

# How to backup/sync all of your dotfiles with Github

---

**DEV** [dev.to/jeffshomali/how-to-backup-sync-all-of-your-dotfiles-with-github-e1c](https://dev.to/jeffshomali/how-to-backup-sync-all-of-your-dotfiles-with-github-e1c)

Jeff Shomali

As a developer, one of the challenges that we are facing every day is to backup or sync up all of our `dot(.)` files with a new or another environment. There are many dotfiles in my machine such as `.alias` , `.vimrc` , `.zshrc` , `.gitconfig` etc...

I'm not going to explain how to get a back up of all your files because each one of us has a different environment, so I'll explain how to backup your VS-Code setting. I know there is a VS-Code extension that syncs up your setting with Github. Our goal is to understand how to back up #vscode settings, snippets, extensions, and sync it with Github manually.

## Create a Script

---

- Create a new Github repository in your Github.com
- In your local machine go to the desktop and open an empty folder in your terminal
- Initiate an empty Git repository with `$ git init` and connect the local repo to newly created repo with `$ git remote add git@github.com:yourusername/yourrepo.git`
- Create a shell script and name it whatever you want, like `backup.sh` and past the following code on that file.

```
#!/bin/bash
```

```
# check to see if git command line is installed in this machine
```

```
IS_GIT_AVAILABLE="$(git --version)"
```

```
if [[ $IS_GIT_AVAILABLE == *"version"* ]]; then
```

```
    echo "Git is Available"
```

```
else
```

```
    echo "Git is not installed"
```

```
    exit 1
```

```
fi
```

```
# copy Vs-Code files
```

```
cp $HOME/Library/Application\
```

```
Support/Code/User/{keybindings.json,settings.json,spellright.dict} .
```

```
# copy snippets folder
```

```
cp -r $HOME/Library/Application\ Support/Code/User/snippets .
```

```
# copy list of extensions that are currently installed
```

```
code --list-extensions --show-versions >> ../../vscode/extensions.txt
```

```
# copy other dot files
```

```
cp $HOME/{.zshrc,.vimrc} .
```

```
# Check git status
```

```
gs="$(git status | grep -i "modified")"
```

```
# echo "${gs}"
```

```
# If there is a new change
```

```
if [[ $gs == *"modified"* ]]; then
```

```
    echo "push"
```

```
fi
```

```
# push to Github
```

```
git add -u;
```

```
git commit -m "New backup `date +%Y-%m-%d %H:%M:%S`";
```

```
git push origin master
```

- Change the script permission with `$ chmod +x backup.sh` .
- Run the script with `$ ./backup.sh`

If you want to run in background services:

`crontab -e` and add your script path and time you want to run the script for example

```
$ crontab -e # this is open the cronjobs table in vim mode use i for insert mode
```

```
# this is run the script every minutes
```

```
$ 1 * * * * cd /Users/macp15/Projects/dotfiles/scripts/backup && ./backup.sh
```

```
# display list of cron jobs
```

```
$ crontab -l
```

Done (Khalas).