
REPORT

Node를 활용한
Profiler 프로그램 분석

윤소영

20220854

담당 교수

강영명 교수님



성결대학교
SUNGKYUL UNIVERSITY

1. 프로그램 개요

이 프로그램은 Node.js 와 Express를 기반으로 구축된 웹 애플리케이션이다. 프로그램의 주요 목적은 사용자가 파일을 업로드하고, 업로드된 파일의 데이터를 효율적으로 관리하고 시각화 하는 데에 중점을 두고 있다. 이를 통해 데이터의 인사이트를 쉽게 얻을 수 있다.

주요 기능

- 파일 업로드
 - 사용자는 파일을 업로드 할 수 있으며, 업로드된 파일은 서버의 특정 디렉토리에 저장된다.
 - 업로드된 파일의 데이터는 파싱되어 MySQL 데이터 베이스에 저장된다.
- 파일 목록 조회 및 관리
 - 사용자는 업로드 된 파일의 목록을 조회할 수 있다.
 - 업로드 된 파일을 삭제할 수 있다.
- 데이터 조회 및 시각화
 - 저장된 데이터는 다양한 통계치(최소값, 최대값, 평균값)로 요약되어 조회할 수 있다.
 - 선택된 데이터는 차트 형태로 시각화되어 사용자에게 제공된다.

사용된 기술

- 서버 측 (server.js)
 - Express.js 를 사용하여 서버를 설정하고, 엔드 포인트를 정의.
 - MySQL 데이터 베이스에 연결하고, 필요한 테이블을 생성.
 - 파일 업로드, 파일 목록 조회, 파일 삭제, 데이터 조회 등의 기능을 제공.
- 클라이언트 측(upload.js, files.js, chart.js)
 - HTML 폼을 통해 파일 업로드를 처리.
 - 서버에서 데이터를 가져와서 DOM 요소를 업데이트하고, 사용자와의 상호작용을 처리.
 - chart.js를 사용하여 데이터를 시각화.

2. 프로그램의 수행 절차

2.1 사전 수행

```
user@cooooow0 bin % cd /usr/local/mysql/bin
user@cooooow0 bin % ./mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.3.0 Homebrew

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

[그림 1] mysql 접속

터미널을 켜고, MySQL을 설치한 경로로 이동한다. 이후 ‘./mysql -u root -p’ 명령어를 입력하여 mysql에 접속한다.

```
1  -- 데이터베이스 생성
2  CREATE DATABASE IF NOT EXISTS my_database;
3
4  -- 데이터베이스 사용
5  USE my_database;
6
7  -- profile_data 테이블 생성
8  CREATE TABLE IF NOT EXISTS profile_data (
9      core VARCHAR(255),
10     task1 INT,
11     task2 INT,
12     task3 INT,
13     task4 INT,
14     task5 INT
15 );
16
17 -- file_data 테이블 생성
18 CREATE TABLE IF NOT EXISTS file_data (
19     id INT AUTO_INCREMENT PRIMARY KEY,
20     filename VARCHAR(255),
21     originalname VARCHAR(255)
22 );
```

[그림 2] DB 생성

터미널에 위와 같은 명령어를 입력하여 데이터 베이스에 테이블을 생성해준다.

```
// MySQL 데이터베이스 설정
const db = mysql.createConnection({
  host: 'localhost',
  user: 'my_user',
  password: 'password',
  database: 'my_database'
});
```

[그림 3] 파일 내 설정

server.js 파일 상단에 위치한 파일에서 유저 이름과 비밀번호, 데이터 베이스 명을 알맞게 적어준다.

2.2 프로그램 시작

```
47 db.query(createProfileTableQue
문제 출력 디버그 콘솔 터미널 포트
user@coooow0 웹 응용 % node server.js
Server is running on port 3000
MySQL 연결 성공
```

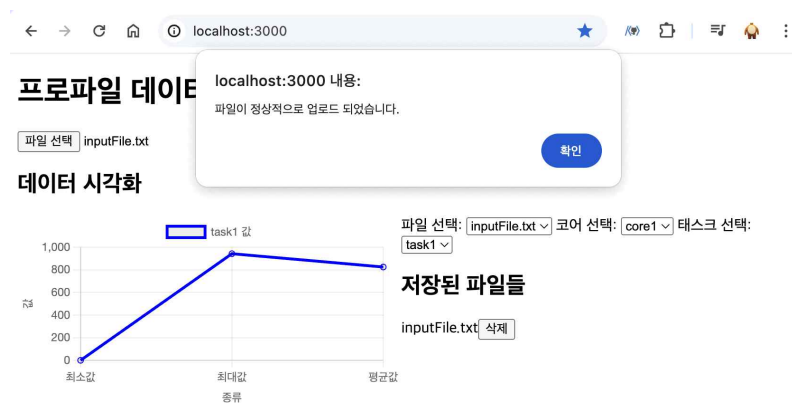
[그림 4] 서버 연결

터미널에 `node server.js`를 작성하고 주소창에 `localhost:3000`를 검색해 들어간다.



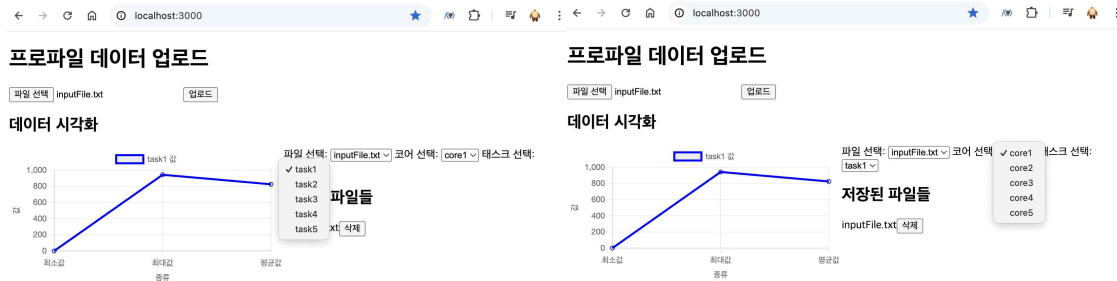
[그림 5] 초기 웹 페이지

파일을 업로드하지 않은 상태의 웹 페이지이다.



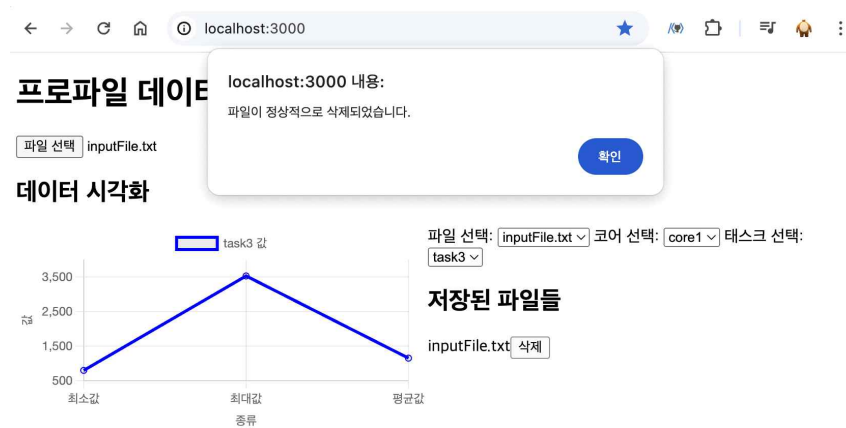
[그림 6] 파일 업로드 후 페이지

파일을 선택하고, 업로드 버튼을 누르면 파일이 정상적으로 업로드 되었다는 창이 뜬다. 업로드 한 파일을 밑에서 확인할 수 있으며 삭제할 수 있다.



[그림 7] 코어/태스크 선택

Core와 Task를 선택하여 최대값, 최소값, 평균값을 볼 수 있다.



[그림 8] 파일 삭제

저장된 파일들 항목에서 파일을 삭제할 수 있다.



[그림 9] 파일 선택

2개 이상의 파일을 업로드 할 수 있고, 파일을 선택하여 코어, 태스크 값을 볼 수 있다.

3. 코드 분석

3.1 server.js

- 목적: Node.js를 사용하여 서버를 설정, 파일 업로드 및 데이터베이스 관리 기능을 제공.
- 기능
 - 서버 설정: Express.js를 사용하여 서버를 설정하고, 파일 업로드 처리.
 - 파일 업로드: Multer 미들웨어를 사용하여 파일 업로드 처리.
 - 데이터베이스 연결: MySQL 데이터베이스에 연결하고, 테이블 생성.
 - 데이터 처리: 업로드된 파일을 읽고, 데이터를 데이터베이스에 삽입.
 - 파일 관리: 업로드된 파일의 목록을 제공, 파일 삭제 가능.

3.2 public/chart.js

- 목적: 데이터 시각화. 선택된 파일, 코어 및 태스크에 따라 차트 생성.
- 기능
 - 옵션 로드: 코어와 태스크 옵션을 로드하여 드롭다운 메뉴에 추가.
 - 파일 목록 로드: 서버에서 파일 목록을 가져와서 파일 드롭다운 메뉴와 파일 목록을 업데이트.
 - 차트 생성: 선택도니 파일, 코어, 태스크에 따른 데이터를 서버에서 가져와 chart.js를 사용해 차트 생성.

3.3 public/files.js

- 목적: 파일 목록 로드, 파일 삭제 기능 제공
- 기능
 - 파일 목록 로드: 서버에서 파일 목록을 가져와서 파일 목록 및 파일 선택 드롭 다운 메뉴 업데이트.
 - 파일 삭제: 파일 삭제 요청을 서버에 보내고, 삭제 후 파일 목록을 갱신.

3.4 public/upload.js

- 목적: 파일 업로드 기능 처리. 업로드 후 코어와 태스크 옵션 및 파일 목록 갱신
- 기능
 - 옵션 및 파일 목록 갱신: 파일 업로드 후 코어와 태스크 옵션 및 파일 목록을 갱신.
 - 초기 로드: 페이지가 로드될 때 자동으로 파일 목록을 가져오고, 코어와 태스크 옵션 설정.