

웹응용프로그래밍 2023-

# Ch01-02. Node and JS Issues : 실습 part2

---

부산대학교 정보컴퓨터공학부  
Prof. 김원석

# 실습

---

7주차 실습

# Todo

## ❖ Promise 및 async/await를 활용 해 사용자 관리 시스템 만들기

- 실습
  - 주어진 Template에 따라서 실습 진행
  - 최대한 실습시간 내에 완성하는 것을 목표로 함
  - 질문 사항 있을 시 조교에게 질문
  - 실습시간 내에 완성하지 못할 경우 **목요일 23:59까지** 제출 허용
- 기본 요구 조건
  - 터미널에서 node \*\*\*\*.js 입력 시 터미널 상에서 실행되어야 함
  - 작성 완료 시 학번\_이름\_week07.js로 Plato에 제출
  - 코드 작성시의 요구조건 필히 준수

# Todo

## ❖ 사용자 관리 시스템

- 요구조건 (1)
  - 사용자의 입력을 받아 사용자 정보 출력 또는 사용자를 추가함
  - 사용자 정보는 다음과 같은 형식으로 저장됩니다.
    - 이름
    - 이메일 주소
  - user.json 파일로 저장
  - 실행 시 '1. 사용자 정보 읽기' '2. 사용자 추가' 를 출력 후 '선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): ' 를 출력합니다.
    - 이 때 1 또는 2 또는 q가 아닌 다른 입력을 받을 시 '잘못된 선택입니다. 다시 선택해주세요.'를 출력합니다.
  - 1을 입력 시 '사용자 정보 읽기...' 을 출력 후 fs module을 활용 해 user.json파일을 읽어 사용자 목록을 출력합니다.
    - 저장된 포맷은 샘플로 제공된 users.json파일 참고
    - 이 때 파일이 없거나 빈 파일일 경우 '사용자가 없습니다' 를 출력합니다.
    - 아니라면 사용자 목록을 출력합니다.
      - 이 때 '사용자 번호 (3001번부터 시작). 이름 (메일주소)'로 출력합니다.
      - ex) 3001. Elon Musk ([Musk123@naver.com](mailto:Musk123@naver.com))

# Todo

## ❖ 사용자 관리 시스템

### • 요구조건 (2)

- 2를 입력 시 새 사용자를 입력 받습니다. 순차적으로 '새 사용자의 이름을 입력하세요 : ' 출력 후, 이름을 입력받고 '새 사용자의 이메일을 입력하세요 : ' 를 출력 후, 이메일을 입력 받습니다.
  - 이 때 이름 또는 메일 주소가 중복될 경우 '이름 : (입력한 이름) 또는 이메일 : (입력한 이메일) 이 중복됩니다.' 를 Error로 출력하고 프로그램이 종료됩니다.
    - Hint : some() 함수로 중복 체크 및 push함수로 배열에 요소 추가 가능
  - 저장 완료 후 delay(time)를 이용해 일정 시간 딜레이 및 '새 사용자가 추가되었습니다.(time)' 출력
- q를 입력 시 '프로그램을 종료합니다.' 를 출력 후 프로그램을 종료합니다.

### • 코드 작성 시 요구조건

- readUserData (사용자 정보 이는 promise chaining 및 catch를 필히 사용하여 완성
  - promise chaining을 이용 할 때 첫번째 then은 읽은 JSON 문자열을 파싱, 두번째 then은 파싱된 결과 반환 catch는 파일이 없을 경우 빈 배열을 반환하고, 그 이외의 오류가 발생할 경우 throw error를 하게 작성
    - Hint : 제공된 fs.readFile()은 promise를 return합니다.
    - Hint : fs.readFile(userDataFile, 'utf8') .then() ~~~
- delay()의 경우 setTimeout을 활용 해 Promise를 return하도록 구현!

# 참조

## ❖ 추가 설명

- JSON.parse(data) : JSON 문자열의 구문을 분석하고, 결과를 이용해 JavaScript 값이나 객체를 생성합니다.

예시 코드

```
1 const json = '{"result":true, "count":42}';
2 const obj = JSON.parse(json);
3
4 console.log(obj.count);
5 // Expected output: 42
6
7 console.log(obj.result);
8 // Expected output: true
9
```

결과

```
> 42
> true
```

- some() 메서드는 배열 안의 어떤 요소라도 주어진 판별 함수를 적어도 하나라도 통과하는지 테스트합니다.

예시 코드

```
1 const array = [1, 2, 3, 4, 5];
2
3 // Checks whether an element is even
4 const even = (element) => element % 2 === 0;
5
6 console.log(array.some(even));
7 // Expected output: true
8
```

결과

```
> true
```

# 예시

## 시작 시

```
1. 사용자 정보 읽기
2. 사용자 추가
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료):
```

## 1 입력 시

```
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): 1
사용자 정보 읽기...
사용자 목록:
3001. Elon Musk (Musk123@naver.com)
```

## q 입력 시

```
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): q
프로그램을 종료합니다.
PS C:\Users\user\Desktop\2023학년도2학기조교\Node>
```

## 이외의 글자 입력 시

```
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): 3
잘못된 선택입니다. 다시 선택해주세요.
```

## 2 입력 시

```
1. 사용자 정보 읽기
2. 사용자 추가
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): 2
새 사용자의 이름을 입력하세요: Jon Smith
새 사용자의 이메일을 입력하세요: Js123@pusan.ac.kr
```

## 2 입력 후 일정 시간 후(예시는 2초)

```
1. 사용자 정보 읽기
2. 사용자 추가
선택하려는 번호를 입력하세요 (또는 "q"를 입력하여 종료): 2
새 사용자의 이름을 입력하세요: Jon Smith
새 사용자의 이메일을 입력하세요: Js123@pusan.ac.kr
새 사용자가 추가되었습니다.(2000)
```

## 2 입력 후 중복 입력 시(에러 발생)

```
Error: 이름 : Elon Musk 또는 이메일 : hi 이 중복됩니다.
    at addUser (C:\Users\user\Desktop\2023학년도2학기조교\Node\week07_answer2.js:61:11)
    at async main (C:\Users\user\Desktop\2023학년도2학기조교\Node\week07_answer2.js:103:7)
Node.js v18.17.1
```