

# 【保姆级教程】如何将Demo上传至GitHub

## 一、什么是GitHub

GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 Git 作为唯一的版本库格式进行托管，故名 GitHub。

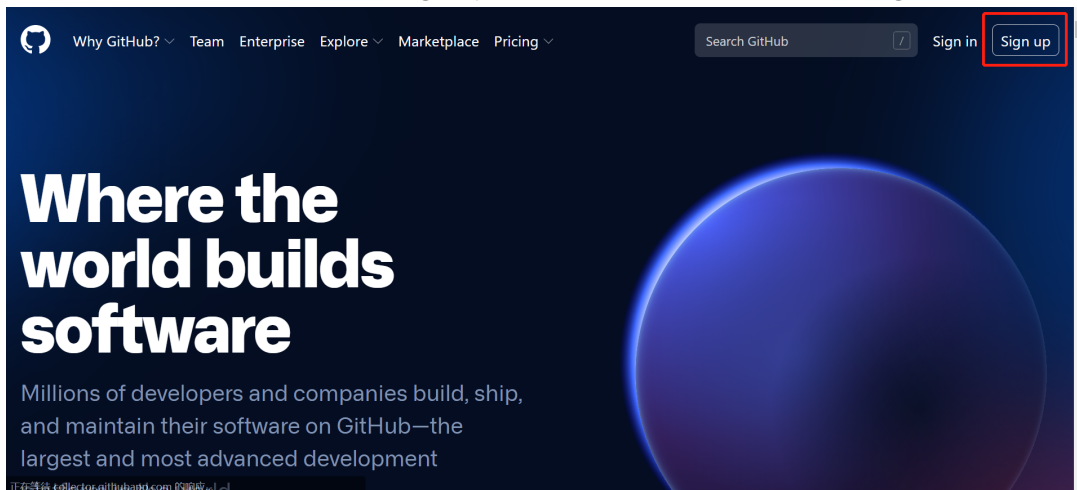
GitHub于2008年4月10日正式上线，除了Git代码仓库托管及基本的Web管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。目前，其注册用户已经超过350万，托管版本数量也是非常之多，其中不乏知名开源项目Ruby on Rails、jQuery、python等。在 GitHub，用户可以通过Explore轻而易举地找到海量的开源代码。因此，称之为程序员的 圣地 也不过吧？

由于GitHub是国外的网站，所以可能需要科学的上网工具进行登录。

## 二、注册GitHub账号

使用GitHub，首先必须需要有一个GitHub账号，

1、首先登录 [GitHub官网](#) 点击“Sign Up”，如果已经有GitHub账号点击“Sign In”



2、填写用户名、邮箱地址、密码等用户信息。

# Create your account

Username \*

用户名

Email address \*

邮箱地址

Password \*

密码

Make sure it's **at least 15 characters** OR **at least 8 characters including a number and a lowercase letter**. [Learn more](#).

Email preferences

是否接受邮件推广，如果不用可以不勾

☐ Send me occasional product updates, announcements, and offers.

3、真人校验，点击“验证”，完成验证之后，点击“Create account”。进入下一步

Verify your account

请回答此问题，以证明您是真人

验证



Create account

4、欢迎加入GitHub~请填写工作、开发经验、编程语言等等个人信息。完成之后点击“complete setup”（英语水平和我一样的同学，请打开“有道词典”）。

## Welcome to GitHub

Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there.

What kind of work do you do, mainly?

Software Engineer  
I write code

Student  
I go to school

Product Manager  
I write specs

UX & Design  
I draw interfaces

Data & Analytics  
I write queries

Marketing & Sales  
I look at charts

5、邮箱验证，打开自己的邮箱~打开GitHub发给你的邮寄，点击“verify email address”。



## Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to 2367262310@qq.com.

[Resend verification email](#)

重新发送邮件

[Change your email settings](#)

更换邮箱



Almost done, @tuyalele! To complete your GitHub sign up, we just need to verify your email address: [2367262310@qq.com](#).

[Verify email address](#)

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

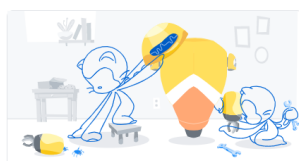
Button not working? Paste the following link into your browser: [https://github.com/users/tuyalele/emails/149313918/confirm\\_verification/34cc3f6cc086dd9ff5a0826b5c64e7ee424ba9dc](https://github.com/users/tuyalele/emails/149313918/confirm_verification/34cc3f6cc086dd9ff5a0826b5c64e7ee424ba9dc)

You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.

6、进入GitHub第一件需要做的事情是什么？当然是创建一个自己的代码仓库啦！

## What do you want to do first?

Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



### Start a new project

Start a new repository or bring over an existing repository to keep contributing to it.

[Create a repository](#)



### Collaborate with your team

Improve the way your team works together and get access to more features with an organization.

[Create an organization](#)



### Learn how to use GitHub

Get started with an "Introduction to GitHub" course in our Learning Lab.

[Start Learning](#)

## 三、创建代码仓库

1、创建代码仓库，设置仓库名称、仓库描述，公开仓库。点击“Create repository”  
GitHub仓库结束必须用英文介绍，精简描述即可。

建议：仓库名统一用小写字母/数字 + “-”的方式命名，如aws-iot-device-sdk-js-v2

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner \* 选择个人Github账户 Repository name \* Github仓库命名

 / 

Great repository names are short and memorable. Need inspiration? How about **fictional-fortnight**?

Description (optional) Github仓库描述

- ☒  **Public** Public为公开仓库，默认选择该项  
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

### Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file** 添加默认的Readme文档模板  
This is where you can write a long description for your project. [Learn more](#).

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

☒ **Choose a license** 选择代码License，默认选择MIT  
A license tells others what they can and can't do with your code. [Learn more](#).

License: MIT License ▼

This will set  **main** as the default branch. Change the default name in Tuya-Community's [settings](#).

Create repository

## 四、编写README.md文件

### 1、什么是README.md文件

了解一个项目，恐怕首先都是通过其Readme文件了解信息。如果你以为Readme文件都是随便写写的那你就错了。github, oschina git git cafe的代码托管平台上的项目的Readme.MD文件都是有其特有的语法的。称之为Markdown语法。

### 2、如何写好一个README呢？

需要提供中英文版本，内容包括：

- 项目的简单介绍
- 项目的使用教程介绍，包括配套的硬件设备，使用的涂鸦SDK类型和版本号，如何运行项目等
- 项目相关的demo照片，App截屏，视频链接等
- 项目使用的开源License文件，如MIT
- 项目作者名称（可选）

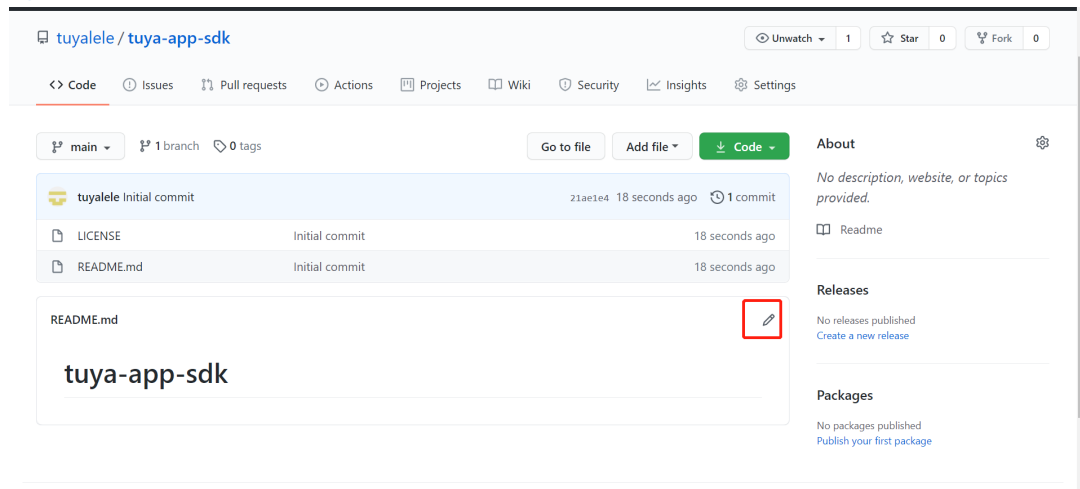
**必须加上这句话描述在Readme文档开头：**

This project is developed using Tuya SDK, which enables you to quickly develop branded apps connecting and controlling smart scenarios of many devices.

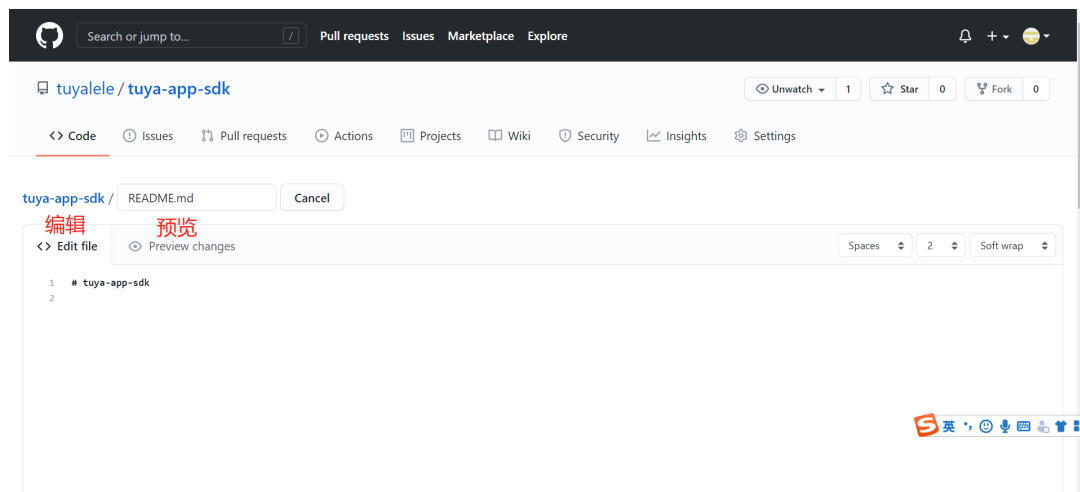
For more information, please check Tuya Developer Website.

3、如何在Git Hub上编辑README.md文件，找到代码库中的README.md文件，点击红框内的图标。

如果不习惯在Git Hub上编写可以选用在线剪辑软件：<https://maxiang.io/> 或者 <http://mahuajser.me/> 都可以。有道云笔记也有相应的功能。



#### 4、编辑README文件



#### MD的基础语法：

**大标题（一级标题）：**在文本下面加等于号，那么上方的文字就变成了大标题，等于号的个数无限制，但一定要大于0

- 1 大标题
- 2 ==

**中标题（二级标题）：**在文本下面加下划线，那么上方的文本就变成了中标题，下划线个数无限制，中标题比大标题低一级

- 1 中标题
- 2 -----

**1~6级标题：**文本大小依次减小，以#号开头，多少个#号就是多少级标题，#号和标题名称要并排写

- 1 #一级标题
- 2 ##二级标题
- 3 ###三级标题
- 4 ####四级标题
- 5 #####五级标题
- 6 #####六级标题

**插入圆点符号：**编辑的时候使用的是星号\*，星号后面要有一个空格，否则为普通星号

- 1 \* 列表一

```
2 * 列表二
3 * 列表三
```

**二级圆点、三级圆点：**多加一个Tab，即第二行一个Tab，第三行两个Tab

```
1 * 列表一
2   * 列表二
3     *列表三
```

**缩进：**

```
1 >缩进一
2 >>缩进二
3 >>>缩进三
4 >>>>缩进四
5 >>>>>缩进五
```

**插入链接：**

```
1 [百度](http://baidu.com)
```

**插入网络图片：**![网络图片链接地址]，即叹号!+方括号[]+括号()，如果不加叹号!就会变成普通文本，方括号里可以加入一些 标识性的信息。

**插入GitHub仓库里的图片：**，即叹号!+方括号[]+括号()，URL写法：<http://github.com/自己的用户名/项目名/raw/分支名/存放图片的文件夹/文件夹里的图片名字>

```
1 ![baidu](http://www.baidu.com/img/bdlogo.gif "百度logo")
```

**插入代码片段：**在代码上下行用```标记

**注意：**`符号是tab键上面那个，要实现语法高亮，则在```后面加上编程语言的名称

```
1 ```Java
2 public static void main(String[] args){}
3 ```
4 ```javascript
5 document.getElementById("ts").innerHTML="Hello"
6 ```
```

**换行：**使用标签<br>

**单行文本：**前面使用两个Tab

**多行文本：**每行行首加两个Tab

**部分文字高亮：**使用``包围，这个符号不是单引号，而是Tab上方，数字1左边那个按键的符号

**文字超链接格式：**[要显示的文字](链接的地址"鼠标悬停显示")，在URL之后用双引号括起来一个字符串，即鼠标悬停显示的文本，可不写。

**更多Markdown语法可以参考：**<https://guides.github.com/features/mastering-markdown/>

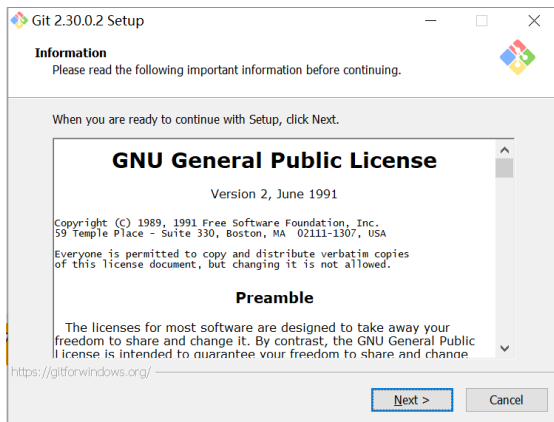
## 五、下载git

**什么是git?**

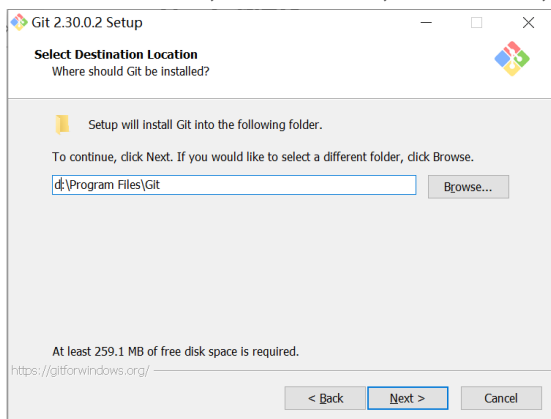
git是目前世界上最先进的分布式版本控制系统。

1、点击 [下载git](#) （也可以在QQ群文件中找到“Git-2.30.0.2-64-bit.exe”下载）

2、安装 git "next>"

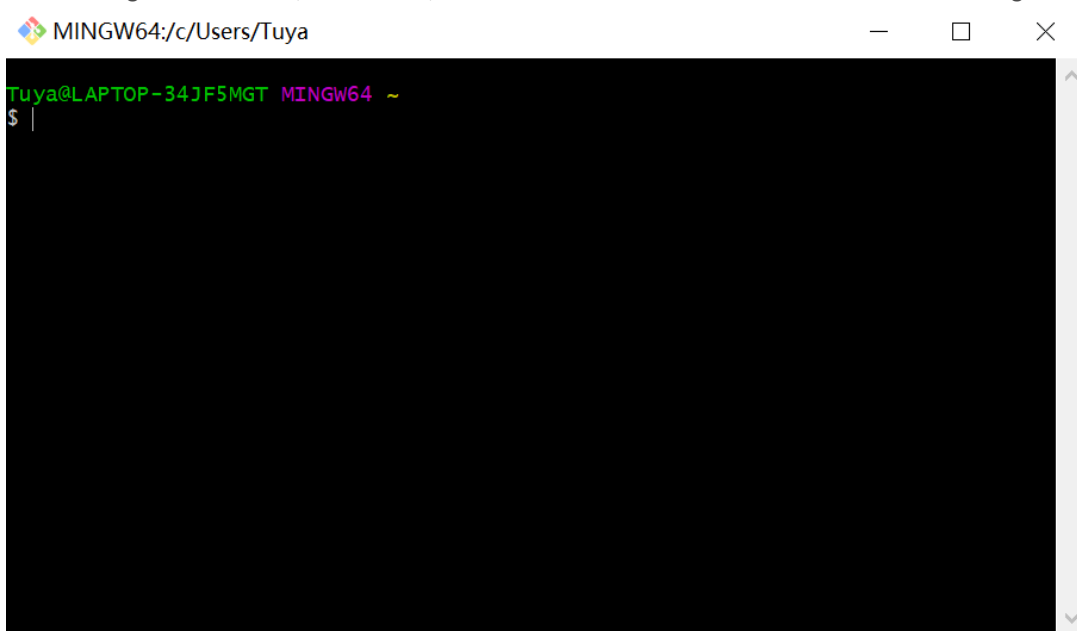


3、设置安装路径，然后”next>”，看不懂选啥，就默认。一路”next>”。



## 六、git上传Demo

1、找到“git Bash”文件，双击打开，出现这么一个命令行文件。说明你已经成功安装git！



2、设置用户名、邮箱。出门在外，岂能没有姓名，不然别人怎么知道demo是你上传的！

**邮箱需要和GitHub注册的邮箱一致！**

输入下面的命令，回车。（姓名和邮箱记得换成你自己的！不填就是查看信息）

```
1 git config --global user.name 姓名
2 git config --global user.email 邮箱
```

```
MINGW64:/c/Users/Tuya
Tuya@LAPTOP-34JF5MGT MINGW64 ~
$ git config --global user.name tuyu

Tuya@LAPTOP-34JF5MGT MINGW64 ~
$ git config --global user.name
tuya

Tuya@LAPTOP-34JF5MGT MINGW64 ~
$ git config --global user.email
3-om

Tuya@LAPTOP-34JF5MGT MINGW64 ~
$
```

3、进入工程目录，我下载了MCU SDK放在了桌面进行演示。

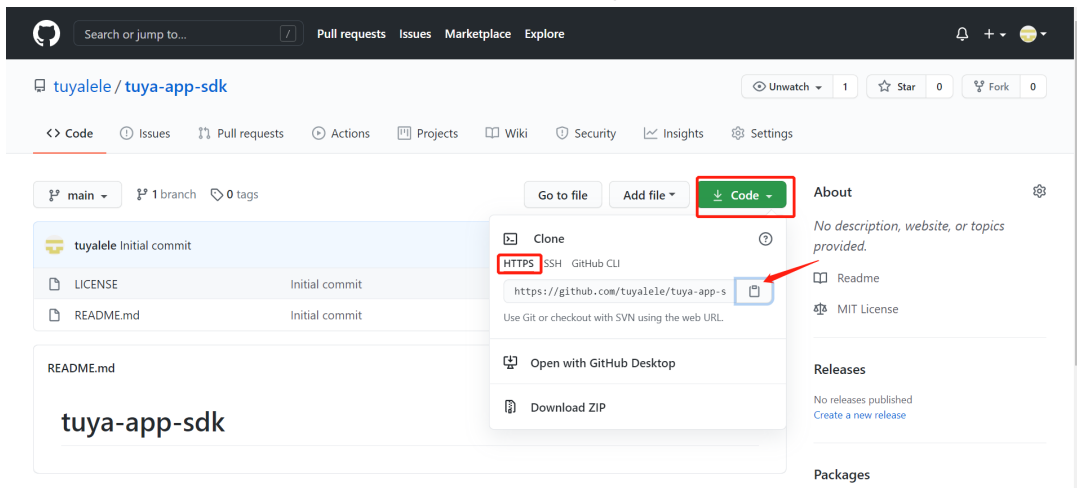
命令 cd：进入相应目录，你们直接自己工程的目录。

```
Tuya@LAPTOP-34JF5MGT MINGW64 ~/Desktop/mcu_sdk
$ cd C:\Users\Tuya\Desktop\mcu_sdk
```

4、从GitHub上clone自己的代码仓库到本地，为了更直观的演示，我就在桌面演示了

- 1 cd Desktop //首先你需要进入桌面
- 2 git clone 你的GitHub地址 //从GitHub上clone代码仓库到本地，运行之后你会看到一个空的文件夹。

仓库地址查找：在GitHub项目中点 code-clone-https 再点这个小图标复制就可以。



这里你会在你的目录下看到一个空的文件夹~然后把你的项目复制到文件夹里。

5、上传demo（不明白为什么的，复制就可以）

- 1 git add --all //将项目上所有的文件添加到仓库中
- 2 git commit -a -m "first commit" //提交demo，引号内的内容自己可以修改
- 3 git push origin //push



```
Tuya@LAPTOP-34JF5MGT MINGW64 ~/Desktop/tuya-mcu-demo (main)
$ git commit -a -m "commit"
[main 96ee308] commit
12 files changed, 7984 insertions(+)
create mode 100644 mcu_sdk/VERSION.md
create mode 100644 mcu_sdk/cJSON.c
create mode 100644 mcu_sdk/cJSON.h
create mode 100644 mcu_sdk/mcu_api.c
create mode 100644 mcu_sdk/mcu_api.h
create mode 100644 mcu_sdk/protocol.c
create mode 100644 mcu_sdk/protocol.h
create mode 100644 mcu_sdk/readme.txt
create mode 100644 mcu_sdk/system.c
create mode 100644 mcu_sdk/system.h
create mode 100644 mcu_sdk/wifi.h
create mode 100644 "\346\225\260\346\215\256\347\202\271\346\226\207\346\241\243_123_202102031746.xls"

Tuya@LAPTOP-34JF5MGT MINGW64 ~/Desktop/tuya-mcu-demo (main)
$ git push origin
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 61.27 KiB | 61.27 MiB/s, done.
Total 15 (delta 0), reused 11 (delta 0), pack-reused 0
To https://github.com/tuyalele/tuya-mcu-demo.git
   leedf3c..96ee308  main -> main
```

这样我们的Demo就成功的上传到GitHub上面啦！！

只要复制代码仓库的地址！

就可以提交你的作品啦！！

## 附一、如何设置GitHub SSH Key

1、验证ssh key是否存在，主要看看有没有这两个文件，如果存在，说明已经有SSH Key。

```
1 cd ~/.ssh
2 ls
```

```
Tuya@LAPTOP-34JF5MGT MINGW64 ~
$ cd ~/.ssh

Tuya@LAPTOP-34JF5MGT MINGW64 ~/.ssh
$ ls
id_rsa id_rsa.pub

Tuya@LAPTOP-34JF5MGT MINGW64 ~/.ssh
$ ll
total 5
-rw-r--r-- 1 Tuya 197121 2590 Feb  3 19:30 id_rsa
-rw-r--r-- 1 Tuya 197121  558 Feb  3 19:30 id_rsa.pub
```

2、如果没有文件，需要生成一下，一路回车就好。

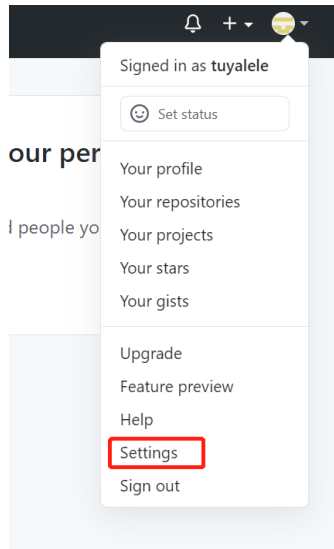
```
1 ssh-keygen -t rsa -C "lele" // ""内的内容可以替换成自己的姓名
```

3、执行之后继续执行第一步命令来获取SSH Key，找到“ssh-rsa”开头的字符串，这就是你的秘钥，复制它。

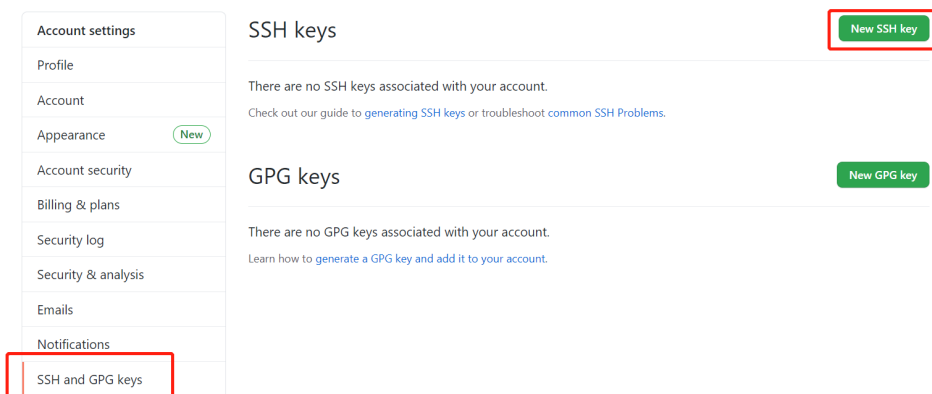
```
1 cat id_rsa.pub
```

```
Tuya@LAPTOP-34JF5MGT MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDcwPyXMBdcoOzB8bF6FmN9aiDFNsapnYZvAgVQLK3m
SAelTH13pkm41pd9KRQyxwMqZfyY1Iz07UzX4Umtxi9hS7XgbSf4nkAC1yjdX4+MDci4oBCeN2w1/CXq
dTxmZxS1K1ec4ELQ/bNDTLd+7nEYIcITdT2XY8dPEYDb7c+h7pHTW4jSTNMnaFo1A09QhCvSniATZzx
p1dx7t2vmwpFogHrivPUJtoPtXDxJc433w9HqBvRRezLs28Y0qPaw8LcItSg/tADRB/SFFWpNrglQpG
bF2UdYDPZnpj/WJX/fuwMk1PFIdcsq2ksCTF2tkMtjZTp8Mi2HD+xDMDQter0INLhAAiJY8v5YZvyx5/
AeqST5oSqbFOYuDzVgKTSkD8RF1FFKhCvADsvaCMoz40MKmyCWF/9yhw7xKrhDomHFQdjXj5YSkAETit
d6+vkoGYADROvf7wq3pdCf5T+83cwjecjeBDSFQHoij3CH7TLNYWRitsWkyIZ06+8j50mU= lele
```

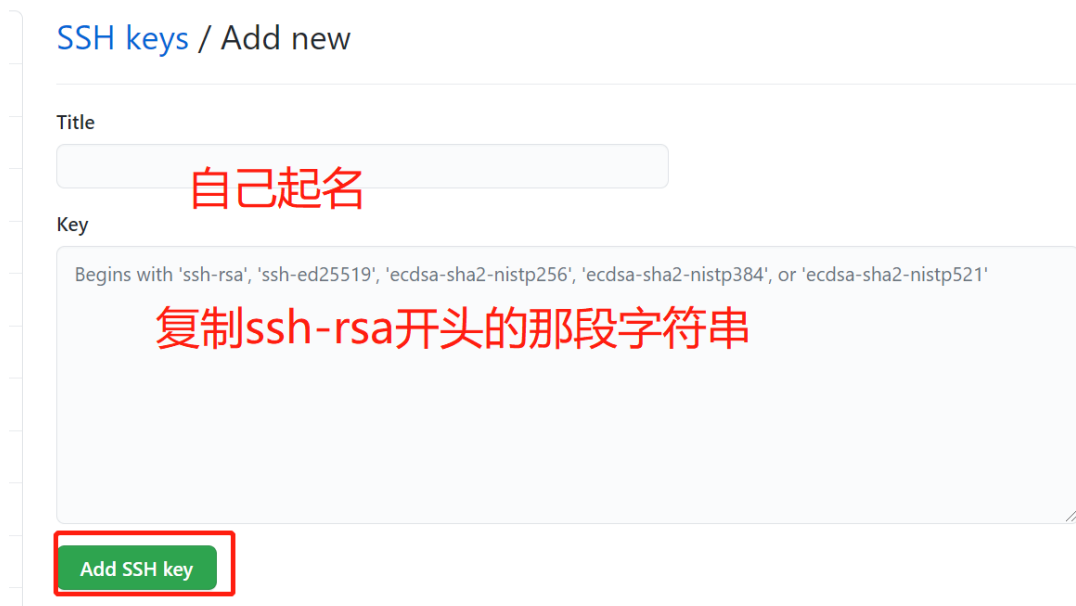
4、登录GitHub，点击右上角头像，选择Settings



5、左边选择”SSH and GPG key”，点击”New SSH key”。



6、取个名字，把之前拷贝的秘钥复制进去，添加就好啦。



7、测试是否成功配置SSH Key，出现下面这句话，就说明设置成功啦！

```
1 ssh -T git@github.com //验证是否配置成功
```

```
IG Tuya@LAPTOP-34JF5MGT MINGW64 ~/.ssh
$ ssh -T git@github.com
Hi tuyalele! You've successfully authenticated, but GitHub does not provide shell
access.
Tuya@LAPTOP-34JF5MGT MINGW64 ~/.ssh
```

如果出现下面这段话，记得输入yes，回车稍等一会就可以啦。

