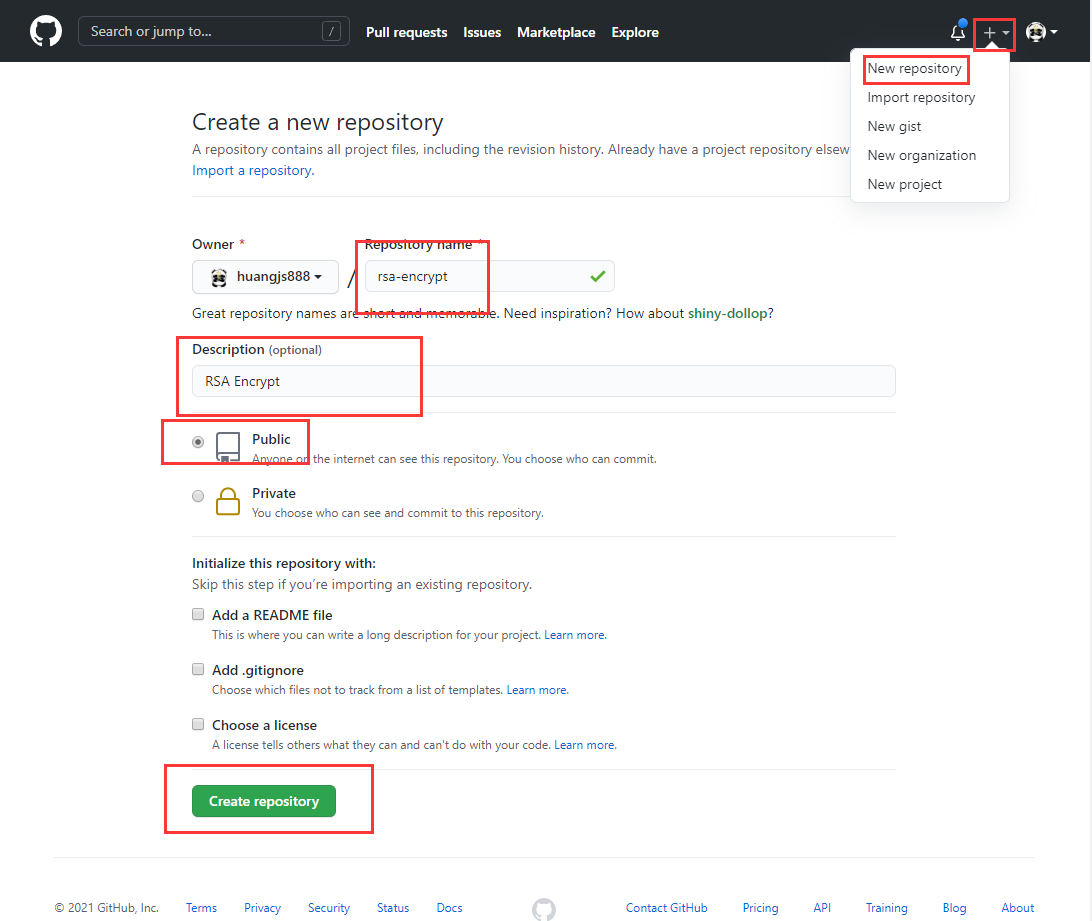
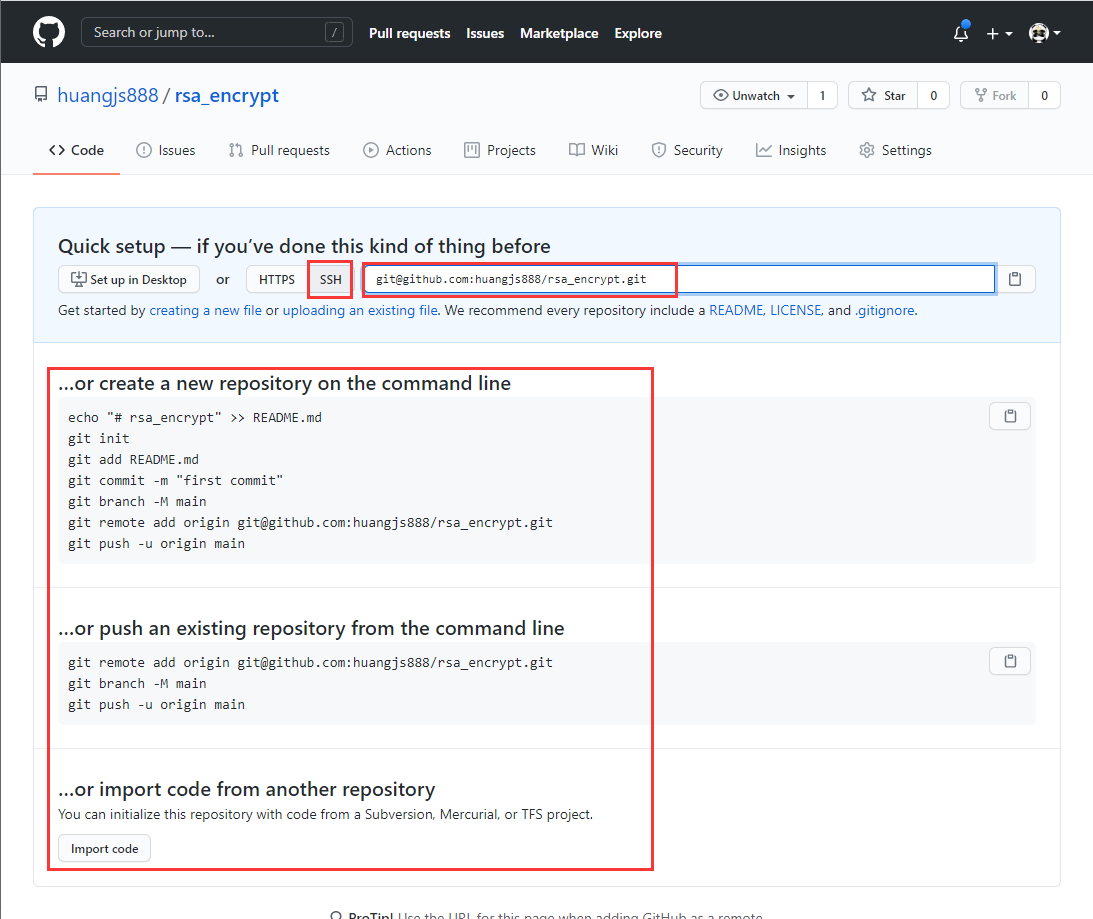
**1，在github上的创建一个仓库**

1，登录github账号进入主页后，依次按下图，从上到下，最后创建（不要勾选readme，.gitgnore等文件，否则仓库会自己创建分支及代码，我们这里创建一个完全空的仓库，后面使用已有项目进行上传，直接使用git命令即可）。



2，这个是完全空的仓库，下面有几种方式提示如何上传代码，我们拷贝对应的https或者ssh的地址（具体请查阅“GitLab(GitHub)使用Https和SSH方式管理代码”文档），然后使用git命令进行上传代码。



3，根据上面图片可以知道三种上传代码方式：1，还未被代码库管理的初始代码库上传。2，已经使用git管理的项目代码上传。3，其他代码库管理的项目。

第一种，我们可以首先安装git及git相关的代码管理工具并且进行一些列配置之后，然后在项目的根目录右键选择git bash打开命令行：

git init

git add \*

git commit –m ‘Init code’

git remote add origin git@github.com:huangjs888/rsa\_encrypt.git

git push -u origin master

第二种，代码库已经被git管理，表示项目已经被git init过，在项目的根目录右键选择git bash打开命令行，首先确认所有代码已经被commit过，然后git config –list 查阅下配置的name和email，如果和自己的github账号邮箱一致可忽略，如果不一致，则为了使上传的代码是自己的github用户邮箱这里需要使用git config命令修改下name和email（主要是因为可能存在多个git仓库账户问题，比如自己的和公司的账户不一致，system，global，local具体查阅git 使用）:

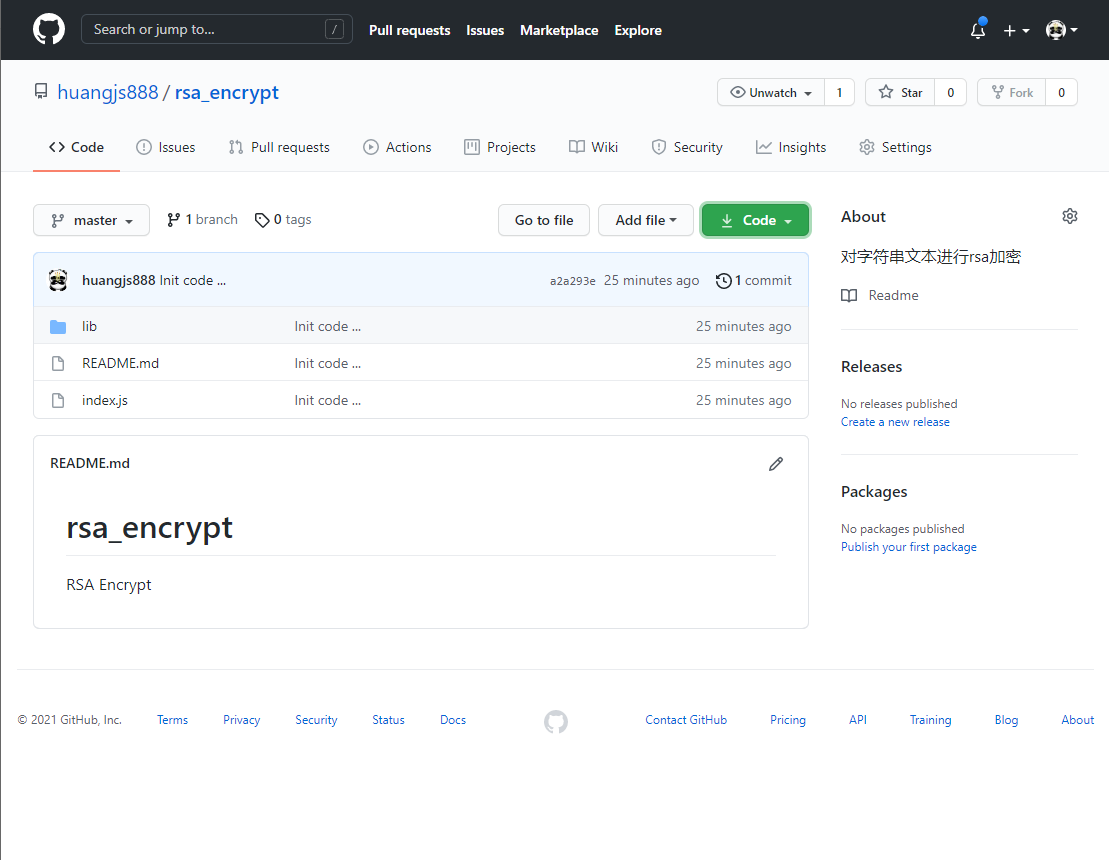
git remote add origin git@github.com:huangjs888/rsa\_encrypt.git

git push -u origin master

第三种，我可以选择导入项目，具体查阅github文档，还可以绕道而行，就是将代码脱离原有的代码库管理，然后按照第一个方法执行，不过会丢失提交历史。

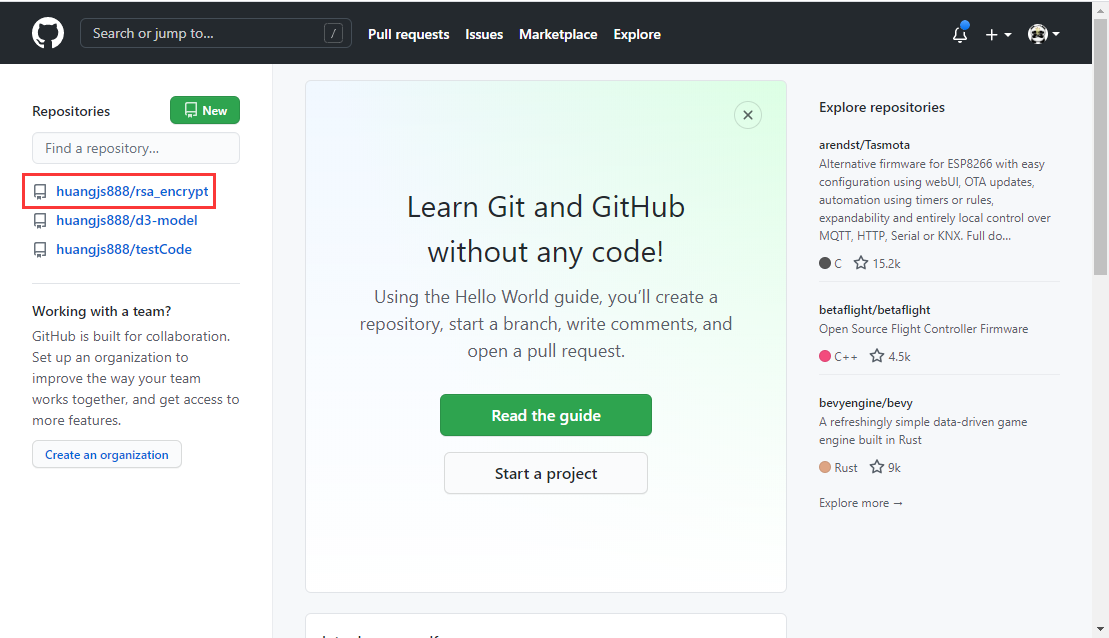
说明：-u 将当前master分支追踪远程master分支

5，进入github并查看仓库内容，发现已经上传上去，并且可以随时联网情况下使用git bash对该项目的修改进行提交push等，如果有其他人修改代码提交上去，可以pull合并，具体使用和git代码库一样。

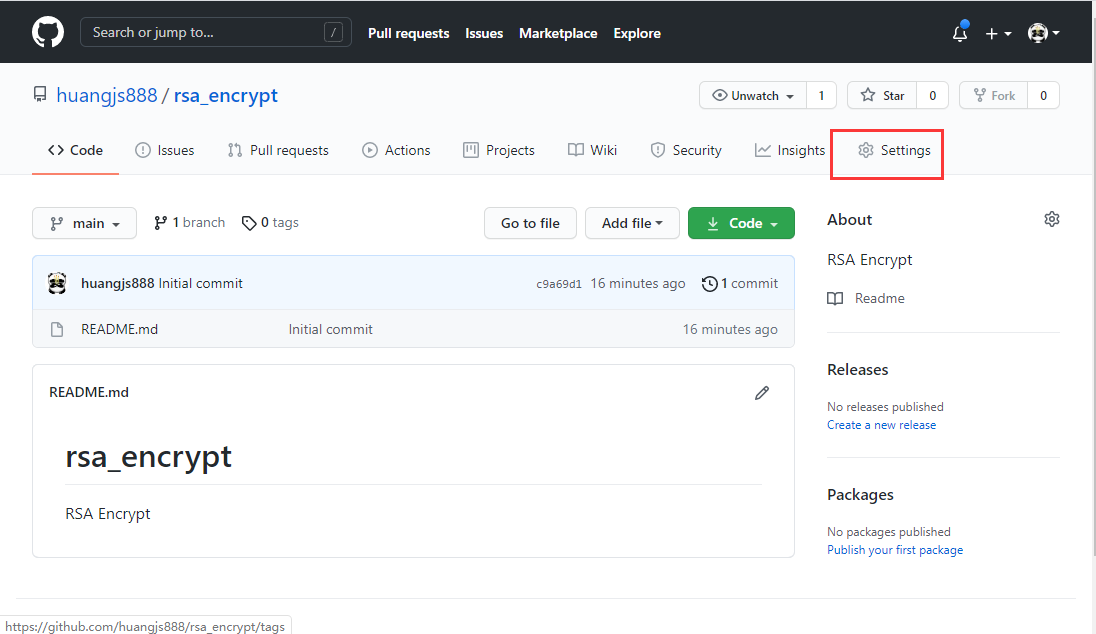


**2，删除github上的某一个仓库**

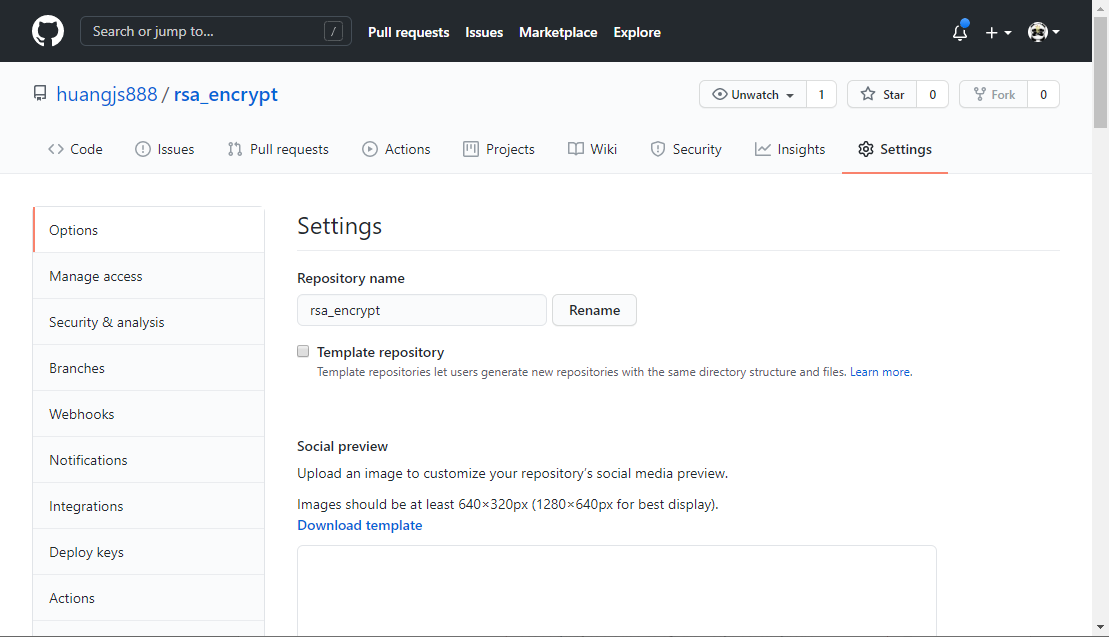
1，登录github账号进入主页后，点击要删除的仓库。

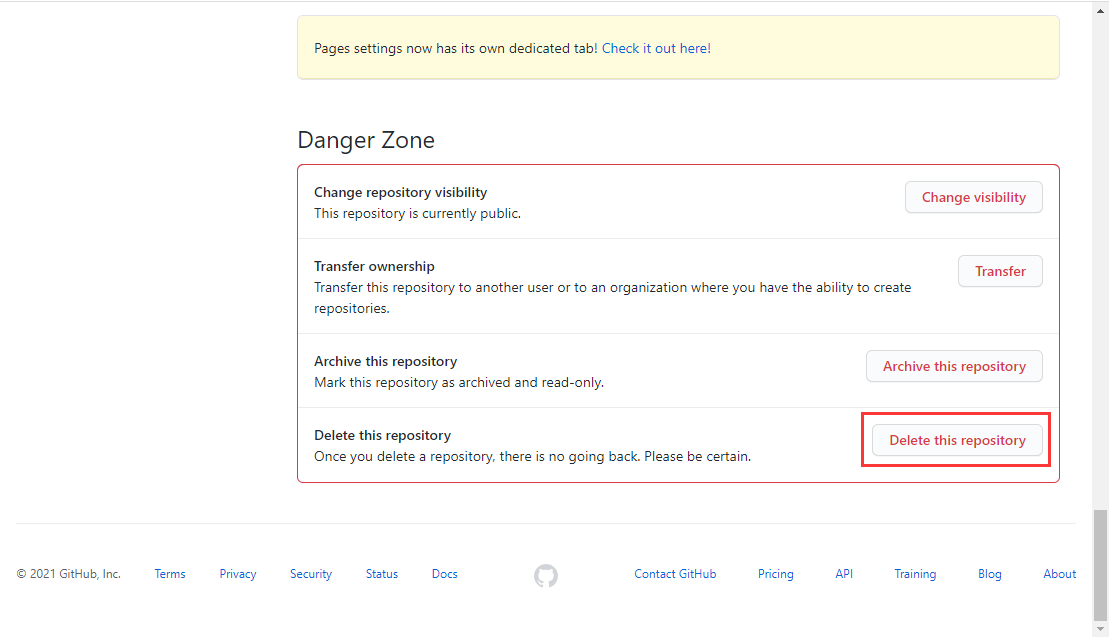


2，进入仓库后，点击“Setting”按钮。

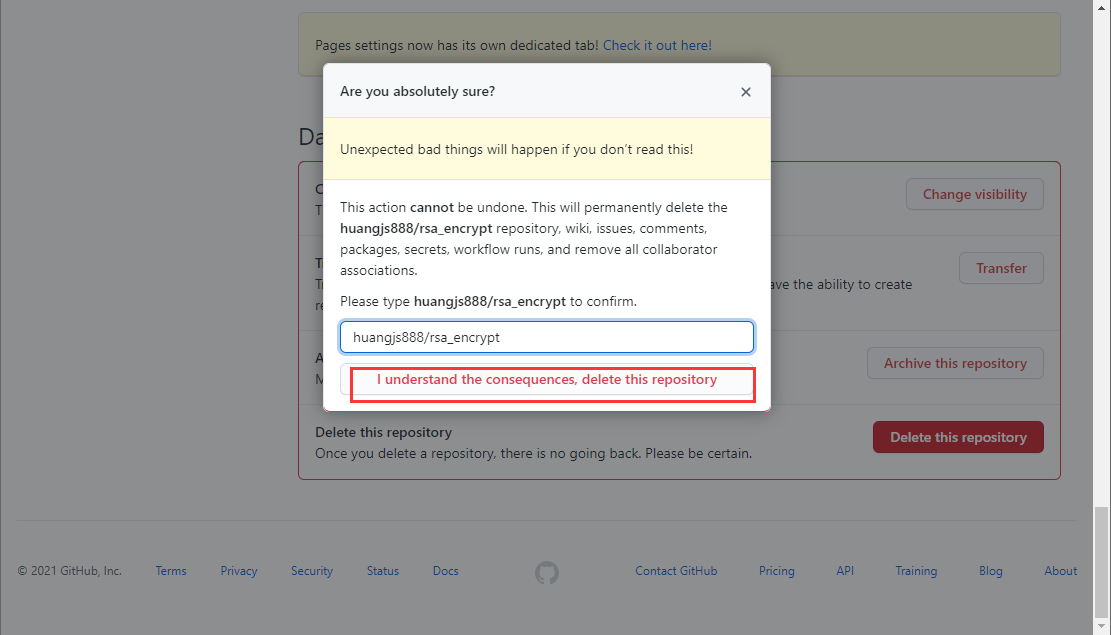


3，进入设置后滚动到页面最底部，找到“Danger Zone”即危险区域，点击“Delete this repository”按钮

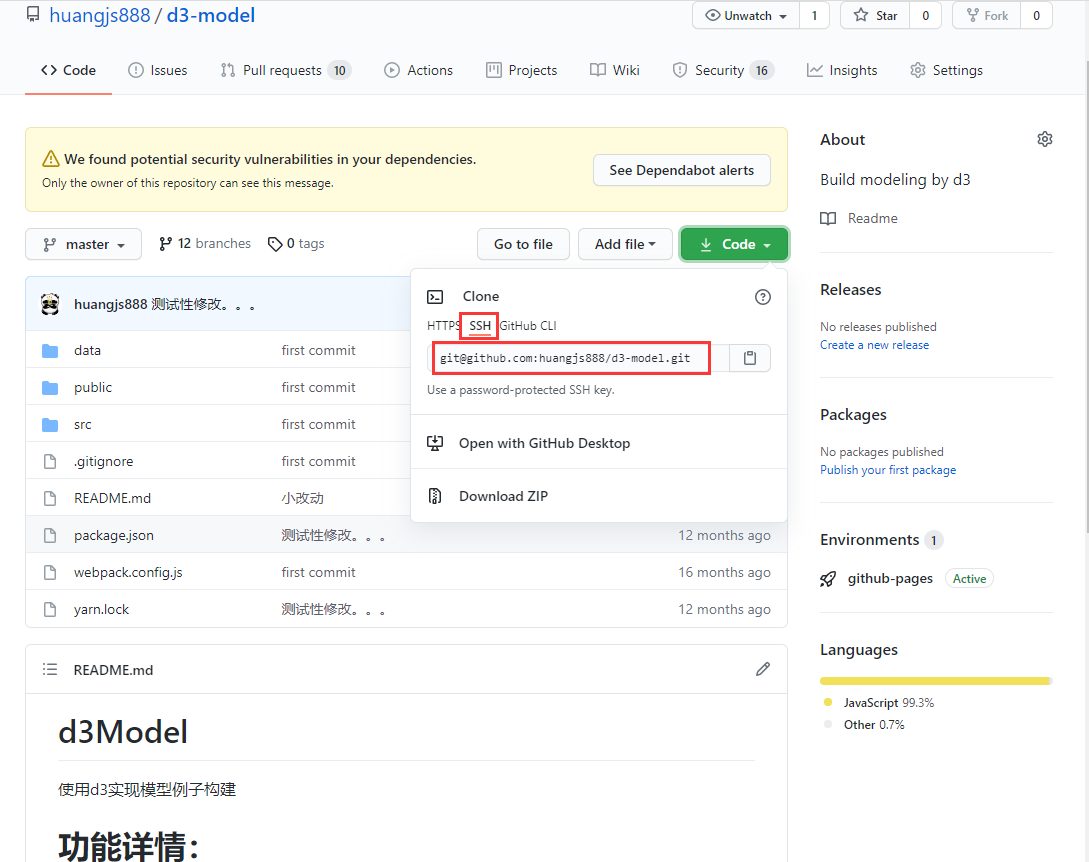




4，在弹出框里输入仓库全称确认，然后点下面的按钮即可。



**3，关联github上的某一个仓库到本地并更新仓库代码**

1, 拷贝对应的https或者ssh的地址，然后使用git命令进行关联。

2，在要建立项目的文件夹下，右键git bash进入执行：

git clone (https://github.com/xxx|git@github.com:xxx.git) project

相当于执行以下几步：

0: 新建项目目录project，cd进入project

1: git init

2: git remote add origin (https://github.com/xxx|git@github.com:xxx.git)

3: git pull origin master

第3步相当于执行：

1: git fetch origin master

2: git checkout master

3: git branch -u origin/master (将当前master分支追踪远程master分支)

说明：

a：git clone 可以不指定project，其会默认使用远程项目名称，同时其会默认指定远程地址的别名为origin，同时会自动设置本地 master 分支跟踪远程仓库的 master 分支，并且会默认拉取所有分支内容。

b：分步骤方式，只会拉取master分支并建立追踪，如果需要拉取其它分支，例如有个devlop分支，则：git checkout -b devlop，然后git pull origin devlop ( git fetch origin develop,然后 git branch -u origin/develop)，使用分步骤方式可以拉去自己想要跟踪的分支，不需要把远程不想要的分支都拉下来。

c：可以使用git branch –avv查看所有分支，及追踪分支。

3，有修改后，使用add和commit命令后push到origin master即可。注意要查询git config 看看name和email注意提交人的身份。、