

인터페이스 가이드 문서 작성

클라이언트 개발자와 협업하는 방법

개정이력

버전	변경일	변경사유	변경내역
1.0	2024-01-11	최초작성	최초작성

데이터 공통사항

name	desc	type	etc
userId			

Response

아래와 같은 규칙을 따른다.

- JSON 포맷의 데이터이며, 필드명은 소문자로 시작하는 camel notation을 사용.
- 다음 json object type으로 응답

```
{
  "userId" : 1,
  "userName" : "hong Gil Dong",
  "department" : "HRM",
  "postCnt" 2,
  "logDate" : [
    {
      "date": 2023-1-11 00:12:00
    },
    {
      ...
    },
  ]
},
{
  ...
}
```

name	desc	type	note
userId	요청 유저 아이디	number	
userName	요청 유저 이름	String	
department	요청 유저 부서	String	
postCnt	요청 유저 게시글 작성 수	number	
logDate	요청받은 유저 방문 기록	json array	

SW 활용률 API

어떤 사용자가 우리의 SW를 활용하는지 정보 제공

Request Parameters

name	Type	desc	Mandatory	
userId	number	number	o	

Response Body

```
{
  "userId" : 2,
  "userName" : "hong Gil DDong",
  "department" : "IT",
  "postCnt" 1,
  "logDate" : [
    {
      "date": 2023-1-11 00:20:00
    },
    {
      "date": 2023-1-09 10:09:00
    },
  ]
}
```

2주차 직무 부트캠프

1) Rest API

- ✓ Rest란 Respresentational State Tranfer의 약어로서, 자원을 이름으로 구분하여 자원의 상태를 주고 받는 모든 것을 의미
- ✓ HTTP URI를 통해 자원을 명시하고, POST, GET, PUT, DELETE, PACTH를 통해 URI(자원)에 대한 CRUD Operation을 적용하는 것
- ✓ Rest API 설계 예시
 - URI는 동사보단 명사, 대문자보다 소문자 사용
 - 마지막에 /를 포함 x
 - _대신 - 사용
 - 파일확장자는 URI에 포함 x

2) HTTP 통신

- ✓ HTTP란 Hypertext Transfer Protocol의 약어로서, 텍스트 기반의 통신 규약으로 인터넷에서 데이터를 주고받을 수 있는 프로토콜
- ✓ 클라이언트-서버 모델을 따르며, TCP/IP위에서 작동

3) 브라우저에 URL을 입력 후 요청하여 서버에서 응답하는 과정

- ① 웹 브라우저에 URL 입력
 - https://는 통신 프로토콜
 - TLS를 사용하여 서버에 연결하도록 지시
 - HTTPS를 사용하면 암호와 같이 위험한 정보들이 브라우저와 서버가 데이터를 교환할 때 암호화 됨.
- ② 웹 브라우저가 도메인명으로 IP주소 조회
 - 브라우저는 URL을 가지고 인터넷에서 연결할 서버를 파악하기 위해 입력한 도메인을 사용하여 웹 사이트를 호스팅하는 서버의 IP주소를 알아냄 → DNS 사용
 - 고유 캐시 → OS 캐시 → 라우터의 로컬 네트워크 캐시 → 회사 네트워크 또는 ISP의 DNS 캐시 확인
- ③ 웹 브라우저가 서버와의 TCP 연결 시작
 - 웹 브라우저가 인터넷에서 서버를 찾으면 웹 서버와 TCP 연결을 설정하고, HTTPS를 통해 통신 시작

④ 웹 브라우저가 HTTP 요청을 서버로 전송

- 웹 브라우저가 서버에 연결되면, HTTPS 프로토콜에 대한 통신 규칙 따름.
- HTTP 요청

<요청 라인>
GET /table/user HTTP/2
- GET, POST, PUT, DELETE 등 요청 메서드 : GET
- 요청된 리소스를 가리키는 경로 : /table/user
- 통신할 HTTP 버전 : HTTP/2

- 웹 서버가 클라이언트로부터 요청을 받으면 서버는 요청을 처리하고 응답을 다시 전송

⑤ 웹 서버가 요청을 처리하고 응답을 전송

- 웹 서버는 콘텐츠를 가져오고 응답을 생성하여 클라이언트로 다시 전송

⑦ 웹 브라우저가 콘텐츠 렌더링

- 웹 브라우저가 서버로부터 응답을 받으면 응답 헤더를 검사하여 렌더링 정보 확인