

1. 요구 사항

- 매거진 게시물 삭제 (매거진 내용, 첨부한 사진파일)
- 파일에 대한 정보를 AJAX를 통해 JSON으로 파싱해서 스프링에서 인식이 되게 한 뒤 파일을 삭제한다.
- 게시글에 사진이 여러 개 이므로 이를 판별할 수 있는 방법을 구상한다.

2. 구현사항

2-1. Controller

```
// 매거진 삭제
@DeleteMapping("/magazine/modify/{board_id}")
public ResponseEntity<String> magazineDelete(BoardPrdctImageVO bPrdctImageVO) {
    ResponseEntity<String> entity = null;
    log.info("magazineDelete...");

    String[] deletefiles = bPrdctImageVO.getDeletefiles(); // 삭제할 이미지 정보
    int board_id = bPrdctImageVO.getBoard_id(); // 삭제할 게시글 번호

    try {
        // 삭제는 업로드의 역순으로 진행한다
        for (String s : deletefiles) {
            boardService.magazineImageRemove(board_id, s); // 이미지 삭제(N)
        }

        boardService.magazineRemove(board_id); // 텍스트 삭제(1)

        entity = new ResponseEntity<String>("SUCCESS", HttpStatus.OK);
    } catch (Exception e) {
        e.printStackTrace();
        entity = new ResponseEntity<String>(e.getMessage(),
            HttpStatus.BAD_REQUEST);
    }
    return entity;
}
```

2-2. Service

```
// 매거진 이미지 삭제
public int magazineImageRemove(int board_id, String s) throws IOException;
```

```
// 매거진 이미지 삭제
@Override
public int magazineImageRemove(int board_id, String str) throws IOException {
    String image_name = str;
    log.info("image_name: ", image_name);

    // 삭제할 File 객체를 생성(썸네일 파일)
    // 삭제할 폴더 이름, 삭제할 파일 이름
```

```

File deleteFile = new File(UPLOAD_PATH, image_name);

// 해당 파일이 존재하면 삭제
if (deleteFile.exists() == true) {
    deleteFile.delete();
}

return boardMapper.magazineImageRemove(board_id, image_name);
}

```

2-3. Mapper

```

// 매거진 이미지 삭제
public int magazineImageRemove(int board_id, String image_name);

```

```

<!-- 매거진 이미지 삭제 -->
<delete id="magazineImageRemove">
<![CDATA[
    DELETE FROM prdct_image WHERE board_id = #{board_id} AND image_name = #
    {image_name}
]]>
</delete>

```

2-4. View

```

$(document).ready(function () {
    $('#magazine_delete').click(function(event) {
        event.preventDefault();

        var formData = new FormData(); // FormData 객체 생성

        // 이미지 파일
        var fileNoArray = new Array(); // 여러 파일번호를 담을 배열 생성
        var fileNameArray = new Array(); // 여러 파일명을 담을 배열 생성

        // uploadimagenumber 키워드가 붙은 class의 개수를 가져온다
        var imageCnt = $(".class*=uploadimagenumber").length;
        console.log("imageCnt: " + imageCnt);

        // 이미지 개수만큼 해당 태그내의 요소 값을 배열안에 push한다
        for (var i = 0; i < imageCnt; i++) {
            var temp = i.toString();
            fileNoArray.push($('.uploadimagenumber' + temp).html());
            fileNameArray.push($('.deletefiles' + temp).html());
        }

        // push한 데이터 확인
        console.log("fileNoArray: " + fileNoArray);
        console.log("fileNameArray: " + fileNameArray);

        // 배열 길이만큼 formData에 해당 인덱스 값을 append한다
        for (var i = 0; i < fileNoArray.length; i++) {
            formData.append("image_number", fileNoArray[i]);
            formData.append("deletefiles", fileNameArray[i]);
        }
    });
}

```

```

console.log("formData: " + formData);

$.ajax({
    type : 'DELETE',
    url : $(this).attr("href"),
    cache : false,
    processData: false,
    contentType: false,
    data: formData,
    success: function(result) {
        console.log(result);
        if(result == "SUCCESS") {
            alert("파일을 삭제합니다.");
            $(location).attr('href',
                '${pageContext.request.contextPath}/board/magazine')
        }
    },
    error: function(e) {
        alert("파일을 삭제할 수 없습니다.");
        console.log(e);
    }
});
});
});

```

```

<!-- 간소화 -->
<c:forEach items="${magazine_image}" var="image" varStatus="image_status">
    <!-- varStatus="image_status"를 통해 해당 리스트의 인덱스를 class 이름 뒤에 붙인다 -->
    <span class="uploadimagenumber${image_status.index}">${image.image_number}
</span>
    <span class="deletefiles${image_status.index}">${image.image_name}</span>
</c:forEach>

<button type="button" id="magazine_delete">삭제하기</button>

```

3. 새롭게 알게 된 정보

- JQuery로 해당 클래스 키워드가 붙은 것들을 찾아서 가져올 수 있다.

```

// uploadimagenumber 키워드가 붙은 class의 개수를 가져온다
var imageCnt = $("[class*='uploadimagenumber']").length;
console.log("imageCnt: " + imageCnt);

```

- html()을 통해 해당 태그에 있는 정보를 가져온다.

```

// 이미지 개수만큼 해당 태그내의 요소 값을 배열안에 push한다
for (var i = 0; i < imageCnt; i++) {
    var temp = i.toString();
    fileNoArray.push($('.uploadimagenumber' + temp).html());
    fileNameArray.push($('.deletefiles' + temp).html());
}

```

- JSTL의 foreach에서 varStatus="" 속성이 있는데 이를 통해 해당 리스트의 인덱스를 추출할 수 있다.

```
<c:forEach items="${magazine_image}" var="image" varStatus="image_status">
    <!-- varStatus="image_status"를 통해 해당 리스트의 인덱스를 class 이름 뒤에 붙인다 -->
    <span class="uploadimagenumber${image_status.index}" style="display: none;">${image.image_number}</span>
    <span class="deletefiles${image_status.index}" style="display: none;">${image.image_name}</span>
    ...
</c:forEach>
```

- Java상에서 파일을 삭제할 때 다음과 같은 방법을 이용한다.

```
// 해당 파일이 존재하면 삭제
if (deleteFile.exists() == true) {
    deleteFile.delete();
}
```