git规范

一、 git开发规范

1.master分支作为生产环境对应分支，**禁止任何push操作。**对于在测试服务器测试通过的dev分支代码，**禁止直接合并到master分支**，需要请求分支合并并且指定前端/后端负责人作为审核人，审核通过后则会合并。

2.dev(test)分支作为测试环境对应分支，**禁止直接在dev上进行开发操作。**开发者在开发新功能以及修复BUG的时候，需要**新建一个分支**，在该分支开发完成并且本地测试通过之后，将该分支合并到dev分支，开始在测试服务器上部署进行联调测试。

3. 开发者每次push代码前，必须先pull dev分支代码（每次开发都需要**先**拉取最新代码），并且合并到自己的开发分支中。

二、 git提交规范

Commit message包括三部分。

1. 类型说明

feat：新增功能；
fix：修复bug；
docs：修改文档；
refactor：代码重构，未新增任何功能和修复任何bug；
build：改变构建流程，新增依赖库、工具等（例如maven、webpack修改）；
style：仅仅修改了空格、缩进等，不改变代码逻辑；
perf：改善性能和体现的修改；
chore：非src和test的修改；
test：测试用例的修改；
ci：自动化流程配置修改；
revert：回滚到上一个版本；

2.修改内容所属模块
所属前端、后端 (前后端分离后，此步骤可以略过)
所属具体 业务/技术 模块
多个模块以英文逗号(,)间隔

3.内容描述
对本次commit的详细描述，可分多行

————————————————————

例子，修改登录功能BUG
git commit -m "fix(前端-登录模块,**模块):修复点击登录的弹出框不显示二维码"