

# 로그인(Sign in) 구현

1 Next-auth

회원가입을 통해서  
Google에서 제공해준  
사용자의 정보를 DB에 저장했다.

이걸로 뭘 할 수 있지?

이메일 주소로  
서비스 이용 안내 이메일을 보내줄 수는 있을듯...

결국은 로그인을 해야한다  
로그인은 어떻게 처리해줘야 할까?

로그인 과정을 거치면  
사용자는 스스로를  
우리 DB에 저장된 사용자 A임을 입증할 수 있다.

로그인을 거친 이후에는  
해당 사용자의 모든 동작이  
사용자 A가 했다는 도장을 찍어줘야 한다.

세션(session)은 컴퓨터 과학에서, 특히 네트워크 분야에서 반영구적이고 상호작용적인 **정보 교환**을 전제하는 둘 이상의 통신 장치나 컴퓨터와 사용자 간의 대화나 송수신 **연결상태를 의미**하는 보안적인 다이얼로그(dialogue) 및 시간대를 가리킨다.

Session은 어디에서 관리하는게 좋을까?

- 브라우저에 저장되면 좋겠고
- 모든 api 요청에 담겨야 하고
- 유효기간을 설정해줄 수 있어야 하고

Cookie를 선택!

회원가입 / 로그인 관련된

여러 기능을 제공하는 라이브러리를 활용해보자

Next.js 에서는 Authentication(인증)을 어떻게 가이드하고 있는지 확인  
(<https://nextjs.org/docs/authentication>)

# NextAuth.js

## Authentication for Next.js

(<https://next-auth.js.org/>)

(<https://next-auth.js.org/getting-started/example>)



## Prisma Adapters

(<https://next-auth.js.org/adapters/prisma>)

```
yarn add @next-auth/prisma-adapter
```

prisma 명령어

(<https://www.prisma.io/docs/reference/api-reference/command-reference>)

prisma/schema.prsima

Account

Session

User

VerificationToken

yarn prisma db push

components/GoogleLogin.tsx

SessionProvider

pages/api/auth/[...nextauth].ts

pages/auth/login

실제로 로그인을 해보자

그리고 TABLE 들을 조회해보자

Account

User

Session

Login 후 화면마다  
Session 정보 노출

```
const { data: session } = useSession()
```

api 요청을 해보자

search에 뭐라도 입력해보자

network 탭에

request 쿠키에 뭔가가 남는다.

next-auth.session-token=4493df47-e03b-4348-b88f-f31830fb2a9b



Session으로 사용자를 구분할 수 있다.

next-auth.session-token

로 사용자를 특정할 수 있다.

## Planet Scale Cli

(<https://github.com/planetscale/cli>)

인증

pscale auth login

pscale shell [database] [branch]

## 정리

### 로그인 구현

로그인

기존에 가입한 사용자를 식별

Session

일정 기간 동안 유지되는 정보

next-auth

next.js를 위한 인증 라이브러리

pscale cli

웹 콘솔을 대체하는 도구