**Secret**

보안포털 개발표준가이드(React)

Ver. 1.1

|  |  |
| --- | --- |
|  | |
|  |  |
|  |  |
| Copyright ⓒ LG CNS  LG CNS의 사전 승인 없이 본 내용의 전부 또는 일부에 대한 복사, 배포, 사용을 금합니다. | |

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| V1.0 |  | 최초작성 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목 차

[1. 개요 1](#_Toc176957693)

[1.1 목적 1](#_Toc176957694)

[1.2 적용범위 1](#_Toc176957695)

[1.2.1 Vite 1](#_Toc176957696)

[1.2.2 TypeScript 1](#_Toc176957697)

[1.2.3 React 1](#_Toc176957698)

[2. 로컬 개발 환경 세팅 1](#_Toc176957699)

[2.1 Node.js 설치(LTS 20 버전) 1](#_Toc176957700)

[2.2 IDE 설치 2](#_Toc176957701)

[2.2.1 IntelliJ Community Edition 설치 2](#_Toc176957702)

[2.2.2 Visual Studio Code 설치 2](#_Toc176957703)

[2.3 Git 설치 4](#_Toc176957704)

[2.4 프로젝트 구동 4](#_Toc176957705)

[2.4.1 소스코드 다운로드 4](#_Toc176957706)

[2.4.2 프로젝트 구동 4](#_Toc176957707)

[3. 개발표준 6](#_Toc176957708)

[3.1 디렉토리 구조 6](#_Toc176957709)

[3.2 명명 규칙 6](#_Toc176957710)

[4. 개발 방법 7](#_Toc176957711)

[4.1 배포 환경별 환경 변수 설정 7](#_Toc176957712)

[4.1.1 공통 환경변수(.env) 7](#_Toc176957713)

[4.1.2 배포 환경별 환경변수(.env.dev / .env.prd) 8](#_Toc176957714)

[4.1.3 환경 변수 파싱 8](#_Toc176957715)

[4.2 model 개발 8](#_Toc176957716)

[4.3 service 개발 9](#_Toc176957717)

[4.4 components, pages 개발 9](#_Toc176957718)

[4.5 utils 개발 9](#_Toc176957719)

[4.6 scss 개발 10](#_Toc176957720)

[5. 개발 예제 11](#_Toc176957721)

[6. 빌드 및 배포 11](#_Toc176957722)

[6.1 빌드 11](#_Toc176957723)

# 개요

## 목적

본 문서는 보안 관련 포털(privacy center) 구축 프로젝트에 투입되는 개발자들에게 필요한 개발표준을 제공하기 위한 문서이다. 각 단계는 절차 중심으로 작성되어 개발자들이 쉽게 이해할 수 있도록 하였다.

## 적용범위

Vite + TypeScript + React 개발 환경에 대한 가이드이다.

### Vite

빠르고 현대적인 frontend 빌드 도구로, 빠른 개발 서버와 최적화된 빌드를 제공

### TypeScript

JavaScript에 정적 타입을 추가하여 코드의 안정성과 가독성을 높이는 자바스크립트 슈퍼셋

### React

사용자 인터페이스(UI) 구축을 위한 컴포넌트 기반의 자바스크립트 라이브러리

# 로컬 개발 환경 세팅

## Node.js 설치(LTS 20 버전)

Node.js는 자바스크립트 런타임 환경으로, Fronted 개발 후 이를 실행할 수 있는 환경을 제공한다

공식 사이트에서 **LTS 20 버전**을 다운받아 설치하도록 한다.

(다운로드 링크: <https://nodejs.org/en/download/prebuilt-installer>)

Node.js를 설치하면 자바스크립트 패키지 관리자인 npm도 함께 설치된다.

텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명

## IDE 설치

본 가이드에서는 IntelliJ Comminy Edition과 Visual Studio Code 설치 및 세팅 방법에 대하여 안내한다.

IntelliJ Community Edition은 Backend 개발 툴과 동일하다는 장점이 있고(자바스크립트 개발을 위한 플러그인 및 기능은 Ultimate 버전에서만 지원한다.), Visual Studio Code는 자바스크립트 개발을 위한 다양한 플러그인 및 기능을 지원한다는 장점이 있다.

따라서 두 IDE 중 적합한 IDE를 선택하여 설치를 진행하도록 한다.

### IntelliJ Community Edition 설치

자바 및 자바스크립트를 포함한 다양한 프로그래밍 언어를 위한 통합 개발 환경(IDE)이다. 공식 사이트에서 **Community Edition**을 다운받아 설치하도록 한다.

(다운로드 링크: <https://www.jetbrains.com/ko-kr/idea/download/?section=windows>)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

### Visual Studio Code 설치

Visual Studio Code는 소스 코드 에디터로, Frontend 개발에 필요한 다양한 확장 프로그램을 제공한다.

공식 사이트에서 설치 프로그램을 다운받아 설치하도록 한다.

(다운로드 링크: <https://code.visualstudio.com/download>)

#### 확장 프로그램 설치

Visual Studio Code에서 React 개발 환경 구성을 위해 기본적으로 3개의 플러그인(ESLint, Prettier, ES7+) 이 필요하다.

아래 3개의 VSCode 플러그인을 Extenstions에서 검색하여 설치한다.

**- ESLint**

텍스트, 스크린샷, 소프트웨어, 멀티미디어이(가) 표시된 사진

자동 생성된 설명

**- Prettier**

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**- ES7+**

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

#### 설정 변경

**ctrl + ,** 입력 > Settings 활성화 > default formatter 검색 > Prettier – Code Formatter로 변경

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

format on save 검색 > 체크

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이후 Visual Studio Code를 종료 후 재실행 한다.

## Git 설치

Git은 분산 버전 관리 시스템으로, 소스코드의 변경 이력을 관리하고 협업을 용이하게 한다.

공식 사이트에서 설치 프로그램을 다운받아 설치하도록 한다.

(다운로드 링크: <https://git-scm.com/downloads>)

## 프로젝트 구동

### 소스코드 다운로드

Repository 주소: https://github.com/LGU-CTO/nppmo-ui.git

CMD 또는 git bash에서 다음 명령어 실행

- 명령어 실행: git clone https://github.com/LGU-CTO/nppmo-ui.git

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

### 프로젝트 구동

#### IntelliJ

IntelliJ 실행 후 다운로드 받은 프로젝트 폴더 오픈

alt + F12 버튼을 눌러 Terminal 실행

**npm install 입력(최초 한번만 수행)**

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

**npm run 배포환경** 입력(ex. npm run dev, npm run prd)

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

브라우저가 자동으로 열리지 않는다면 ctrl을 누른 채로 Local 옆의 <http://localhost>:(포트번호)/ 클릭

#### VSCode

소스코드 다운로드 이후 해당 폴더에서 git bash를 실행한 후 **code .** 명령어를 입력하여 VSCode 실행

VSCode 실행 후, **ctrl + `** 을 입력하여 Terminal 실행

**npm install** 입력**(최초 한번만 수행)**

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**npm run 배포환경** 입력(ex npm run dev, npm run prd)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

브라우저가 자동으로 열리지 않는다면 ctrl을 누른 채로 Local 옆의 <http://localhost>:(포트번호)/ 클릭

# 개발표준

## 디렉토리 구조

페이지 라우팅 기준으로 구성하는 pages 구조이다.

| **폴더 구조** | | | | **용도** |
| --- | --- | --- | --- | --- |
| {프로젝트 모듈 경로} | /public | | | 공개 정적 자산 포함(URL로 접근 가능) |
| /src | /assets | | 이미지, 아이콘 등 정적 자산 포함 |
| /components | /atoms | 재사용 컴포너트 관리(Button, Radio 등) |
| /errors | 에러 페이지 관리 |
| /layouts | 전체 페이지 레이아웃 관리(Header, Footer 등) |
| /hooks | | 커스텀 hook 관리 |
| /pages | /(page단위) | 페이지 단위의 컴포넌트 폴더로 구성 (ex. login – Login.tsx, Login.scss  main – Main.tsx, Main.scss) |
| /router | | 페이지 라우터 구성 관리 |
| /service | | Backend API를 호출하기 위한 함수 관리  업무 단위별로 파일을 만들어 구성하되, 예외가 있을 수 있다.  (ex. MemberAPI.ts) |
| /styles | | Global Style 저장 |
| /utils | | 공통 함수 관리 |
| \*\*/model | | 데이터 구조 인터페이스 또는 클래스 정의(필요시) |

## 명명 규칙

모든 이름은 코드만으로 기능과 의미를 알 수 있도록 작성한다(Self-Document).

각 구분에 따라 표기법이 달라지며, 아래 표에 기준하여 작성한다.

| **구분** | **표기법** | **예시** | **비고** |
| --- | --- | --- | --- |
| 변수명 | Camel | let memberList | boolean 타입은 is/has/are 등으로 시작하도록 작성 |
| Snake | const MAX\_NUMBER | readonly 상수는 대문자와 언더스코어(\_)를 사용하여 작성 |
| 메소드/함수명 | Camel | function getMemberList(){…} | boolean 타입은 is/has/are 등으로 시작하도록 작성 |
| 파일명 | Pascal | Example.ts  Example.tsx | typescript만 사용할 경우 .ts 확장자 사용  React Component가 포함될 경우 .tsx 확장자 사용 |

**타입 스크립트 파일은 React Component가 포함될 경우 확장자가 달라짐(ts -> tsx)에 유의**

# 개발 방법

## 배포 환경별 환경 변수 설정

import.meta.env 객체를 이용해 환경 변수에 접근할 수 있으며, 공통 또는 배포 환경별로 환경 변수를 설정하여 사용할 수 있다.

import.meta.env로 가져오는 환경 변수는 빌드 시 정적으로 대체되며, 문자열로 파싱 되므로 boolean 또는 숫자를 파싱했을 경우 코드에서 원하는 타입으로 변환해야 한다.

### 공통 환경변수(.env)

프로젝트 루트 폴더의 .env 파일에서 설정할 수 있다.

배포 환경에 관계없이 공통으로 사용하거나, default 값이 필요할 경우 여기에 설정한다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

### 배포 환경별 환경변수(.env.dev / .env.prd)

프로젝트 루트 폴더의 .env.dev(개발환경) 또는 .env.prd(운영환경) 파일에서 설정할 수 있다.

개발 환경별 필요한 환경 변수를 설정하며, .env(공통)와 **환경변수가 중복될 경우 배포 환경 별 환경변수(.env.dev / .env.prd)로 덮어 씌워진다.**

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

### 환경 변수 파싱

**import.meta.env.환경변수명**으로 소스코드 내에서 파싱할 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

## model 개발

정보를 전달하는 목적으로 만들어진 Entity로, 필요한 요소가 무엇인지 파악한 후 작성하여야 한다. 주로 interface로 구현하며 Constant로 구성되어 있을 경우에는 class로 구현한다.

.ts 확장자를 사용한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

## service 개발

Rest API 방식을 사용하여 서버와 요청을 주고받는 부분으로, 응답에 대한 예외처리를 해준다.

.ts 확장자를 사용한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

## components, pages 개발

전달받은 데이터를 토대로 화면을 구성하는 부분이다. components 요소는 재사용이 가능하도록 작성되어야 하며, 데이터를 가져와 화면에 그리는 일을 전달한다.

.tsx 확장자를 사용하며, **함수형 컴포넌트 사용을 권장한다.**

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

## utils 개발

공통 함수를 정의한다.

.ts 확장자를 사용한다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

## scss 개발

공통 적용될 스타일은 /src/styles 경로의 App.scss에 적용한다.

폰트, 텍스트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

컴포넌트별 개별 적용될 스타일은 해당 컴포넌트와 동일한 위치에 적용한다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

스타일을 적용할 때는 해당 .scss 파일을 import 한다.



.scss 내부의 클래스명은 파일 경로와 이름이 다르더라도 클래스명이 동일하면 최종적으로 생성되는 css에서는 동일한 클래스명으로 취급되기 때문에, 이름이 중복되지 않도록 유의한다.

생성한 스타일을 한 파일에만 적용하기 위해서는 .module.scss 파일 생성하여 사용하는 방법도 있으나 코드 소스코드가 복잡해지므로 추천하지 않는다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

# 개발 예제

프로젝트 구동 후 개발 샘플 버튼을 클릭하거나 아래 주소를 입력하여 개발 예제를 확인할 수 있다.

<http://localhost:3000/examples>

텍스트, 폰트, 스크린샷, 그래픽이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

# 빌드 및 배포

## 빌드

터미널 실행 후 **npm run build:배포환경**(ex. npm run build:dev) 입력

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

빌드 완료 시 dist 폴더 생성

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

빌드 후 npm run preview로 빌드한 앱을 테스트할 수 있다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

1. 변경 내용: 변경이 발생되는 위치와 변경 내용을 자세히 기록(장/절과 변경 내용을 기술한다.) [↑](#footnote-ref-1)