



# Git/Jira Convention



## 목차

### [1. Git Convention](#)

#### [1\) branch 전략 - Git Flow](#)

#### [2\) Git Commit Convention](#)

#### [3\) Git 주의사항](#)

### [2. Jira Convention](#)

#### [1\) Epic 단계](#)

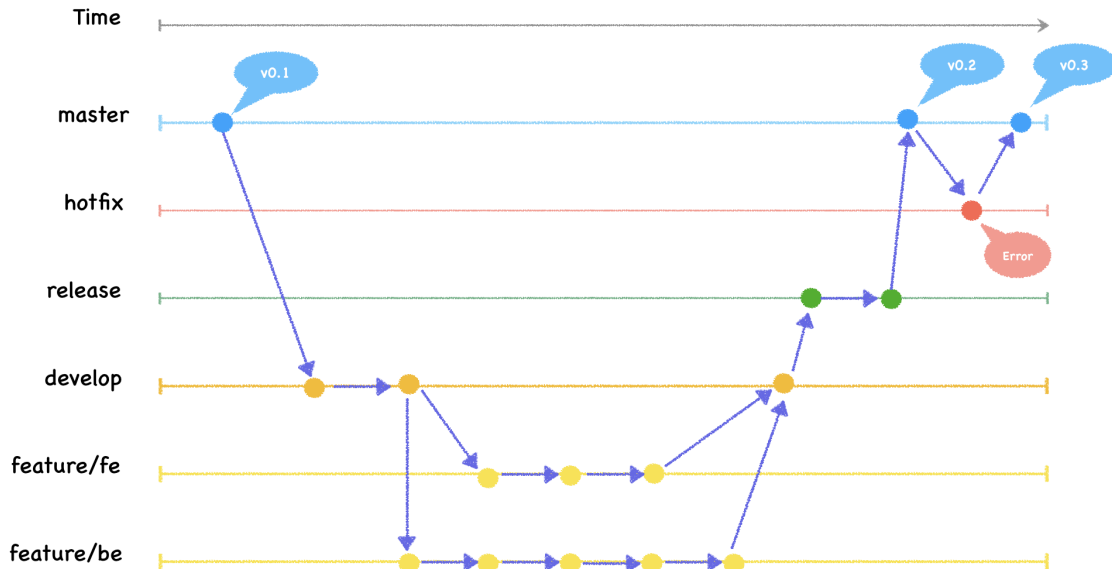
#### [2\) Story Point](#)

---

## 1. Git Convention

### 1) branch 전략 - Git Flow

- master
  - 배포 버전이 표기되는 최상위 브랜치
  - 새로운 feature가 등록하는 것 금지
  - 배포에서 에러가 발견될 경우 hotfix 브랜치로 분기됨
- hotfix
  - 최종 배포된 버전에서 에러가 발생할 경우 에러 해결을 위해 만들어지는 브랜치
  - 새로운 feature가 등록되는 것 금지
- develop
  - master에서 분기되어 feature 개발, 에러 수정 등 배포전까지의 활동이 이루어지는 브랜치
- release
  - develop에서 개발이 완료되면 master에 최종 배포하기 전 테스트 배포를 위해 사용되는 브랜치
  - 테스트가 완료되면 master로 합쳐진다
  - 새로운 기능이 추가되는 것은 안되나 release 브랜치에서 버그 수정은 가능하다.



## 2) Git Commit Convention

다음 양식에 맞추어 커밋메시지 등록해주세요!



[jira issue 번호(#)][포지션-이름] 작업 종류: 작업 내용

- [포지션-이름] ▶ BE-호진 / FE-예지
- 작업 종류: 작업 내용 ▶ docs: 시연시나리오
  - 작업 종류



**feat** : 새로운 기능에 대한 커밋  
**fix** : 파일 수정에 대한 커밋  
**build** : 빌드 관련 파일 수정에 대한 커밋  
**chore** : 그 외 자잘한 수정에 대한 커밋  
**docs** : 문서 수정에 대한 커밋  
**style** : 코드 스타일 혹은 포맷 등에 관한 커밋  
**refactor** : 코드 리팩토링에 대한 커밋  
**test** : 테스트 코드 수정에 대한 커밋  
**ci** : CI(Continuous Integration - 지속적 통합) 관련 설정 수정에 대한 커밋

## 3) Git 주의사항

- Jira와 연동되기 때문에 따로 git issue를 발생시켜 merge를 진행하지는 않습니다.
- git issue는 에러 발견과 같은 다양한 용도로 사용가능합니다.

## 2. Jira Convention

### 1) Epic 단계

- Epic 총 5단계로 구분되어 있습니다.
  - 개발 준비
  - 기획 및 설계
  - 개발
    - 개발은 기능에 따라 Epic이 여러개 생길 수 있습니다.
    - **Epic - 회원관리**
  - 테스트
  - 릴리즈

### 2) Story Point

- 주 40 스토리 포인트가 권장됩니다.
- 업무 강도에 따라 스토리 포인트를 구분합니다.
  - 4: 강
  - 3: 약간의 강도 있음
  - 2: 보통
  - 1: 약함



다음을 클릭하시면 지라 사용법을 정리한 페이지로 이동합니다.

[Jira 사용법](#) 