

## 1. 이슈사항

- DefaultHandlerExceptionHandler 발생
- JSON으로 파싱할때 타입이 안맞아서 생기는 예외로 보임(VALUE\_STRING token)

```
WARN 10900 --- [nio-8989-exec-7] .w.s.m.s.DefaultHandlerExceptionHandler
: Resolved [org.springframework.http.converter.HttpMessageNotReadableException
: JSON parse error
: Cannot deserialize instance of
`[Lorg.springframework.web.multipart.MultipartFile;` out of VALUE_STRING token;

nested exception is com.fasterxml.jackson.databind.exc.MismatchedInputException
: Cannot deserialize instance of
`[Lorg.springframework.web.multipart.MultipartFile;` out of VALUE_STRING token
at [Source: (PushbackInputStream); line: 1, column: 106] (through reference
chain: edu.bit.ex.joinvo.BoardImageUploadVO["uploadfiles"])]
```

## 2. 해결사항

- View 페이지에서 AJAX 처리시 Type 파라미터를 FormData로 변경(JSON.stringify => FormData)

```
// 기존
$.ajax({
  dataType: 'json',
  type : "POST",
  url : $(this).attr("action"),
  cache : false,
  contentType:'application/json; charset=utf-8',
  processData: false,
  data: JSON.stringify({form, formData}),
  success: function (result) {
    $(location).attr('href',
    '${pageContext.request.contextPath}/board/magazine')
  },
  error: function (e) {
    console.log(e);
    alert('업로드를 할 수 없습니다.');
```

```
// 변경사항
$.ajax({
  type : "POST",
  url : $(this).attr("action"),
  cache : false,
  contentType:'application/json; charset=utf-8',
  processData: false,
  contentType: false,
  data: formData,
  success: function (result) {
```

```

        $(location).attr('href',
        '${pageContext.request.contextPath}/board/magazine');
    },
    error: function (e) {
        console.log(e);
        alert('업로드 실패');
        location.reload();
    }
});

```

- JSON.stringify로 받지 않는다는건 말그대로 문자 형태의 데이터를 넘기지 않는다는 의미이므로 Controller단에 있는 파라미터의 @RequestBody 어노테이션을 없애준다.

```

// 매거진 작성
@Transactional
@PostMapping("/magazine/write")
public ResponseEntity<String> magazineWrite(BoardPrdctImageVO bPrdctImageVO)
{
    ResponseEntity<String> entity = null;
    log.info("magazineWrite..");

    MultipartFile[] uploadfiles = bPrdctImageVO.getUploadfiles();

    try {
        boardService.setMagazineWrite(bPrdctImageVO); // 텍스트 등록(1)

        for (MultipartFile f : uploadfiles) {
            boardService.setMagazineImage(f); // 이미지 등록(N)
        }

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

    return entity;
}

```

### 3. JSP파일

```

<!-- 작성 폼 스크립트 -->
<script>
    $(document).ready(function () {
        $("#writeForm").submit(function(event){
            event.preventDefault();

            var formData = new FormData();

            // 텍스트 입력 영역
            var mbr_id = $("#mbr_id").val();
            var board_name = $("#board_name").val();
            var board_content = $("#board_content").val();

```

```

        console.log(mbr_id);
        console.log(board_name);
        console.log(board_content);
        console.log($(this).attr("action"));

        formData.append("mbr_id", mbr_id);
        formData.append("board_name", board_name);
        formData.append("board_content", board_content);

        // 파일저장 영역
        var inputFile = $("#uploadfiles");
        var files = inputFile[0].files;

        for (var i = 0; i < files.length; i++) {
            console.log(files[i]);
            formData.append("uploadfiles", files[i]);
        }

        // 파일 넣을때 JSON.stringify()는 적용이 안된다...
        $.ajax({
            type : "POST",
            url : $(this).attr("action"),
            cache : false,
            contentType:'application/json; charset=utf-8',
            processData: false,
            contentType: false,
            data: formData,
            success: function (result) {
                console.log("UPLOAD SUCCESS!")
                $(location).attr('href',
                    '${pageContext.request.contextPath}/board/magazine');
            },
            error: function (e) {
                console.log(e);
                alert('업로드 실패');
                location.reload(); // 실패시 새로고침하기
            }
        })
    });
})
</script>

<!-- 업로드 폼(간소화) -->
<form id="writeForm" method="post"
action="${pageContext.request.contextPath}/board/magazine/write"
enctype="multipart/form-data">
    <!-- 회원 id 가져오기 -->
    <input type="hidden" id="mbr_id" value="${magazine_write.mbr_id}">
    <!-- 글제목 -->
    <input class="form-control" type="text" id="board_name" name="board_name"
placeholder="글제목을 입력하세요">
    <!-- 글내용 -->
    <textarea class="form-control" cols="3" id="board_content"
name="board_content" placeholder="글내용을 입력하세요"></textarea>
    <!-- 첨부파일 -->

```

```

        <input type="file" id="uploadfiles" name="uploadfiles" placeholder="첨부 사진"
multiple/>
        <button type="submit" class="btn btn-primary">작성하기</button>
    </form>

```

## 4. Controller

```

// 매거진 작성
@Transactional
@PostMapping("/magazine/write")
public ResponseEntity<String> magazineWrite(BoardPrdctImageVO bPrdctImageVO) {
    ResponseEntity<String> entity = null;
    log.info("magazineWrite..");

    MultipartFile[] uploadfiles = bPrdctImageVO.getUploadfiles();

    try {
        boardService.setMagazinewrite(bPrdctImageVO); // 텍스트 등록(1). 우선 텍스
트 부분을 선행으로 한다.

        for (MultipartFile f : uploadfiles) {
            boardService.setMagazineImage(f); // 이미지 등록(N). 텍스트 부분이 선행되
면 이미지를 올린다.
        }

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

    return entity;
}

```

## 5. Service

```

// 매거진 작성
public void setMagazinewrite(BoardVO boardVO);

// 매거진 첨부사진 작성
public void setMagazineImage(MultipartFile file);

```

```

// 매거진 작성
@Override
public void setMagazinewrite(BoardVO boardVO) {
    log.info("setMagazinewrite");
    boardMapper.setMagazinewrite(boardVO);
}

// 매거진 첨부사진 작성

```

```

@Override
public void setMagazineImage(MultipartFile file) {
    // 파일 이름 변경(중복방지)
    UUID uuid = UUID.randomUUID();
    String saveName = uuid + "_" + file.getOriginalFilename();
    log.info("image_name: ", saveName);

    // 저장할 File 객체를 생성(썸네기 파일)
    // 저장할 폴더 이름, 저장할 파일 이름
    File saveFile = new File(UPLOAD_PATH, saveName);

    try {
        // 업로드 파일에 saveFile이라는 썸네기 입힘
        file.transferTo(saveFile);
    } catch (IOException e) {
        e.printStackTrace();
    }

    boardMapper.setMagazineImage(saveName);
}

```

## 6. Mapper

```

// 매거진 작성
public void setMagazineWrite(BoardVO boardVO);

// 매거진 첨부사진 작성
public void setMagazineImage(String saveName);

```

```

<!-- 매거진 작성 -->
<insert id="setMagazineWrite" >
<![CDATA[
    INSERT INTO board (board_id, board_name, board_content, mbr_id,
board_type_number, board_like, board_hit)
    VALUES (board_seq.nextval, #{board_name}, #{board_content}, #{mbr_id}, 2, 0,
0)
]]>
</insert>

<!-- 매거진 첨부사진 작성 -->
<!-- 매거진 작성을 거친뒤 첨부사진을 올리는 상황이므로 업로드하는 board_id의 현재 번호를 가
리키는 board_seq.currval을 insert 한다. -->
<insert id="setMagazineImage" >
<![CDATA[
    INSERT INTO prdct_image (image_number, image_name, image_path, board_id)
    VALUES (prdct_image_seq.nextval, #{uploadfiles},
'D:/Others/Programming/Project
space/branches/branches_project/src/main/resources/static/prdct_img/',
board_seq.currval)
]]>
</insert>

```

## 7. VO

```
@AllArgsConstructor
@NoArgsConstructor
@Setter
@Getter
@ToString
// 게시판 이미지 VO
public class BoardPrdctImageVO {
    private String board_name; // 게시글제목
    private String board_content; // 게시글내용
    private String mbr_id; // 회원id
    private int board_type_number; // 게시판유형번호
    private int inquiry_number; // 문의유형번호
    private Date board_date; // 게시글작성일자
    private String prdct_id; // 상품id
    private int order_number; // 주문번호
    private int board_like; // 게시글추천수
    private int board_hit; // 게시글조회수
    private int board_starrate; // 게시글별점수

    private int board_id; // 게시글번호 부모pk, 자식fk

    private int image_number; // 상품이미지번호
    private String image_name; // 상품이미지명
    private String image_path; // 상품이미지경로

    private MultipartFile[] uploadfiles; // 게시판이미지업로드
}
```