

GitLab Pull Request 工作流

Step 1: 创建gitlab项目

注意：如果是Group项目，也就是多人配合的项目，建议Namespace使用Group name，Visibility默认为Internal, **必须填写项目描述信息**。

Step 2: 源仓库的构建

源仓库建立后，我们需要注意这时这个项目是属于整个Group的，每个开发同事需要fork一份源代码到自己的GitLab账号中，作为自己开发使用的仓库。点击右上角的fork就可以发现代码成功fork到了自己的名下：

这时你就可以在自己的项目下随便折腾了。

Step 3: 在自己的项目上进行开发

git clone , git add , git commit , git push , 如果需要的话在自己的项目上建立branch，一切都由自己的习惯决定。

Step 4: 向管理员提交pull request

假设我完成了某项功能，需要向线上版本进行提交，首先进行git fetch --all，拉取下来upstream，也就是线上版本的代码，并且将线上的代码合并到自己的分支上，解决冲突。

再次行测试，觉得没问题后就可以push到GitLab中了。接下来请求管理员把自己仓库的分支合并到原仓库的分支下，这就是pull request。

进入merge request，并且点击New Merge Request，将自己的分支提交到主代码的分支下。

提交一个Merge Request，请记得**写清楚提交的理由，分配需要为你去做Review的同事**。

此外，还可以在评论框中去@其他的同事，也可以在Commit和Change里看到最新的改变。

Step 5: Code Review

被assign或者@的同事都会收到邮件要求review，，点击进入就可以进行Code Review了。

在这里，各位可以针对某一行提出自己的意见，也可以在评论里发表意见。如果没有问题，可以输入lgm(looks good to me)，如果everybody都认为没问题，就accepted，这时也就提交到了主代码上。

如果代码被评论过后，评论者也会收到消息，修改后push的代码会自动提交到同一个merge request里。

大家注意要养成多次少量commit的习惯，不要让每个commit承担太多的内容。1是避免给code review同事造成太大压力，2是避免断电死机等意外情况造成的代码丢失。

6. star

可以star自己关注的项目，这样就可以在自己的消息流中收到动态，看到所关注项目的最新进展。