

# Problem 1

## Notes API 테스트

엔드포인트 기본: http://localhost:3000

### 1) 목록 조회 / 단건 조회

GET /notes	id 입력
	GET /notes/:id

### 2) 생성 (POST /notes)

title	content
-------	---------

CREATE

### 3) 수정 (PUT /notes/:id)

id	new title (optional)
new content (optional)	

UPDATE

### 4) 삭제 (DELETE /notes/:id)

id	DELETE
----	--------

public/index.html 페이지 기능이 구현되는 서버 만들기

## 목표

- http + fs/promises로 **REST API** 직접 구현
- **async/await**로 파일 I/O 처리
- 상태코드/에러/라우팅 기능 구현

## 엔드포인트 (notes 리소스) [각 +7점]

- Front와 서버 연결
- GET /notes : 전체 목록
- GET /notes/:id : 단일 조회
- POST /notes : 생성({title, content}) → **201 Created**
- PUT /notes/:id : 수정
- DELETE /notes/:id : 삭제 → **204 No Content**

## 체크리스트 [각 -7점]

1. **fs/promises** + **async/await**만 사용 (동기 함수 금지)
2. data/notes.json에 영속화 (없으면 생성)
3. **상태코드** 정확 전달: 200/201/204/400/404/500
4. **JSON Body** 파싱하여 구성
5. **라우팅** 직접 구현 (express 사용x)
6. **에러 공통 처리** (try/catch, 에러객체에 status 가능)

## Problem 2

페이징 링크 처리 Route 구현

- board1.html 하단의 페이지 버튼 기능을 정상 작동하도록 함 [14]
- 1 -> board1.html
- 2 -> board2.html
- 3 -> board3.html 으로 연결됨
- 페이지에서 자신의 페이지 버튼을 누르는 경우에 대해서도 문제 없도록 함 [14]

- Express를 사용한 서버 만들기
- board X.html 내의 버튼은 학교 링크와 연결되며, 이는 정상임
  - html 코드는 수정하지 않음
  - 파일 제출 시 public 코드는 삭제하고 제출함 (채점자의 public 폴더 사용)