
Coding Standard, Repository Management and Review Proces

ALBASIDAE

팀명: UOS Works		프로젝트명: 알바시대	
2014920047 임진우	2020920031 서제임스	2020920058 조성채	
2022920010 김동하	2022920024 박동찬	2022920045 이승목	

Modification History

버전	일자	변경 내역	작성자
v01	2024-11-24	최초 작성	전체

Table of Contents

- 1 Coding Standard: Naming Convention
- 2 Github Branch (Naming) Rules
- 3 Code Review Process

1 Coding Standard: Naming Convention

1.1 명명 규칙

1.1.1 파일명, 클래스: PascalCase

1.1.2 변수, 함수, 메소드: camelCase

1.1.3 목업데이터, 상수: UPPER_CASE

1.2 메서드 이름

1.2.1 일반적으로 소문자로 시작하는 동사구

1.2.2 함수 이름은 일반적으로 값을 설명하는 명사구로

1.2.3 필드의 값을 설정하는 함수는 “set”

1.2.4 필드의 값을 접근하여 리턴하는 함수는 “get”

1.2.5 조건을 묻는 부울 함수의 이름은 대개 “is”로 시작

1.3 변수 이름

1.3.1 일반적으로 소문자로 시작

1.3.2 모호한 변수명 사용 x

1.3.3 길어도 충분한 의미를 담을 수 있도록

1.4 패키지 이름

1.4.1 일반적으로 모두 소문자이며 명사

1.5 API Convention

1.5.1 Response body :

```
public class ApiResponse<T> {  
    private boolean result;  
    private String message;  
    private T data;  
}
```

2 Github Branch (Naming) Rules

2.1 Notion 에서 새로운 업무 단위마다 티켓을 생성 후, Description 에 어떤 내용인지 작성해주세요.

2.2 각 티켓의 ID 명과 동일한 branch 를 생성해주세요.

2.3 하나의 branch 의 내용이 너무 많지 않도록 만들어주세요.

2.4 새로운 branch 를 만들때, main branch 에서 git pull 을 하고 생성해주세요.

2.5 Commit message 구조

2.5.1 메세지 끝에 마침표 불가

2.5.2 <Type>: <Subject>

2.5.3 Type

2.5.3.1 feat : 새로운 기능 추가, 기존의 기능을 요구 사항에 맞추어 수정

2.5.3.2 fix : 기능에 대한 버그 수정

2.5.3.3 build : 빌드 관련 수정

2.5.3.4 chore : 패키지 매니저 수정, 그 외 기타 수정 ex) .gitignore

2.5.3.5 ci : CI 관련 설정 수정

2.5.3.6 docs : 문서(주석) 수정

2.5.3.7 style : 코드 스타일, 포매팅에 대한 수정

2.5.3.8 refactor : 기능의 변화가 아닌 코드 리팩터링 ex) 변수 이름 변경

2.5.3.9 test : 테스트 코드 추가/수정

2.5.3.10 release : 버전 릴리즈

2.5.4 Subject

2.5.4.1 Type 과 함께 헤더를 구성합니다. 예를들어, 로그인 API 를 추가했다면
다음과 같이 구성할 수 있습니다.

2.5.4.2 ex) feat: add login api

3 Code Review Process

3.1 개별 branch 에서 작업을 모두 한 뒤, 모든 commit 을 push 하고 github 에서 PR(Pull Request)을 합니다.

3.2 main branch 에 push 금지

3.3 branch PR 후, main branch 에 merge 가능

3.4 PR merge 시 peer review 한 명 이상 해야함

3.5 PR description