

개발환경 가이드

☀ 상태

진행중

1. 설치 가이드



서버 환경 설정 command 양식

1. 서버 환경 구성시 필요한 모듈 리스트

(Linux 실행 command 로 설치하셨으면 command로 주시면 더 좋아요~)

ex) node v18.15.0 또는 `sudo apt install nodejs` 같은 형식

2. 해당 서버 실행하는 명령어 (command 로 주시면 됩니다. ex) `node app.js`)

설치 순서 맞춰서 작성해주세요!

▼ 설치가이드

1. Web Front

1. Curl 설치

- `sudo apt-get install -y curl`

2. Node.js 설치(`v.18.15.0`)

```
sudo apt update
```

```
curl -fsSL https://deb.nodesource.com/setup_18.15.0 | sudo -E bash - &&\nsudo apt-get install -y nodejs
```

2. Web Back

1. Web Front와 같습니다.

- Node version은 18.12.1
- Npm version은 9.6.1

3. Validator Server

- Node.js (LTS) 18.12.1

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash - &&\nsudo apt-get install -y nodejs
```

- package.json 에 설정된 라이브러리 설치
 - `/API_validator_server` 디렉토리에서 `npm i`

4. MySQL

- MySQL 8.0.20

```
sudo apt update\nsudo apt install mysql-server
```

2. JIRA Convention 정의

▼ JIRA Convention

- 단위
 - Epic
 - Story
 - Task
- Epic
 - 가장 큰 단위
 - 하나의 기능 및 제공되는 서비스 목록
 - 네이밍 규칙
 - 명사로 마무리

- 맨 앞에 대괄호를 사용해 카테고리 설정(영어의 경우 대문자로)

- Story Point는 0
- Epic Name과 Summary는 동일하게
- 카테고리

카테고리 명	내용
DEV	개발
PLN	기획
DSN	디자인
TEST	테스트
STUDY	학습
ETC	기타

- 예시

- [DEV] 사용자 API
- [PLN] 기획
- [DSN] 디자인
- [TEST] 사용자 API
- [STUDY] 학습
- [ETC] 팀 기술 블로그

- Story

- Epic과 관련한 개발 외적인 업무
- 네이밍 규칙
 - 명사로 마무리
 - 맨 앞에 대괄호를 사용해 카테고리 설정(영어의 경우 대문자로)
- 최대 Story Point는 4
- 카테고리

카테고리 명	내용
FE	프론트엔드
BE	백엔드

카테고리 명	내용
DB	데이터베이스
IoT	라즈베리파이
HW	외관 설계 및 전장 디자인
DSN	디자인
공통	공통
SVR	서버

- 예시
 - [공통] PPT 작성
 - [공통] 포팅 메뉴얼 작성
- Task
 - Epic 기능에 해당하는 개발 업무
 - 네이밍 규칙
 - 명사로 마무리
 - 맨 앞에 대괄호를 사용해 카테고리 설정(영어의 경우 대문자로)
 - 구체적인 구현 예정 기능 명시
 - 최대 Story Point는 4
 - 카테고리

카테고리 명	내용
FE	프론트엔드
BE	백엔드
DB	데이터베이스
IoT	라즈베리파이
HW	외관 설계 및 전장 디자인
DSN	디자인
공통	공통

- 예시
 - [BE] 사용자 조회 API
 - [FE] 마이페이지 조회

- [공통] 프론트 백 연동

3. Git Convention 정의

▼ Git Convention

GIT branch convention

Branch 종류

- main
 - 배포 가능한 상태의 결과물
- develop
 - 구현한 기능을 병합하기 위한 브랜치
 - 통합 폴더의 기능
- feature
 - 개별 기능 구현 브랜치
 - 기능 개발 완료 시 삭제
 - 네이밍 규칙
 - feat/파트/기능 명세서 id
 - feat/fe/12
 - feat/be/1

Merge 관련

- MR은 Git lab을 통해 요청 및 수락
- MR시
 - title : merge하려는 브랜치 이름
 - description : 비우기
 - 머지 후 브랜치 삭제(local에서도 삭제)
 - git branch -d [Branch 이름]

GIT commit convention

각자아이콘 [백프론트:커밋타입] 어떤 커밋인지 내용

🔧 [FE: fix] 메인 영상 삽입

☀️ [FE: feat] 애니메이션 추가

- 각자 이모티콘

1. Logan : 🥑

2. Brian : 🍊

3. Hanson : 🍷

4. Julie : ✨

5. May : 🐼

6. Juoh : 🍀

- 개괄

- 모든 커밋 메시지는 `한글` 로 작성

- 구조

- 기본적으로 커밋 메시지는 아래와 같이 제목/본문/꼬리말로 구성

- 커밋 타입(Type)
 - feat : 새로운 기능 추가
 - fix : 버그 수정
 - docs: 문서 내용 변경
 - 예시
 - "fix" --> '[BE] fix'
 - study: 공부
 - settings: 환경설정
 - style: 포맷, 세미콜론 수정 등 코드가 아닌 스타일에 관련된 수정
 - refactor: 리팩토링 코드
 - test: 테스트 코드 추가 및 리팩토링 테스트 등
 - chore: build task 수정, 프로젝트 매니저 설정 수정 등
 - 타입은 소문자로 시작
 - 타입은 항상 대괄호 안에 파트를 입력하여 시작
- 제목(Subject)
 - 제목은 50자 이내로, 대문자로 시작하며 모두 소문자로 작성
 - 파일명의 경우에는 대소문자 고려하지 않음
 - 마침표로 끝나지 않도록 함
 - 과거시제를 사용하지 않고 명령어로 작성
 - 예시
 - "feat : Logined" --> "feat : Login"
 - "feat : Added" --> "feat : Add"
- 본문(Body)
 - 선택사항
 - 부연 설명 필요 시 작성
 - 100자 미만 작성 권장
 - 영어
- 꼬리말(Footer)
 - 선택사항
 - issue tracker id를 작성할 때 사용
 - JIRA Code 작성 시 사용
- 예시


```

```jsx
♥[FE: feat] Login DDesign.py

한글 설명쓰기
- 선택사항

JIRA Code : S08P21S002-7

```

```

4. URL Convention 정의

▼ URL Convention

1. 주소 뒤, 도메인을 우선 붙이도록 한다.

- 도메인 = DB 테이블

주소/도메인/...

2. 도메인에 대해 복수를 사용한다.

- GET /user/123 (x)
- GET /users/123 (o)

```
GET /users <-> users 리스트 조회
POST /users <-> users.push(data)
GET /users/12 <-> select * from users where id = 12;
```

3. 명사를 사용한다. (동사x)

- 동사보다는 명사를 사용해 api 를 더 깔끔하게 관리한다.

```
GET /getUsers (x) --> GET /users (o)
POST /createNewUser (x) --> POST /users (o)
GET /getAccountsOfUserByUserId/123 (x) --> GET /users/123/accounts (o)
```

4. 하이픈(-) 을 사용한다.

- 주소에서 표현하는 리소스가 길어지는 경우, 하이픈(-) 을 이용해 구분해 가독성을 높이도록 한다.

```
GET /device-management/managed-devices
```

5. 소문자를 사용한다.

- URI 경로에서는 대문자보다는 소문자를 사용한다.

6. 컬렉션 필터링은 URL 쿼리를 통해 구현한다.

- where name = ? 처럼 필터링이 필요한 경우, 새로운 API 생성 보다는 URL 쿼리를 통해 필터링할 수 있도록 한다.

```
GET /users?region=ko
GET /users?gender=female
GET /users?region=ko&gender=female
```

5. React Convention 정의

▼ React 버전

- Node.js : 18.15.0 (LTS)
- React : 18.2.0
- Mui : 5.11.12

▼ React Convention

- 참고 사이트

[react] react 코딩 컨벤션

최근 면접을 보면서 나의 코딩 스타일에 대해 생각하게 되었다.
나는 많은 인강을 보면서 react를 독학한 케이스인데 아무래도 인강 강사님의 코드 스타일을 따라가게 되었고 무의식적으로 그렇

 <https://phrygia.github.io/react/2022-04-05-react/>

Welcome to my
Awesome Blog ✨

▼ CSS Convention

- 참고 사이트

CSS naming methodology(BEM)

✓ Naming Case CSS Naming 방법론을 알아보기 전, 대표적인 Naming Case는 어떤 것들이 있는 지 알아보자 kebab-case(케밥 표기법) kebab-cas

v <https://velog.io/@2seunghye/CSS-naming-methodology>
BEM

velog