

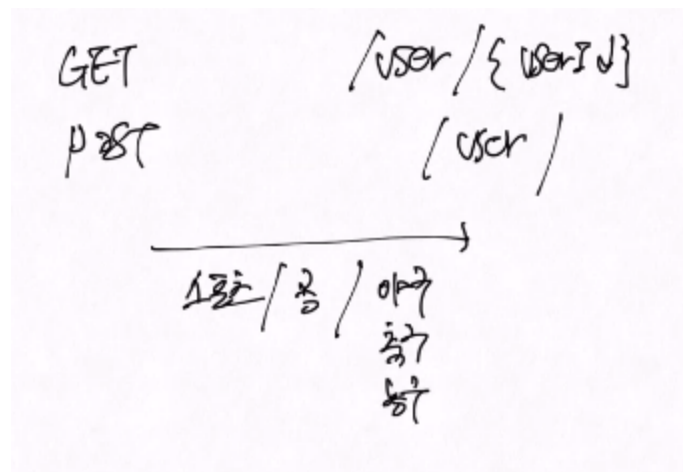
6주차 수업

HTTP 5가지 메소드

- POST
- PATCH
- GET
- DELETE
- PUT

REST API

5가지 메소드만 보고도 어떤 역할을 하는지 아주 깔끔하고 규칙성 있게 만든 url 구성이다.



서버를 사용하는 로그인

- 쿠키&세션: 쿠키는 클라이언트 저장소, 세션은 서버의 저장소이다. 로그인을 하면 서버에서 SID이라고 하는 세션 ID를 만들어서 세션에 저장하고 쿠키에 전달한다. 이 두 SID를 비교해서 이용자인지 확인한다. 단 SID는 모든 정보에 접근할 수 있어서 유저의 모든 정보에 접

근할 수 있다. 보안이 안 좋아서 쿠키엔 세션 방식은 지금 안 쓴다. SID에 유저의 모든 정보가 걸려있다.

구현이 쉽다! 보안이 약하다!

- OAuth: OAuth는 털리면 일부만. 로그인 시 사용자가 어떤 정보까지 접근이 가능하게 할지 정한다. 동작 방식은 JWT와 비슷하다.

보안은 좋다! 구현이 어렵다!

- JWT: Json Web Token의 약자이다. 토큰안에 몇번째 유저이고 유저가 어떤 사람인지 정보를 약간 가짐.

OAuth와 JWT의 공통적 동작 방식

둘 다 토큰이 있다.

1. Access token: 서버에 들어 갈 수 있는 지 자격을 주는 token
수명이 짧다 몇분~몇십분 그 시간이 지나면 만료돼서 못 씀
2. Refresh token: 재발급 토큰
수명이 길다 몇달, access token이 만료되면 refresh token으로 다시 받는다.

해커가 열심히 해킹해서 access token을 훔어도 금방 만료되어서 보안성에 도움이 된다.

JWT 웹에서 만들어 볼 수 있다. 토큰이 바뀌는 것을 볼 수 있다.

페이징

서버에서 정보를 불러올 때 한번에 다 불러오는 것이 아니고 페이지 단위로 불러오는 것이다.

구글링을 해보면 정보가 엄청 많다. 앱에서 한번에 큰 정보를 불러오면 버벅이기 때문에 이걸 막기 위해 페이징이 필요하다.

페이징3라이브러리가 있는데 어렵다... live data, 코루틴, 등등..

싱글톤 패턴

어떤 화면에서도 다 쓰는 부분이 있으면 basedp aksemdjwnsek.

biniding을 무조건 bull destroy binding null

비동기 interface를 implement하는 방식으로 비동기를 처리

팁

firebase 스토리지

로컬의 사진을 firebase에 옮기고 그 링크만 서버에 보낸다.

초반 서버 개발 속도 느림 3~4일 api 안 나옴

그 때 화면을 많이 만들어야 함

API 10 몇 개는 연결

꼭 필요한 화면 먼저 만들자

2주안에 다 만들기는 힘들다. 제한 된 시간에 얼마나 효율적인 수업을 하면 좋다

서버도 클라이언트를 모르고 클라이언트도 서버를 모른다. 애매하면 질문을 하자! 서버는 서버 개발 리더에게 물어보자

언제 시간

2시까지 기획하고 밥 먹음 밥 먹으면서 계속 만져보자

UI를 짜다가 갈아 엮으면 안된다.

질문 처음에는 없는데 나중에 많아질거다 미리 하자

commit과 개발 일지 하루 한번 감점 요

아이콘은 그냥 캡처!

아이콘은 원래 서버에서 받는 것이 맞다. 합의를 하자. 서버로 힘들다.

캡처는 내가 하고 서버에서 받자 등등

앱을 삭제해서 남아야 하는 것은 서버에 저장해야 한다. sharesprf는 삭제해서 없어지는 거

지금 홈 api 빠르게 필요하면 우선적으로 ㄱㄱ

다음주 토요일 시작 13

피드백은 2

1차 4-5일: 회원가입 홈화면 구성

postman을 사용하면 서버가 잘못된 것을 알 수 있다.