

GitHub リポジトリに大量のファイルを一括登録する方法

著者：関勝寿 公開日：2016年6月3日 ソース：<https://sekika.github.io/2016/06/03/github-many-files/>

GitHub で1ヶ月7ドルの **Personal Plan** でプライベートリポジトリ作成無制限となったので、いろいろなファイルをバックアップ用にプライベートリポジトリに入れることとした。その時に、はまったところがあるので、はまるポイントを回避しつつ、大量のファイルをリポジトリに一括登録する方法をまとめた。

1. 100MB 以上のファイルと追加料金

GitHub では、100MB 以上のファイルをリポジトリに push しようとするとエラーとなる。**Git Large File Storage (LFS)** を使うと、100MB 以上のファイルを扱えるようになる。ここで、Personal Plan では 100MB 以下のファイルについては容量無制限だが、100MB 以上の LFS 管理のファイルについては 1GB までの容量制限があり、オーバーすると LFS の容量を追加購入しなければ LFS が使えなくなる。LFS は容量 50GB と帯域 50GB / month を \$5/month で購入可能である。

そこで、100MB 以下のファイルだけを git リポジトリに入れる、という使い方と、LFS で全てのファイル（あるいは一部の大容量ファイル）を入れる、という使い方が考えられる。ここでは、両方の使い方に対応して記述する。

2. リポジトリの作成

GitHub にサインインして New repository ボタンからリポジトリを作成する。通常はリポジトリを初期化して git clone から始めるが、ここではすでに手元にリポジトリに登録しようとするファイルがあるという状況の話をしているので、Initialize this repository with a README にはチェックを入れずにリポジトリを作成する。まず、README.md が存在しなければ、仮のファイルを

```
echo "# test" >> README.md
```

として作成する。続いて、次の一連のコマンドでリポジトリを初期化し、README.md をリポジトリに追加し、初期設定をする（USER と REP は書き変える）。

```
git init
git add README.md
git commit -m "First commit"
git remote add origin git@github.com:USER/REP.git
git push -u origin master
```

3. ファイル名からスペースを取り除く

ファイル名にスペースが入っていると色々と面倒なので、このコマンドで、カレントディレクトリ以下のファイル名について、スペースをアンダースコア「_」に一括置換してしまうのが良いと思う（それによって何らかの問題が生じないのであれば）。

```
for A in $(find . | grep " " | sed -e s/" "/x3Exe/g) ; do mv "$(echo $A | sed -e s/x3Exe/'_' /g)" "$(echo $A | sed -e s/x3Exe/'_' /g)"; done
```

ディレクトリ名にもスペースがある場合には、エラーが出ることもある。その時には、エラーが出なくなるまで何回か同じコマンドを繰り返す。

4. LFS 管理をするファイルの選択

LFS を使わないのであればここはとばす。LFS を使う場合、まずは、LFS をインストールする。Homebrew であれば

```
brew install git-lfs
```

である。そして、Git に LFS をセットアップする。

```
git lfs install
```

次に、100MB 以上のファイルを一覧する。

```
find . -size +100M | xargs du -sh
```

この中から、LFS 管理とするファイルを適宜指定する。例えば、拡張子 .psd のファイルをまとめて指定するには

```
git lfs track "*.psd"
```

5. git で管理しないファイルの設定

LFS を使わない場合、あるいは LFS を一部のファイルのみにしか使わない場合には、git で管理をしないファイルを .gitignore に設定する。100MB 以上のファイルをすべて .gitignore ファイルに加えるには

```
find . -size +100M | sed -e 's/^\.\.\/' >> .gitignore
```

6. バッファサイズの設定

大きいファイルを git push するとといったエラーが出やすいので、エラーを出にくくするために HTTP post バッファサイズを上げると良い。50MB に上げるには

```
git config http.postBuffer 52428800
```

7. リポジトリに追加

git add -A; git commit; git push で追加できれば良いのだけれど、何ギガバイトもあるような大量のファイルをリポジトリに追加しようとすると、バッファサイズを上げても fatal: The remote end hung up unexpectedly のエラーが出ることもある。そこで、段階的にファイルをリポジトリに追加するスクリプト gitadd を作成した。

[Gist: gitadd](#)

```
#!/bin/sh
#
# gitadd - safe run of (git add .; git commit; git push)
#
# Usage: gitadd [commit message]
#
# When adding too many files to github, error may arise such as
# remote: fatal: pack exceeds maximum allowed size
# or
# fatal: The remote end hung up unexpectedly
#
# This command, gitadd, avoids such error by committing less than
# 100MB data
# for each step.
#
# Increasing post buffer size by the following command is also
# recommended.
#
# git config http.postBuffer 52428800
```

```
#
# Related article: Uploading many files to GitHub repository
# English:
http://sekika.github.io/2016/06/06/github-many-files/
# Japanese:
http://sekika.github.io/2016/06/03/github-many-files/

# Log file
AddLog="/dev/null"
CommitLog="/dev/null"
PushLog="/dev/null"

# Commit message
message=$@
if [ -z "$message" ]; then
    message="commit by gitadd command"
fi

# Commit step by step
while read a b c
do
    total=`find . -type f -size +$a -size -$b | grep -v " ^\./\.\.git/" | wc
-l | sed -e 's/ //g'`
    if [ $total -gt "0" ]; then
        echo "$total Files < $b          "
        fi
        find . -type f -size +$a -size -$b | grep -v " ^\./\.\.git/" | cat -n |
while read num file
do
    echo "Adding: "`expr $num \* 100 / $total`"%" ($num/$total)\r\c"
    git add "$file" 1>>$AddLog 2>>$AddLog
    if [ `echo $num | grep " $c" ` ]; then
        echo "Committing $num          \r\c"
        git commit -m "$message" 1>>$CommitLog 2>>$CommitLog; git push
1>>$PushLog 2>>$PushLog
        fi
    done
    if [ $total -gt "0" ]; then
        echo "Last commit of this stage          \r\c"
        git commit -m "$message" 1>>$CommitLog 2>>$CommitLog; git push
1>>$PushLog 2>>$PushLog
        fi
done << _LIST_
0 8k 0000$
8k 80k 000$
80k 800k 00$
800k 8M 0$
8M 100M $
_LIST_

# Commit all the files
#
# Basically all the files < 100MB has been committed, and files
larger
# than 100MB are left, which should be handled with LFS in GitHub.
# If you prefer not managing files of this size, the files should be
# specified in .gitignore
#
echo "All files          "
git add . 1>>$AddLog 2>>$AddLog
git commit -m "$message" 1>>$CommitLog 2>>$CommitLog; git push
1>>$PushLog 2>>$PushLog

echo "Finished"
```

最初に git add -A; git commit; git push をやってエラーが出ててしまった時には git reset HEAD~ で commit と index を戻してから、このスクリプト gitadd を実行する。

8.100MB 以上のファイルがあつてエラーとなる場合

Filename というファイルが 100MB 以上で、File Filename is

230.01 MB; this exceeds GitHub's file size limit of 100.00 MB というようなエラーが出て git push できないとき、そのファイルを git 管理から除き、履歴から完全に削除する。

```
git rm --cached Filename
git commit --amend -CHEAD
git filter-branch --index-filter 'git rm --cached
--ignore-unmatch Filename' HEAD
git push origin master
```

git filter-branch --index-filter 'git rm --cached --ignore-unmatch Filename' HEAD に対して A previous backup already exists in refs/original/ のようなメッセージが出た時には、

```
git update-ref -d refs/original/refs/heads/master
```

で refs/original/ を消す。

必要に応じて LFS 管理に入れてコミットし直す。

```
git lfs track Filename
git add Filename
git commit -m lfs
git push origin master
```

9. 管理

```
git pull; git add -A; git commit -m "commit" && git push
```

とすればアップデートされていくので、お好みで crontab に入れる。