

Node.js Server

Express Practice

2023-1 T.G.WinG 강의식 스터디

배승호



KHU

TCWinG

Rest API 통신

- ❖ 클라이언트와 서버와의 통신을 지금까지 REST API에 기반하여 하였음
- ❖ 이는 서버와 서버의 통신에서도 마찬가지로 REST API를 사용할 수 있음
 - EX) A서버에서 B서버에 데이터를 요청하는 행위
 - 서버와 서버의 통신이지만, 해당 통신에서는 A서버는 B서버의 클라이언트가 됨
 - 즉, 통신에서 절대적인 서버는 없고 서버도 클라이언트가 될 수 있음
 - 실제 서비스에서 예를 들자면 Web Server와 Database Server를 예를 들 수 있음



Axios

❖ REST API를 요청할 수 있는 라이브러리

❖ 모듈 설치

```
// terminal  
npm install axios
```

❖ 모듈 импорт

```
// app.js  
const axios = require("axios")
```



Axios 사용 예제

- ❖ Axios는 Promise 객체를 반환하기 때문에 **async 키워드로 선언된 함수 내에서 사용해야 편함**

```
const axios = require("axios")

axios.get("https://www.naver.com")
  .then((data) => {
    console.log(data.data)
  })
```



```
const axios = require("axios")

const func = async () => {
  const data = await axios.get("https://www.naver.com")
  console.log(data.data)
}

func()
```

라우터 콜백에서 Axios 사용하기

- ❖ Axios는 라우터 콜백 내부에서도 사용할 수 있음

```
app.get('/', async (req, res) => {  
  const data = await axios.get("https://www.naver.com")  
  res.send(data.data)  
})
```

- ❖ www.naver.com에서 반환된 응답을 다시 응답해주는 코드



실습 1

- ❖ 163.180.118.157:5500/study/node/members에 GET요청을 보내 멤버 데이터를 받아온 후 클라이언트에게 멤버 데이터를 다시 보내주는 라우터



실습 2

- ❖ 163.180.118.157:5500/numbers에 GET요청을 보내 숫자 리스트를 받은 후 모든 수를 더한 값을 응답하여라



실습 3

- ❖ 1:1 코딩 대결 게임을 만들고자 한다
- ❖ 이용자들이 게임 매칭 버튼을 누르면 게임 서버에 자신의 rating정보와 닉네임을 전송한다
- ❖ 게임은 비슷한 실력의 유저와 이루어져야 하기 때문에 자신과 50점 이상 차이는 사람과는 매칭하지 않는다
- ❖ 대기열에는 매칭되지 못한 사람들의 정보가 들어있다
 - 데이터는 (GET)163.180.118.157:5500/game/queue에서 확인이 가능하다
- ❖ 자신의 캐릭터가 대기열의 유저와 매칭이 가능하면 상대 유저의 데이터를 응답한다
- ❖ 만일 매칭할 수 없으면 "대기열에 등록되었습니다" 를 응답한다
- ❖ 자신의 캐릭터 정보는 GET요청의 querystring으로 전송한다

감사합니다

