

네트워크응용프로그래밍

중간평가 프로젝트 - 설계

중간평가 과제물 개요

- 과제물 제목: 중간평가 프로젝트
- 과제 유형: 개인 과제
- 과제물 내용
 - API 설계
 - API 구현
 - 로그인
 - 공공데이터 포털/네이버API 활용
 - CRUD
 - 웹크롤링
- 제출 기한: 10월 30일(월)
- 제출물:
 1. 프로젝트 결과보고서 (hwp 형식)
 2. 프로젝트 내용 설명 및 시연 동영상: 10분 내외
 - 팀즈로 직접 설명(화면 자막 사용 금지)과 시연을 녹화하고 이를 유튜브에 올린 다음
 - 링크를 제출

- 서버 API 설계
- 서버 API 구현
 - 로그인 기능
 - express-session 모듈 이용
 - 공공데이터포털/네이버API 등 외부 API 연동
 - 1개이상
 - API 구현
 - 포털의 내용을 가공하여 웹클라이언트에게 제공
 - axios 모듈 이용
 - To do list API 구현
 - To do list에 대하여
 - post, put, delete 동작 구현
 - Postman을 이용하여 테스트
 - 웹 크롤링 API 구현
 - crawling.js 예제 코드 활용

- 평가 기준
 - 구현 완성도(30%)
 - 구현 설명, 이해도(50%)
 - 보고서(20%)
- 각 영역별 점수 배정
 - API 설계 명세: 10%
 - 로그인 구현: 20%
 - 공공포탈 활용: 25%
 - CRUD: 25%
 - 웹클롤링: 20%

서버 API 설계 예시

메소드	경로	설명	
GET	/devices	모든 device 정보를 조회합니다.	1
GET	/devices/:id	특정 device 정보를 조회합니다.	2
POST	/devices	device를 추가 합니다. { "deviceId": "name": "latitude": "longitude": "description": }	3
POST	/devices/:id	특정 device에 해당하는 방문자 정보를 업로드합니다. { "deviceId": "visitor": "temperature": }	4
POST	/devices/alarms/:id	특정 device의 경보 상황을 전달합니다. { "deviceId": "visitor": "contextId": "contextValue": }	5
DELETE	/devices/:id	특정 device의 정보를 제거합니다.	6

서버 API 설계 예시...

처리 번호	요청	비즈니스 로직 (요청을 처리하기 위한 내부 절차)	응답
1			
2			
3			

서버 API 설계 예시...

처리 번호	요청	비즈니스 로직 (요청을 처리하기 위한 내부 절차)	응답
4			
5	경보 상황 정보	경보 상황 정보를 관리자 화면에 표시	정상 요청에 대하여 HTTP 정상 응답
6			

데이터 테이블 설계

```
const board = {  
  "id": ++numOfBoard,  
  "user_id": req.body.user_id,  
  "date": new Date(),  
  "title": req.body.title,  
  "content": req.body.content  
};
```


데이터베이스 설계

```
var DeviceSchema = mongoose.Schema({ //스키마(Schema) 설정
  deviceId: {type: Number, require:true},
  name: { type: String, require: true},
  latitude: { type: Number, require: true},
  longitude: { type: Number, require: true},
  description: { type: String, require: true},
  register_date: { type: Date, default:Date.now}
});
```

```
var DataSchema = mongoose.Schema({ //스키마(Schema) 설정
  deviceId: {type: Number, require:true},
  temperature: { type: Number, require: true},
  humidity: { type: Number, require: true},
  register_date: { type: Date, default:Date.now}
});
```