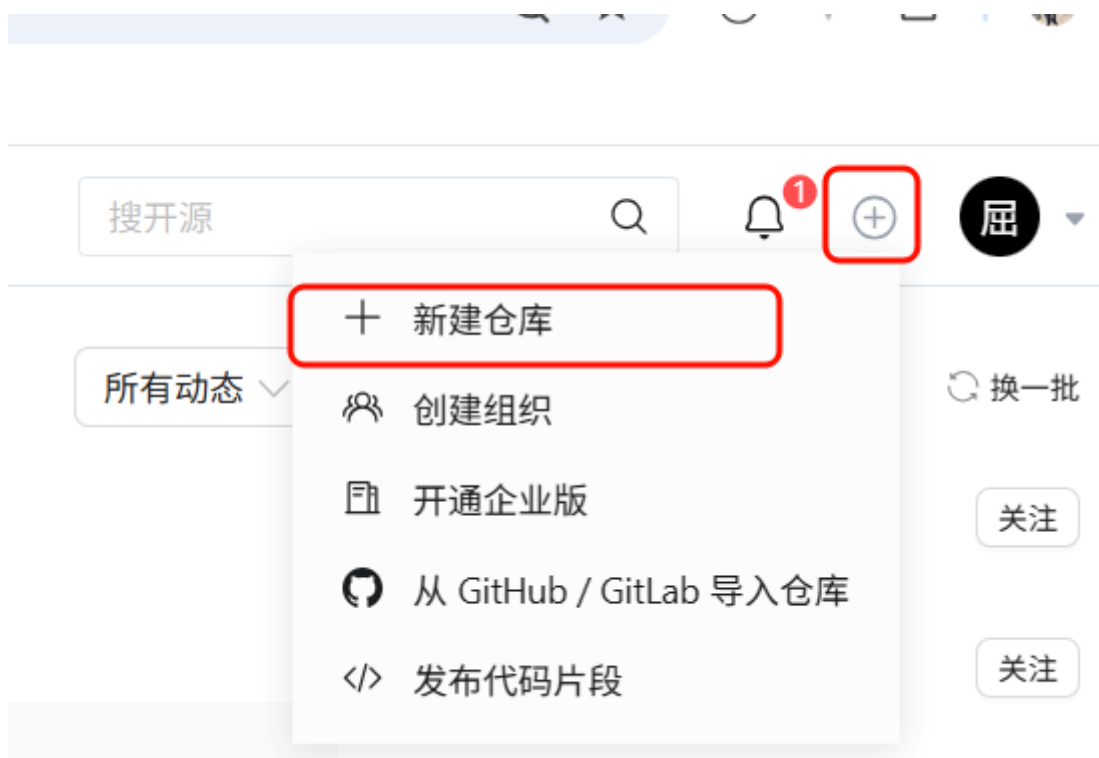


# 1、分组、选择小组实现的项目

## 2、在gitee上创建项目的基本的仓库



创建项目仓库的时候进行初始化，这时远程仓库本质是一个空仓库，将来将前端和后代的代码都可以提交到当前的仓库中。

- ☒ 开源 (所有人可见) ②
- ☐ 私有 (仅仓库成员可见)

☒ 初始化仓库 (设置语言、.gitignore、开源许可证)

选择语言

Java

添加 .gitignore

Actionscript

添加开源许可证

MIT

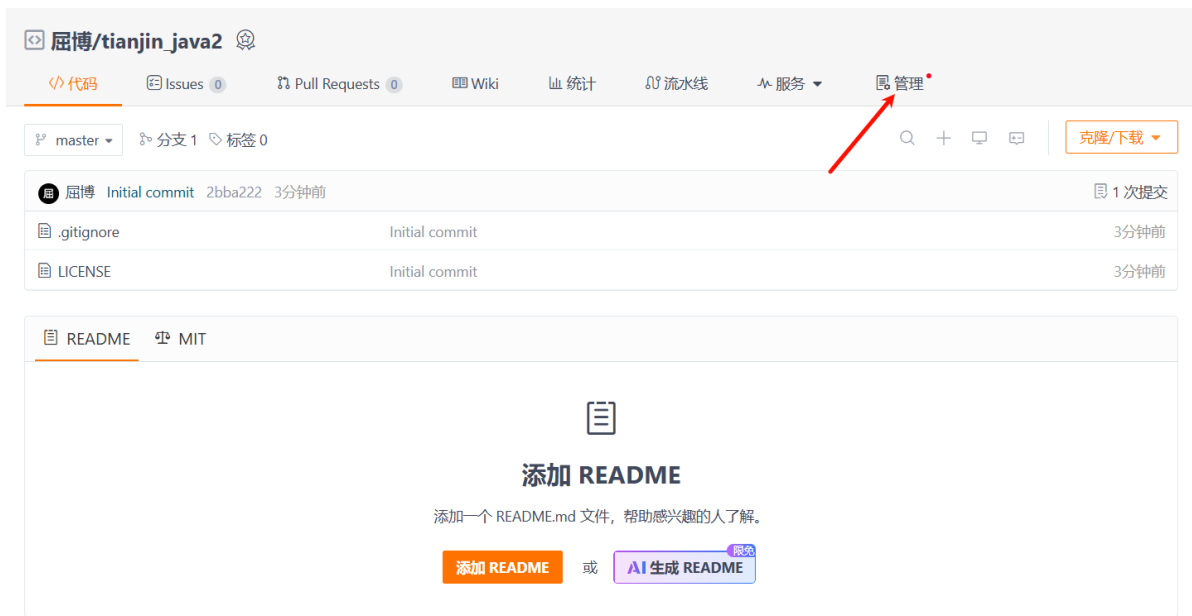
☐ 设置模板 (添加 README、Issue、Pull Request 模板文件)

☒ 选择分支模型 (仓库创建后将根据所选模型创建分支)

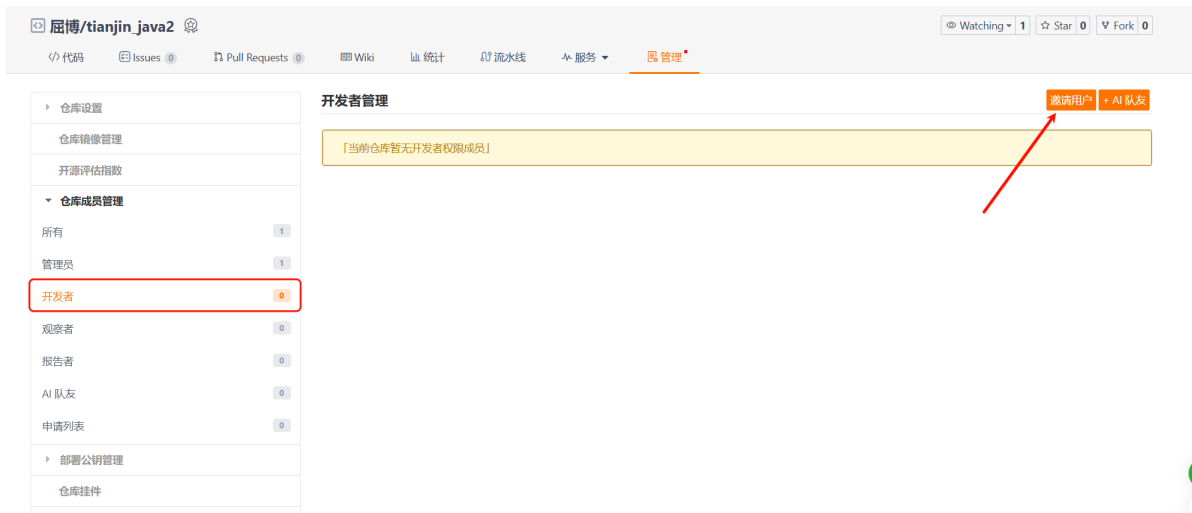
单分支模型 (只创建 master 分支)

创建

将本组的其他人的账号添加到项目中



选择管理员或者开发者，添加账号



当有组员通过链接等方式加入仓库的时候，需要仓库的创建者（管理员）进行审核



组长将仓库的地址还给本组其他人员：[https://gitee.com/qubome\\_admin/tianjin\\_java2.git](https://gitee.com/qubome_admin/tianjin_java2.git)

```
qubo@QuBo MINGW64 /e/东软/太原理工/24-软件工程行业案例分析与实现/代码
$ git clone https://gitee.com/qubome_admin/tianjin_java2.git
Cloning into 'tianjin_java2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.

qubo@QuBo MINGW64 /e/东软/太原理工/24-软件工程行业案例分析与实现/代码
$
```

```
MINGW64: /e/东软/太原理工/24-软件工程行业案例分析与实现/代码/tianjin_java2
qubo@QuBo MINGW64 /e/东软/太原理工/24-软件工程行业案例分析与实现/代码
$ git clone https://gitee.com/qubome_admin/tianjin_java2.git
Cloning into 'tianjin_java2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (4/4), done.
qubo@QuBo MINGW64 /e/东软/太原理工/24-软件工程行业案例分析与实现/代码
$ cd tianjin_java2/
qubo@QuBo MINGW64 /e/东软/太原理工/24-软件工程行业案例分析与实现/代码/tianjin_java2 (master)
$ -
```

克隆完成之后，默认窗口在仓库的外面，若需要git仓库，必须进入仓库中

进入仓库之后，可以看到当前所在仓库哪个分支上

## 3、需求分析

根据选择的项目，借助AI，进行需求分析、功能设计、功能总图、原型设计（草图、低保真、高保真）、

## 4、数据库设计

根据需求分析、功能总图、原型设计、分析数据库表（表之间的关系、表中拥有的字段）表的SQL语句、ER图（总图、每张表ER图）

建库：数据库中创建数据库仓库，然后再基于这个仓库，创建用户（非root账号），给创建用户分配指定库的相关权限

## 5、项目开发

分组分工完成项目前后端开发

## 6、测试环节

白盒测试（单元测试）、接口测试、功能（界面）测试、性能（压力）测试