

# Lang ChainJS AI 챗봇 개발 With Next.js

## Basic

발표자료실 : <https://bit.ly/3yFgzf9>

실습소스 깃허브 주소: <https://github.com/eddykang1074/next-langchain.git>

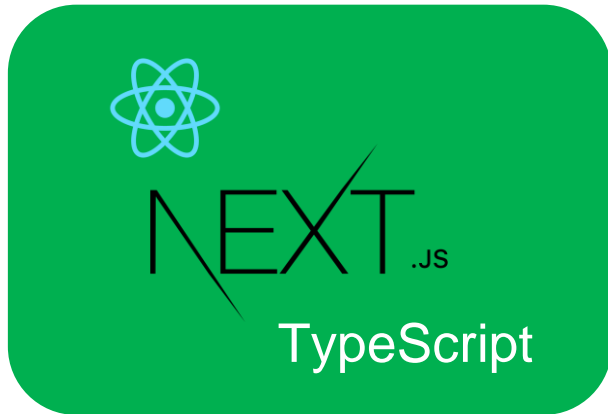
강창훈

지니공공아카데미

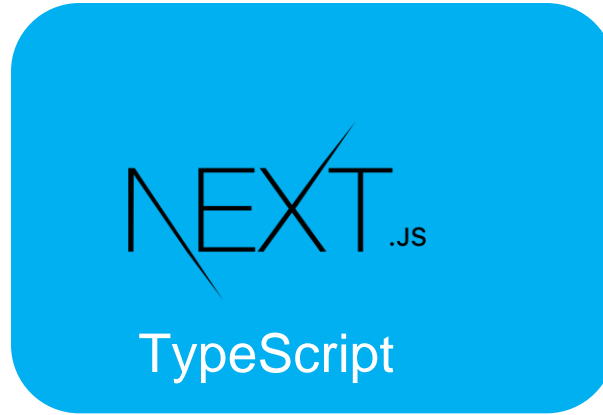


<https://bit.ly/3yFgzf9> 

## 기술 세미나 개요 – Web Full Stack + AI



**Web Frontend**



**Backend**

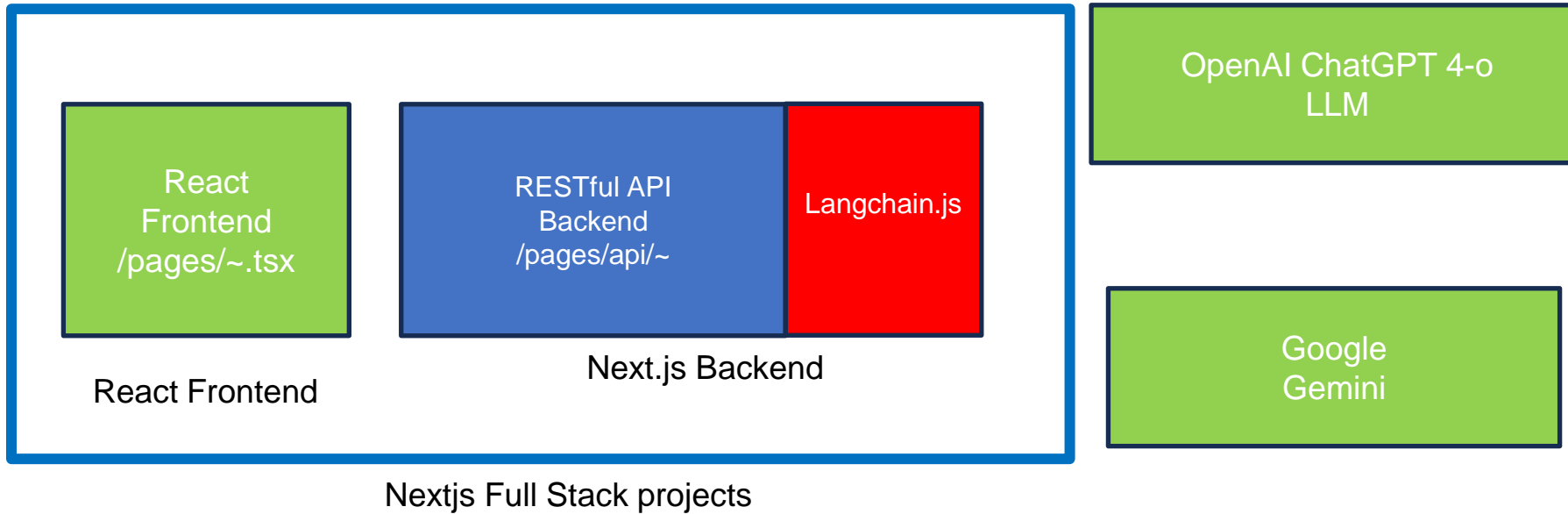


**LLM Framework**



**LLMs**

## 기술 세미나 개요 – Web Full Stack + AI



# 기술 세미나 개요 – Web Full Stack + AI



데모 샘플 : <https://chatbotmaker.co.kr/>

회원가입 또는 테스트 계정정보

id: [test1@test.co.kr](mailto:test1@test.co.kr) pwd: 1234

# Contents

## 01

### Next.js 풀 스택 개발환경 준비하기

1. Node Framework & VSCode 설치하기
2. Next.js 소개 및 프로젝트 만들기
3. 챗봇 UI 및 Backend API 구현하기

## 02

### LLM 서비스 신청하기

1. OpenAI LLM서비스 API 신청하기
2. API 인증키 발급
3. 프로젝트 환경변수 등록하기

## 03

### LangChain.js 기반 AI 챗봇 개발하기

1. Lang Chain 소개 및 패키지 설치하기
2. LangChain.js 기초 모듈 활용하기
3. Agent 기반 챗봇 구현하기

# 01

## Next.js 풀 스택 개발환경 준비하기

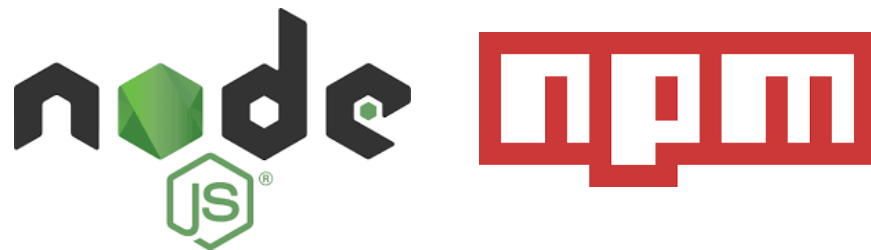
1. Node Framework & VSCode 설치하기
2. Next.js 소개 및 프로젝트 만들기
3. 챗봇 UI 및 Backend API 구현하기

## 01 Next.js 풀스택 개발환경 준비하기

# 1. Node Framework & VSCode 설치하기

### 1) Node Framework 설치 : 2024년 07월 기준 Version 20.15.0 LTS 권장 설치

- <https://nodejs.org/ko/>
- 20.15.0 LTS 다운로드 설치 진행
- **Node.js 18.17 or later for Next.js**
- 오픈소스는 되도록 최신버전의 LTS(Long Term Support)버전설치 필수 3년간 안정적 사용환경지원



### 2)Node, NPM 설치 버전 확인하기

- Node 버전확인 : `node -v`
- NPM 버전확인 : `npm -v`

## 01 Next.js 풀스택 개발환경 준비하기

# 1. Node Framework & VSCode 설치하기

### 3) Visual Studio Code(VS Code) 설치

- <https://code.visualstudio.com/>
- **Windows x64 or macOS Stable** 다운로드 설치
- 설치 이후 한국어 언어 팩 설치 권장 시 설치
- 최초 설치 후 각종 설정 화면 나타나면 하단에 Skip 진행 건너뛰





## 01 Next.js 풀스택 개발환경 준비하기

### 개발환경 구축하기 - 실습 작업 폴더 만들기

- 실습 작업 폴더 만들기
- C:₩**Next-LangChainJS**₩ or D:₩ **Next-LangChainJS**₩
- macOS는 사용자 폴더내 실습 폴더명만 동일하게 지정해줌

\*\* 실습 및 개발 작업 환경을 일치시켜

강사와 수강생 간의 원활 한 커뮤니케이션을 위해 **실습 폴더명은 반드시 동일하게** 만들어주세요.



# 2. Next.js 소개 및 프로젝트 만들기

### 1)리액트 프론트 엔드 개발 라이브러리

- 페이스북에서 개발 보급하는 오픈소스 프론트엔드 개발 자바스크립트 라이브러리
- 가상돔 제어 기술을 기반으로 웹 페이지의 필요한 부분만 빠르게 DOM 렌더링해주는 기술제공
- 컴포넌트 단위로 UI 요소 재사용 극대화 기술제공.
- 페이스북의 강력한 기술주도와 관련 수많은 커뮤니티 지원으로 지속적으로 발전
- 리액트를 지원하는 수많은 라이브러리와 확장 가능한 프론트엔드 개발 프레임워크와 쉽게 통합
- 최신버전 v18.x - 2024.07 기준

### 2)어떤 회사들에서 사용할까?

- 넷플릭스
- 에어비엔비
- 드롭박스
- 트위터
- 에버노트
- 우버

### 3)확장 가능한 개발 생태계



CRA



Vite

Next Generation Frontend Tooling



NEXT.JS



## 01 Next.js 풀스택 개발환경 준비하기

# 2. Next.js 소개 및 프로젝트 만들기-3가지 방식

리액트 프로젝트를 만들고 개발할수 있는 방식은 아래와 같이 총 3가지 방식이 제공됩니다.

### 1)CRA(CreateReactApp) 방식 : 가장 올드한 방식

- create-react-app 패키지를 설치하고 해당 패키지를 통해 리액트 프로젝트 생성/개발
- 리액트 기술 초기 프로젝트 템플릿 생성 환경제공
- 현재 CRA 관련 기술 지양(비추천)하고 있음

### 2) VITE 방식 : Next.js 이전 추천 되던 방식

- 기존 CRA 프로젝트 생성방식의 번들링 과정을 획기적으로 개선 개발시 개발 생산성/효율성 제공
- 기존 CRA 프로젝트 기반의 비효율성(번들링) 개선 과 현대적 프론트엔드 애플리케이션 요구사항 반영
- Next.js 도입전까지 가장 많이 사용되던 리액트 프로젝트 생성/개발 방식

### 3) NEXT.JS 방식 : 추천

- 가장 최신의 리액트 개발 방식으로 추천
- 현대적 거대 웹 프론트엔드 애플리케이션 개발시 필요한 다양한 최신기술 제공(SEO,SSR...)
- 리액트 기반 웹 프론트엔드 개발 표준 기술로 자리잡아 가고 있음( 백엔드 개발기능도 제공함)
- Next.js는 리액트기반 프론트엔드 개발과 백엔드 기능 개발도 가능한 웹 풀스택 개발프레임워크를 지향



2020 년 이전



Vite + React



2019-2022년

NEXT.js

2022~2024년~

# 2. Next.js 소개 및 프로젝트 만들기-CRA 방식

### 1)create-react-app 패키지 필수 설치

- 리액트 개발 프로젝트 자동 생성 지원 패키지
- yarn,npm 명령어 통해 설치 가능

**yarn global add create-react-app or npm install -g create-react-app**

### 2) 리액트 CRA 프로젝트 만들기

**yarn create react-app cra-react-app:** 반드시 사전에 CRA 패키지가 로컬 개발자 컴퓨터에 설치되어 있어야함.

**Or**

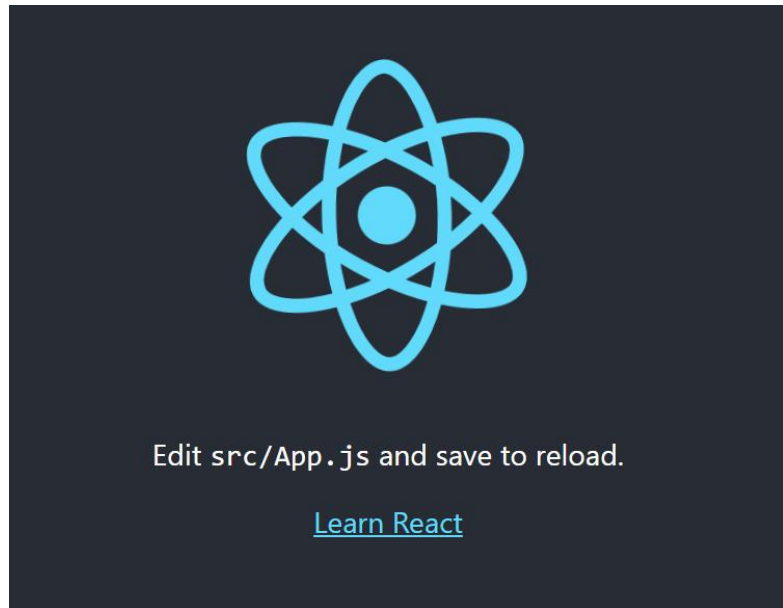
**npx create-react-app cra-react-app:** 프로젝트를 만들때마다 CRA패키지를 다운로드 설치하고 CRA기반으로 리액트 프로젝트 생성후 CRA패키지를 삭제합니다.

**Or**

**npx create-react-app cra-react-app --template typescript :** 프로젝트를 만들때마다 CRA패키지를 다운로드 설치하고 CRA기반 + TypeScript 기반으로 리액트 프로젝트 생성후 CRA패키지를 삭제합니다. : CRA 권장사항

### 3) 리액트 CRA 프로젝트 실행하기

- 프로젝트 폴더이동: **cd cra-react-app**
- 프로젝트 실행위한 기본 패키지 복원 설치-필요시(프로젝트생성시 자동 패키지 복원설치됨) : **yarn or yarn install**
- 프로젝트 시작 : **yarn start or npm run start**
- **프로젝트 중지: Control + C 실행모드 중단하기**
- 프로젝트 빌드 배포 - 서비스를 위한 최종 정적 리소스를 생성하는 과정- 번들링과정 포함 : **yarn build**
- **Build후 프로젝트 build폴더에 최종 번들링된 소스 생성**



# 2. Next.js 소개 및 프로젝트 만들기-Vite 방식

### 1)Yarn 패키지 관리자 기반 Vite 프로젝트 생성

- 하기 명령어 기반 CLI 프롬프팅 방식으로 프로젝트 생성

**yarn create vite : 프로젝트 생성옵션 방식 결정 가능**

**-순차적을 프로젝트 생성**

**-Project name : vite-react-app**

**-Select a framework: React**

**-Select a variant : JavaScript Or TypeScript Or JavaScript + SWC Or TypeScript + SWC**  
(SWC는 Rust로 개발된 빌드 컴파일러로 프로젝트 빌드/번들링시 속도 개선 효과 제공 )

Or

**-직접 프로젝트 생성옵션 지정 생성하기**

**yarn create vite vite-react-app --template react**

### 2) 리액트 Vite 프로젝트 실행하기

- 프로젝트 폴더이동: **cd vite-react-app**
- 프로젝트 실행위한 기본 패키지 복원 설치 : **yarn or yarn install**
- 프로젝트 시작 : **yarn dev**
- **http://localhost:5173/**
- **프로젝트 중지: Control + C 실행모드 중단하기**
- 프로젝트 빌드 배포 - 서비스를 위한 최종 정적 리소스를 생성하는 과정- 번들링과정 포함 : **yarn build**
- **Build후 프로젝트 폴더내 dist폴더 생성번들링 결과물 생성됨**



## Vite + React

count is 0

Edit src/App.jsx and save to test HMR

Click on the Vite and React logos to learn more

## 01 Next.js 풀스택 개발환경 준비하기

### Tips) Next.js 특징

#### 1) Next.js

- Next.js는 풀 스택 웹 애플리케이션을 구축하기 위한 React 프레임워크
- 현대적인 웹 애플리케이션 개발 과정 간소화 환경제공
- 서버 사이드 렌더링 기능제공
- 정적 사이트 생성 기능제공
- API 개발에 대한 쉬운 솔루션을 제공.
- 고급 성능 최적화와 내장된 개발 기능 제공
- 개발자들이 쉽게 빠르고 확장 가능
- SEO 친화적인 웹 애플리케이션을 구축할 수 있도록 지원
- **Node.js 18.17 or later**

#### 2) Next.js 주요 특징

- 직관적인 라우팅 시스템-dynamic routes지원
- 페이지 자동 최적화
- 데이터가 필요한 페이지를 SSR 할 수 있게 지원
- 빠른 페이지 로드를 위한 자동 코드 스플라이팅
- HMR을 지원
- **Node.js 18.17 or later.**



<https://nextjs.org>  
<https://nextjs.org/docs>

## 01 Next.js 풀스택 개발환경 준비하기

# 2. Next.js 소개 및 프로젝트 만들기-NextJS 본

### 1)리액트 NEXT.JS 프로젝트 생성하기

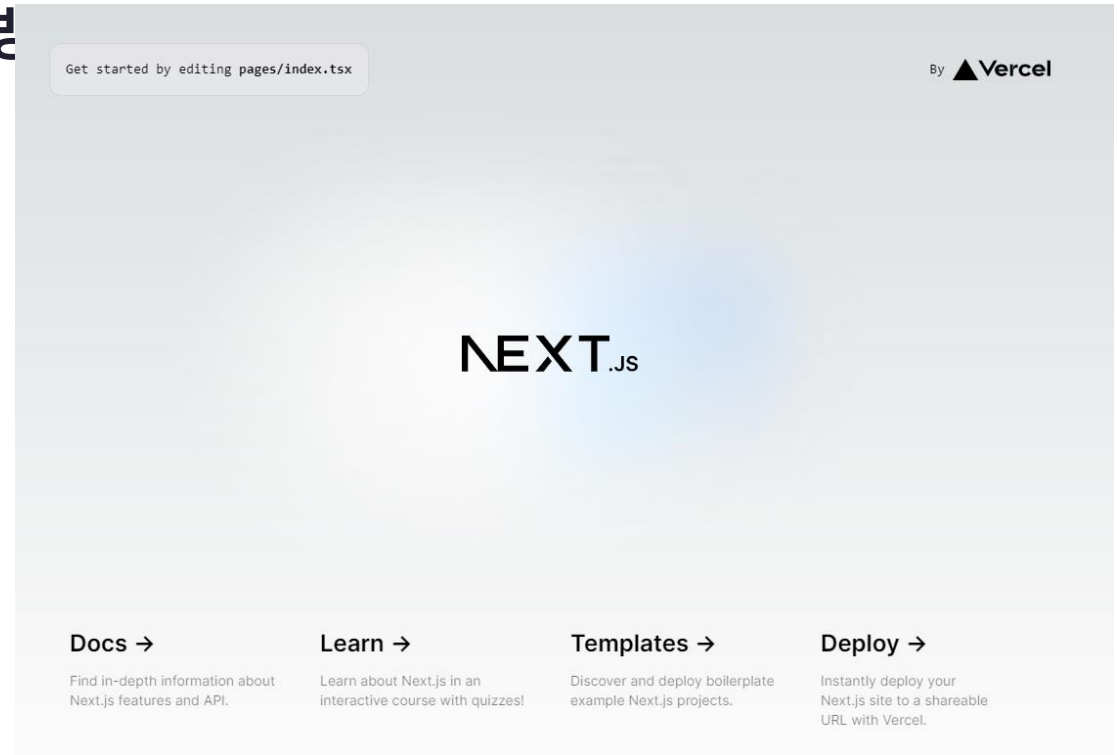
- 하기 명령어 기반 CLI 프롬프팅 방식으로 프로젝트 생성

**`npx create-next-app@latest next-react-app`**

- ✓ Would you like to use TypeScript? ... Yes
- ✓ Would you like to use ESLint? ... Yes
- ✓ Would you like to use Tailwind CSS? ... Yes
- ✓ Would you like to use ``src/`` directory? ... No
- ✓ Would you like to use **App Router? (recommended)** ... No
- ✓ Would you like to customize the default import alias (@/\*)? ... Yes
- ✓ What import alias would you like configured? ... @/\*

### 2) 리액트 Next.js 프로젝트 실행하기

- 프로젝트 폴더이동: **`cd next-react-app`**
- 프로젝트 개발 모드 시작 : **`npm run dev`**
- **프로젝트 중지: Control + C 실행모드 중단하기**
- 개발소스 만 빌드처리 빌드결과물은 프로젝트 .next폴더에 저장됨 : **`npm run build`**
- 이미 빌드된 결과물을 서버를 통해 실행 : **`npm run start`**
- **`http://localhost:3000`**

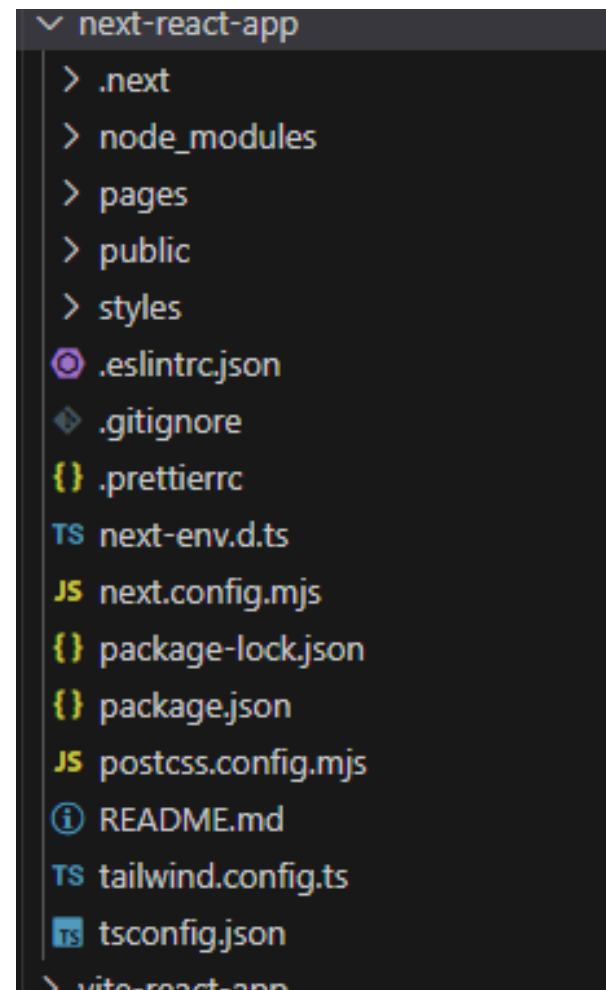


**Tips) npm run 명령어는 package.json파일내 scripts 속성내 npm 명령어 별칭을 실행시킬때 사용하는 npm 명령어중 하나입니다.**

# 2. Next.js 소개 및 프로젝트 만들기-NextJS 방식

### 3)리액트 NEXT.JS 프로젝트 폴더 구조

- .next : 소스 빌드 결과 저장 폴더
- node\_modules : 프로젝트 기능/개발 노드 패키지 설치 폴더
- pages : 페이지 컴포넌트/API 파일 저장 폴더
  - └ api : 백엔드 RESTful API 기능 구현 및 라우팅 폴더
  - └ \_app.tsx : NextJS 앱 서버측 최초 실행 파일/공통레이아웃구성,넥스트앱 설정/실행 구성
  - └ \_document.tsx : 클라이언트에 최종 렌더링 되는 Single HTML 페이지 생성 및 구성기능 제공
  - └ index.tsx : 메인 화면 페이지 컴포넌트
- public : 정적 리소스 폴더
- styles : 전역 스타일 폴더
- .eslintrc.json : eslint 설정파일
- .gitignore: 깃소스 제어무시 대상 설정파일
- .next-env.d.ts : 타입스크립트 컴파일러가 NEXTJS타입을 호출해주는 파일
- next.config.mjs : next.js 설정 파일
- package.json : 프로젝트 패키지 설정파일
- postcss.config.mjs : css 후처리기 설정파일
- tailwind.config.ts : 테일윈드 스타일 설정파일
- tsconfig.js : 타입스크립트 설정파일

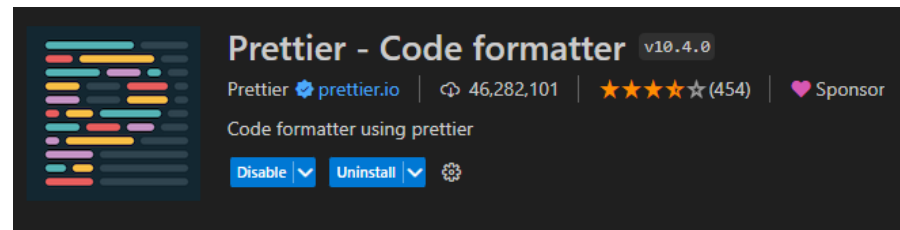




# Tips) Prettier-Code formatter 설치/구성

## 1) Prettier-Code formatter 설치

- 리액트 코드 작성시 자동 코드 정리 처리, 코딩 컨벤션 규칙 적용 등 개발 생산성/효율성 제공
- VSCode 확장 툴(Extensions)에서 Prettier 검색 설치 : Prettier - Code formatter



## 2) Prettier-Code formatter 설정하기

### A) 수동 코드정리 방법

- Control + Shift + P => 명령어 팔레트 => Format 검색어 입력 => Format Document 선택

### B) Formatter 이용 자동 코드정리 : 개발소스 저장 시 자동으로 해당 파일 정리기능 제공

Step1) VSCode 환경설정(File>Preferences > Settings : 검색박스 에서 formatOnSave 검색 >

**Editor: Format On Save 옵션 : 체크박스 선택**

Step2) 기본 포맷터를 Prettier로 지정하기: Settings: 상단 검색박스 에서 Default Formatter 검색 >

**Editor: Default Formatter 옵션 : Prettier-Code formatter 로 설정**

Step3) 프로젝트 루트에 **.prettierrc** 파일 생성 하고 아래 코딩 규칙 정의

```
{
  "semi": true,
  "singleQuote": true,
  "trailingComma": "all",
  "useTabs": false,
  "tabWidth": 2,
  "printWidth": 80,
  "arrowParens": "always"
}
```

## 01 Next.js 풀스택 개발환경 준비하기

### Tips) ESLint 확장툴 설치 및 구성

#### 1) VSCode ESLint 확장 툴 설치

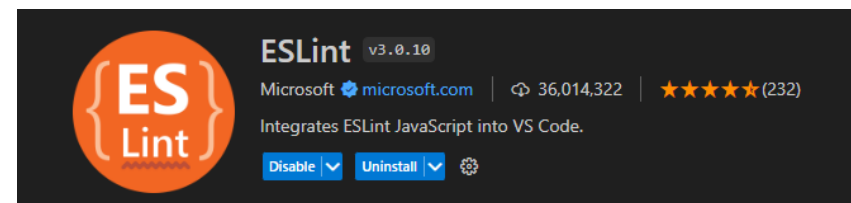
- ESLint 는 기본 코딩 스타일 가이드를 제공
- ESLint 는 코딩 컨벤션에 위배되는 코드나 안티 패턴을 자동 검출하는 도구
- JavaScript 및 JSX 코드에서 문제를 식별하고 일관된 코딩 스타일을 적용하기 위한 정적 분석 도구
- 코드 작성 중에 발생할 수 있는 오류나 잠재적인 문제를 미리 검출 제공
- 팀 내에서 일관된 코딩 규칙을 적용하여 코드 품질 향상 목적
- 개발자/팀별 코딩 컨벤션/스타일 가이드 작성 적용 가능
- **VSCode 확장툴 Extensions > ESLint 조회 설치**

#### 2) 프로젝트 ESLint 설치 및 구성

- 프로젝트별로 개발지원 패키지로 개별 프로젝트에 설치해서 사용함
- **Next.js 버전 11.0.0 부터 Next 프로젝트 생성시 자동 프로젝트별로 설치되어 별도 설치할 필요 없음**

#### 3) 개인/팀별 코딩 컨벤션 적용하기 (Airbnb)

- 팀별로 별도 컨벤션 규칙을 수립해 적용하고 싶으면 아래 링크를 참고해서 설정해보세요.
- <https://velog.io/@zzacodez/next.js-14-typescript에-eslint-prettier-husky-적용하기>



# Tips) 코드 JSX Snippets 확장 툴 설치 및 사용

### 1) Code Snippets 확장툴 설치

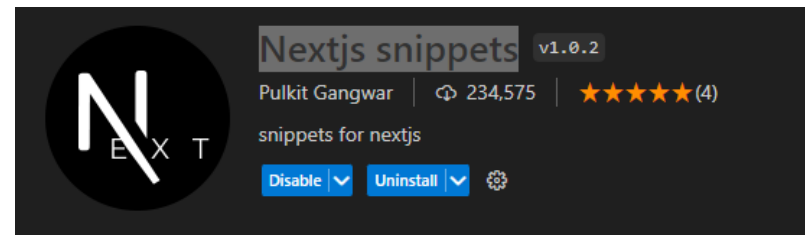
- Code Snippets은 코드 조각이란 의미로 단축키/예약어를 통해 코드 블록(조각)을 자동으로 생성해주는 기능을 말합니다.
- 리액트 개발시 컴포넌트 파일을 만들고 해당 파일의 기본적인 코드 구조를 빠르게 만들 때 유용합니다.

- **VSCode 확장툴 Extensions > Nextjs snippets 검색 설치**

### 2) Nextjs snippets 단축키 소개

- 리액트에서 컴포넌트 파일을 생성하고 (.jsx .js .tsx)
- 아래 명령어를 입력/선택(탭)하면 자동으로 코딩조각이 추가됩니다.
- **naf**: 화살표함수 기반 함수형 컴포넌트 생성
- **nafe: export default** 기반 화살표 함수형 컴포넌트 생성
- **napage** : 기본 함수형 컴포넌트 생성하기
- **natemplate**: 자식요소 포함한 함수형 컴포넌트 생성하기
- **nf**: 기본함수형 컴포넌트
- **nfe**: export default 기반 함수형 컴포넌트 생성
- **nspage**: getServerSideProps 함수 포함한 템플릿
- **nstaticpage**: getStaticPaths, getStaticProps 함수 포함한 템플릿
- **nstaticpage** (nextjs page with getStaticProps and getStaticPaths)

<https://marketplace.visualstudio.com/items?itemName=PulkitGangwar.nextjs-snippets>

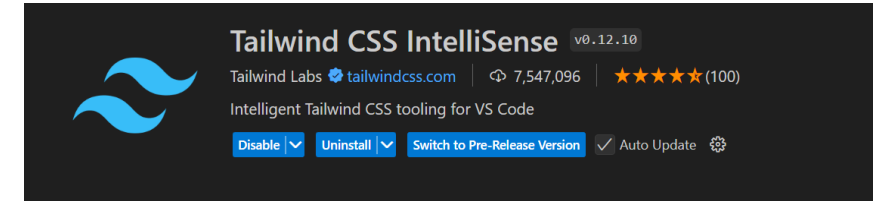


## 01 Next.js 풀스택 개발환경 준비하기

### Tips) Tailwind CSS IntelliSense VS Code 확장툴 설치

#### 1) Tailwind CSS IntelliSense

- Tailwind CSS 코드 자동완성 기능제공
- Tailwind Utility Class 를 순수 CSS 코드로 보기 기능제공



# 3. 챗봇 UI 및 Backend API 구현하기

## 1) Tailwind 기반 챗봇 UI 구현하기

```
<div className="m-4">
  SimpleBot
  {/* 메시지 입력 전송영역 */}
  <form className="flex mt-4">
    <input
      type="text"
      placeholder="Please Input Message"
      className="block rounded-md w-[500px] border-0 py-1 pl-2 text-gray-900 shadow-sm ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 "
    />
    <button
      type="submit"
      className="rounded-md bg-indigo-600 px-3 py-2 ml-4 text-sm font-semibold text-white shadow-sm hover:bg-indigo-500
        focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600"
    >
      Send
    </button>
  </form>
  {/* 대화이력 표시영역 */}
  <div className="mt-4">
    <ul>
      <li></li>
    </ul>
  </div>
</div>
```

# 3. 챗봇 UI 및 Backend API 구현하기

## 2) Backend RESTful API 구현하기

//기본호출주소: http://localhost:3000/api/articles  
import type { NextApiRequest, NextApiResponse } from "next";

```
type APIResultData = {  
  code: number;  
  data: IArticle[] | null | IArticle;  
  msg: string;  
};
```

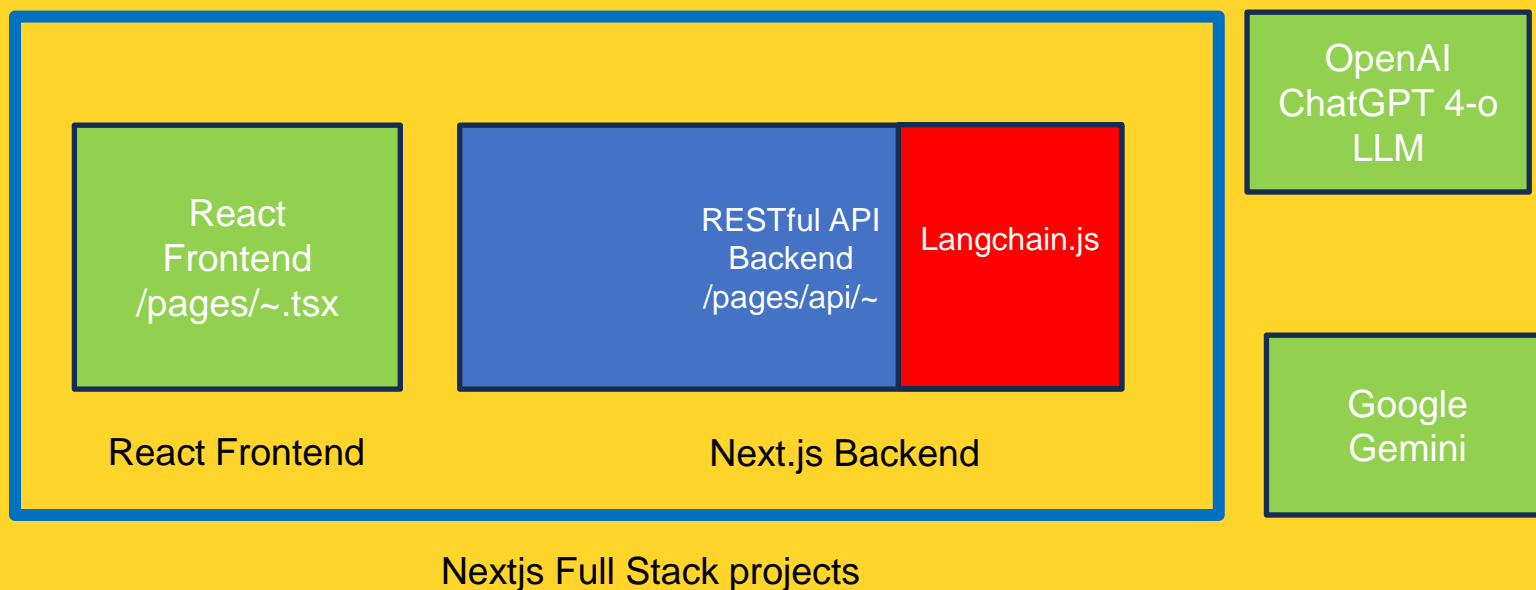
```
interface IArticle {  
  id: number;  
  title: string;  
  contents: string;  
  view_cnt: number;  
  display: boolean;  
  created_at: string;  
  created_id: string;  
}
```

```
export default function handler(  
  req: NextApiRequest,  
  res: NextApiResponse<APIResultData>  
) {  
  try {  
    //req.body = JSON.parse(req.body);  
    if (req.method === "GET") {  
      const articles: IArticle[] = [  
        {  
          id: 1,  
          title: "제목1",  
          contents: "내용1",  
          view_cnt: 0,  
          display: true,  
          created_at: "2022-01-01",  
          created_id: "admin",  
        }  
      ];  
  
      res.status(200).json({ code: 200, data: articles, msg: "Ok" });  
    } else if (req.method === "POST") {  
      //req.body = JSON.parse(req.body);  
  
      const article = {  
        id: req.body.id,  
        title: req.body.title,  
        contents: req.body.contents,  
        view_cnt: req.body.view_cnt,  
        display: req.body.display,  
        created_at: Date.now().toString(),  
        created_id: req.body.created_id,  
      };  
  
      console.log(req.body);  
      res.status(200).json({ code: 200, data: article, msg: "Ok" });  
    }  
  } catch (e) {  
    console.log(e);  
    res.status(200).json({ code: 500, data: null, msg: "Server Error" });  
  }  
}
```

## 02

# LLM 서비스 신청하기

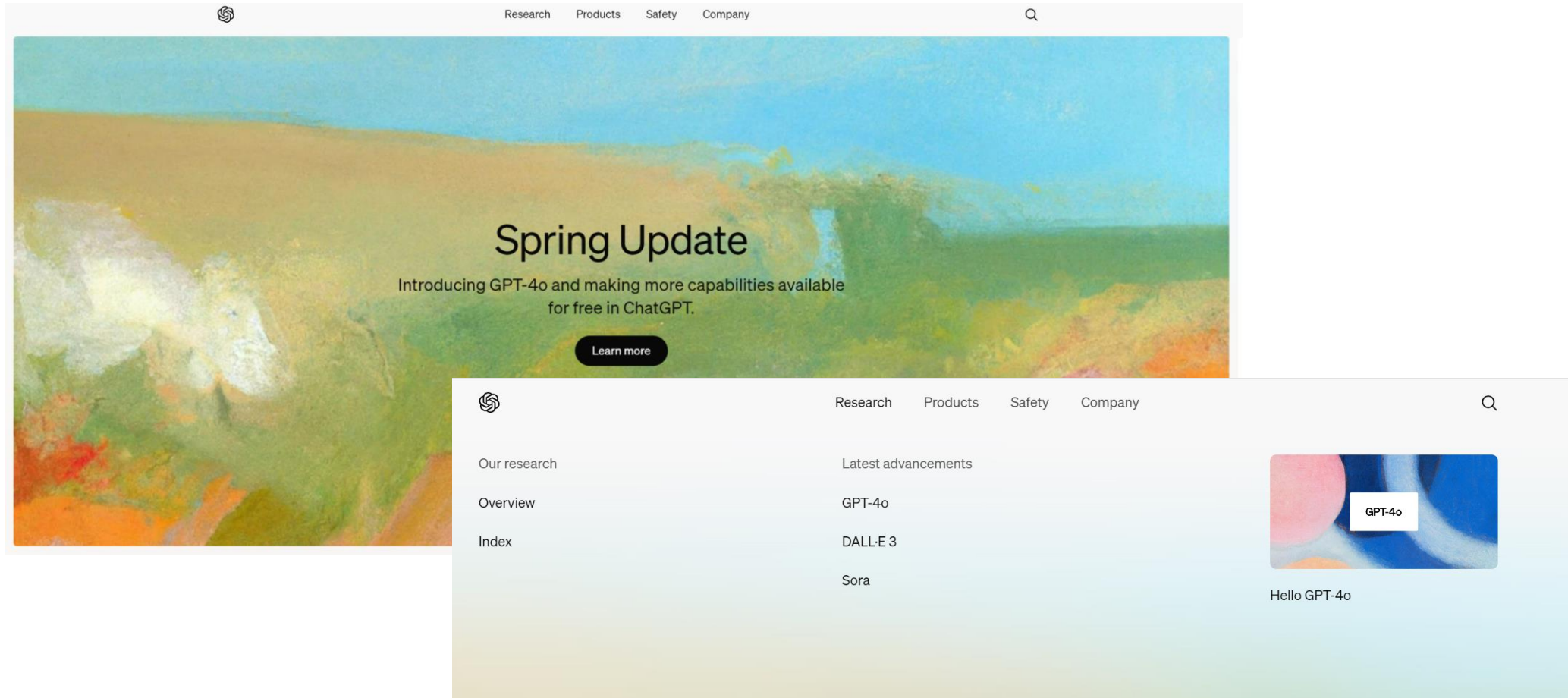
1. OpenAI LLM서비스 API 신청하기
2. API 인증키 발급
3. 프로젝트 환경변수 등록하기



## 02 LLM 서비스 신청하기

# 1. OpenAI LLM서비스 API 신청하기

- <https://openai.com/> : OpenAI 웹사이트

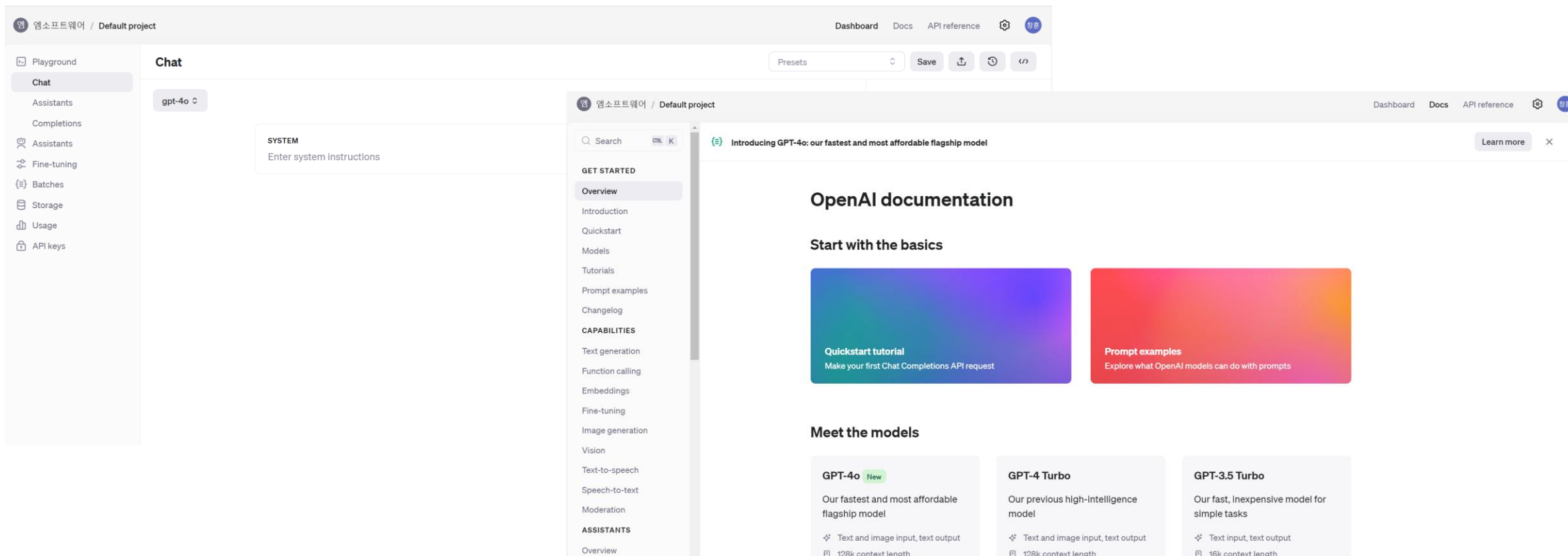




## 02 LLM 서비스 신청하기

# 1. OpenAI LLM서비스 API 신청하기

- <https://platform.openai.com> : API 키 발급링크-Dashboard
- <https://platform.openai.com/docs/overview> OpenAI 서비스 사용자 매뉴얼 (Docs)



## 02 LLM 서비스 신청하기

# 2. API 인증키 발급

- **OpenAI LLM서비스 API 신청하기** **OpenAI API 서비스 신청하기 – API 개발자 가이드 및 유료 서비스 신청**
- <https://platform.openai.com/docs/api-reference> : 각종 API 개발자 가이드
- <https://platform.openai.com/settings/organization/general> : 상단 우측 설정(Settings) > 조직 및 프로젝트 생성, 사용자 결제 유료서비스 신청

The image shows a screenshot of the OpenAI API documentation and settings page. The left sidebar contains a search bar and a navigation menu with categories like API REFERENCE, ENDPOINTS, and ASSISTANTS. The main content area is divided into two sections: 'Introduction' and 'Authentication'. The 'Introduction' section explains how to interact with the API and provides commands for installing the Python and Node.js bindings. The 'Authentication' section explains how to create API keys. The right sidebar shows the 'Organization settings' page, which includes details about the organization name, ID, and integrations.

**Introduction**

You can interact with the API through HTTP requests from any language, via our official Python library, or a [community-maintained library](#).

To install the official Python bindings, run the following command:

```
pip install openai
```

To install the official Node.js library, run the following command in your Node.js project directory:

```
npm install openai@^4.0.0
```

**Authentication**

The OpenAI API uses API keys for authentication. You can create API keys at a user or organization level. API keys are tied to a "bot" individual and should be used to provision access for production systems. Here are the different types of API keys:

1. **Project keys** - Provides access to a single project (**preferred option**); access [Project keys](#) project you wish to generate keys against.
2. **User keys** - Our legacy keys. Provides access to all organizations and all projects that the user belongs to.

**Organization settings**

**Details**

**Organization name**  
Human-friendly label for your organization, shown in user interfaces

**Organization ID**  
Identifier for this organization sometimes used in API requests

**Integrations**

**Weights and Biases**  
Your organization's Weights and Biases API Key. If set, enables the Weights and Biases integration for the [fine-tuning API](#). This key will be used to log training runs in your specified W&B project. See the [documentation](#) for more information.

**Features and capabilities**

**Threads**  
Threads page shows messages created with the Assistants API and Playground.

☒ Hidden

☐ Visible to organization owners

☐ Visible to everyone

**Usage dashboard visibility**  
Usage dashboard shows activity and costs for your organization.

☐ Visible to organization owners

☒ Visible to everyone

## 02 LLM 서비스 신청하기

# 2. API 인증키 발급

- OpenAI 신규 회원 가입 및 API 서비스 신청 절차 블로그 안내 링크  
<https://mixedcode.com/blog/6>

MIXEDCODE

공지사항

로그인 | 회원가입

KO


HOME WEB WEBFRONTEND MOBILE BACKEND DATA ARCHITECTURE DEVOPS/CLOUD AI TRENDS EVENT

Index

GPT-4

DALL·E 3

Sora



Introducing Sora: Creating video

OpenAI 회원가입 및 API Key 발급받기

By AIDA(아이다)

조회수 | 284

2024-04-29 07:28

안녕하세요. AI 서비스 활용 헬퍼 AIDA(아이다)입니다.  
개발과 다양한 AI 서비스를 연결하고 활용하는 방법들에 대해 순차적으로 글을 연재고자합니다

RELATED POST


DALL·E 3

DALL·E 3 understands significantly more nuance and detail than our previous systems, allowing you to easily translate your ideas into exceptionally accurate images.

DALL·E

Node.js 백엔드에 OpenAI DALL·E3 API 서비스 연결하기

EVENT



## 02 LLM 서비스 신청하기

### 3. 프로젝트 환경변수 등록하기

- .env 환경변수 설정 파일 추가
- API 키값 설정하기

# 03

## LangChain.js 기반 AI 챗봇 개발하기

1. LangChain 소개 및 주요 패키지 설치하기
2. LangChain.js 기초 모듈 활용하기
3. Agent 기반 챗봇 구현하기

## 03 LangChain.js 기반 AI 챗봇 개발하기

# 1. Lang Chain 소개 및 패키지 설치하기



### 1) LLM 사용의 한계점

- 매개변수 조정
- 프롬프트 보완
- 응답결과 조정 필요
- LLM 상태정보저장 및 대화컨텍스트 유지기능
- 이전 대화 컨텍스트 유지방안 제공
- 특정 기능에 특화된 다양한 모델을 함께 사용 가능해야함
- 각종 언어모델을 기반으로한 프로그래밍 환경제공 필요

### 2) LangChain?

- 언어 모델 기반의 애플리케이션을 개발하는 프레임워크
- 언어 모델은 랭체인을 통해 데이터 소스와 연결되고 해당 환경과 상호작용함
- Langchain은 OpenAI, Cohere, HuggingFace 등의 여러 대형 언어 모델(LLM)의 상호 작용과 체이닝을 간소화하도록 설계된 강력한 툴킷
- AI 모델, 에이전트, 벡터 저장소 및 검색 증강 생성(RAG)을 위한 기타 데이터 소스와 함께 작업하기 위한 도구와 추상화를 제공하는 오픈 소스 프레임워크
- LangChain을 사용하면 서로 다른 모델을 연결하여 다양한 AI 애플리케이션을 만들 수있다.

- <https://www.langchain.com/>
- <https://github.com/langchain-ai>

## 03 LangChain.js 기반 AI 챗봇 개발하기

# 1. LangChain 소개 및 패키지 설치하기

### 1) Models

- 언어 모델과의 인터페이스 제공, LLM 모델과 연동기능 제공
- 프롬프트 관리 및 언어모델별 인터페이스를 통해 모델 호출 및 정보 추출

### 2) Chains

- 각종 LLM 간 또는 다른 구성요소와 체인으로 연결하여 대화 흐름이나 태스크 흐름을 관리하는 단위
- 체인은 실행순서를 하드코딩을 통해 구현하는 방법

### 3) Agents

- 상위 지시문이 주어지면 체인이 사용할 툴을 선택할 수 있도록 함
- 추론엔진을 기반으로한 특정작업을 특정시점에서 특종 에이전트를 통해 수행할때 사용

### 4) Index

- LLM이 문서를 쉽게 탐색할 수 있도록 구조화 하는 모듈
- Document Loaders, Text Splitters, Vectorstores, Retrievers

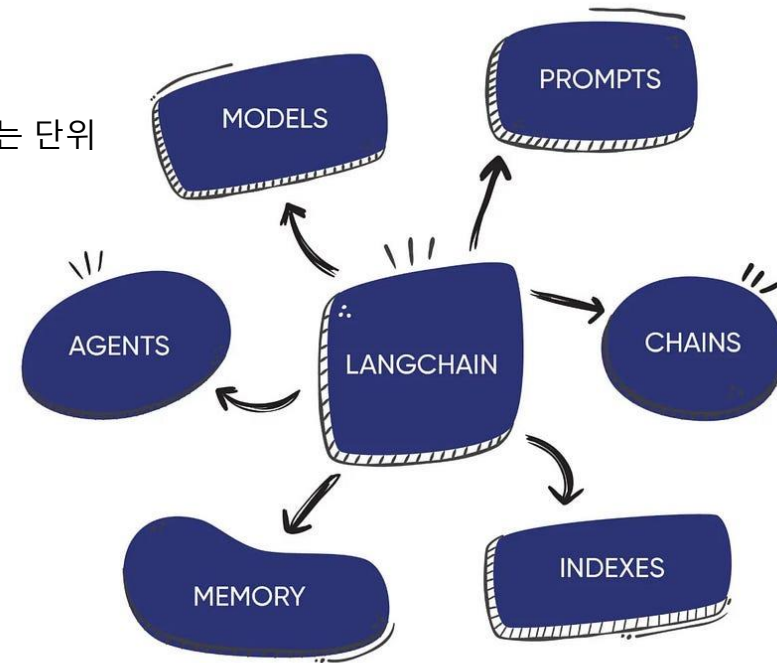
### 5) Prompts

- 초거대 언어모델에게 지시하는 명령문
- Prompt Templates, Chat Prompt Template, Example Selectors, Output Parsers

### 6) Memory

- 체인 실행 간 애플리케이션 상태 유지
- LLM과 프롬프트 기반 대화내용과 상태정보를 기억하고 저장하는 방법제공
- 과거 메시지에 직접 접근 가능한 환경제공
- 대화의 문맥관리에 사용

Structure of LangChain



## 03 LangChain.js 기반 AI 챗봇 개발하기

# 1. LangChain 소개 및 패키지 설치하기

### 1) Python

- 파이썬 언어지원
- Python LangChain 0.2 버전 지원 2024년 05월 업데이트
- <https://python.langchain.com/v0.2/docs/introduction/>
- <https://github.com/langchain-ai/langchain>

### 2) JavaScript/TypeScript

- 타입스크립트로 작성된 LangChain.js 지원
- LangchainJS 0.2 버전 지원 2024년 05월 업데이트
- LangChain.js + Next.js 환경 최적화 지원
- Node.js(ESM 및 CommonJS) - 18.x, 19.x, 20.x 지원
- **Vercel/Next.js(브라우저, 서버리스 및 Edge 함수) 지원**
- Deno 지원
- <https://js.langchain.com/v0.2/docs/introduction/>
- <https://github.com/langchain-ai/langchainjs>
- **npm i langchain**
- **npm install @langchain/core**
- **npm install @langchain/community**
- **npm install @langchain/openai**



## 03 LangChain.js 기반 AI 챗봇 개발하기

# 2. LangChain.js 기초 모듈 활용하기

### 1) Message 객체

- 랭체인에서 사용자 프롬프트로 전달된 메시지를 처리하기 위한 다양한 메시지 유형 객체 제공
- 사용자 메시지 와 시스템 메시지, LLM에서 반환된 AI메시지 등 각각의 용도에 맞게 사용권장

```
import { HumanMessage, SystemMessage } from "@langchain/core/messages"
```

- HumanMessage : 사용자 입력 프롬프트를 처리하기 위한 메시지 객체
- AIMessage : LLM모델에서 반환한 메시지 객체
- SystemMessage : LLM 시스템에 추가적인 정보 전달을 위해 사용하는 시스템 알림 메시지 객체

### 2) Model(LLM) Input/Output

- Model Input : 언어모델을 호출하는 행위
- Model Output : 호출한 언어모델에서 반환된 결과 의미
- (LLM)모델 생성

```
import { ChatOpenAI } from "@langchain/openai";
```

```
const model = new ChatOpenAI({  
  model: "gpt-4o",  
  temperature: 0.9,  
  apiKey: process.env.OPENAI_API_KEY,  
});
```

```
const result = await model.invoke("LangChain 컨셉을 알려줘");
```

- (LLM)모델 Input : **model.invoke('사용자메시지');**
- (LLM)모델 Output : **AIMessage**

# 2. LangChain.js 기초 모듈 활용하기

### 3) Prompt Template, Chain 활용

- 미리 정의된 프롬프트 구조 제공 템플릿/LLM에 최적화된 프롬프트 생성 기능제공
- 프롬프트 템플릿을 통해 좀더 LLM에 최적화된 프롬프트 데이터 생성해 LLM에 전달가능
- 프롬프트 템플릿 형식 및 실행

사용자 입력메시지--> 전용 프롬프트 템플릿 생성 -> 프롬프트 데이터생성 -> 프롬프트 데이터 언어모델에 전달 -> 언어모델

### 4) Output Parser 활용

- LLM(Model) 에서 전달된 응답값을 사용자에게 전달하기 전에 원하는 포맷의 응답값 포맷으로 파싱가능
- 다양한 포맷의 LLM Output Parser 제공됨

# 2. LangChain.js 기초 모듈 활용하기

### 5) Chain 활용하기

- 단순 LLM기반 챗봇의 경우 사용자 질문을 받아 LLM에 전달하고 LLM에서 반환된 응답 메시지를 사용자에게 반환한다.
- 데이터 처리로직, 업무로직등이 포함된 복잡한 구조의 LLM 챗봇 의 경우
- 하나의 질의/응답 프로세스내에서 다양한 사전/사후 작업이 처리 되어야합니다.
- 체인이란 대화 질의응답과 관련한 사전/사후 작업의 개별 단위를 말하며
- 복잡한 작업을 여러 단일 작업단위로 정의하고 정의된 작업들을 하나의 파이프 라인으로 연결해 대화흐름을 제어하는 방법을 제공함.

### 6) 대화이력 관리하기

- LLM모델 자체는 상태개념이 없음.그래서 대화흐름 기억못함.
- LangChain Memory 모듈을 이용해 대화이력 관리기능 제공
- 전체 대화 기록을 LLM모델에 전달가능

# 2. LangChain.js 기초 모듈 활용하기

### 7) Agent 활용하기

- 언어모델 자체로는 단순 대화만 가능하지 비즈니스 작업을 수행할 수 없다.
- LangChain의 에이전트는 비즈니스에서 목표하는 바를 구현 할 수 있는 다양한 방법 제공
- 에이전트는 LLM을 사용하여 어떤 작업을 어떤 순서로 수행할지 결정하는 역할제공
- LLM에서 제공하는 이미학습된 정보가 아닌 다른 리소스 툴들을 이용해 특정작업을 수행하는 역할을 Agent라고함
- 에이전트는 필요 시 LLM외적인 각종 도구들을 사용하여 그 결과를 반환받아 활용함
- Agent란 각종 툴(Tool)을 이용해 LLM에 어떤 작업을 어떤 순서로 수행할지 결정하는 역할제공
- 도구(Tool): 에이전트가 외부 인터페이스와 상호작용하는 데 사용할 수 있는 함수
- 도구의 종류 : Google 검색, 데이터베이스 조회, Python REPL, 기타 체인,위키디피아사전,네이버 API,공공데이터 API
- Agent는 LLM을 추론 엔진으로 사용하여 어떤 작업을 수행할지와 해당 작업에 대한 입력이 무엇인지 결정하는 시스템
- 그런 다음 해당 작업의 결과를 에이전트에 다시 피드백하여 추가 작업이 필요한지 또는 완료해도 되는지 여부를 결정합니다.
- LangChain은 각종 도구와 상호작용하여 에이전트를 구현 빌드합니다.

### 3. Agent 기반 챗봇 구현하기

#### 1) 검색엔진 기반 검색 Agent 구현하기

- AI Tool 전용 검색엔진 (Tavily) 타빌리 활용
- 타빌리는 고급 알고리즘과 전문가 팀을 활용하여 포괄적이고 신뢰성 있는 연구 결과를 제공함으로써 연구 과정을 간소화하고 가속화하겠다고 약속하는 AI 연구 도구
- 타빌리의 핵심 제공 기능 중 하나는 언어 모델 (LLM)에 중립적인 디자인으로, 모든 LLM과 통합이 가능하도록 설계된 검색 API입니다.
- 이 API는 여러 소스를 검토하여 각각에서 가장 관련성 높은 콘텐츠를 찾아내어 LLM 컨텍스트를 최적화하는 특징을 가지고 있습니다

<https://app.tavily.com/sign-in>

- 회원가입/키발급(무료가능)
- 회원가입후 디폴트키 발급됨/ 키 복사함

#### 2) Vector Store 기반 Agent 구현하기

- 랭체인에서 제공하는 메모리 기반 Vector 저장소를 Agent로 활용 가능
- 웹페이지 문서를 로딩하고 단어 기반으로 문자로 분할한 문서를 벡터 스토어에 벡터화하여 저장 후 조회사용함
- `npm i cheerio` 웹페이지 문서 로딩 오픈소스

# 감사합니다.

Lang ChainJS AI 챗봇개발하기 With Next.js

지니공공아카데미

강창훈 | 믹스드코드닷컴

[010-2760-5246](tel:010-2760-5246)

[ceo@msoftware.co.kr](mailto:ceo@msoftware.co.kr)

<https://mixedcode.com>

<https://jiny.academy>