

【GitHub】相關

- [【Git】Github 新增專案檔案](#)
- [【Git】解決github項目體積過大的問題](#)

【Git】Github 新增專案檔案

create a new repository on the command line

```
echo "# sample_proj" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/treemanou/sample_proj.git
git push -u origin main
```

push an existing repository from the command line

```
git remote add origin https://github.com/treemanou/sample_proj.git
git branch -M main
git push -u origin main
```

【Git】解決github項目體積過大的問題

首先查找出大文件

```
$ git rev-list --objects --all | grep "$(git verify-pack -v .git/objects/pack/*.idx | sort -k 3 -n | tail -5 | awk '{print$1}')
```

“

- objects：列出該提交涉及的所有文件ID。
- all：所有分支的提交，相當於指定了位於/refs下的所有引用。
- verify-pack命令用於顯示已打包的內容，我們用它來找到那些大文件。
- v (verbose) 參數是打印詳細信息。

前面我們通過rev-list得到了文件名-ID的對應關係，通過verify-pack得到了最大的5個文件ID。用後者篩選前者便能得到最大的5個文件的文件名，比如：

```
$ git rev-list --objects --all | grep "$(git verify-pack -v .git/objects/pack/*.idx | sort -k 3 -n | tail -5 | awk '{print$1}')"
b96695fd6e62fff41723ab324368da394f3a504e package-lock.json 76ec4d5d5fd79da922b322d53fbd1298dc128dd3
node_modules/acorn/dist/acorn.mjs.map 286dc92a240dac765d81dfbda451fd19f9611e84
node_modules/ajv/dist/ajv.bundle.js d2cc86eb230313af9bbdbd3faed7f8e7c7247788 node_modules/source-map-
support/node_modules/source-map/dist/source-map.min.js.map aad0620d70e16717ec338fc1d332279739e0d97c
node_modules/terser/node_modules/source-map/dist/source-map.debug.js
```

然後我們需要刪除他們

```
$ git filter-branch --force --prune-empty --index-filter 'git rm -rf --cached --ignore-unmatch YOU-FILE-NAME' --tag-name-filter cat -- --all
```

“ filter-branch命令可以用來重寫Git倉庫中的提交

- index-filter參數用來指定一條Bash命令，然後Git會檢出 (checkout) 所有的提交，執行該命令，然後重新提交。
- all參數表示我們需要重寫所有分支 (或引用)。
- YOU-FILE-NAME 你查找出來的大文件名

重複幾次，直到大文件全部刪除完畢。

如果你確定某一個文件夾下面都不是你需要的，那麼你可以直接刪除整個文件夾，比如：

```
$ git filter-branch --force --prune-empty --index-filter 'git rm -rf --cached --ignore-unmatch node_modules' --tag-name-filter cat -- --all
```

完成後，以強制覆蓋的方式推送你的repo, 命令如下：

```
$ git push --force --all
```

最後，雖然上面我們已經刪除了文件，但是我們的repo裡面仍然保留了這些objects，等待垃圾回收(GC)，所以我們要用命令徹底清除它，並收回空間，命令如下：

```
$ rm -rf .git/refs/original/
$ git reflog expire --expire=now --all
$ git gc --prune=now
```

```
Enumerating objects: 116, done.
Counting objects: 100% (116/116), done.
Delta compression using up to 4 threads Compressing objects: 100% (53/53), done. Writing objects: 100% (116/116), done.
Total 116 (delta 55), reused 116 (delta 55)
```

