

Lokale Git Ordner in einem Rutsch auf den neuesten Stand bringen

Beitrag von „ozw00d“ vom 3. Dezember 2021, 11:11

Moinsen,

da ich immer genervt bin wenn ich etwas mehr als einmal durchführen muss habe ich mir überlegt, wie man am mac oder hack alle lokalen Git Ordner aktualisieren kann.

Da ich aktuell auf macOS Monterey bin, bezieht sich diese Anleitung nur auf dieses System. Es funktioniert natürlich auch unter allen anderen macOS Versionen welche zsh nutzen.

Bash sollte ebenso funktionieren, bei entsprechender Anpassung der .bashrc.

Zuerst sollte man einen Platz für alle Gits haben:

dazu gibt man im terminal folgendes ein, bitte beachtet vorab in den Ordner zu wechseln wo ihr den Ordner anlegen wollt:

Code

1. mkdir git

nun mit:

Code

1. cd git

ins Verzeichnis wechseln.

Nun laden wir uns die entsprechenden Repos, wie das geht findet man im netz, ein kleines beispiel:

Code

1. `git clone https://github.com/acidanthera/OpenCorePkg.git`

womit dann ein entsprechendes git Verzeichnis lokal abgelegt wird.

in meinem Fall befinden sich bereits einige gits im lokalen Pfad:

```
0zzy@FLUX > ls /Volumes/WORK/git
ls
EasyPeasy/
GenSMBIOS/
NetCease/
New-KrbtgtKeys.ps1/
OCConfigCompare/
OcBinaryData/
OpenCanopyIconPacks/
OpenCanopyIcons/
OpenCorePkg/
PowerSploit/
Pwned/
SSDTTime/
USBMap/
gibMacOS/

mslearn-ha-application-storage-with-grs/
sd-card-speed-test-master/
.DS_Store
msrsync
```

in der theorie, müsste man nun in jedes Verzeichnis wechseln und dort ein:

Code

1. `git pull origin master`

ausführen.

Da ich einfach zu lazy bin dies ständig zu wiederholen, nutze ich folgenden Code um mir einen alias im terminal zu erstellen:

Code

1. `alias gitpull='find . -type d -depth 1 -exec git --git-dir={}/.git --work-tree=$PWD/{ } pull origin master \;'`

in der .zshrc schaut das bei mir (ich habe da eine eigene Ordnung drin, wie ihr das im Endeffekt in der Einstellungsdatei sortiert überlasse ich euch 😊) so aus:

```
alias gitpull='find . -type d -depth 1 -exec git --git-dir={}/.git --work-tree=$PWD/{ } pull origin master \;'
```

Mit

Code

1. `source ~/.zshrc`

laden wir den alias.

wenn wir nun in das Verzeichnis wechseln, wo unsere Gits liegen (mit `cd <Pfad/zum/gitdir>`),
und

Code

1. `gitpull`

im Terminal ausführen, lädt euch git alle updates der repos herunter, was dann (leider hatte ich das schon durchgeführt) so aussieht:

```
> gitpull
From https://github.com/corpnewt/USBMap
 * branch      master      -> FETCH_HEAD
Already up to date.
From https://github.com/microsoft/New-KrbtgtKeys.ps1
 * branch      master      -> FETCH_HEAD
Already up to date.
From https://github.com/sameera-madushan/Pwned
 * branch      master      -> FETCH_HEAD
Already up to date.
From https://github.com/corpnewt/OCConfigCompare
 * branch      master      -> FETCH_HEAD
Already up to date.
From https://github.com/p0w3rsh3ll/NetCease
 * branch      master      -> FETCH_HEAD
Already up to date.
fatal: not a git repository: './sd-card-speed-test-master/.git'
fatal: not a git repository: './item24/.git'
From https://github.com/PowerShellMafia/PowerSploit
 * branch      master      -> FETCH_HEAD
Already up to date.
```