

부가기능(트랜잭션/ 예외처리)

2022년 10월 31일 월요일 오후 4:52

트랜잭션

1. 선언적 트랜잭션
 - 설정으로 트랜잭션 관리
 - a. 어노테이션을 이용한 선언적 방식
 - b. AoP를 이용한 선언적 방식
2. 프로그래밍 트랜잭션 (권장 x)
 - 프로그래밍적으로 트랜잭션 관리

Layered Architecture 중 Business layer 에서 트랜잭션을 관리함

예외처리

Try- catch 를 사용해야만 하는 checked exception

Rethrowing 가능한 unchecked exception 이 있음

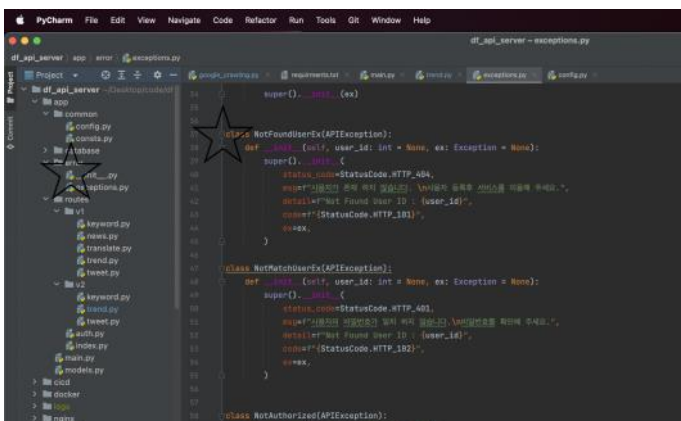
스프링은 어떤 예외가 발생했을때 수행될 메소드를 미리 만들어두고

컨트롤러에서 예외가 발생하면 해당 메소드가 실행

해당 메소드는 Exception 클래스를 인자로 받음

	Checked Exception	Unchecked Exception
처리여부	반드시 예외를 처리해야 함	명시적인 처리를 강제하지 않음
확인시점	컴파일 단계	실행단계
예외발생시 트랜잭션 처리	roll-back 하지 않음	roll-back 함
대표 예외	Exception의 상속받는 하위 클래스 중 Runtime Exception을 제외한 모든 예외 <ul style="list-style-type: none">• IOException• SQLException	RuntimeException 하위 예외 <ul style="list-style-type: none">• NullPointerException• IllegalArgumentException• IndexOutOfBoundsException• SystemException

FastApi 예외처리 예제



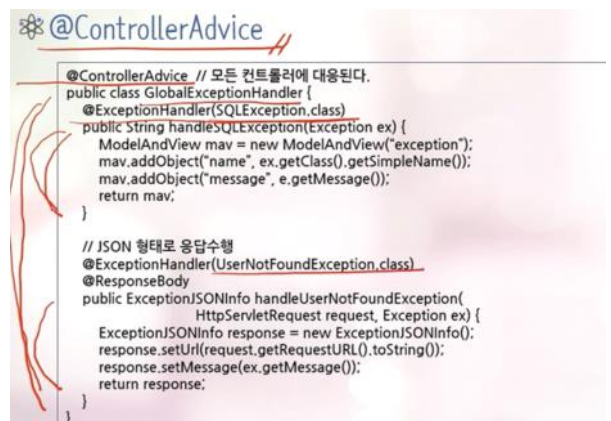
```
from fastapi.exceptions import HTTPException
from fastapi.responses import JSONResponse
from fastapi.routing import APIRoute

class NotFoundException(HTTPException):
    """NotFoundException"""
    status_code = 404
    detail = "Not Found"

def not_found_handler(request, exc: NotFoundException):
    return JSONResponse(status_code=exc.status_code,
                        detail=exc.detail)

class APIRoute(APIRoute):
    """APIRoute"""
    def get_route_handler(self):
        handler = super().get_route_handler()
        if handler:
            return handler
        return not_found_handler
```

Spring예외처리 예제



```
@ControllerAdvice
public class GlobalExceptionHandler {
    @ExceptionHandler({NotFoundException.class})
    public String handleNotFoundException(Exception ex) {
        ModelAndView mav = new ModelAndView("exception");
        mav.addObject("name", ex.getClass().getSimpleName());
        mav.addObject("message", ex.getMessage());
        return mav;
    }

    // JSON 형태로 응답수행
    @ExceptionHandler({UserNotFoundException.class})
    @ResponseBody
    public ExceptionJSONInfo handleUserNotFoundException(
        HttpServletRequest request, Exception ex) {
        ExceptionJSONInfo response = new ExceptionJSONInfo();
        response.setUrl(request.getRequestURL().toString());
        response.setMessage(ex.getMessage());
        return response;
    }
}
```