

Git Rebase

省流版本

```
1 git init
2 git remote add origin https://github.com/Republic1024/TestRepo.git
3
4 git pull origin main
5
6 rm .DS_Store # 例如移除未跟踪的文件
7 git add .
8 git commit -m "Initial commit or merge or rebase remote changes"
9
10 git fetch origin
11 git rebase origin/main # 或者如果更喜欢, 使用 git rebase origin/main
12 git push origin main
```

1. 初始化Git仓库并设置远程地址:

```
1 git init
2 git remote add origin https://github.com/Republic1024/TestRepo.git
```

2. 从远程仓库拉取更新:

```
1 git pull origin main
```

3. 移除未跟踪的文件（如果有必要）并提交:

```
1 rm .DS_Store # 例如移除未跟踪的文件
2 git add .
3 git commit -m "Initial commit or merge remote changes"
```

4. 推送本地更改到远程仓库:

```
1 git push origin main
```

5. 处理分支分歧（如果遇到）:

```
1 git fetch origin
2 git rebase origin/main # 或者如果更喜欢, 使用 git rebase origin/main
3 git push origin main
```

这个步骤假设每个步骤都顺利执行:

- **初始化和设置:** 在您的目录中初始化Git, 并设置远程仓库。
- **拉取更新:** 从远程仓库获取并集成更新到您的本地分支（`main`）。
- **提交更改:** 在解决冲突或进行必要调整后, 提交您的更改。

- **推送更改：**最后，将提交的更改推送回远程仓库。
- **处理分支分歧：**如果遇到分歧的分支，请获取最新的更改，根据需要进行合并或变基，然后再次推送您的更改。

这个总结涵盖了一个典型的工作流程，确保您的本地仓库与远程仓库保持同步，同时有效地处理任何冲突或分歧历史。