github api

基于GitHub API的GitHub文件上传工具

jgit是java实现的git版本控制。主要的模块如下：

org.eclipse.jgit 核心实现，包括git命令、协议等

org.eclipse.jgit.archive 支持导出各种压缩的格式

org.eclipse.jgit.http.server 支持http协议的服务器，主要提供GitServlet

maven配置

<dependency>

<groupId>org.eclipse.jgit</groupId>

<artifactId>org.eclipse.jgit</artifactId>

<version>4.11.0.201803080745-r</version>

<dependency>

windows安装curl：<https://blog.csdn.net/qq_37958578/article/details/79973265>

jgit学习：https://blog.51cto.com/5162886/2094475 我可以用来创建本地仓库，向远程提交代码

https://blog.csdn.net/yulin\_Hu/article/details/81673446

Github Java API：http://github-api.kohsuke.org/project-reports.html 可以用来查看远程项目提交情况

GitHub API 的实例使用：<https://blog.csdn.net/qq_15174755/article/details/83306497>

<https://segmentfault.com/a/1190000015144126>

GitHua API 英文版本：<https://developer.github.com/v3/>

GitHua API 中文版本：<https://blog.csdn.net/xiaotao745324325/article/details/41621891>

github自己有API，如果自己写的爬虫没有特色或者没有很高的效率，不如使用API，数据获取可以来得快一些。github不像社交网站那样拥有大量用户和内容，所以数据规模相对小一点，使用API也相对便捷一些。

直接使用github的API，没有token的情况下，只有60次访问，而且有的IP直接被限制。除此之外，有一些API在访问的时候必须要认证的参数，所以还是进行认证好一些。

支持的验证访问方式有：使用token；使用auth认证。

对于生成的token还有访问权限范围的设置，避免滥用token。这是纯粹是为了开发，如果只是简单的获取点信息，没有什么设置的必要，可以全部关闭。

使用API之后，就可以拥有每小时5000次的请求。

Git和Github协同开发：<https://www.cnblogs.com/yhaing/p/8473746.html>

git checkout和git reset的区别：

<https://blog.csdn.net/longintchar/article/details/82314102>

shiro如何进行密码校验：

<https://blog.csdn.net/chenyidong521/article/details/80245362>



--------------------------------------------------------------------- \* ## 常用命令 ------------------ ``` git clone [url/ ssh] --- 克隆仓库

git init --- 初始化仓库

git add [name] --- 添加文件到暂存区

git add . --- 添加所有文件到暂存区 git add \*.html --- 添加某种后缀名的所有文件，以html文件为例

git commit -m [tip] --- 提交到当前分支

git remote origin master --- 将本地仓库关联到远程仓库  
origin 为远程仓库的默认名称

git push -u origin master --- 将当前分支推送到远程仓库的master分支

\* ## 版本撤销与文件删除恢复

git log --- 历史版本列表

git status --- 查看工作区状态

git diff --- 查看工作区与版本库(暂存区和master)有无区别

git log --- 提交记录(最近提交的三条记录)

git reflog --- 执行命令记录(提交的历史操作记录)

git reset --- 版本回退 (版本回退)

git reset HEAD [] --- 撤销暂存区修改

git rm [name] --- 删除文件

1.如果你用的rm删除文件，那就相当于只删除了工作区的文件，如果想要恢复，直接用git checkout -- 就可以  
2.如果你用的是git rm删除文件，那就相当于不仅删除了文件，而且还添加到了暂存区，需要先git reset HEAD ，然后再git checkout -- 3.如果你想彻底把版本库的删除掉，先git rm，再git commit 就ok了

\* ## 分支管理

git checkout -- [name] --- 撤销工作区修改

git checkout [name] --- 切换分支

git checkout -b [name] --- 创建并切换分支

git branch --- 列出所有分支

git branch -d [name] --- 删除分支

git branch [name] --- 创建分支

git merge [name] --- 合并分支

\* ## 其它命令

git stash --- 保存当前状态

git stash --- 查看状态列表

git stash drop --- 删除保存的状态

git stash pop --- 恢复工作区并删除保存的状态

git pull ---拉取