

GitHub 私有仓库多人协作教程（含 SSH、分支保护、PR 工作流）

一、每个人都要做的事：配置 SSH 密钥

## 1. 生成 SSH 密钥

- 打开终端（Windows 推荐 Git Bash），输入：  
`ssh-keygen -t rsa -b 4096 -C "你的GitHub邮箱"`
- 一路回车即可。会生成 `id_rsa`（私钥，不能泄露）和 `id_rsa.pub`（公钥，用来上传）。

## 2. 添加公钥到 GitHub

- 打开 `~/.ssh/id_rsa.pub` 文件，复制全部内容。
- 登录 GitHub → 右上角头像 → Settings → SSH and GPG keys → New SSH key
- Title 随便填，Key 区域粘贴公钥内容 → Add SSH key

## 3. 测试 SSH 连接

```
ssh -T git@github.com
```

- 成功会显示：Hi xxx! You've successfully authenticated...

二、仓库管理员操作：添加协作者

- 打开你的仓库页面（如：<https://github.com/chengsq/-helloWorld>）
- 进入 Settings → Collaborators 或 Manage access
- Add people，输入队友 GitHub 用户名，发送邀请
- 队友接受邀请后获得权限

三、分支保护设置（推荐）

- 仓库页面 → Settings → Branches → Branch protection rules
- 新建一条规则，保护 main 分支，勾选：
  - Require a pull request before merging
  - Require approvals（可选）
  - Include administrators（推荐）
- 保存

四、协作流程（PR 工作流）

- 克隆仓库（用 SSH 地址）

```
git clone git@github.com:chengsq/-helloWorld.git
cd -helloWorld
```

- 基于 main 创建新分支开发

```
git checkout main
git pull origin main # 保证本地最新
git checkout -b feature-xxx # xxx为本次开发的功能名
```

- 编码、提交

```
git add .
```

```
git commit -m "描述你的修改"
```

#### 4. 推送你的分支

```
git push origin feature-xxx
```

#### 5. 在 GitHub 网页发起 Pull Request

- 打开仓库页面 → “Pull requests” → “New pull request”
- base 选 main, compare 选 feature-xxx
- 填写说明, 创建 PR

#### 6. 代码 Review 和合并

- 有分支保护时, 不能直接 push 到 main, 必须等 PR 审核通过、无冲突后, 点击 “Merge” 合并到 main

### 五、遇到 “历史无关” 无法 PR 的特殊情况

如提示 main 和分支 “entirely different commit histories”, 说明两分支没共同祖先。

这时只能本地合并:

```
git fetch origin feature-xxx
```

```
git checkout main
```

```
git merge feature-xxx --allow-unrelated-histories
```

```
# 解决冲突 → git add . → git commit
```

```
git push origin main
```

合并后大家都基于最新 main 分支拉新分支, 就不会再遇到这个问题。

### 六、重要建议

- 不要把 id\_rsa (私钥) 给任何人, 只上传 id\_rsa.pub (公钥)
- 每个人必须有自己的 SSH key, 并添加到各自的 GitHub 账号
- 日常开发都从 main 拉分支, 提交 PR, 主分支永远保持干净