



Git Convention

전체 작업 흐름

1. 기능 명세서를 기반으로 팀장이 epic을 생성해두었음
2. 각 파트 리드가 story를 생성함
 - 필수 지정 요소 : 담당자 , 컴포넌트
 - 권장 요소 : 우선순위
3. 해당 story의 담당자는 sub task를 등록함
 - 레이블 , 우선순위 선택하는 것 권장
4. 담당자는 story 번호를 넣어 gitlab에서 branch를 생성함
 - ▲ 생성할 branch는 본인의 파트의 브랜치에서 분기할 것 !
5. 담당자가 sub task에 등록한 기능 단위로 커밋
 - sub task의 번호를 넣어 커밋 메시지를 작성
 - 약소한 규모의 커밋의 경우 sub task를 생성하지 않고, 이슈 번호 작성을 생략해도 무방
6. story에 해당하는 기능 개발을 완료한 경우, Merge Request 생성
 - source branch (작업한 branch) / target branch (파트의 branch)
 - ▲ title에는 story의 번호를 넣어 작성할 것 !!!
 - assign은 본인에게 , reviewer는 각 파트 리드를 지정
 - ▲ 셀프로 merge 금지
7. 각 파트 리드가 확인 후 리뷰 혹은 승인 후 merge

🔔 커밋, 브랜치 이름, Merge Request에서 컨벤션 및 템플릿을 따를 것 !!

- jira의 추적 활용 목적

1. 커밋 규칙

커밋 타이밍

- 기능 단위로 커밋
- 빌드 성공이 확인된 상태에서 커밋
- 실행 가능한 상태로 커밋

커밋 메시지 구조

```
<type>: <subject> [<jira-issue-number>]
```

```
<body>
```

제목(필수) : 커밋타입 : 명료한 제목 [지라 이슈번호]

본문(선택) : 작업한 내용

간단한 수정같이 jira의 하위 이슈로 등록하기 애매한 커밋의 경우는 이슈번호 생략 가능

커밋 타입

- `feat` : 새로운 기능 추가
- `fix` : 버그 수정
- `docs` : 문서 수정
- `style` : 코드 포매팅, 세미콜론 누락
- `refactor` : 코드 리팩토링
- `test` : 테스트 코드 추가/수정

- `chore` : 빌드 업무, 패키지 매니저 수정

예시

`feat`: 회원가입 API 구현 [`PROJECT-123`]

- 이메일, 비밀번호 유효성 검증 추가
- 중복 이메일 체크 기능 구현
- 회원가입 완료 메일 발송 기능 추가

템플릿 자동화 방법 :

<https://velog.io/@cnsrn1874/git-커밋-메시지-템플릿>

2. 브랜치 규칙

브랜치 생성은 JIRA 스토리 단위로

브랜치 명명 규칙

`<type>/<jira-issue-number>-<description>`

브랜치 타입

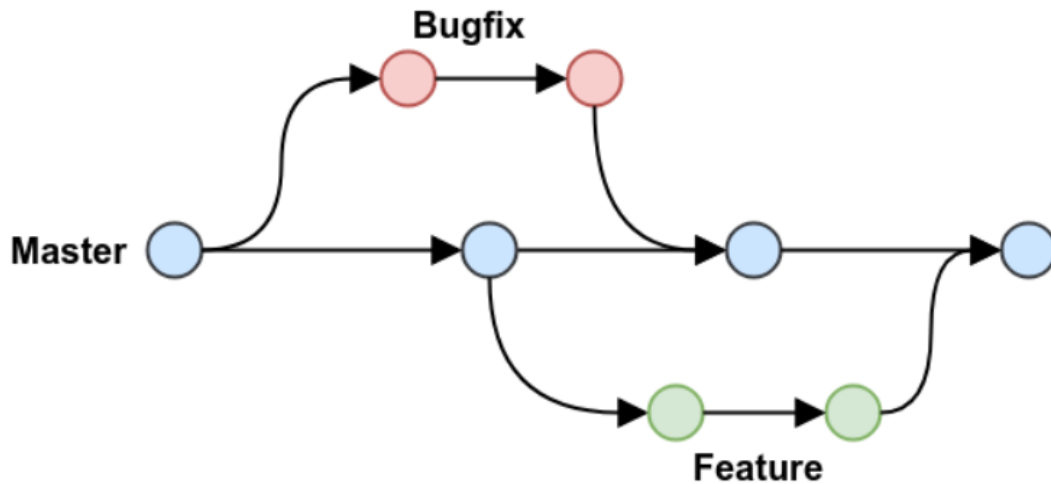
- `feature` : 새로운 기능 개발
- `fix` : 버그 수정
- `hotfix` : 긴급 버그 수정
- `refactor` : 코드 리팩토링

예시

기능 추가 위한 브랜치: `feature/PROJECT-123-implement-login`

버그 수정 위한 브랜치: `fix/PROJECT-456-login-validation`

▼ 이런 방식을 github flow 라고 함



3. Merge Request 규칙

MR 생성 시기

- 코드 리뷰가 필요한 작업 완료 시
- 테스트 완료 후 (한다면)

MR 제목 형식

[#jira 스토리 이슈번호 (브랜치에 사용한 이슈번호)] 브랜치에서 작업한 간단 명료 타이틀

MR 템플릿

작업 내용

- 구현한 기능 설명
- 변경사항 요약

테스트 결과 (수행 시)

- 테스트 방법
- 테스트 결과

리뷰 요청 사항

- 중점적으로 봐야 할 부분
- 논의가 필요한 부분

MR 설정 항목

- Reviewers: 코드 리뷰어 지정
- Source/Target Branch 확인

예시

작업 내용

- 회원가입 페이지 UI 구현
- 회원가입 API 연동
- 유효성 검증 로직 추가

테스트 결과

- 이메일 중복 체크 테스트 완료
- 비밀번호 유효성 검증 테스트 완료
- 회원가입 flow 테스트 완료

리뷰 요청 사항

- 비밀번호 유효성 검증 로직 확인 부탁드립니다
- 에러 처리 방식이 적절한지 봐주세요