

Git Branch Requirements

In order to ensure code quality, improve development efficiency, reduce multi-branch conflicts, and be more friendly for QA mates to test, we decide to gradually standardize the process of Git branch development and collaboration.

Branch

1. There are 4 kinds of branches

a. **master/main** PROTECTED

- i. Main branch: Ensure the code in this branch is stable and ready to deliver to client
- ii. Protected Branch, **FORBID PUSH TO THIS BRANCH**

b. **feature/***

- i. Personal development branches. All new features should be developed based on master and checked out to a new feature branch
- ii. Generally speaking, the name of the branch is named according to the content of the requirement.
- iii. This is the development branch. **Strictly forbidden to merge into the Master.**

c. **dev** PROTECTED —> CD GCP

- i. Before testing, you need to raise **PR** from your feature branch to **dev** branch for code review
- ii. After the development of a sprint is complete. Before we merge from dev to master, we need to **rebase master and confirm the diff from master**
- iii. Protected Branch, **FORBID PUSH TO THIS BRANCH**

d. **hotfix/***

- i. The hot fix code for online problems in **dev/master branch**.
- ii. Generally speaking, one Hotfix branch only fixes one problem, and multiple problems open multiple branches.
- iii. Repair branch: Merge into the **dev** branch after the repair is complete.

3. COMMIT Msg Requirement

- a. feature: New Feature
- b. fix: Bug fix
- c. docs: Add documentation
- d. style: Change code style

- e. refactor: Refactoring (that is, it is not a new feature, nor is it a code change to modify a bug)
- f. test: Add test cases

New Feature Procedure

1. Firstly, Check if you are building with the Latest Master Branch.
 - a. `git checkout master && git pull --rebase`
2. Then, create your own feature branch
 - a. `git checkout -b feature/your-feature`
 - b. `git push --set-upstream origin feature/your-feature`
4. After development, commit & push your code to your own feature branch.
 - a. `git checkout feature/your-feature`
 - b. `git commit -m "feature: your changes"`
 - c. `git push`
 - d. Raise PR on bitbucket for code review: FROM: feature/your-feature To: dev
5. Before raising PR in 4.d, rebase Master and then raise PR to dev branch。
 - a. `git checkout feature/your-feature`
 - a. `git rebase master`
 - a. `git push --set-upstream origin feature/your-feature`
6. Tester do tests
7. After Testing & Code Review, Approve PR and merge into Dev
8. After completing all features in a sprint, do code Review and **Merge Master branch**。