



Error Handling

@병주 박 추가해줘잉

🤖 Common API response

예외 처리 핸들링

- Spring Boot 내에서 발생하는 예외에 대한 처리 방식
- 일반적인 통신 데이터와 동일하게 예외처리에 대한 리턴도 통일을 하기 위함
- @RestControllerAdvice 를 통해서 Spring Container내에서 발생하는 예외에 대한 핸들링을 지원
- 컨테이너 외에서 발생하는 예외(필터)에 대한 처리는 최상단 서블릿 필터에서 예외처리 진행

새로운 Error Exception 추가법

1. 예외 정의

a. common/error/ResponseType.java

i. enum 속성값 추가

ii. `SUCCESS ("SU", HttpStatus.OK, "요청이 성공적으로 처리되었습니다.")`

iii. 속성 구조 → 호출이름(코드, Http상태, 메세지)

2. 핸들링 추가

a. common/error/ControllerExceptionHandler.java

```
@ExceptionHandler(NotFoundException.class)
public ApiResponse<?> notFoundPageHandler(NotFoundException e) {
    return new ApiResponse.builder<Object>()
        .errorStatus(ResponseType.NOT_FOUND_PAGE)
        .message(e.getMessage())
        .build();
}
```

```

        .build();
    }

```

- b. @ExceptionHandler 어노테이션 사용
- c. 리턴타입 일치시키기

ResponseType.java

```

import lombok.Getter;
import org.springframework.http.HttpStatus;

@Getter
public enum ResponseType {
    // HTTP Status 200
    SUCCESS("SU", HttpStatus.OK, "요청이 성공적으로 처리되었습니다."),

    // HTTP Status 400
    NOT_FOUND_PAGE("NFP", HttpStatus.BAD_REQUEST, "페이지를 찾을 수 없습니다."),
    INVALID_EMAIL_AUTH("IEA", HttpStatus.BAD_REQUEST, "이메일 인증 코드가 유효하지 않습니다."),
    EXPIRED_EMAIL_AUTH("EEA", HttpStatus.BAD_REQUEST, "이메일 인증 코드가 만료되었습니다."),
    NOT_MATCH_ID_AND_EMAIL("NMIE", HttpStatus.BAD_REQUEST, "아이디와 이메일이 일치하지 않습니다."),
    INVALID_OLD_PASSWORD("IOP", HttpStatus.BAD_REQUEST, "기존 비밀번호가 올바르지 않습니다."),
    INVALID_NEW_PASSWORD_LENGTH("INPL", HttpStatus.BAD_REQUEST, "새 비밀번호 길이가 올바르지 않습니다."),
    UNREADABLE_IMAGE("URI", HttpStatus.BAD_REQUEST, "이미지를 읽을 수 없습니다."),
    NOT_FOLLOWING("NF", HttpStatus.BAD_REQUEST, "팔로우 취소를 요청했습니다."),
    ALREADY_FOLLOWING("AF", HttpStatus.CONFLICT, "팔로우 추가를 요청했습니다."),
    INVALID_ORDER("IO", HttpStatus.BAD_REQUEST, "order 옵션이 잘못되었습니다."),
    PLAYLIST_TITLE_REQUIRED("PTR", HttpStatus.BAD_REQUEST, "플레이리스트 제목이 필요합니다."),

    // HTTP Status 401
    INVALID_CREDENTIALS("IC", HttpStatus.UNAUTHORIZED, "올바른 아이디와 비밀번호를 입력해주세요."),
    EMAIL_NOT_VERIFIED("ENV", HttpStatus.UNAUTHORIZED, "이메일 인증이 필요합니다."),
    INVALID_REFRESH_TOKEN("IRT", HttpStatus.UNAUTHORIZED, "Refresh Token이 유효하지 않습니다."),
    EXPIRED_REFRESH_TOKEN("ERT", HttpStatus.UNAUTHORIZED, "Refresh Token이 만료되었습니다."),

    // HTTP Status 403

```

```

// HTTP Status 404
USER_NOT_FOUND("UNF", HttpStatus.NOT_FOUND, "찾는 회원이 없습니다.");
FOLLOWER_NOT_FOUND("FNF", HttpStatus.NOT_FOUND, "찾는 팔로워가 없습니다.");
TRACK_NOT_FOUND("TNF", HttpStatus.NOT_FOUND, "페이지에 해당하는 트랙이 없습니다.");
TRACK_NOT_FOUND("TNF", HttpStatus.NOT_FOUND, "요청한 트랙이 없습니다.");
PLAYLIST_NOT_FOUND("PNF", HttpStatus.NOT_FOUND, "요청한 플레이리스트가 없습니다.");

// HTTP Status 409
DUPLICATE_NICKNAME("DNN", HttpStatus.CONFLICT, "이미 사용 중인 닉네임입니다.");
DUPLICATE_ID("DID", HttpStatus.CONFLICT, "이미 사용 중인 아이디입니다.");
DUPLICATE_EMAIL("DEM", HttpStatus.CONFLICT, "이미 사용 중인 이메일입니다.");

// HTTP Status 500
DATABASE_ERROR("DBE", HttpStatus.INTERNAL_SERVER_ERROR, "데이터베이스 오류입니다.");
SERVER_ERROR("SER", HttpStatus.INTERNAL_SERVER_ERROR, "서버 내부 오류입니다.");
EMAIL_SEND_FAILED("ESF", HttpStatus.INTERNAL_SERVER_ERROR, "이메일 전송에 실패했습니다.");
IMAGE_UPLOAD_FAILED("IUF", HttpStatus.INTERNAL_SERVER_ERROR, "이미지 업로드에 실패했습니다.");

private final String code;
private final HttpStatus status;
private final String message;

ResponseType(String code, HttpStatus status, String message) {
    this.code = code;
    this.status = status;
    this.message = message;
}
}

```