

웹 어플리케이션 보안 (캡스톤디자인)

4. Social Login

중부대학교 정보보호학과
이병천 교수
sultan@joongbu.ac.kr

전체 목차

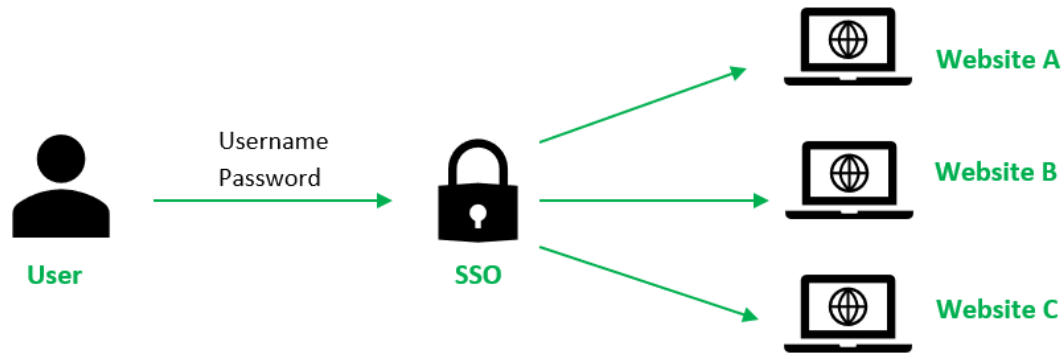
1. 강의 개요
 - 캡스톤디자인 프로젝트 추진 방법
2. Next.js
 - 프론트엔드 프로그래밍
 - 백엔드 프로그래밍
 - NextAuth
3. 예제 웹서비스 : ecommerce
4. 소셜로그인
5. 예제 웹서비스 : 암호 활용 Forge
6. 캡스톤디자인 프로젝트 발표

4. 소셜로그인



4.1 소셜로그인이란?

싱글사인온(SSO)



한번 로그인하면 SSO 도메인내의
모든 사이트에 로그인 처리됨

많이 사용하는 SNS 서비스의 로그인을 통해
인증을 승인. 인증정보의 관리를 편리하게...

소셜로그인

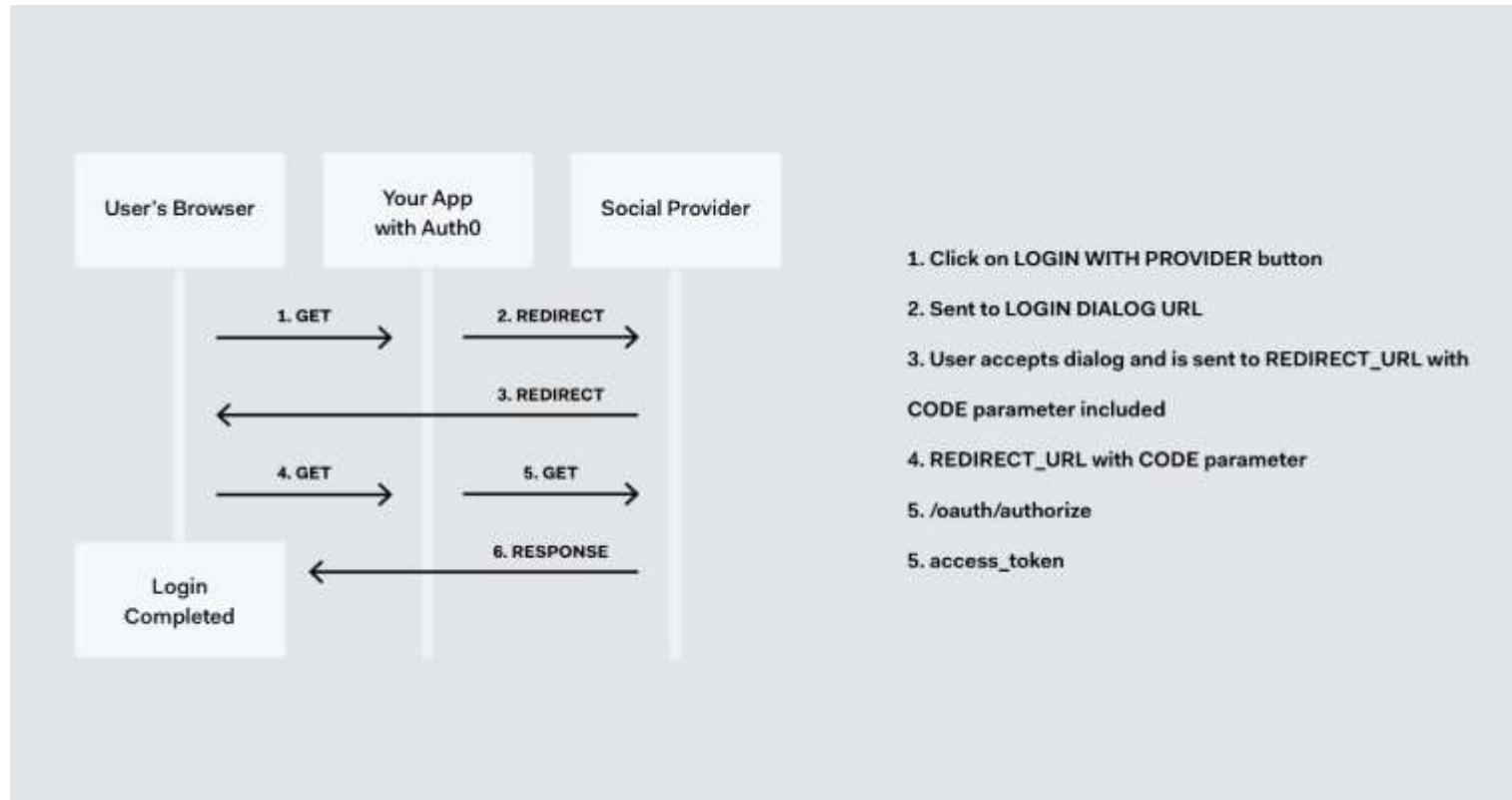


소셜 로그인의 프로토콜

- OAuth 2.0
 - <https://oauth.net/>

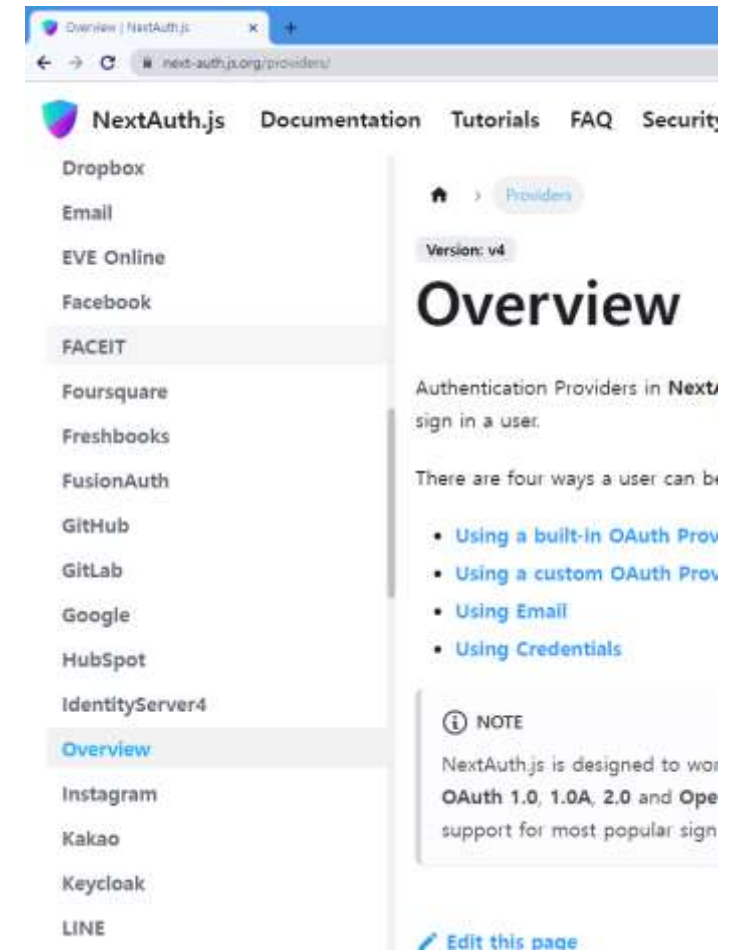
An **open protocol** to allow **secure authorization** in a **simple** and **standard** method from web, mobile and desktop applications.

[Learn more about OAuth 2.0 »](#)

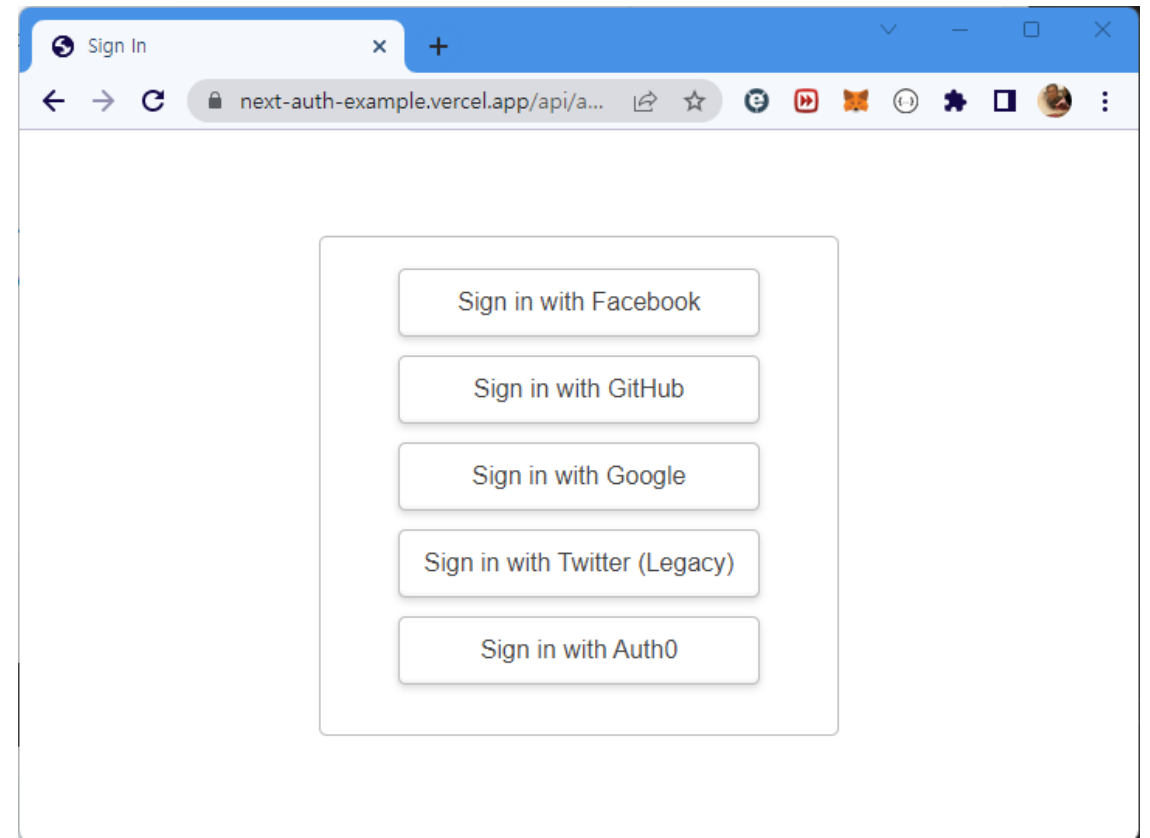
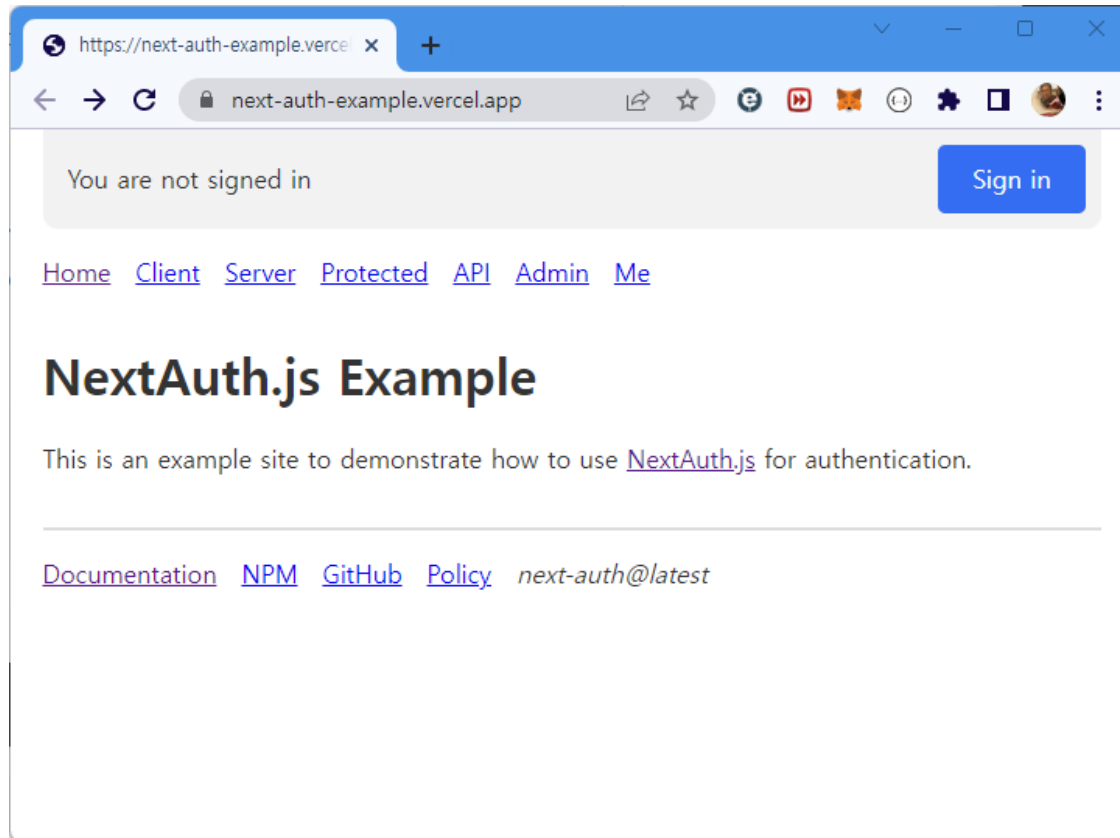


NextAuth Providers

- NextAuth는 많은 소셜로그인 프로바이더를 제공
 - Facebook
 - Github
 - Google
 - Kakao
 - Naver
 - Twitter
 - 등등



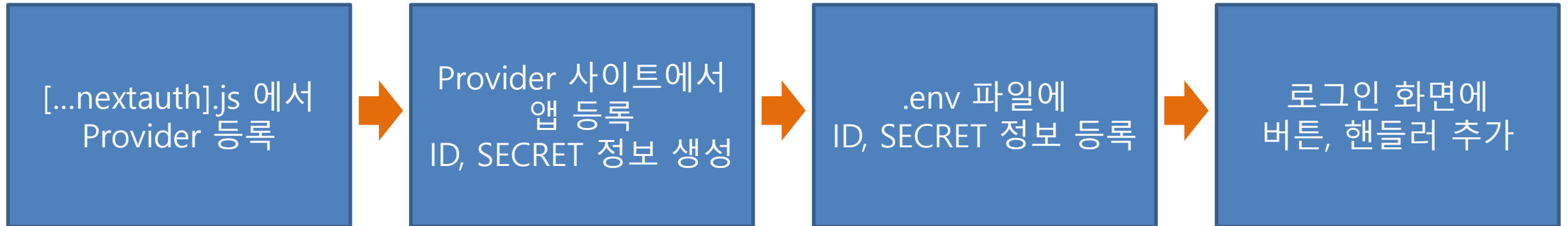
NextAuth 예제 사이트



<https://next-auth-example.vercel.app/>

작업 순서

.env에 새로 추가한 정보를
사용해야 하는 경우
서버를 재시작해야 함.



Github
Google
Kakao
Naver

4.2 Github 로그인

- Github provider 등록
 - <https://next-auth.js.org/providers/github>

```
import GitHubProvider from "next-auth/providers/github";
...
providers: [
  GitHubProvider({
    clientId: process.env.GITHUB_ID,
    clientSecret: process.env.GITHUB_SECRET
  })
]
...
```

Github 로그인

/pages/api/auth/[...nextauth].js
에서 github provider 추가

```
import bcryptjs from 'bcryptjs'
import NextAuth from 'next-auth'
import CredentialsProvider from 'next-auth/providers/credentials'
import User from '../models/User'
import db from '../utils/db'
import GithubProvider from 'next-auth/providers/github'

export default NextAuth({
  ....중략....
  providers: [
    CredentialsProvider({
      async authorize(credentials) {
        await db.connect()
        const user = await User.findOne({
          email: credentials.email,
        })
        await db.disconnect()
        if (user && bcryptjs.compareSync(credentials.password, user.password)) {
          return {
            _id: user._id,
            name: user.name,
            email: user.email,
            image: 'f',
            isAdmin: user.isAdmin,
          }
        }
        throw new Error('Invalid email or password')
      },
    },
  ],
  GithubProvider({
    clientId: process.env.GITHUB_ID,
    clientSecret: process.env.GITHUB_SECRET,
  }),
})
```

Github.com에서 앱 등록

- <https://github.com/settings/developers> 에서
- New OAuth App 등록
 - Application Name: next-ecommerce
 - Homepage URL:
<http://localhost:3000/>
 - Callback URL:
<http://localhost:3000/api/auth/callback/github>
 - Register application 버튼 클릭

Register a new OAuth application

Application name *

next-ecommerce

Something users will recognize and trust.

Homepage URL *

<http://localhost:3000/>

The full URL to your application homepage.

Application description

Application description is optional

This is displayed to all users of your application.

Authorization callback URL *

<http://localhost:3000/api/auth/callback/github>

Your application's callback URL. Read our [OAuth documentation](#) for more information.

☐ Enable Device Flow

Allow this OAuth App to authorize users via the Device Flow.

Read the [Device Flow documentation](#) for more information.

Register application

Cancel

Github 로그인

- 다음 2가지 정보를 복사하여 .env에 등록
 - Client ID
 - Client Secret

```
gear .env
1 MONGODB_URI=mongodb+
2 NEXTAUTH_SECRET=fdk1
3 NEXTAUTH_URL=http://
4 PAYPAL_CLIENT_ID=AWh
5
6 GITHUB_ID=0941a71df9
7 GITHUB_SECRET=c16c24
```

next-e-commerce



lbcsultan owns this application.

Transfer ownership

You can list your application in the [GitHub Marketplace](#) so that other users can discover it.

List this application in the Marketplace

0 users

Revoke all user tokens

Client ID

0941a71df990d9ce2abe

Client secrets

Generate a new client secret

Make sure to copy your new client secret now. You won't be able to see it again.



Client secret

✓ ~~0941a71df990d9ce2abe~~

Added 2 minutes ago by lbcsultan

Never used

You cannot delete the only client secret. Generate a new client secret first.

Delete

로그인 페이지에 버튼, 핸들러 추가

```
const githubLoginHandler = async () => {  
  try {  
    const result = await signIn('github', {  
      redirect: false,  
    })  
    console.log('Github login: ' + result)  
  } catch (err) {  
    toast.error(getError(err))  
  }  
}
```

중략

```
<div className="p-5 bg-gray-500 rounded-lg">  
  <div className="mb-4">  
    <button  
      className="primary-button w-full"  
      type="button"  
      onClick={githubLoginHandler}  
    >  
      Github Login  
    </button>  
  </div>  
</div>  
</form>  
</Layout>
```

/pages/login.js

```
await signIn('github', {  
  redirect: false,  
})
```

Login

Email

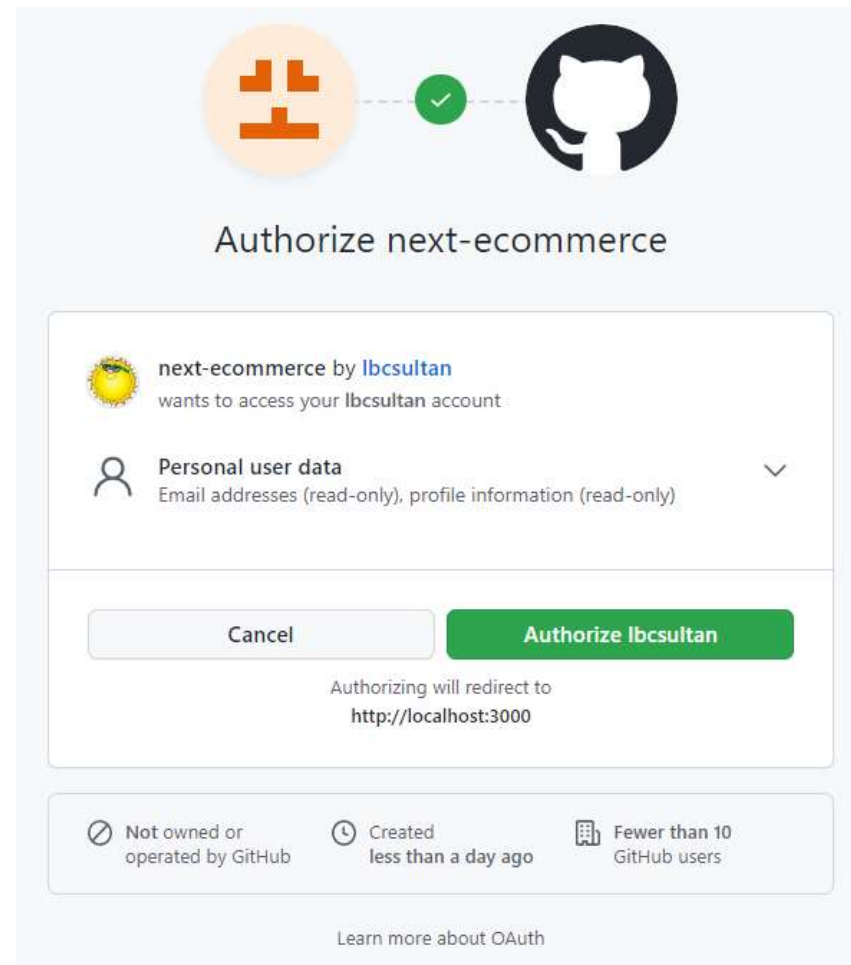
Password

Login

Don't have an account? [Register](#)

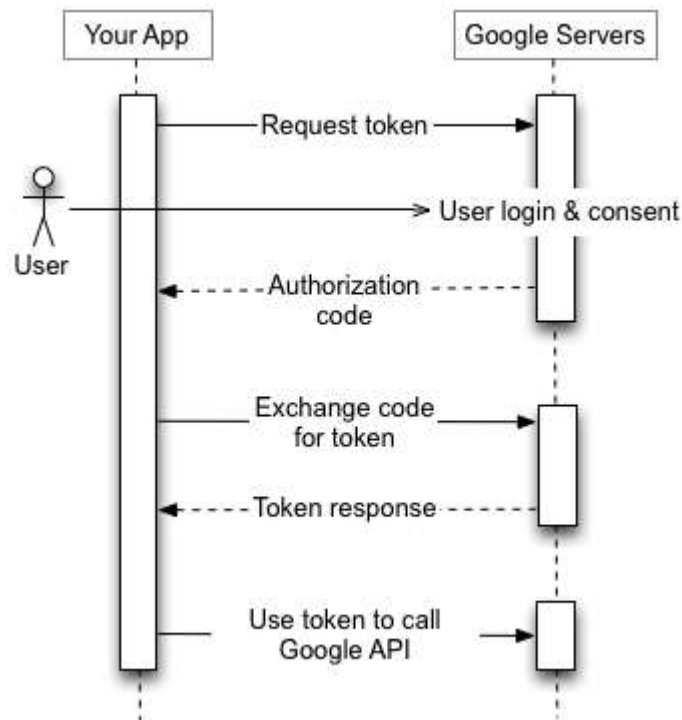
Github Login

로그인 성공



4.3 Google 로그인

- Provider
 - <https://next-auth.js.org/providers/google>



참조



Version: v4

Google

Documentation

<https://developers.google.com/identity/protocols/oauth2>

Configuration

<https://console.developers.google.com/apis/credentials>

The "Authorized redirect URIs" used when creating the credentials must include your full domain and end in the callback path. For example;

- For production: `https://{YOUR_DOMAIN}/api/auth/callback/google`
- For development: `http://localhost:3000/api/auth/callback/google`

Google provider 등록

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

```
import GithubProvider from 'next-auth/providers/github'  
import GoogleProvider from 'next-auth/providers/google'
```

중략

```
GithubProvider({  
  clientId: process.env.GITHUB_ID,  
  clientSecret: process.env.GITHUB_SECRET,  
}),  
GoogleProvider({  
  clientId: process.env.GOOGLE_CLIENT_ID,  
  clientSecret: process.env.GOOGLE_CLIENT_SECRET,  
  }),
```


Google Cloud에서 앱 등록

- <https://console.developers.google.com/apis/credentials> 접속, 로그인
 - Google Cloud 에 접속
 - 프로젝트 생성
 - 사용자 인증 정보
 - 사용자 인증 정보 만들기
 - OAuth 클라이언트 ID 만들기

Google Cloud nextauth

검색 제품, 리소스, 문서(/)

API API 및 서비스

OAuth 클라이언트 ID 만들기

클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요. OAuth 클라이언트 유형을 [자세히 알아보세요](#).

애플리케이션 유형 *
웹 애플리케이션

이름 *
next-ecommerce

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

아래에 추가한 URI의 도메인이 [승인된 도메인](#)으로 OAuth 동의 화면에 자동으로 추가됩니다.

Google Cloud에서 앱 등록

- URL 등록
 - URL:
<http://localhost:3000/>
 - Callback URL:
<http://localhost:3000/api/auth/callback/google>
 - 저장 버튼 클릭

승인된 자바스크립트 원본 ?

브라우저 요청에 사용

URI 1 *
<http://localhost:3000/>

+ URI 추가

승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI 1 *
<http://localhost:3000/api/auth/callback/google>

+ URI 추가

.env에 등록

```
gear .env
1 MONGODB_URI=mongodb+srv://nex
2 NEXTAUTH_SECRET=fdkljsdlkjfsk
3 NEXTAUTH_URL=http://localhost
4 PAYPAL_CLIENT_ID=AWh-cKNvyTm8
5
6 GITHUB_ID=0941a71df990d9ce2ab
7 GITHUB_SECRET=c16c24faa025441
8
9 GOOGLE_CLIENT_ID=558386476392
10 GOOGLE_CLIENT_SECRET=GOCSPX-L
11
```

OAuth 클라이언트 생성됨

API 및 서비스의 사용자 인증 정보에서 언제든지 클라이언트 ID와 보안 비밀에 액세스할 수 있습니다.



OAuth 액세스는 [OAuth 동의 화면](#)에 나열된 [테스트 사용자](#)로 제한됩니다.

클라이언트 ID

558386476392-uvgggbvg2a7uvtckhbck4uhvm1nq6d4u.apps.gc



클라이언트 보안 비밀번호

[REDACTED]



↓ JSON 다운로드

[확인](#)

로그인 메뉴, 핸들러 등록

```
await signIn('google', {
  redirect: false,
})
```

/pages/login.js

```
const googleLoginHandler = async () => {
  try {
    // eslint-disable-next-line no-unused-vars
    const result = await signIn('google', {
      redirect: false,
    })
  } catch (err) {
    toast.error(getError(err))
  }
}

<div className="p-5 bg-gray-500 rounded-lg">
  <div className="mb-4">
    <button
      className="primary-button w-full"
      type="button"
      onClick={githubLoginHandler}
    >
      Github Login
    </button>
  </div>

  <div className="mb-4">
    <button
      className="primary-button w-full"
      type="button"
      onClick={googleLoginHandler}
    >
      Google Login
    </button>
  </div>
</div>
```

Google 로그인 성공

- 로그인 성공

Login

Email


Password

Login

Don't have an account? [Register](#)

Github Login

Google Login

 Google 계정으로 로그인

계정 선택

[next-auth-demo](#)(으)로 이동



홍익보안

lbcsultan@gmail.com



다른 계정 사용

계속 진행하기 위해 Google에서 내 이름, 이메일 주소, 언어 환경설정, 프로필 사진을 [next-auth-demo](#)과(와) 공유합니다.

4.4 Kakao 로그인

Version: v4

Kakao

Documentation

<https://developers.kakao.com/product/kakaoLogin>

Configuration

<https://developers.kakao.com/docs/latest/en/kakaologin/common>

Options

The **Kakao Provider** comes with a set of default options:

- [Kakao Provider options](#)

Kakao Login

Login

Kakao 로그인

- Kakao provider 등록
 - <https://next-auth.js.org/providers/kakao>

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

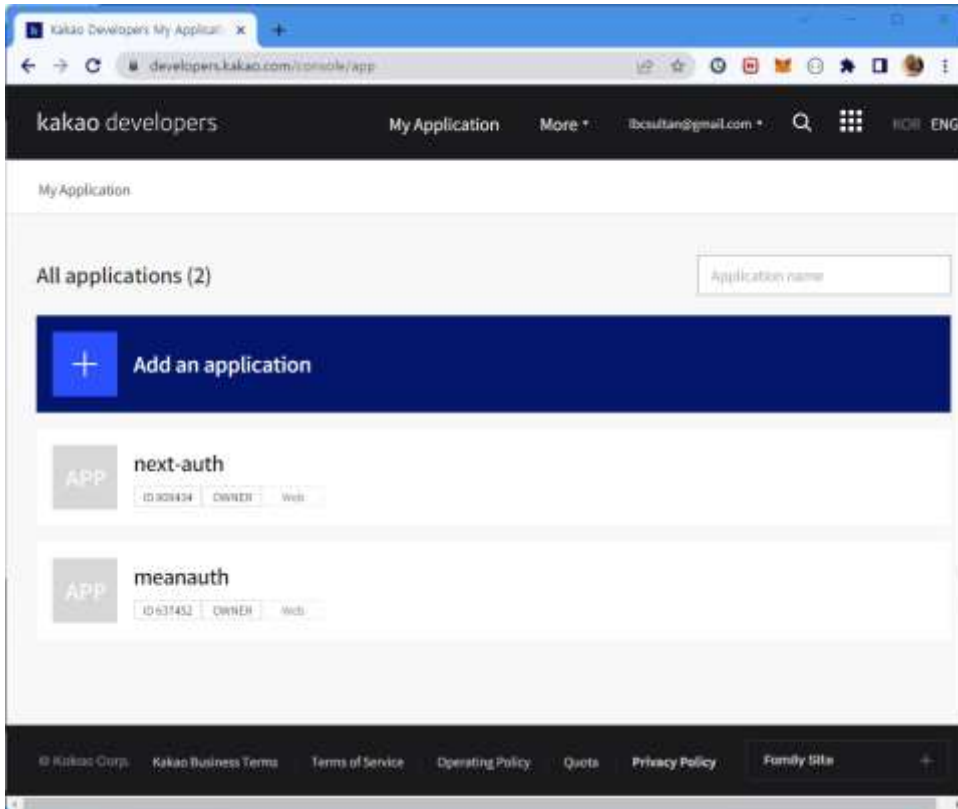
```
import GithubProvider from 'next-auth/providers/github'  
import GoogleProvider from 'next-auth/providers/google'  
import KakaoProvider from 'next-auth/providers/kakao'
```

중략

```
GoogleProvider({  
  clientId: process.env.GOOGLE_CLIENT_ID,  
  clientSecret: process.env.GOOGLE_CLIENT_SECRET,  
}),  
KakaoProvider({  
  clientId: process.env.KAKAO_CLIENT_ID,  
  clientSecret: process.env.KAKAO_CLIENT_SECRET,  
}),
```

앱 등록

- <https://developers.kakao.com/console/app>



Add an application

App icon

Image Upload

Select a file

JPG, GIF, PNG
Recommended size: 128px, up to 250KB

App name

next-ecommerce

Company name

중부대학교

- The information you enter in here is displayed when a user attempts Kakao Login.
- If you input wrong information, the use of the service may be restricted.

☒ This app does not violate the Operating Policy associated with [Categories Prohibited from Service Use](#), [Prohibited Content](#), and [Prohibited Activities](#).

Cancel

Save

KAKAO_CLIENT_ID, KAKAO_CLIENT_SECRET

kakao developers My Application

My Application > App Settings > Summary

APP next-ecommerce ID 814201 OWNER

App Keys

Native app key	26aef4f5abbfde48c2e06d8eec02c993
REST API key	e67fed23797d6dcec7616381b6f3f14f
JavaScript key	10ab18d11810ccbd95fa033284a3c309
Admin key	51b95c70b8df7ec0aef4ea6d6b18df78

KAKAO_CLIENT_ID

kakao developers 내 애플리케이션

내 애플리케이션 > 제품 설정 > 카카오 로그인 > 보안

플랫폼 팀 관리 APP next-ecommerce ID 814201 OWNER

제품 설정

카카오 로그인 OFF

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

연결 끊기

사용자 프로퍼티

보안

고급

Client Secret

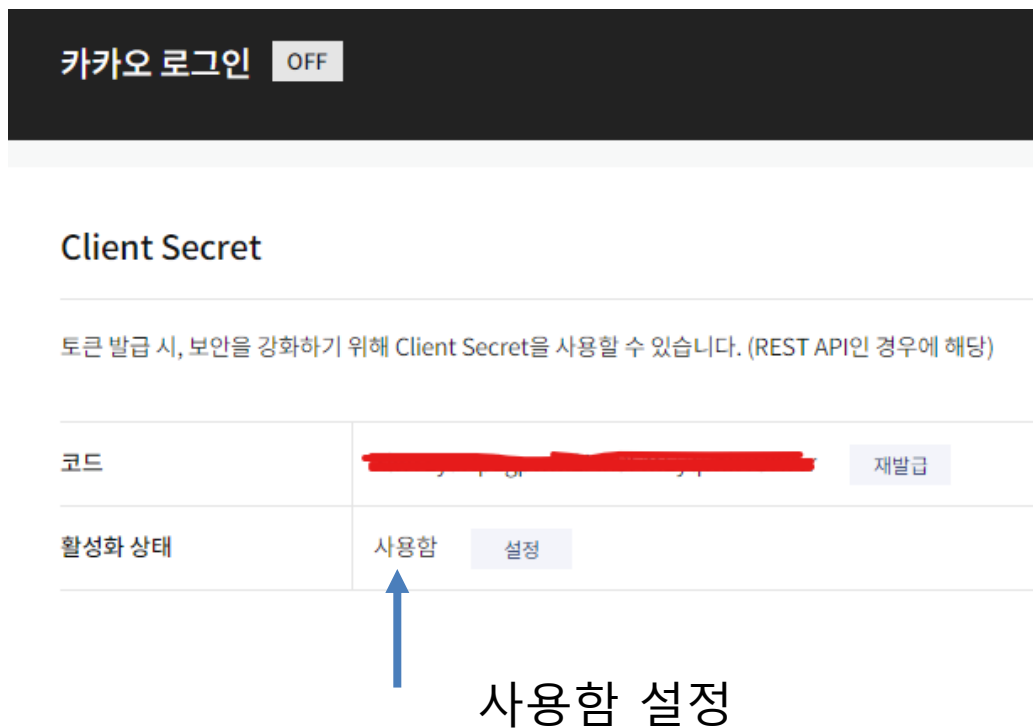
토큰 발급 시, 보안을 강화하기 위해 Client Secret을 사용할 수 있습니다. (REST API인 경우에 해당)

코드 생성

KAKAO_CLIENT_SECRET

KAKAO_CLIENT_ID, KAKAO_CLIENT_SECRET

- .env 에 ID, SECRET 등록
 - KAKAO_CLIENT_ID=
 - KAKAO_CLIENT_SECRET=



```
GITHUB_ID=0941a71df990c
GITHUB_SECRET=c16c24faa
GOOGLE_CLIENT_ID=558380
GOOGLE_CLIENT_SECRET=GO
KAKAO_CLIENT_ID=e67fed1
KAKAO_CLIENT_SECRET=Xlv
```

카카오 로그인 활성화, 동의항목 설정

kakao developers

내 애플리케이션 > 제품 설정 > 카카오 로그인

플랫폼
팀 관리

제품 설정
카카오 로그인
동의항목
간편가입
카카오톡 채널
개인정보 국외이전
연결 끊기
사용자 프로퍼티
보안
고급

APP next-ecommerce
ID 814201 OWNER

카카오 로그인 ON

활성화 설정

상태 **ON**

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버 상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결

APP next-ecommerce
증부대학교

전체 동의하기
전체동의를 선택목적에 대한 동의를 포함하고 있으며, 선택목적에 대한 동의를 거부해도 서비스 이용이 가능합니다.

lbcsultan@gmail.com
next-ecommerce 서비스 제공을 위해 회원번호와 함께 개인정보가 제공됩니다. 보다 자세한 개인정보 제공항목은 동의 내용에서 확인하실 수 있습니다. 정보는 서비스 탈퇴 시 지체없이 파기됩니다.

✓ [필수] 필수 제공 항목 보기

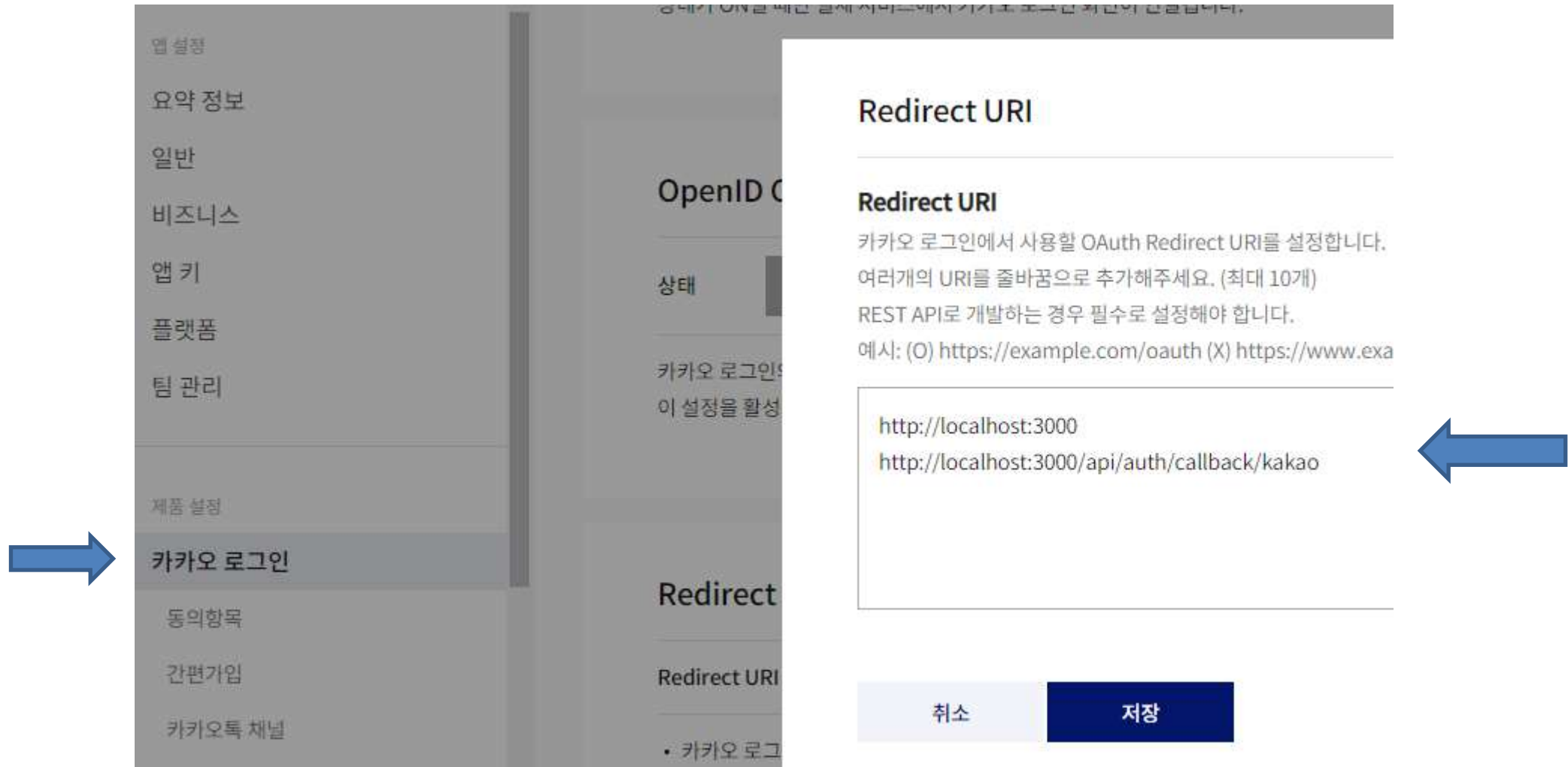
닉네임

[선택] 선택 제공 항목 보기

✓ 카카오계정(이메일)

동의하고 계속하기

Redirect URI 등록



앱 설정

요약 정보

일반

비즈니스

앱 키

플랫폼

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

OpenID C

상태

카카오 로그인

이 설정을 활성

Redirect

Redirect URI

카카오 로그

Redirect URI

카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다.
여러개의 URI를 줄바꿈으로 추가해주세요. (최대 10개)
REST API로 개발하는 경우 필수로 설정해야 합니다.
예시: (O) `https://example.com/oauth` (X) `https://www.exa`

`http://localhost:3000`
`http://localhost:3000/api/auth/callback/kakao`

취소 저장

로그인 메뉴, 핸들러 등록

```
await signIn('kakao', {
  redirect: false,
})
```

/pages/login.js

```
const kakaoLoginHandler = async () => {
  try {
    // eslint-disable-next-line no-unused-vars
    const result = await signIn('kakao', {
      redirect: false,
    })
  } catch (err) {
    toast.error(getError(err))
  }
}
```

중략

```
<div className="mb-4">
  <button
    className="primary-button w-full"
    type="button"
    onClick={kakaoLoginHandler}
  >
    Kakao Login
  </button>
</div>
```

Kakao 로그인 성공

Login

Email

Password

Login

Don't have an account? [Register](#)

Github Login

Google Login

Kakao Login

kakao



next-ecommerce

중부대학교

✓ 전체 동의하기

전체동의를 선택목적에 대한 동의를 포함하고 있으며, 선택목적에 대한 동의를 거부해도 서비스 이용이 가능합니다.



lbcsultan@gmail.com

[계정 변경](#)

next-ecommerce 서비스 제공을 위해 회원번호와 함께 개인정보가 제공됩니다. 보다 자세한 개인정보 제공항목은 동의 내용에서 확인하실 수 있습니다. 정보는 서비스 탈퇴 시 지체없이 파기됩니다.

✓ [필수] 필수 제공 항목

[보기](#)

닉네임

[선택] 선택 제공 항목

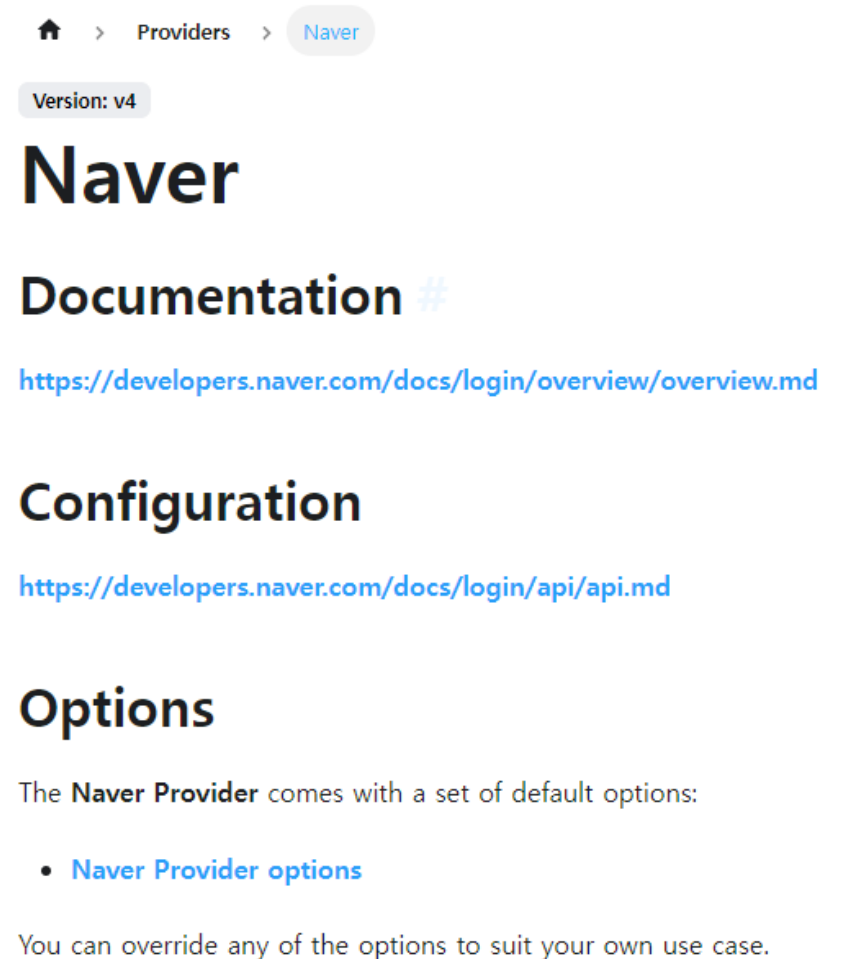
[보기](#)

✓ 카카오회계정(이메일)

동의하고 계속하기

4.5 Naver 로그인

- Naver provider
 - <https://next-auth.js.org/providers/naver>
- Naver Login API
 - <https://developers.naver.com/docs/login/api/api.md>



Naver provider 등록

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

```
import GithubProvider from 'next-auth/providers/github'
import GoogleProvider from 'next-auth/providers/google'
import KakaoProvider from 'next-auth/providers/kakao'
import NaverProvider from 'next-auth/providers/naver'
```

중략

```
KakaoProvider({
  clientId: process.env.KAKAO_CLIENT_ID,
  clientSecret: process.env.KAKAO_CLIENT_SECRET,
}),
NaverProvider({
  clientId: process.env.NAVER_CLIENT_ID,
  clientSecret: process.env.NAVER_CLIENT_SECRET,
}),
```


Naver App 등록

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

애플리케이션 이름 ⇄

✓

- 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.
- 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/" , "-", "_", 만 입력 가능합니다.

사용 API ⇄

선택하세요. ▼ ✓

네이버 로그인

제공 정보 선택(이용자 식별자는 기본 정보로 제공) ⓘ

필수 항목은 개인정보보호법 제3조 제1항, 제16조 제1항 등에 따라 서비스 제공을 위해 필요한 최소한의 개인정보만을 선택해야 합니다.

권한	필수	추가
회원이름	<input checked="" type="checkbox"/>	<input type="checkbox"/>
이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>
별명	<input checked="" type="checkbox"/>	<input type="checkbox"/>
프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>

✕

네이버 개발자센터

<https://developers.naver.com/apps/#/list>

<https://developers.naver.com/apps/#/register>

로그인 오픈 API 서비스 환경

환경추가 : PC 웹

서비스 URL : <http://localhost:3000/>

네이버 로그인 Callback URL :

<http://localhost:3000/api/auth/callback/naver/>

등록하기

Naver App 등록

next-ecommerce

개요	API 설정	네이버 로그인 검수상태	멤버관리	로그
----	--------	-----------------	------	----

애플리케이션 정보

Client ID	<input type="text" value="kjIPiMqSNVzYOPeqU5eS"/>
Client Secret	<div><input type="password" value="....."/></div> <div>보기</div>

Client ID, Client Secret을 .env에 등록

```
NAVER_CLIENT_ID=  
NAVER_CLIENT_SECRET=
```

로그인 메뉴, 핸들러 등록

```
await signIn('naver', {
  redirect: false,
})
```

/pages/login.js

```
const naverLoginHandler = async () => {
  try {
    // eslint-disable-next-line no-unused-vars
    const result = await signIn('naver', {
      redirect: false,
    })
  } catch (err) {
    toast.error(getError(err))
  }
}
```

종락

```
<div className="">
  <button
    className="primary-button w-full"
    type="button"
    onClick={naverLoginHandler}
  >
    Naver Login
  </button>
</div>
```

Naver 로그인 성공

N 네이버 로그인

 술탄 ▾



next-ecommerce

✓ 전체동의

next-ecommerce에서 **ibcsultan** 회원님의 개인정보에 접근합니다.

이용자 식별자 및 제공된 개인정보는 이용자 식별, 통계, 계정 연동 및 CS 등을 위해 서비스 이용기간 동안 활용/보관됩니다. 본 제공 동의를 거부할 권리가 있으나, 동의를 거부하실 경우 서비스 이용이 제한될 수 있습니다.

✓ 필수 제공 항목 (필수)

- ✓ 이름 ✓ 이메일 주소 ✓ 별명
- ✓ 프로필사진

동의 후에는, 해당 서비스의 이용약관 및 개인정보처리방침에 따라 정보가 관리됩니다.

취소

동의하기

요약

- 소셜로그인
 - 각 사이트별 인증정보(아이디, 비밀번호) 관리의 어려움을 해결
 - Next-auth는 소셜로그인 기능을 기본 내장

Login

The image shows a login form with a light blue background. It contains two input fields: 'Email' and 'Password'. Below the password field is a yellow 'Login' button. At the bottom of the form, there is a link that says 'Don't have an account? Register'. Below the main form, there is a separate grey box containing four yellow buttons for social login: 'Github Login', 'Google Login', 'Kakao Login', and 'Naver Login'.

Vercel.com에 서비스 등록

- .env의 환경변수들을 vercel에서 environment variable(환경변수)로 등록
- 각 소셜로그인 서비스별로 URL 정보를 실제 주소로 변경
 - 서비스 URL
 - Callback URL



Configure Project

PROJECT NAME

next-e-commerce

FRAMEWORK PRESET

Next.js

ROOT DIRECTORY

./

Edit

> Build and Output Settings

Environment Variables

NAME

EXAMPLE_NAME

VALUE (WILL BE ENCRYPTED)

I9JU23NF394R6HH

Add

Learn more about [Environment Variables](#)

Deploy