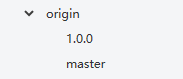
Git流程操作演示

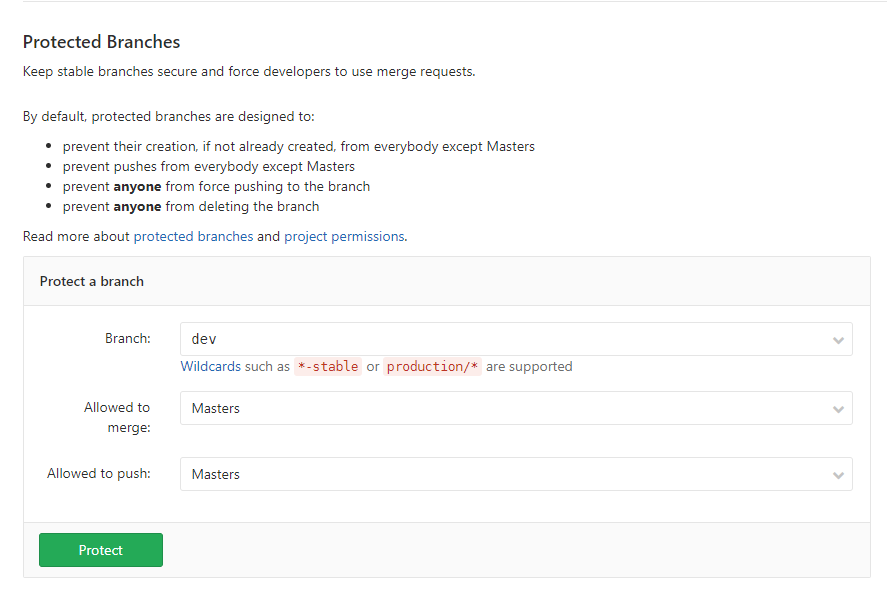
## 项目初始阶段

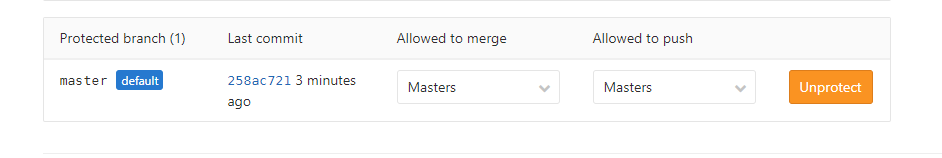
1. 项目起始阶段(刚成立)：

项目master创建master分支并导入项目初始代码，然后基于master分支拉取迭代分支分支(分支名称即为本次迭代版本号)



创建完成后将master分支设为Protected Branches。



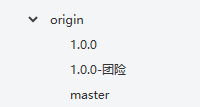


## 项目迭代开发阶段(dit/sit)

1. 项目迭代开始：

项目master由迭代分支拉取开发分支，分支命名规范：ReleaseVersion-功能名称，例如：1.0.0-团险，如果功能并行，拉取多条开发分支并按照命名规范命名。

Ps：如果同一个迭代，不同功能代码间不会有相互干扰，建议在同一个分支上提交，不建议一个功能就拉一个分支。



1. 项目开发到一定阶段后，项目master决定该部分的代码可以发布DIT/SIT环境，由项目master在当前开发分支上打一个tag，开发分支上的tag命名规范为： ReleaseVersion/D(N)，N为feature上的tag编号，从0开始，依次增加，例如1.0.0/D0

Tag打好在jira的对应issue上更新tag号字段，点击发布到DIT环境或者提交申请发布SIT环境。

## 项目发布阶段（UAT）

1. 项目测试阶段测试完成后，准备发uat环境，由项目master先将迭代分支代码合并到开发分支，处理冲突并提交到开发分支，打tag发测试环境进行回归测试，没有问题后合并到迭代分支。合并完成后在迭代分支打tag（命名规范为ReleaseVersion/U(X)，例如1.0.0/U0）。Tag打好后更新jira对应issue的Tag号字段，提交申请发布UAT。

## 项目上线阶段

1. uat环境测试通过之后，从迭代分支最后一次tag打包发到prd环境，同时由项目master将代码合并到master分支(打tag，命名规范：m\_ReleaseVersion,例如：m\_1.0.0)。

## 线上hotfix阶段

1. 如果线上出现紧急hotfix，由项目master基于master分支拉取hotfix分支，hotfix分支命名规范：hotfix/ReleaseVersion，hotfix的版本号一般是在线上版本号末尾加1。例如：hotfix/1.0.1。

hotfix分支创建完成后，在hotfix分支上做修改，修改完成后，先发本地hotfix环境，测试没有问题，根据tag发到prd，并将代码合并到master分支(打tag)，合并完成后，当前hotfix结束，hotfix分支删除 ,master分支tag命名规范：m\_ReleaseVersion,例如：m\_1.0.1。