开源社区项目开发简明使用手册

(版本 2.7)

目录

目录.....II

1 TRUSTIE 平台概览.....2

2 “项目”板块.....3

3 “个人中心”板块.....4

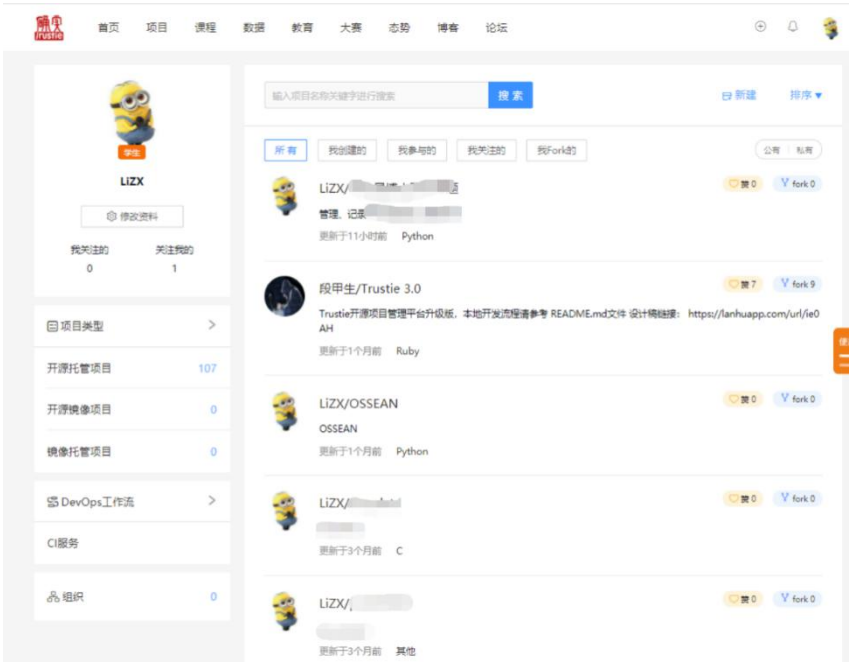
4 项目开发.....6

4.1 功能概述.....6
4.2 项目管理.....7
4.3 开发任务管理.....11
4.4 版本管理.....15
4.6 常见错误.....21

1 开源社区项目协同开发平台概览

（1）“项目”板块：主要汇聚和展示平台上所有公开的开源项目，方便用户查找感兴趣的开源项目，参与开源开发。

（2）“个人中心”板块：主要汇聚和展示用户个人创建、管理或者参与的项目，让用户快速进入个人的项目工作台开展项目开发工作。



2 “项目”板块

“项目”模块汇聚和管理了所有 Trustie 平台上的托管项目和镜像项目，用户可以输入项目名称关键字进行搜索，也可以根据项目类别对项目进行筛选。

进入“项目”模块，左侧列出了项目类型和项目类别。其中，项目类型主要包括开源托管项目和开源镜像项目两类。项目类别主要包括：大数据、机器学习、人工智能、智慧医疗、其他。

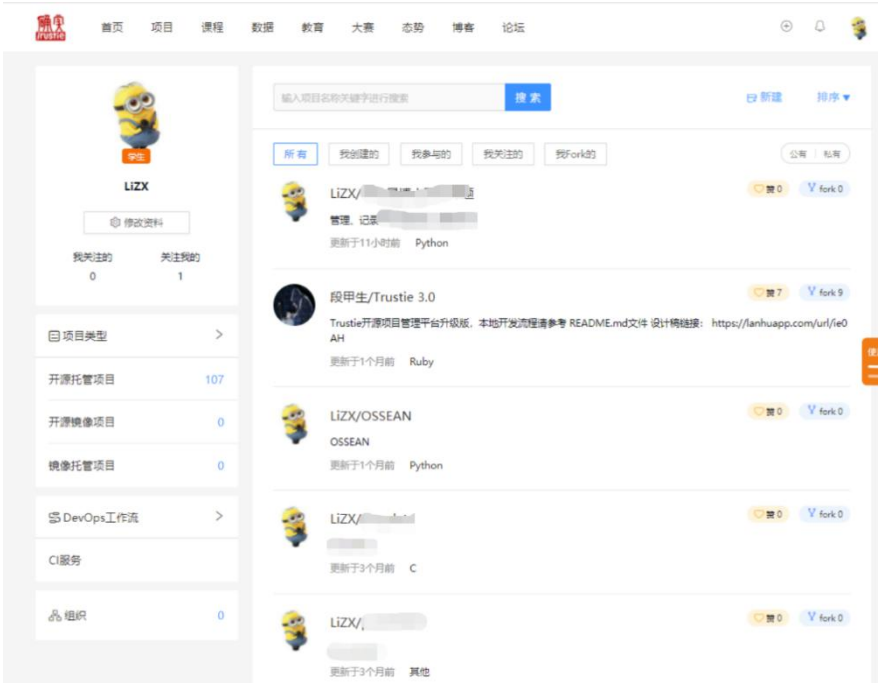
右侧展示了所有项目的基本信息，包括创建者、项目名、项目简介、浏览量、项目类别、更新时间、点赞数量、Fork 数量等信息，用户可以通过关键字搜索查找特定的项目，也可以按照更新时间、创建时间、Fork 数量、点赞数量等对项目进行排序。

用户点击项目名称，即可进入到项目详情，查看和参与开源项目开发。

此外，平台提供了“新建”按钮，用户可以通过点击快速从零开始创建新的公开或者私有项目。

3 “个人中心”板块

“个人中心”项目板块汇聚和管理了所有跟已登录用户相关的项目，包括自己创建的、参与的、关注的和复刻的项目等。未登录用户访问时，只能看到该用户下的公开项目。



进入“个人中心”板块 （鼠标移动到用户头像会出现下拉菜单），左侧列出了与用户相关项目的类型，主要包括开源托管项目、开源镜像项目、镜像托管项目三类。具体差别如下：

(1) 开源托管项目是在开源社区 ~~Trustie~~ 平台上创建和开发的项目；

(2) 开源镜像项目是从其他社区如 GitHub、Gitee 中镜像到 Trustie 平台的项目，此类镜像项目只能查看以及同源项目进行同步，但是不能进行代码修改和提交；

(3) 镜像托管项目是从其他社区如 GitHub、Gitee 中镜像到 Trustie 平台的项目，此类镜像项目可以同源项目进行同步，也可在 Trustie 平台上进行代码修改和提交。

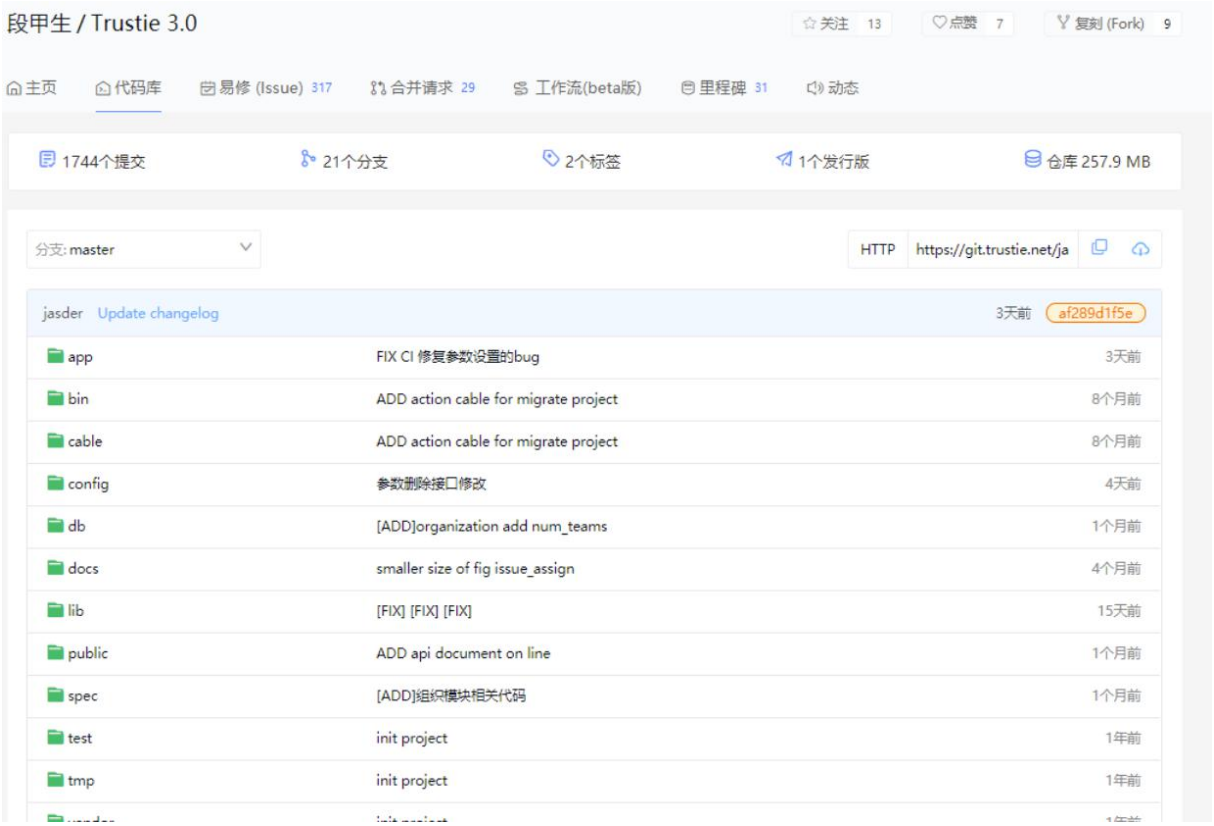
右侧展示了与用户相关的所有项目，用户可以通过关键字搜索查找特定的项目，也可以按照更新时间、创建时间、Fork 数量、点赞数量等对项目进行排序。同时，平台提供了“所有”、“我创建的”、“我参与的”、“我关注的”和“我 Fork 的”分组，用户选择相应的分组即可查看该分组下的所有项目。在分组筛选的基础上，用户可以进一步筛

选“公有”和“私有”两类进行区分查看。

4 项目开发

4.1 功能概述

进入一个具体的软件项目，平台提供了代码版本管理、协同开发过程管理等服务，支持开展大规模群体协同。项目详情界面如下图所示：



其中，各部分的核心功能和服务如下：

- 主　　页：提供对项目主要信息的展示，如背景、特性等。
- 代 码 库：提供基于 Git 的代码版本管理服务；
- 易修（Issue）：提供协同开发任务管理服务，包括任务发布、指派、评论等；
- 合并请求：提供代码合并请求的管理、审阅、合并等服务；
- 工 作 流：提供 DevOps 工作流服务
- 里 程 碑：提供项目版本发布规划、任务规划管理等服务；
- 动　　态：提供项目最新动态的展示和跟踪等服务；
- 仓库设置：提供项目管理设置服务，仅对项目管理者可见；
- 关　　注：关注项目即可将该项目相关动态发送给关注者；
- 点　　赞：对认可的项目点赞支持；

- 复刻（Fork）：完整复制项目并生成一个属于自己的项目副本。

具体的，针对项目开发核心操作和处理流程如 4.2 和 4.3 介绍。

4.2 项目管理

每一个项目可以看做是用户创建的对应于一个小组或团队的协同实践社区，由项目管理员、开发者和报告者组成，提供任务跟踪、版本管理、里程碑、资源托管、交流研讨等功能，支持团队和小组开展各种协作活动。

4.2.1 新建项目

团队（或小组）的核心成员首先创建一个项目，可以选择新建镜像项目或新建托管项目。镜像项目是指基于 GitHub 等其他社区开源项目在 Trustie 平台创建项目镜像；托管项目是指从零开始基于 Trustie 平台创建和管理的项目。

点击顶部导航栏的“+”符号可以进行项目新建操作。

（1）新建镜像项目时可以对项目进行配置的内容包括：镜像版本库地址、项目名称、项目介绍、仓库名称、项目类别、项目语言等信息。如下图：

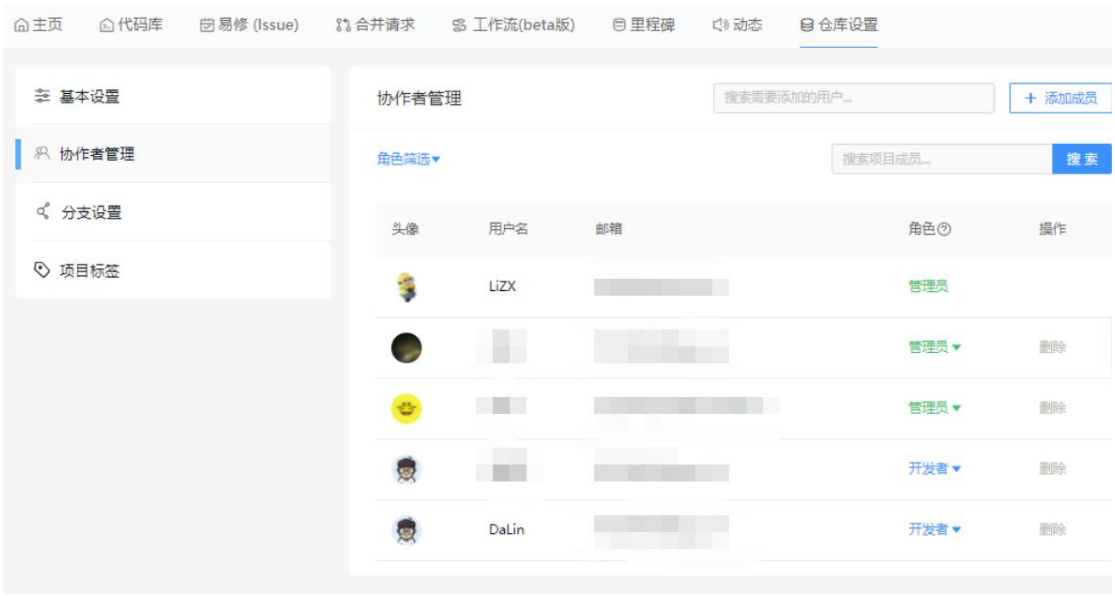
（2）新建托管项目时可以对项目进行配置的内容包括：拥有者、项目名称、项目介绍、仓库名称、项目类别、项目语言等信息。如下图：

项目创建完毕后，项目管理者可以通过“仓库设置”中的“基本设置”修改项目基本信息。另外“仓库设置”中还包含“项目标签”的设置，项目管理者可以自定义标签名字、描述和颜色信息，如下图：

4.2.2 协作者管理

项目创建成功后，项目管理员可以在“仓库设置”中管理项目的成员。

进入需要管理的项目主页，点击“仓库设置”按钮，再点击“协作者管理”标签页，将进入下图所示界面：



项目管理员在搜索框搜索需要添加的用户，选中后点击“添加成员”，然后为其选择右下角的角色（管理员、开发者、报告者），即可将用户添加至协作者。

注意：项目成员的不同角色具有不同权限。

- ✓ 管理员：仓库设置功能，代码库读、写操作；
- ✓ 开发者：代码库读、写操作；
- ✓ 报告者：代码库读操作。

4.2.3 删除项目

管理员可以删除项目。步骤如下：

（1）进入“仓库设置”界面；

（2）点击“基本设置”标签页下方的“删除本仓库”按钮，即可删除当前项目。

4.2.4 项目主页

管理员可以编辑项目的主页信息。步骤如下：

（1）点击并进入“主页”页面；

（2）点击右上角的“编辑”按钮。

（3）填写想要展示的项目信息，然后点击底部的“确定”按钮。

4.2.5 项目简介

项目代码库中的 README.md 默认展示在代码库页面，通常可在 README.md 文件放置关于项目简介的内容。项目成员可以通过在项目代码库中添加和编辑“README.md”添加和修改项目简介。步骤如下：

（1）进入“代码库”界面；

（2）点击并进入“README.md”

（3）点击右上角的编辑按钮，即可编辑当前项目简介。

4.2.6 上传和新建文件

项目成员可以在项目的代码库界面中点击“上传文件”、“新建文件”按钮，完成基于浏览器的在线文件上传和新建，如下图：

注意：

- ✓ 上传文件目前仅支持纯文件格式，不支持图片、excel 等无法以纯文本格式读取的文件；
- ✓ 文件名请使用英文；

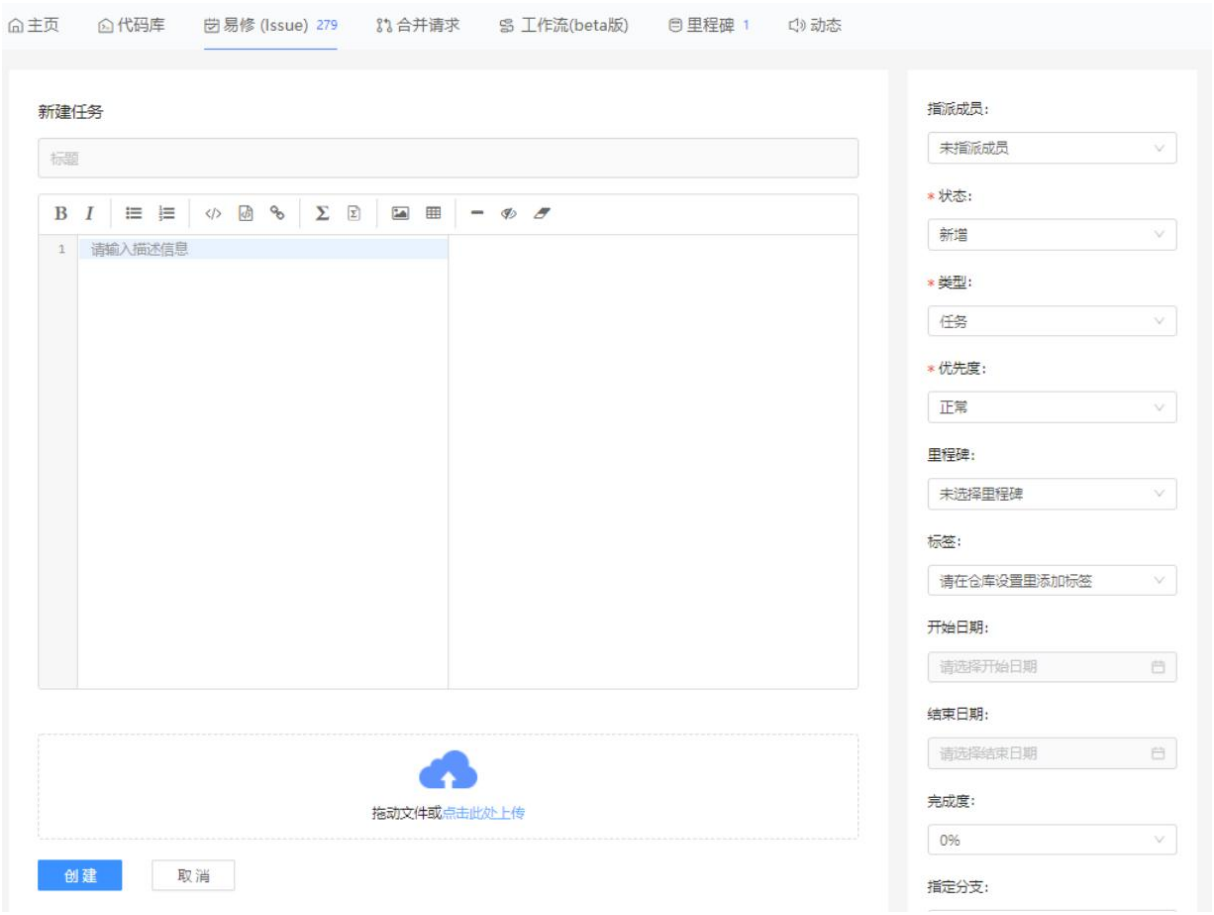
- ✓ 文件大小不得超过 5MB；
- ✓ 文件路径和文件描述必填。

4.3 开发任务管理

4.3.1 发布易修（Issue）

易修（Issue）管理模块主要为项目组成员提供任务发布、指派、跟踪等功能服务。

进入需要发布任务的项目，点击上方的“易修（Issue）”按钮，然后点击“创建任务”即可进入任务发布界面。如下图：

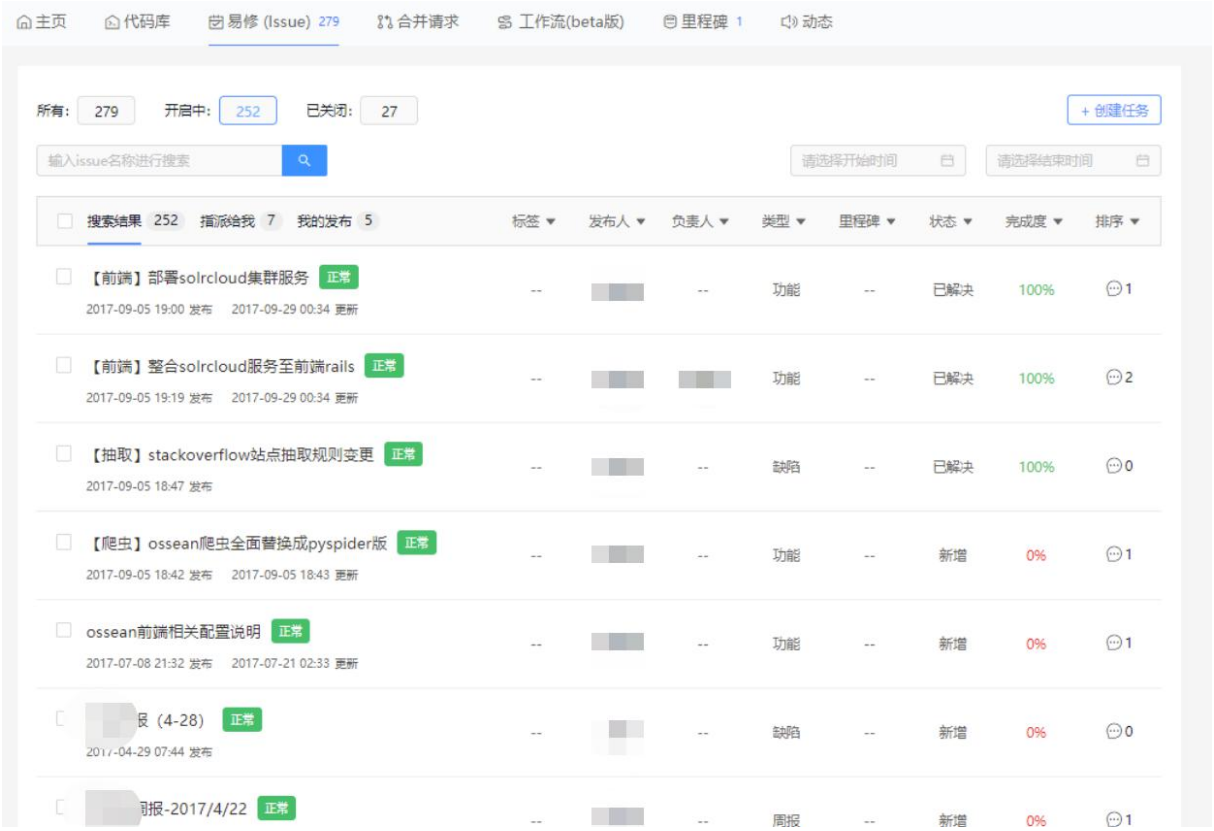


注意：

- ✓ 每个开发任务都相当于一个可以进度追踪的帖子，因此支持回复和评论等；
- ✓ 开发任务的类型包括：缺陷、功能、任务、支持、周报；
- ✓ 在问题截止日期的前一天，系统将对开发的发布人和被指派人发送提醒消息。

4.3.2 查询任务

进入需要查询任务的项目社区，点击上方的“易修（Issue）”按钮，即可进入任务列表，用户可以输入任务名称进行搜索，也可以根据开始、结束时间对任务列表进行筛选。任务列表展示了所有开发任务的基础信息，点击具体任务即跳转到该任务界面。如下图：



4.3.3 新建里程碑

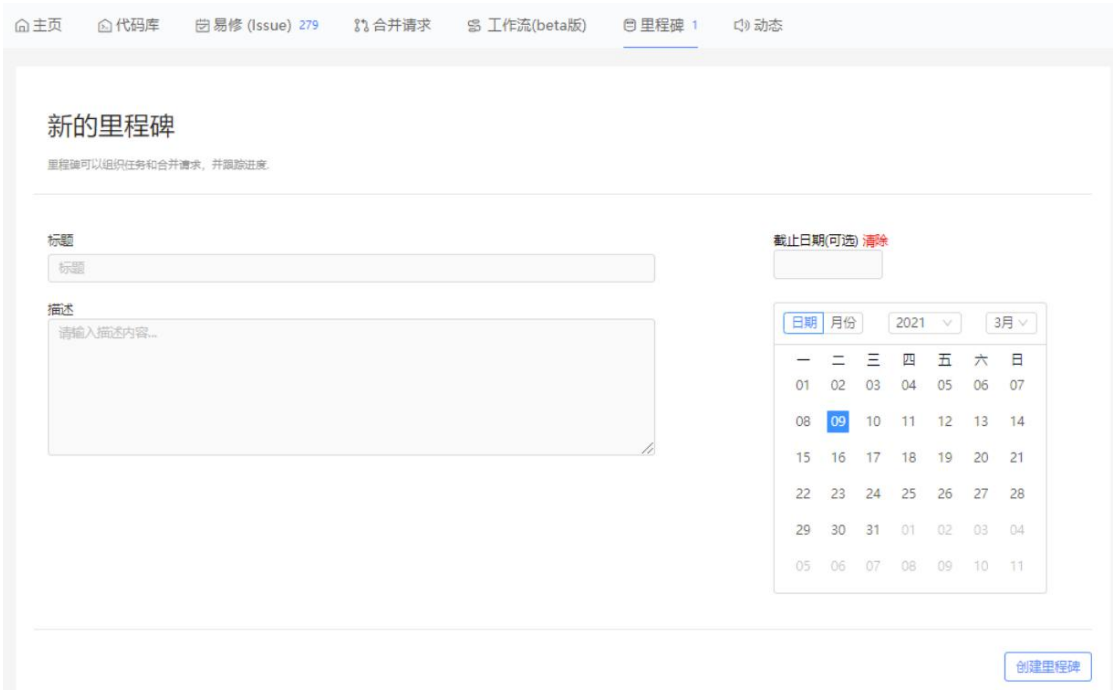
里程碑主要用于项目组对项目开发和版本发布提供支持，每一个里程碑可以关联多个开发任务。

管理员可以新建里程碑。步骤如下：

1、进入“里程碑”界面；

2、点击“新的里程碑”按钮，即可新建一个里程碑；

3、设置新建里程碑的截止日期等属性。



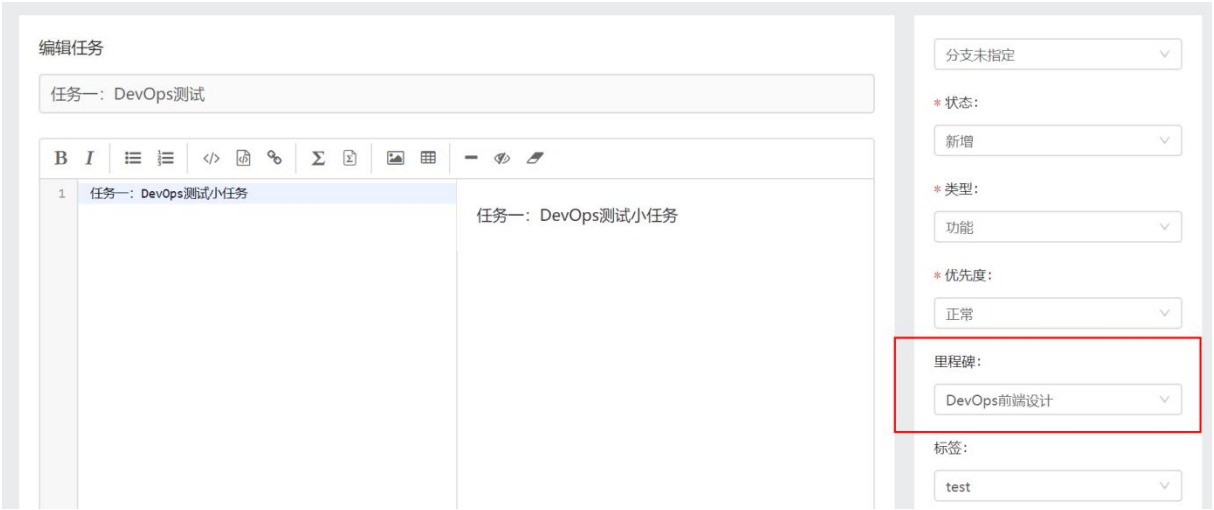
4.3.4 关联里程碑

项目成员可以将项目任务关联到里程碑，从而使里程碑包含明确的任务列表。主要步骤如下：

1、进入某个任务的页面，点击“编辑”，编辑其属性；

2、从“里程碑”下拉框中选择需要关联的里程碑；

3、点击“提交”。

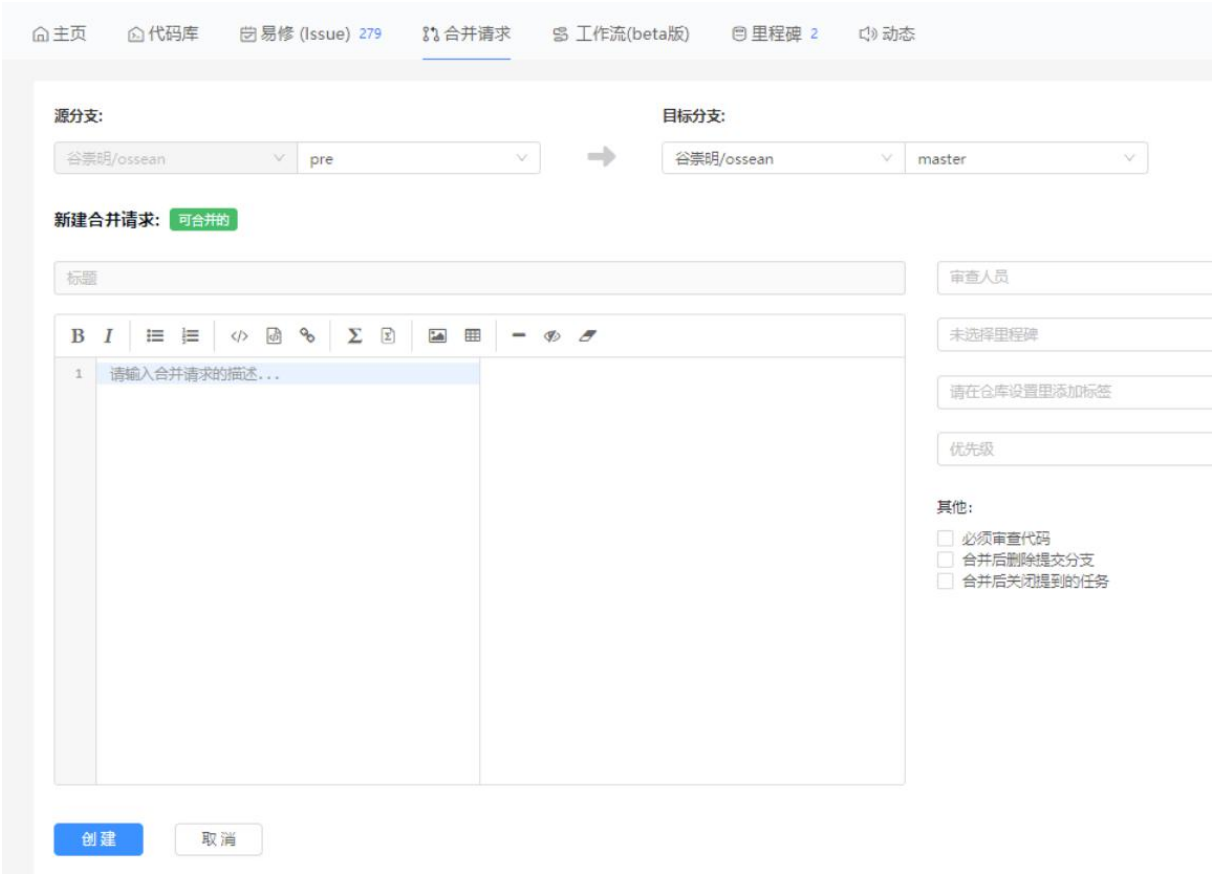


4.3.5 合并请求（Pull-Request）

合并请求板块提供合并请求创建和管理两方面的功能。一方面支持向源项目或者同

一个项目其他分支创建（发起）代码合并请求；另一方面也为项目管理者对他人发送到本项目的合并请求进行管理、审阅并确定是否纳入项目某个分支。

创建合并请求：进入需要发起合并请求的项目，点击上方的“合并请求”按钮，然后点击“新建合并请求”即可进入合并请求创建界面。如下图所示。其中，源分支即为已完成代码开发，准备发起合并请求的代码分支；目标分支即为要进行代码合并的分支。选中分支后，用户需要填写相关信息，包括标题和详细的描述。此外，用户还能够在右侧边栏中指定审查人员、关联里程碑等操作。信息填写完成后，用户点击底部的“创建”按钮即可。



管理合并请求：如果有人针对本项目发起了合并请求，则在“合并请求”模块可以看到相应的合并请求。项目管理员可以查看、审阅和决定如何处理该合并请求，包括打回重新修改、拒绝或者直接合并等。

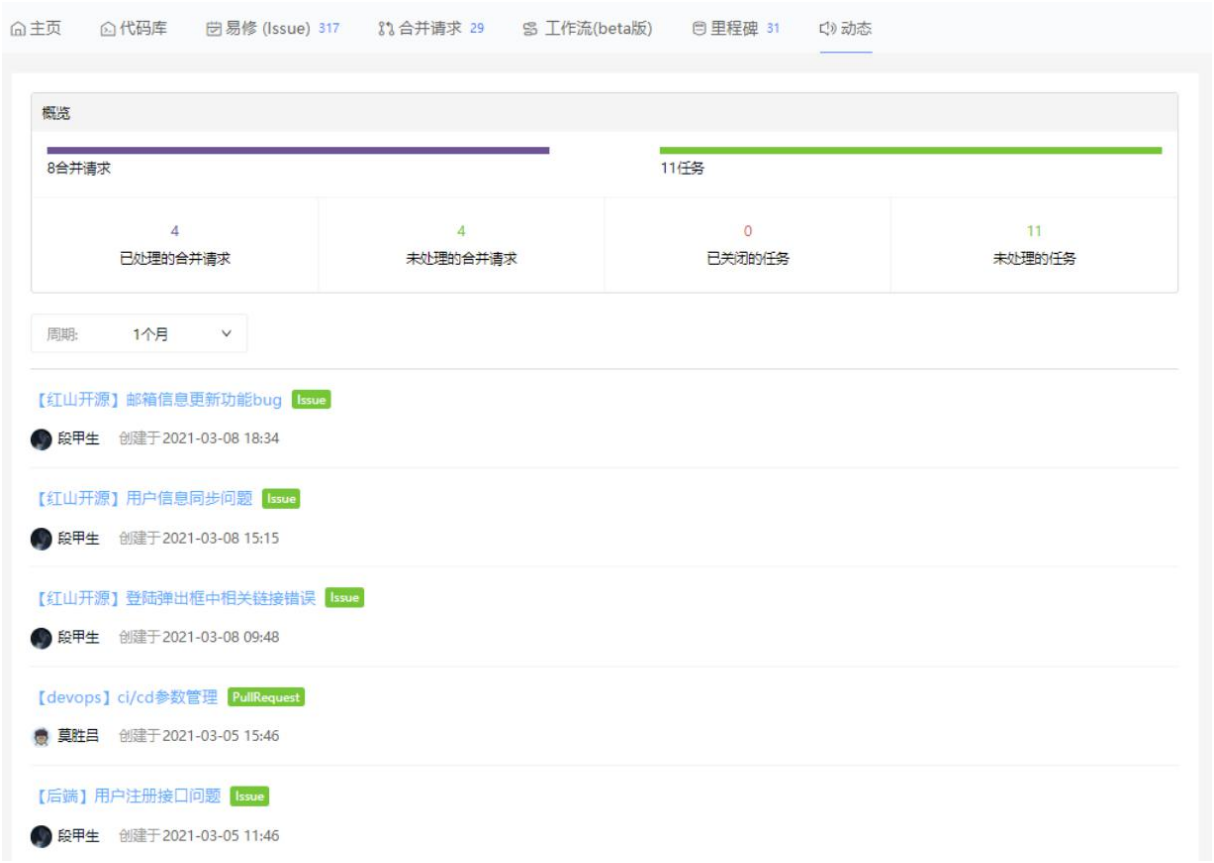
注意：

- ✓ 源分支和目标分支不能相同；
- ✓ 选择审查人员后，该审查人员将收到合并请求的审阅通知；

✓ 合并请求通过审阅后，审查人员点击“合并请求”按钮完成代码合并，该合并请求的状态将变为“已合并”。

4.3.6 开发动态

点击项目社区上方的“动态”按钮，即可进入开发动态展示界面。用户可以方便地查看当前合并请求和开发任务的概览，也可以选择周期查询具体的开发动态。如下图：



4.4 版本管理

版本管理又称版本控制，是一种分布式协同编程的重要工具。没有掌握好版本管理工具，团队式协同开发是不可想象的。

Trustie 项目社区集成了 Git 版本管理系统，可以提供标准的 Git 版本库操作流程。因此，在使用本平台提供的 Git 仓库时，可以直接参考本手册，也可以在互联网中搜索相关问题的解决方法。Trustie 版本管理基本操作流程如下：

（1）在平台上新建远程版本库或 Fork 一个远程版本库；

（2）将远程库 clone 到本地（可选）；

（3）在本地库中创建分支（缺省为 master）；

（4）提交（commit）代码到本地分支；

（5）推送（push）代码到远程库的分支 / 下拉（pull）远程库的代码到本地；

（6）解决冲突；

（7）合并分支。

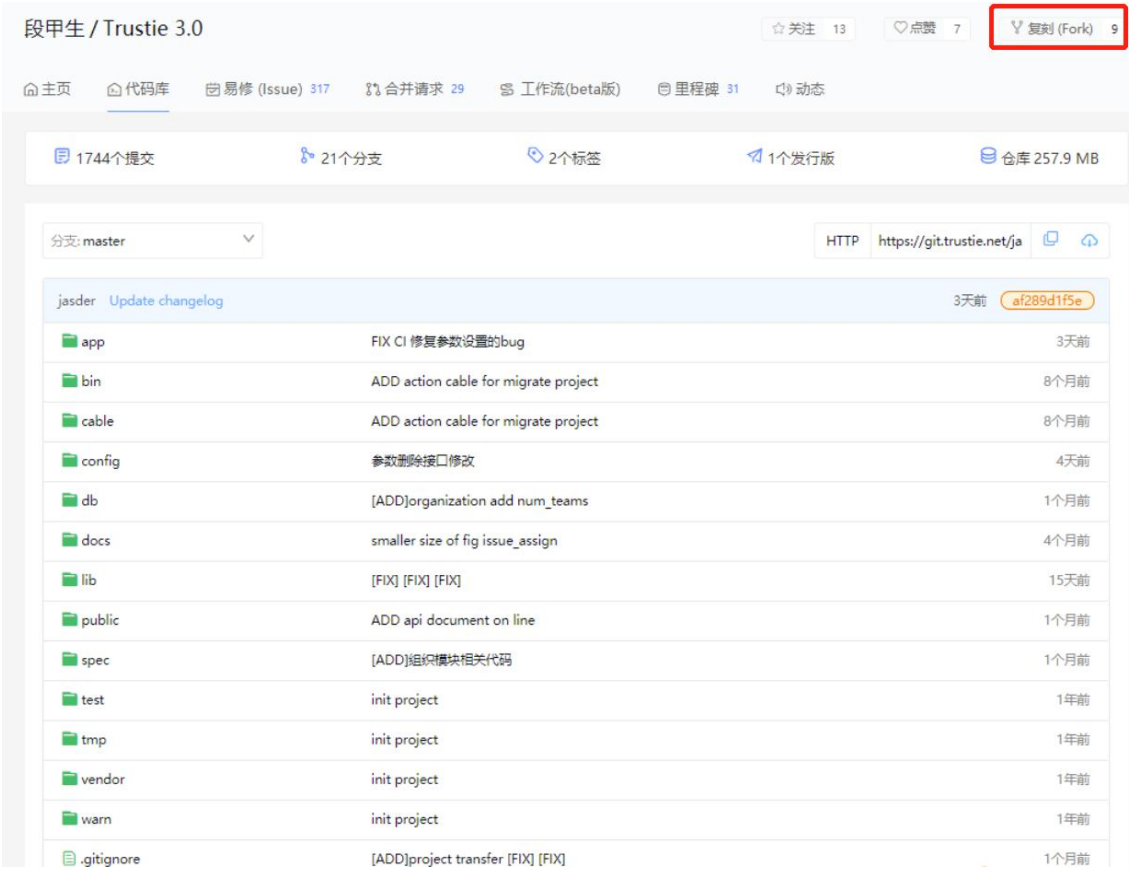
以上是一个在实际操作中反复进行的基本流程，具体操作处理如下介绍。

注意：

- ✓ 在多人协同编程的情况下，每个成员在开发自己的模块时，尽可能先从 master 或 developer 分支下拉最新代码，并与本地分支合并。否则时间久了，合并分支将是一个灾难。
- ✓ 上述基本流程可以使用 Git 命令行客户端、Tortoise Git 客户端等工具完成。

4.4.1 Fork 版本库

用户对某一个版本库的 Fork 操作将会为该用户创建一个属于该用户的同名项目和版本库，用户可以在该项目下进行开发，并通过向源项目发起合并请求来提交代码贡献。操作步骤：进入需要 Fork 的版本库页面，点击右上角的“复刻（Fork）”按钮即可。



注意：

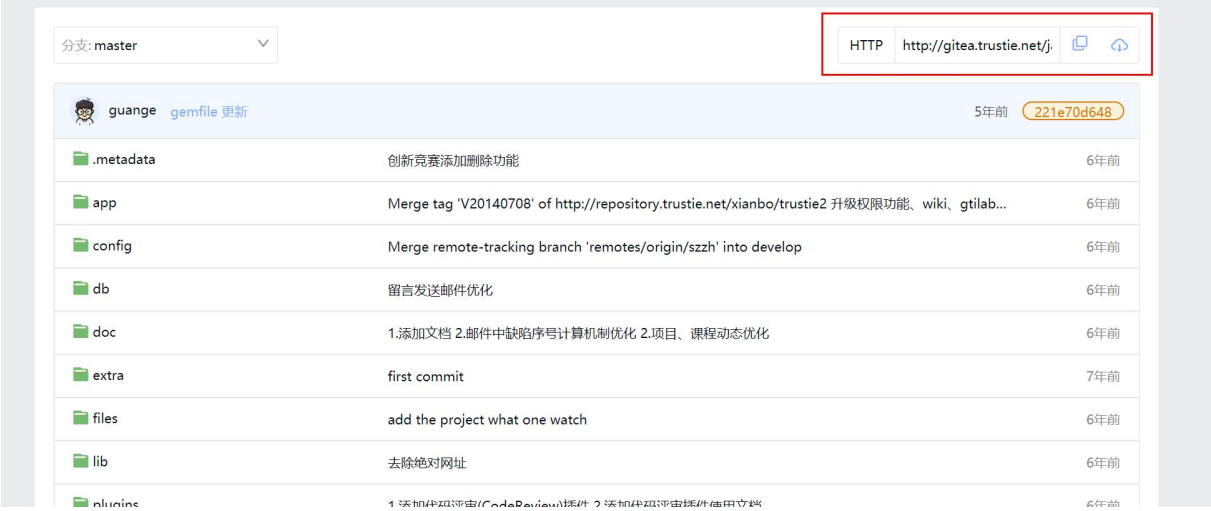
✓ 任何用户都可以 Fork 一个公开的版本库；

✓ 连续两次 Fork 同一版本时，系统将直接跳转到第一次 Fork 后的版本库。

4.4.2 克隆版本库（clone）

克隆版本库是将项目社区的版本库完整地克隆到本地的过程，其中包括了版本库的所有提交记录。步骤如下：

（1）确定需要克隆的版本库地址：从项目的版本库页面中即可找到该地址；



（2）在本地的 git 命令行执行：

git clone https://bdgit.educoder.net/jacknudt/demo.git

其中“https://bdgit.educoder.net/jacknudt/demo.git”是版本库的地址。

（3）如果项目是私有的，根据提示输入版本库用户名和密码，输入当前用户的 Trustie 平台登录名和密码即可。

注意：

✓ 如果项目公开，无需输入口令，可以直接克隆；

✓ 对于私有项目，只有项目成员才具有克隆的权限。

4.4.3 新建分支、切换分支

（1）新建分支

新建分支是指在本地库创建一个新的分支（用于开发新的功能），并且将其同步到远程库。主要步骤如下：

a) 从已有的分支创建新的分支(如从 master 分支)，例如创建一个 dev 分支：

```
git checkout -b dev
```

b）创建后可以查看一下当前分支，确认当前分支已经切换到 dev：

```
git branch
    * dev
    master
```

c) 提交该新的分支到远程仓库：

```
git push origin dev
```

把本地的 dev 分支直接推送到远程仓库(如果是第一次 push 该分支时，执行 push 操作会自动在远程仓库创建一个对应的 dev 分支。

（2）切换分支

切换分支是指将当前工作分支换为另一个分支，即切换到另一个分支进行开发。

假设，本地仓库有 master 和 dev 两个分支，当前分支是 master（其实也可以是其他分支），那么切换到 dev 分支的命令是：

```
git checkout dev
```

注意：用户可以使用 git branch 命令确认当前分支是否已经切换到 dev。

4.4.4 提交代码（commit）

项目成员在本地编辑代码或修改文件后，可以将新修改的文件提交到本地库。

（1）将新修改的所有文件添加到本地版本库的暂存区（stage）：

```
git add .
```

注意：如果仅添加单个文件，或一类问题件，可以：

```
git add somefile.txt        # 添加单个文件
git add *.txt               # 添加所有 txt 文件
```

（2）将提交到暂存区的修改提交到本地的当前分支：

```
git commit -m “my first commit”
```

其中，“my first commit”中的内容是关于本次提交的说明文字。

注意：commit 只能将修改的文件提交到本地库，与远程库没有关系。

4.4.5 推送代码（push）

项目成员可以将本地提交后的改动，推动到远程库。

假设用户需要将本地库的 dev 分支推送至远程库的 dev 分支，执行：

```
git push -u origin dev
```

如果用户的当前的本地分支已经是 dev，那么执行：

```
git push
```

如果用户需要将本地的全部分支推送到远程库，那么执行：

```
git push --all origin
```

注意：

- ✓ push 代码时会提示输入用户名和密码，请输入 Trustie 平台登录名和密码即可。
- ✓ push 代码之前通常要先 pull，解决冲突后才能提交，否则不是最新的文件就提交不上去。

4.4.6 下拉代码（pull）

下拉代码是指从远程版本库获取代码文件到本地分支。例如，用户希望将远程版本库的 dev1 分支下拉到本地的 dev 分支，需要执行：

```
git pull origin dev1:dev
```

注意：下拉代码时可能会出现冲突（即两个分支对同一个文件的同一行代码做了不同的修改）。此时需要利用 merge 来解决冲突。

4.4.7 合并分支（merge）

合并分支是指将一个分支合并到当前分支。通常有两种情况。

第一种情况：将一个本地分支合并到当前分支：

例如，假设用户希望将本地库的 dev1 分支合并到当前分支 dev，命令如下：

```
git merge dev1
```

第二种情况：将一个远程库的分支合并到当前分支：

例如，假设用户希望将远程库的 dev1 分支合并到当前分支 dev，有多种方法，这里仅给出最基本的：

```
git checkout -b temp          # 新建一个临时分支 temp
git pull origin dev1:temp     # 将远程 dev1 分支下拉到 temp 分支
git checkout dev              # 切换到本地 dev 分支
git merge temp                # 合并，可能会产生冲突
```

合并完成后，应删除 temp 分支。

注意：合并分支可能会出现冲突（即两个分支对同一个文件的同一行代码做了不同的修改）。

4.4.8 解决冲突

冲突是两次不同的提交修改了同一行代码导致的。通常表现为两个分支对同一行代码做了不同的修改。

版本冲突主要出现在以下两种情况：

第一种情况：在对远程版本库进行 pull 操作时

第二种情况：对本地版本库的不同分支进行 merge 操作时

解决冲突的主要方法有两种：

1、对于简单的冲突，直接修改有冲突的文件，然后重新提交即可。

2、对于复杂的冲突，需要用 git merge tool 来解决。

复杂的冲突解决方法（基于 git merge tool）将在近期提供给大家。

4.5 常见错误

新手在使用 Git 时经常遇到各种错误，平台集成的是标准 Git 服务，绝大部分问题大家都可以在互联网搜索疑难问题的解决办法，亦可通过 Trustie 问题反馈群进行反馈和讨论。

为给大家提供方便，这里先列出一种最常见的错误：

✓ 错误 1：fatal: The remote end hung up unexpectedly

Windows 环境解决方法，在.git/config 文件中加入：

[http]

postBuffer = 524288000

Linux 环境解决方法，在命令行执行：

git config http.postBuffer 524288000

注：后续我们将列出更多的常见错误解决办法。

5 Git 常用命令参考

https://www.runoob.com/git/git-basic-operations.html