

웹 시스템 설계 HW#1

추민솔 TA

cmss1217@ajou.ac.kr

2025-10-02

목차

1 과제	2
1.1 목표	2
1.2 프로그램 개요	2
1.3 프로그램 구성	2
1.4 구현 요구사항	2
1.4.1 서버 구현 (<code>server.js</code>)	2
1.4.2 클라이언트 구현 (<code>index.html, style.css</code>)	3
1.5 동작 흐름	3
1.6 예시 실행 시나리오	3
1.7 유의 사항	4
1.8 스켈레톤 코드 공통 설정 설명	4
1.9 제출 요구사항	4
2 Appendix	5

1 과제

1.1 목표

- HTML과 CSS를 활용하여 간단한 웹 페이지를 구성하고 Express 기반 REST API를 호출해 동작을 검증한다.
- Node.js + Express로 항공편/예약 REST API를 설계·구현하고, 요청·응답 형식과 상태코드(2xx/4xx)를 올바르게 사용한다.
- 입력 검증(좌석 형식·범위, 중복 좌석), 에러 메시지 처리(400/409 등), 메모리 기반 데이터 관리의 기본 원리를 학습한다.

1.2 프로그램 개요

- 본 과제는 비행기 좌석 예약 시스템의 서버 API와 간단한 클라이언트 페이지를 구현하는 것이다.
- 사용자는 웹페이지의 폼을 통해 이름과 좌석을 입력해 예약을 등록할 수 있고, 편성 정보/예약 목록을 조회할 수 있다.
- 서버는 항공편 정보와 예약 목록을 제공하며, 예약 생성 시 유효성 검증과 중복 좌석 방지 로직을 수행한다.

1.3 프로그램 구성

- 본 과제는 서버(백엔드)와 클라이언트(프론트엔드)로 구성된다.
 - `server.js` Express 기반 REST API 서버. 항공편 정보 및 예약 생성/조회 기능 제공.
 - `index.html` 클라이언트 페이지. form 기반으로 편성/예약 목록 조회 및 예약 생성 요청을 전송한다.
 - `style.css` 스타일시트. 별도 파일로 작성하여 `index.html`에서 `<link>`로 포함 시킨다.

1.4 구현 요구사항

1.4.1 서버 구현 (`server.js`)

필수 엔드포인트 아래 경로명은 반드시 유지할 것.

- GET `/api/flight`
항공편 정보를 JSON으로 반환한다. 필수 필드는 다음과 같다.
 - `code, from, to, date, departureTime, arrivalTime`
 - `rows: 총 행 수 (예: 12), cols: 사용 가능한 열 배열 (예: ["A", "B", "C", "D", "E", "F"])`
- GET `/api/reservations`
현재 예약 배열을 JSON으로 반환한다. 각 예약 객체의 예시는 `{ id, name, seat, ts }` 형태이다.

- POST /api/reservations

요청 본문(name, seat)을 검증 후 새로운 예약을 생성한다. 티

- name: 앞뒤 공백을 제거한 길이가 2자 이상이어야 한다. (미만이면 400 Bad Request)
- seat: <행번호><열문자> 형식을 만족해야 한다. (형식 불일치 시 400 Bad Request)
- 좌석은 항공편 범위 내여야 한다. 즉, 1 ~ rows 범위의 행과 cols에 포함된 열만 허용한다. (범위 밖이면 400 Bad Request)
- 이미 예약된 좌석으로 요청할 경우 409 Conflict로 거부한다.
- 성공 시 201 Created와 함께 생성된 예약 객체({ id, name, seat, ts })를 반환한다.
 - * id: 고유 예약 식별자
 - * name: 예약자 이름
 - * seat: 좌석 번호
 - * ts: 예약 시각 (new Date().toISOString() 등으로 생성 가능)

1.4.2 클라이언트 구현 (index.html, style.css)

필수 요소(최소 요건)

- 탑승객 이름 입력 필드 (name)와 좌석 입력 필드 (seat)
- 예약 생성 폼: action="/api/reservations" & method="post"로 전송
- 편성 정보 조회 폼: action="/api/flight" & method="get"
- 예약 목록 조회 폼: action="/api/reservations" & method="get"

1.5 동작 흐름

1. 메인 페이지(index.html)에서 /api/flight 조회 버튼을 누르면 현재 항공편 코드, 일정, 좌석 범위를 확인할 수 있다.
2. /api/reservations 조회를 통해 현재까지 생성된 예약 목록(JSON)을 확인한다.
3. 사용자가 이름과 좌석을 입력 후 “예약 생성”을 제출하면 POST /api/reservations가 호출되어 서버 검증을 통과하면 예약이 생성되고, 응답(JSON)으로 결과가 표시된다.
4. 서버는 name 혹은 seat 누락, 좌석 형식 오류, 범위 밖 좌석, 중복 좌석 등에 대해 적절한 4xx 응답과 메시지를 반환한다.

1.6 예시 실행 시나리오

- 초기 화면에서 “편성/좌석 범위 조회”를 눌러 항공편 정보(code, from/to, date, departureTime/arrivalTime, rows/cols)를 확인한다.
- 유효한 이름(≥2자)과 좌석 입력 후 예약 생성 시 201과 함께 { id, name, seat, ts }가 반환된다.
- 이미 예약된 좌석으로 재시도하면 409 Conflict와 오류 메시지가 반환된다.

1.7 유의 사항

- 엔드포인트 경로명은 반드시 유지해야 한다. (/api/flight, /api/reservations, /api/reservations (POST))
- 서버는 메모리 기반으로 동작하며, 재시작 시 예약 목록이 초기화되어도 문제 없다.
- flight.rows와 flight.cols 값이 변경되더라도 검증 로직은 이를 참조하여 좌석 범위를 판단해야 한다.

1.8 스켈레톤 코드 공통 설정 설명

본 과제에서 제공되는 `server.js` 스켈레톤 코드에는 학습 편의를 위해 몇 가지 공통 설정이 포함되어 있다. 이는 REST API 서버 동작에 필요 및 참고 가능한 설정으로, 아래 내용을 참고하여 이해할 수 있다.

path 모듈

- `const path = require("path");`
- Node.js 내장 모듈로, 운영체제와 무관하게 안전하게 파일 경로를 다룰 수 있게 해준다.
- `app.use(express.static(path.join(__dirname)));` 설정을 통해 현재 디렉터리(`__dirname`)를 정적 파일 제공 경로로 지정한다. 따라서 `index.html`, `style.css` 같은 프론트엔드 리소스를 클라이언트가 직접 요청할 수 있다.

app.use(...) 공통 미들웨어

- `app.use(express.json());` 요청 본문(body)이 JSON 형식일 때 자동으로 파싱하여 `req.body` 객체로 접근 가능하게 한다. (예: POST /api/reservations)
- `app.use(express.urlencoded(extended: true));` HTML form 전송 시 기본 URL-encoded 형식을 파싱할 수 있게 한다. 즉, 클라이언트에서 form을 통해 보낸 데이터도 `req.body`로 받을 수 있다.
- `app.use(express.static(...));` 앞서 설명한 대로, `index.html`과 같은 정적 자원을 제공하기 위해 사용된다.

1.9 제출 요구사항

- HW1_이름_학번.zip 형식으로 압축하여 제출

예시

```
HW1_홍길동_202401234.zip/
├── server.js
├── index.html
└── style.css
```

- 제출 기한: 10/15 (수) 23:59 (지각 제출 시 D+1 -40%, D+2 -70%, D+3 -100% 감점)

2 Appendix

- 아래 내용은 간략한 동작 예시로 UI 등을 완벽하게 똑같이 구현할 필요는 없다.
- 아래 예시는 일부 예외 사항의 처리 예시는 포함하고 있지 않다.

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성

탑승객 이름
홍길동
2 글자 이상 입력하세요.

좌석
예: 3C

형식: 행번호 + 열(A~F). Ex. 7B

예약 생성

편성/좌석 범위 조회

현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다.

조회

예약 목록 조회

현재까지 생성된 예약 목록을 조회합니다.

조회

Figure 1: 초기 화면 예시

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성

탑승객 이름

2 글자 이상 입력하세요. 이 입력란을 작성하세요.

좌석

형식: 행번호 + 열(A~F). Ex. 7B

예약 생성

편성/좌석 범위 조회

현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다.

조회

예약 목록 조회

현재까지 생성된 예약 목록을 조회합니다.

조회

(a) 빈 이름 처리

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성

탑승객 이름

2 글자 이상 입력하세요.

좌석

형식: 행번호 + 열(A~F). Ex. 7B

예약 생성

편성/좌석 범위 조회

현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다.

조회

예약 목록 조회

현재까지 생성된 예약 목록을 조회합니다.

조회

(b) 빈 좌석 처리

Figure 2: 빈 값 처리

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성 탑승객 이름 <input type="text" value="Test"/> <small>2 글자 이상 입력하세요.</small> 좌석 <input type="text" value="4F"/> <small>형식: 행번호 + 열(A~F). Ex. 7B</small> 예약 생성	편성/좌석 범위 조회 현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다. 조회
---	---

(a) 예약 요청

pretty print 적용

```
{ "id": 2, "name": "Test", "seat": "4F", "ts": "2025-10-01T01:00:04.981Z" }
```

(b) 예약 요청 응답

Figure 3: 예약 생성

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성

탑승객 이름
홍길동
2 글자 이상 입력하세요.

좌석
예: 3C
형식: 행번호 + 열(A~F). Ex. 7B

예약 생성

편성/좌석 범위 조회

현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다.

조회

예약 목록 조회

현재까지 생성된 예약 목록을 조회합니다.

조회

(a) 좌석 범위 요청

```
pretty print 적용 
{
  "id": "F124",
  "code": "AJ101",
  "date": "2025-10-03",
  "departureTime": "10:30",
  "arrivalTime": "11:40",
  "from": "PAL",
  "to": "HYK",
  "rows": 12,
  "cols": [
    "A",
    "B",
    "C",
    "D",
    "E",
    "F"
  ]
}
```

(b) 좌석 범위 요청 응답

Figure 4: 좌석 범위 조회

비행기 좌석 예약

폼을 제출하면 브라우저가 JSON 응답 화면으로 이동합니다.

좌석 예약 생성

탑승객 이름

좌석

형식: 행번호 + 열(A~F). Ex. 7B

예약 생성

편성/좌석 범위 조회

현재 설정된 항공편 코드, 일정, 좌석 행/열 범위를 조회합니다.

조회

(a) 예약 목록 요청

```
pretty print 적용
[{"id": 1, "name": "홍길동", "seat": "1A", "ts": "2025-09-24T16:18:33.719Z"}, {"id": 2, "name": "Test", "seat": "4F", "ts": "2025-10-01T01:04:54.139Z"}]
```

(b) 예약 목록 요청 응답

Figure 5: 예약 목록 조회