomString+extension);
            String filePathName = UPLOAD_PATH+fileName;

            Path filePath = Paths.get(filePathName);

            try {
                upload.transferTo(filePath);

                // 경로, 원본 파일명, 수정된 파일명 저장
                board.setUploadPath(UPLOAD_PATH);
                board.setOriginalName(originalName);
                board.setFileName(fileName);
                /* board.setIdx((Integer) session.getAttribute("memberIdx")); */
            } catch (IllegalStateException e) {
                e.printStackTrace();
            }
        }
    }
}
```

게시판 구현하기 - 파일 업로드 - 수정1

• BoardController

```
@PostMapping("update.do")
public String updateBoard(MultipartFile upload, Board bo, HttpSession session, @RequestParam(value="boardIdx") int idx) {
    bo.setWriter((String) session.getAttribute("memberName"));
    bo.setIdx(idx);

    // upload가 비어있지 않은 경우 : 새로운 업로드 정보 업데이트
    if(!upload.isEmpty() && bo.getWriter() == session.getAttribute("memberName")) {
        bo = uploadfile.uploadMethod(upload, bo, session);
        int result = boardService.updateUploadBoard(bo);

        if(result>0) {
            session.setAttribute("msg", "수정 되었습니다.");
            session.setAttribute("status", "success");
            return "redirect:/board/list.do";
        } else {
            session.setAttribute("msg", "수정에 실패하였습니다.");
            session.setAttribute("status", "error");
            return "redirect:/board/list.do";
        }
    }
    // upload가 비어있는 경우 : 기존 업로드 정보 유지
    } else if (upload.isEmpty() && bo.getWriter() == session.getAttribute("memberName")) {
        int result = boardService.updateBoard(bo);

        if(result>0) {
            session.setAttribute("msg", "수정 되었습니다.");
            session.setAttribute("status", "success");
            return "redirect:/board/list.do";
        } else {
            session.setAttribute("msg", "수정에 실패하였습니다.");
            session.setAttribute("status", "error");
            return "redirect:/board/list.do";
        }
    }
    } else {
        session.setAttribute("msg", "잘못된 접근입니다.");
        session.setAttribute("status", "error");
        return "redirect:/board/list.do";
    }
}
```

게시판 구현하기 - 파일 업로드 - 수정1

- BoardService / BoardServiceImpl

```
// 게시글 수정
int updateBoard(Board bo);

// 게시글 수정(파일 업로드)
int updateUploadBoard(Board bo);
```

```
@Override
public int updateBoard(Board bo) {
    return boardDao.updateBoard(sqlSession, bo);
}

@Override
public int updateUploadBoard(Board bo) {
    return boardDao.updateUploadBoard(sqlSession, bo);
}
```

게시판 구현하기 - 파일 업로드 - 수정1

- BoardDao

```
public int updateBoard(SqlSessionTemplate sqlSession, Board bo) {  
    return sqlSession.update("boardMapper.updateBoard", bo);  
}  
  
public int updateUploadBoard(SqlSessionTemplate sqlSession, Board bo) {  
    return sqlSession.update("boardMapper.updateUploadBoard", bo);  
}
```

100

- **board-mapper.xml**

```
<update id="updateBoard" parameterType="Board">
    UPDATE board SET b_title = #{title},
                    b_content = #{content}
    WHERE b_idx = #{idx}
</update>

<update id="updateUploadBoard" parameterType="Board">
    UPDATE board SET b_title = #{title},
                    b_content = #{content},
                    upload_path = #{uploadPath},
                    upload_name = #{fileName},
                    upload_origin_name = #{originalName}
    WHERE b_idx = #{idx}
</update>
```


게시판 구현하기 (파일 업로드 - 수정2)

게시판 구현하기 - 파일 업로드 - 수정1

- `kr.co.greenart.common.controller.UploadFileController`

이전에 만들어뒀던 `detailBoard` 메서드를 사용하여 `boardIdx`를 기준으로 정보 가져오기

* 파일 경로, 파일 이름을 가져오기 위함

서버에 저장된 파일 삭제 ←

```
@Autowired
private BoardServiceImpl boardService;

public boolean deleteFile(MultipartFile upload, Board board) {
    board = boardService.detailBoard(board.getIdx());
    if(!Objects.isNull(board)) {
        File deleteFile = new File(board.getUploadPath()+board.getFileName());
        deleteFile.delete();
        return true;
    } else {
        return false;
    }
}
```

게시판 구현하기 - 파일 업로드 - 수정1

- BoardController

```
// upload가 비어있지 않은 경우
// 새로운 업로드 정보 업데이트
if(!upload.isEmpty() && bo.getWriter() == session.getAttribute("memberName")) {
    // 이전에 업로드한 파일 삭제
    uploadfile.deleteFile(upload, bo);

    bo = uploadfile.uploadMethod(upload, bo, session);
    int result = boardService.updateUploadBoard(bo);

    if(result>0) {
        session.setAttribute("msg", "수정 되었습니다.");
        session.setAttribute("status", "success");
        return "redirect:/board/list.do";
    } else {
        session.setAttribute("msg", "수정에 실패하였습니다.");
    }
}
```

게시판 구현하기
(실습 - 삭제 구현하기)