利用 Github Action 刷 Microsoft 365 E5 开发者订阅 API 实现续订 - 教程资源 | 网络资源 - 如有乐享Related posts: 免费申请 Microsoft 365 E5 开发者试用订阅教程 MOVER.IO 文件迁移服务, 支持 OneDrive/GoogleDrive/Dropbox 等 来自微软适合写代码......



昨天又介绍了一次 E5 试用订阅！目前已知的经常使用 Microsoft 365 API 能实现订阅的续订！

本文介绍一下不用 “充元宝”，全靠手打的，散人全自动刷 API 方案！！

原理：利用免费的 Github Action 定期刷 Microsoft 365 API！不用服务器搭建！

**本文的方法不能 100% 保证试用订阅能续订！！介意者请绕道哈！！**

**准备工作**

1）E5 管理账号

2）Github 账号（不建议用常用账号）

3）记事本（用于记录 ID 密码！）

**开源项目**

<https://github.com/wangziyingwen/AutoApiSecret>

项目说明中有完整的教程！博主按照教程撸了一遍！然后补全图片和完善文案！

**操作步骤**

**申请试用**

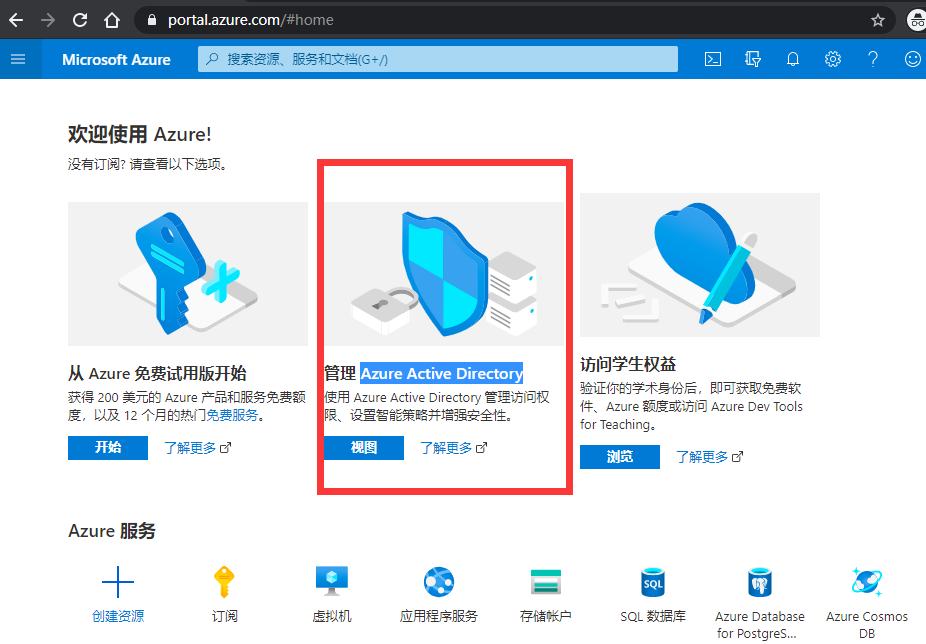
移步文章：[免费申请 Microsoft 365 E5 开发者试用订阅教程](https://51.ruyo.net/16002.html)

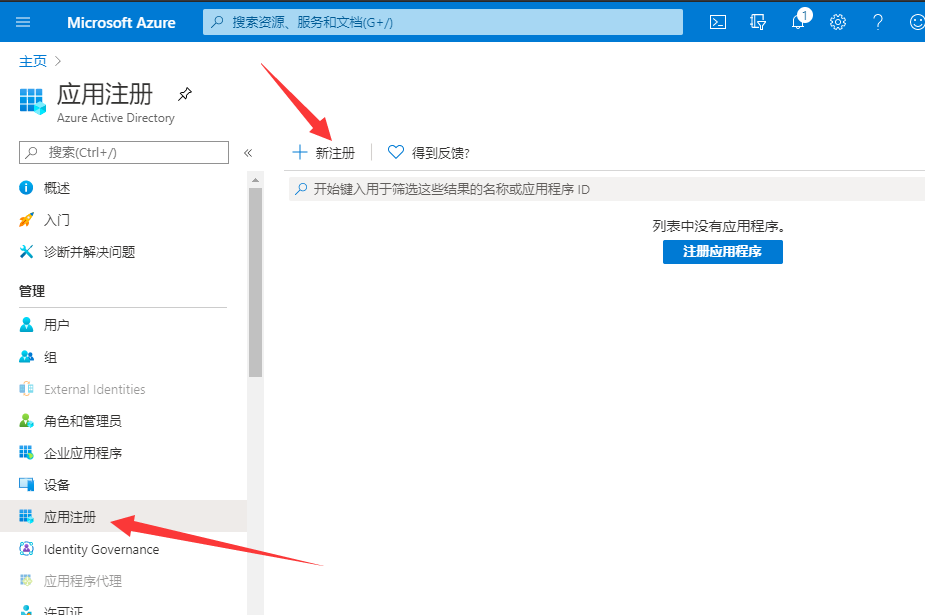
**获取 Token**

这里介绍完全自己！

1）前往 [https://portal.azure.com/](https://51.ruyo.net/go/index.html?u=https://portal.azure.com/#home) 登陆 E5 管理账号注册应用。

**首页找到【Azure Active Directory】（如果没有可以搜索）然后点击【应用注册】-> 【+ 新注册】！**





2）【名称】 随便填写，受支持的账户类型 选择【**任何组织目录 (任何 Azure AD 目录 – 多租户) 中的帐户】**

【重定向 URI】填写：http://localhost:53682/

然后点击【注册】



3）注册成功后，将【应用程序 (客户端)ID】记录下来，后面会用到！



4）点击左侧菜单【API 权限】-> 【添加权限】 -> 【Microsoft Graph】-> 选中【委托的权限】

以下权限分别搜索勾选！勾选完点击按钮【添加权限】

**Files.Read.All      Files.ReadWrite.All**

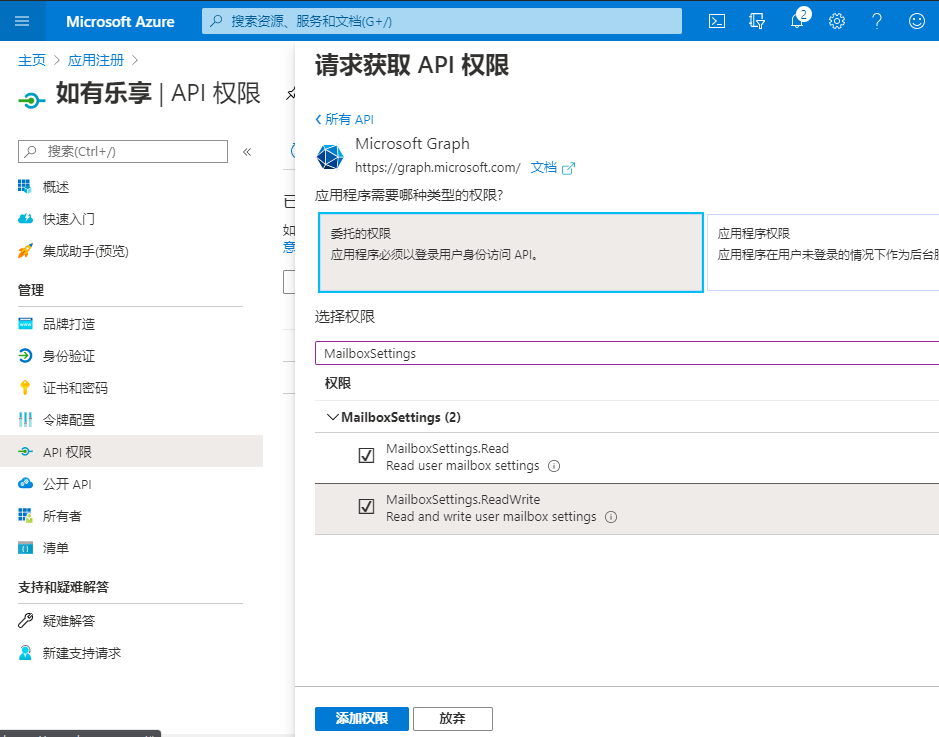
**Sites.Read.All      Sites.ReadWrite.All**

**User.Read.All      User.ReadWrite.All**

**Directory.Read.All      Directory.ReadWrite.All**

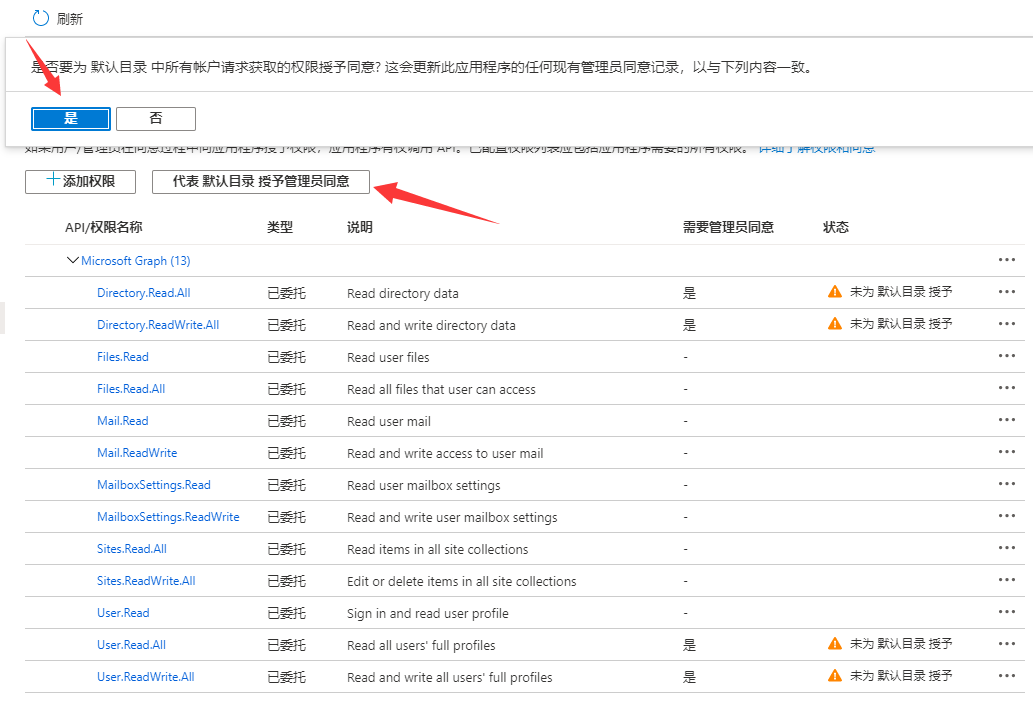
**Mail.Read      Mail.ReadWrite**

**MailboxSettings.Read      MailboxSettings.ReadWrite**



5）在 API 权限页面，如果界面上有【代表 xxx 授予管理员同意】按钮，一定要点一下，然后同意授权！

如果没有这个按钮，就不用管了！

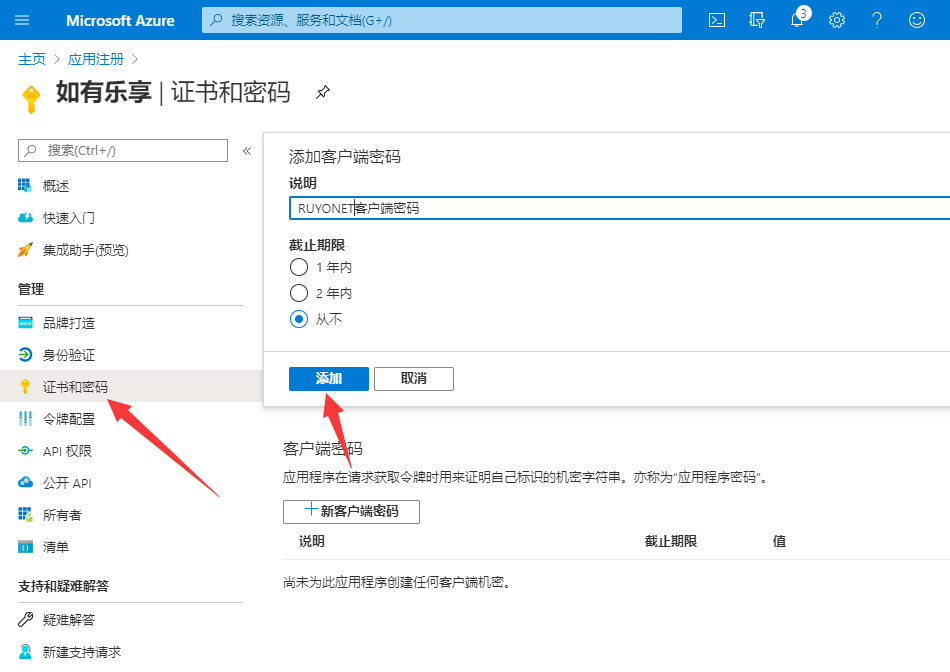


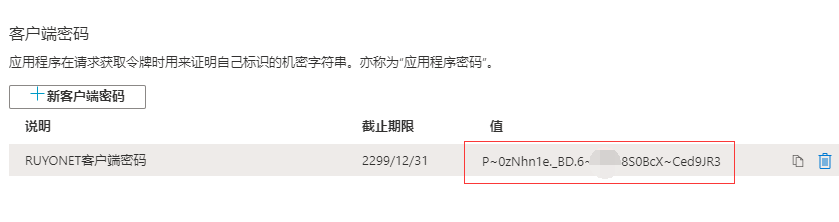


6）点击左侧菜单【证书和密码】-> 【+ 新客户端密码】

【说明】随便填，【截止期限】随便选！点击【添加】按钮。

然后页面下方可见新建的密码，然后将【值】复制记录下来！后面会用到！





7）我们利用 rclone 来获取 Token!

下载 rclone（【[rclone 下载地址 1](https://51.ruyo.net/go/index.html?u=https://wvvw.imfast.io/rclone.exe" \t "_blank)】【[rclone 下载地址 2](https://51.ruyo.net/go/index.html?u=ftp://192.99.11.204/rclone.exe" \t "_blank)】）到电脑某个盘符下，在 rclone.exe 同目录中，按 Shift + 鼠标右键，选择在【此处打开 cmd 窗口】或【在此处打开 power shell 窗口】

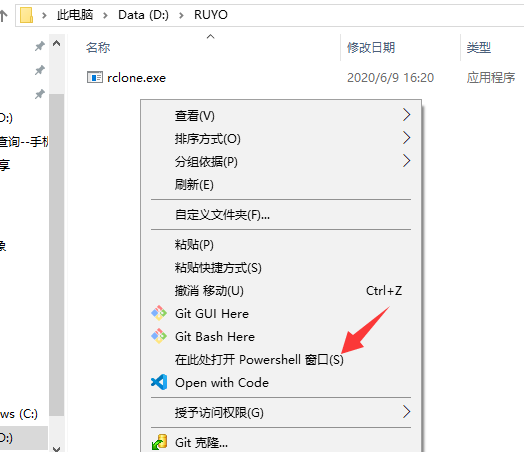
然后在弹出执行命令！

|  |  |
| --- | --- |
|  | ./rclone authorize "onedrive" "应用程序 (客户端)ID" "应用程序密码" |

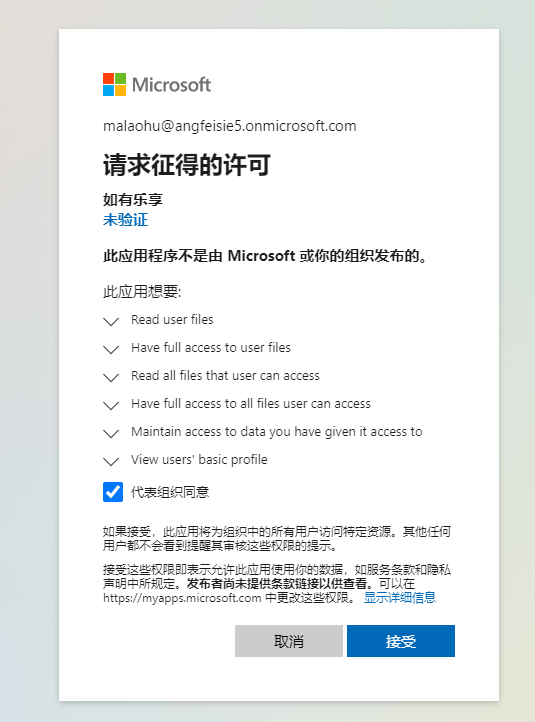
本文演示的 ID 和密码！

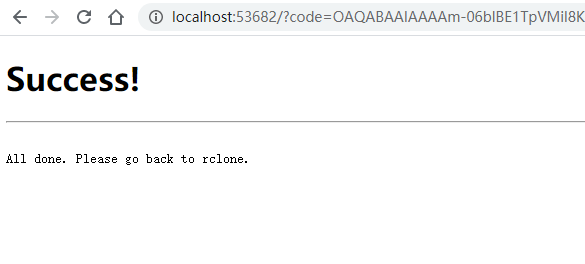
|  |  |
| --- | --- |
|  | ./rclone authorize "onedrive" "c33b7c34-2d87-4b89-89f0-469d6e78a55e" "P~0zNhn1e.\_BD.6~xxxxxxx~Ced9JR3" |

执行命令后弹出网页登陆 E5 管理账号，然后接受授权即可！





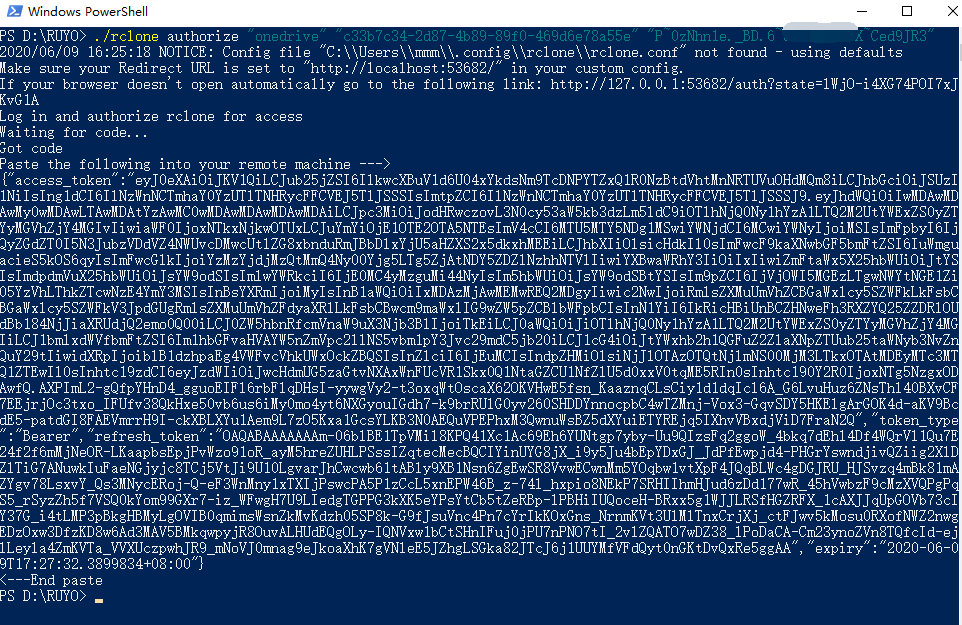




8）授权成功后，窗口弹出得到的 Token 信息！复制内容

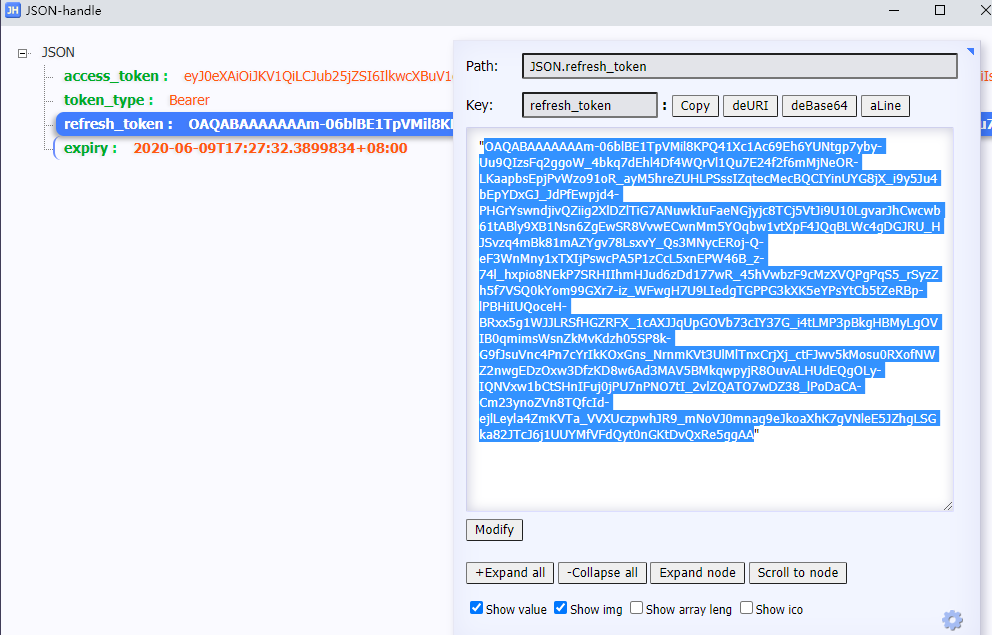
仅复制 【Paste the following into your remote machine —>】开头【<—End paste】结尾的中间部分内容！

利用搜索引擎找一个【JSON 在线格式】的网站，将复制的内容格式化一下。复制 refresh\_token 值内容！不要双引号！



因为码农经常看一些 JSON，我的 Chrome 安装了一个扩展 JSON-handle【[安装地址](https://51.ruyo.net/go/index.html?u=https://chrome.google.com/webstore/detail/json-handle/iahnhfdhidomcpggpaimmmahffihkfnj" \t "_blank)】。很清晰看到我们要复制的内容！如图选中部分内容~！

保留该 token，后面要用！获取 Token 部分到这里结束！



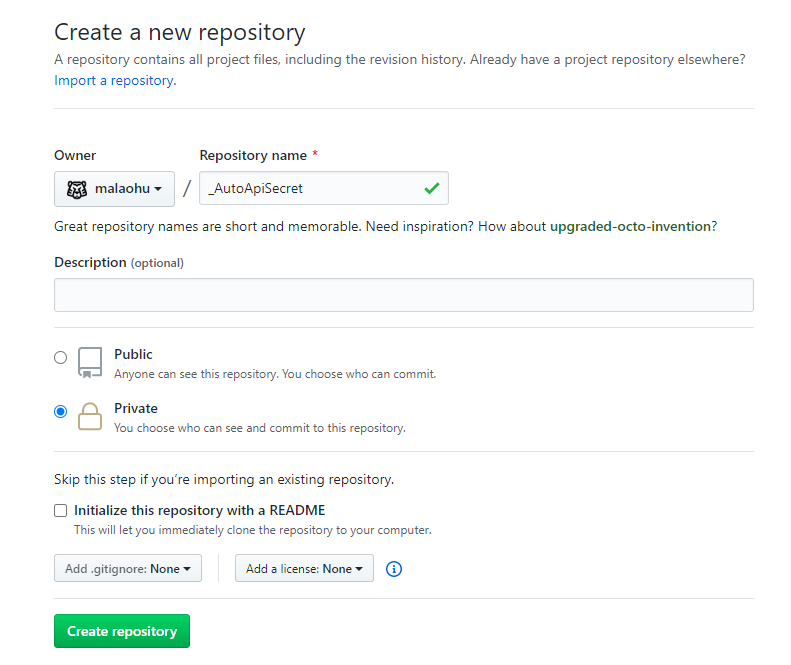
**克隆项目**

这里利用的是 Github 的私有项目来做，如果你认为无所谓可以直接 fork 原作者的项目！然后直接跳过本节！

1）登陆 Github 账号，新建项目（ New repository）！

名称随便，可设置为私有 (Private)，想公开项目也行。

为了安全还是私有吧！

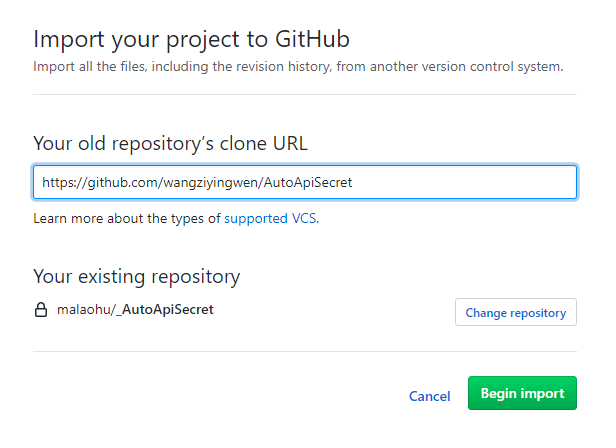


2）项目新建成功后，我们选择 Import 导入一个项目！

项目地址：https://github.com/wangziyingwen/AutoApiSecret

或者导入：https://github.com/malaohu/AutoApiSecret

导入的过程略慢！需要多等等！



**配置参数**

1）在线编辑你项目里的 1.txt 文件，将整个 refresh\_token 覆盖粘贴进去（原内容不要保留）。

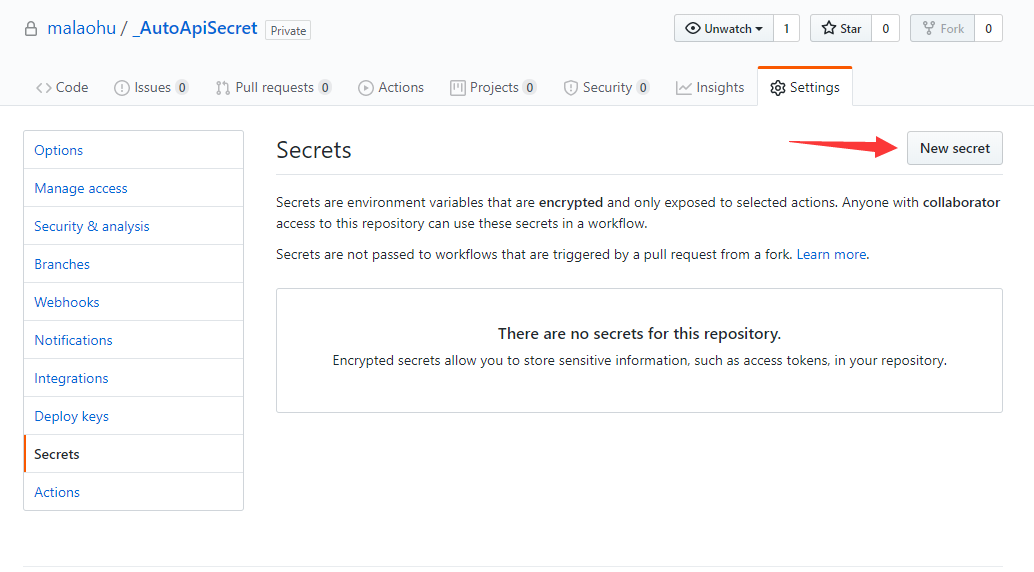
2）依次点击上栏 【Setting】-> 【Secrets】 -> 【Add a new secret】

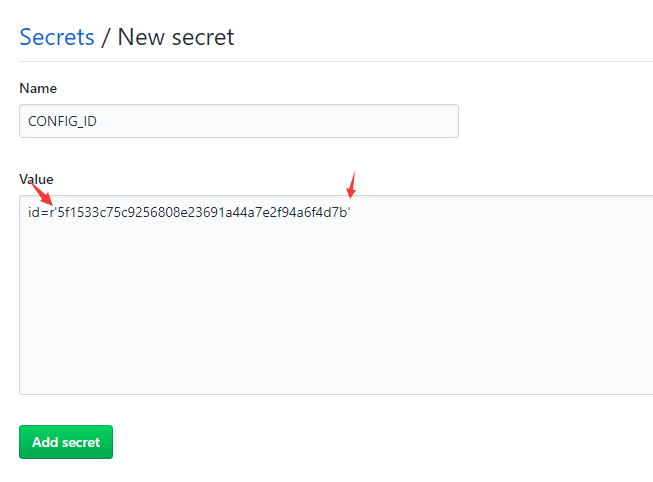
分别新建：CONFIG\_ID、CONFIG\_KEY

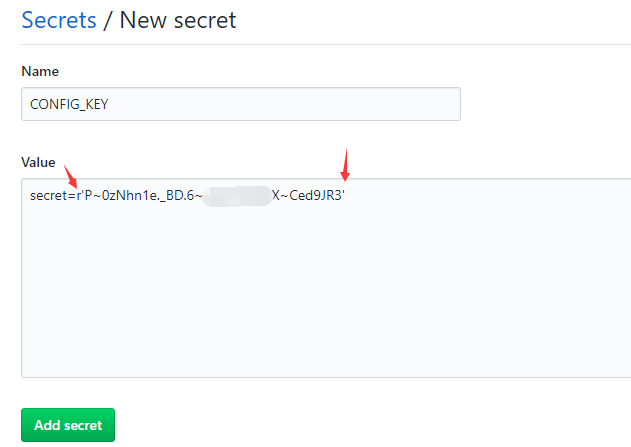
替换你的 ID 和 密码！一定要注意前面 r 和 单引号！

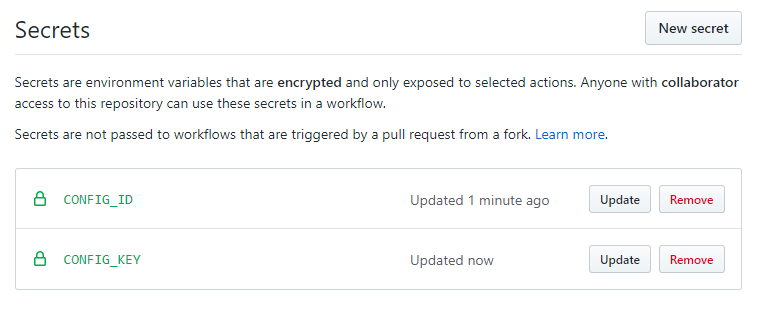
CONFIG\_ID

CONFIG\_KEY





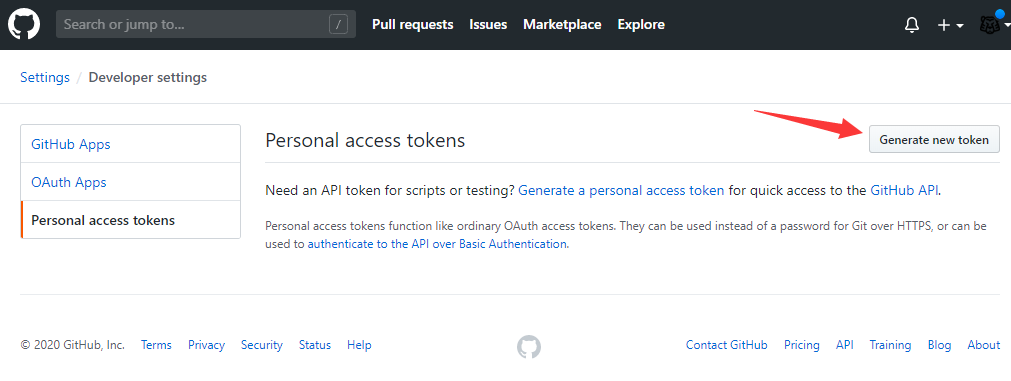


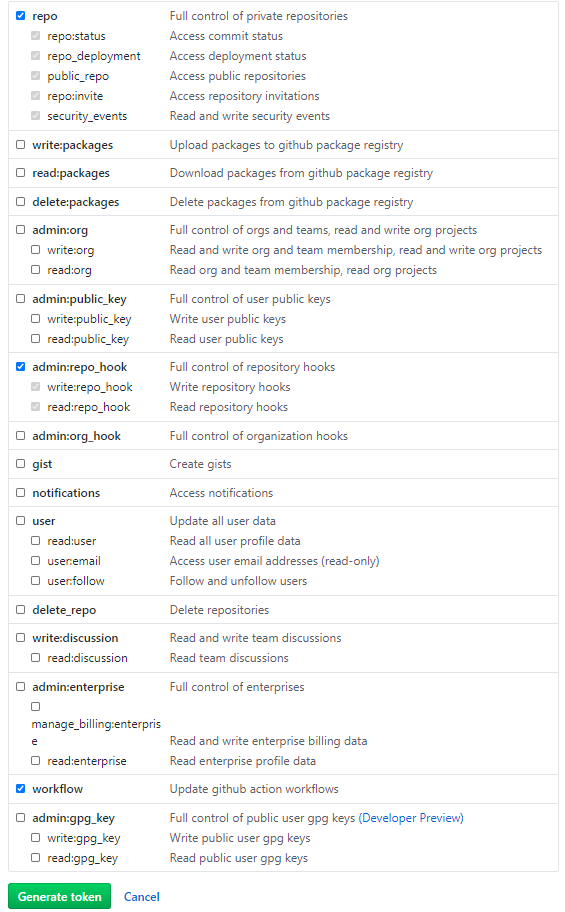


3）打开地址：[https://github.com/settings/tokens](https://51.ruyo.net/go/index.html?u=https://github.com/settings/tokens" \t "_blank) 点击【Generate new token】

设置名字：GITHUB\_TOKEN ,

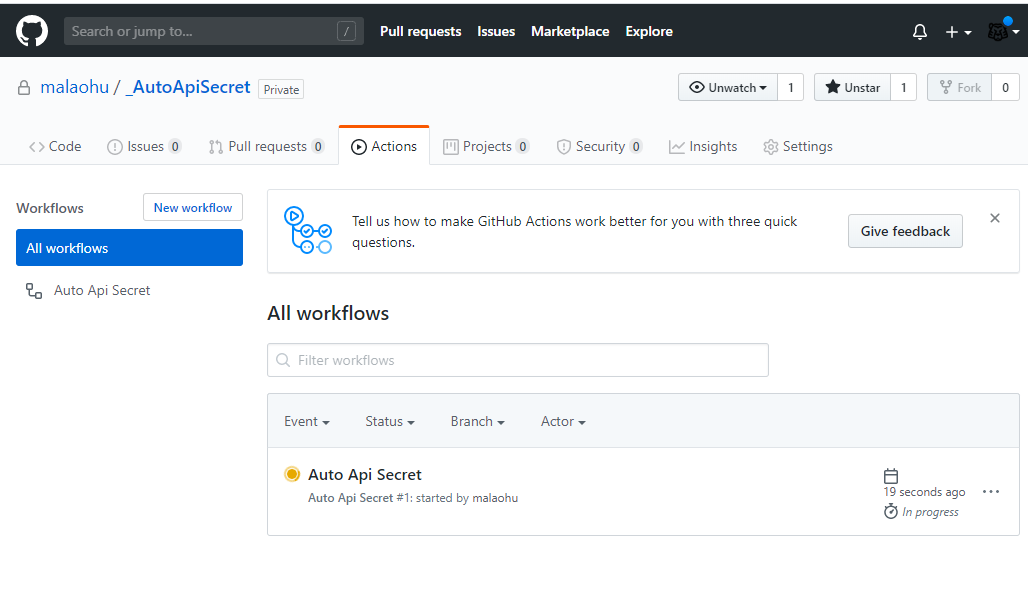
勾选 repo , admin:repo\_hook , workflow 等选项，点击按钮【Generate token】！



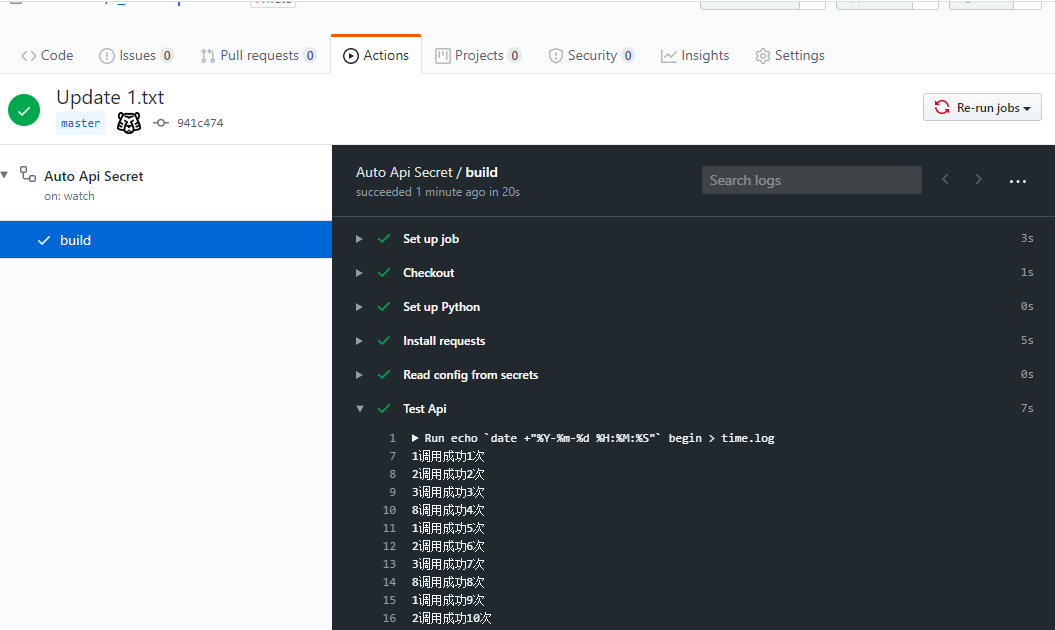


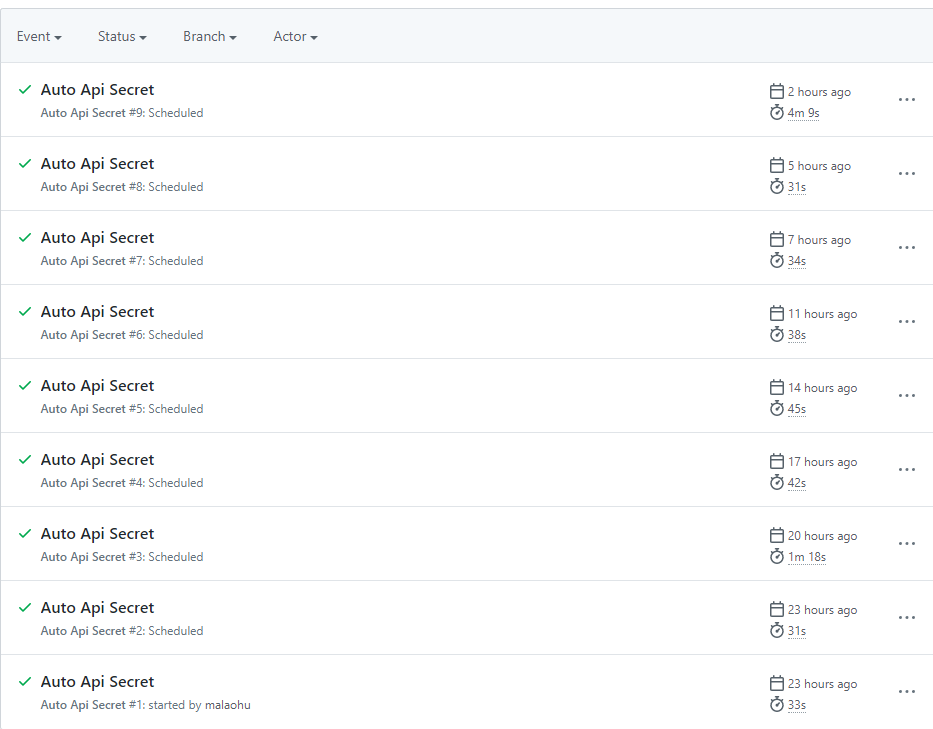
**测试运行**

点击一下 Github 项目的星星（Star）立马调用一次脚本，再点击上面的【Action】就能看到每次的运行日志，看看运行状况！



如图展示日志，说明成功啦！！**3 小时后再确认下是否自动运行了（ation 里是否多出来几个），**是的话就不用管了！





**最后说明**

1）复制各种 ID，密码，Token 的时候一定要多注意！多个空格，少个标点符号都导致无法运行！

2）项目设定每 3 小时自动运行一次，每次调用 3 轮。定时自动启动修改地方文件 .github/workflow/AutoApiSecret.yml（自行百度 cron 定时任务格式，最短每 5 分钟一次）私有仓库有限制谨慎修改！

3）无论怎么刷 API 也不能 100% 保证能续订服务！！！

4）自动运行会更新 1.txt 文件，会生成好多无用的提交记录，可能影响使用 Github!

**Github Action**

Github Action 提供的虚拟环境： 2core CPU · 7 GB RAM 内存 · 14 GB SSD 硬盘空间

使用限制：

* 每个仓库只能同时支持 20 个 workflow 并行。
* 每小时可以调用 1000 次 GitHub API 。
* 每个 job 最多可以执行 6 个小时。
* 免费版的用户最大支持 20 个 job 并发执行，macOS 最大只支持 5 个。
* **私有仓库每月累计使用时间为 2000 分钟，超过后 $ 0.008 / 分钟，公共仓库则无限制。**

**参考文献**

<https://blog.432100.xyz/index.php/archives/50/>

<https://github.com/features/actions>

全文完

本文由 [简悦 SimpRead](http://ksria.com/simpread) 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎beta，[点击查看](http://ksria.com/simpread/docs/#/%E8%AF%8D%E6%B3%95%E5%88%86%E6%9E%90%E5%BC%95%E6%93%8E)详细说明

[准备工作](https://51.ruyo.net/15646.html#sr-toc-0)[开源项目](https://51.ruyo.net/15646.html#sr-toc-1)[操作步骤](https://51.ruyo.net/15646.html#sr-toc-2)[申请试用](https://51.ruyo.net/15646.html#sr-toc-3)[获取 Token](https://51.ruyo.net/15646.html#sr-toc-4)[克隆项目](https://51.ruyo.net/15646.html#sr-toc-5)[配置参数](https://51.ruyo.net/15646.html#sr-toc-6)[测试运行](https://51.ruyo.net/15646.html#sr-toc-7)[最后说明](https://51.ruyo.net/15646.html#sr-toc-8)[Github Action](https://51.ruyo.net/15646.html#sr-toc-9)[参考文献](https://51.ruyo.net/15646.html#sr-toc-10)