

Beste de savoir

Des petits outils sympa

16 avril 2022

Table des matières

	Introduction	1
1.	modd	1
2.	wait-for-it	2
3.	ngrok	2
4.	pandoc	3
	Conclusion	3

Introduction

Le développement est une activité complexe. Pour se faciliter la vie il existe une multitude d'outils. Ici nous allons voir 4 outils que peuvent vous aider.



Le but n'est pas de faire une explication complète sur ces outils mais de les présenter succinctement.

1. modd

`modd` est un outil de *hot reload*.

Le *hot reload* est une fonctionnalité qui permet de recharger une application à chaud. C'est une fonctionnalité très présente dans le développement web (avec `npm`), récemment ajouté à la commande `dotnet` (pour le développement C# par exemple), et aussi lors du développement d'applications Flutter.

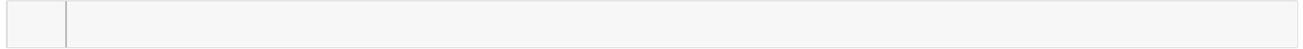
Les avantages de `modd` sont multiples :

- Il est indépendant de la technologie utilisée. Il peut donc être utilisé de la même manière que vous faisiez du Python, du Go, ...
- On choisit les fichiers traqués. Les règles des fichiers observés par l'outil sont à définir et peuvent être très complexes au besoin.
- On personnalise la commande qui est exécutée. Lorsqu'un fichier traqué est modifié, la commande exécutée est définie par l'utilisateur.

Cet outil à un principe de fonctionnement très simple. Il s'utilise avec un fichier de configuration (dans lequel est défini les fichiers à observer et les commandes à exécuter) et la commande `modd`.

Voici un exemple de fichier de configuration :

2. wait-for-it



Ce fichier de configuration observe tous les fichiers ayant pour extension `.go`. Lorsqu'un fichier de ce type est modifié sur le disque, la commande `go build -o bin .` est exécuté.

La configuration de cet outil peut être bien plus complexe. Vous trouverez la documentation complète sur [le dépôt GitHub de modd](#) .

2. wait-for-it

`wait-for-it` est un outil qui permet d'attendre avant l'exécution d'une commande. Il attend la réponse à une connexion TCP.

En informatique il est peu commun de vouloir attendre. Cependant cela peut être utile lors du développement d'applications. Prenons un exemple.

Lorsque vous développez une application avec une base de données, il faut que la base de données soit prête à accepter des connexions. Sans quoi votre application ne peut s'y connecter et ne démarre pas.

`wait-for-it` est un script bash que vous trouverez sur [ce dépôt GitHub](#) .

Une fois le script téléchargé, il s'exécute de la manière suivante

```
1 ./wait-for-it.sh zestedesavoir.com:443 -- echo "ZdS is up"
```

Le script attend une réponse de `zestedesavoir.com` sur le port `443` avant d'exécuter la commande `echo "ZdS is up"`.

La commande sera exécutée même si `zestedesavoir.com` ne répond pas, car un `timeout` sera retourné par le protocole TCP. Pour pallier cela on peut ajouter l'argument `--strict`.

```
1 ./wait-for-it.sh googel.fr:443 --strict -- echo "Googel is up"
```

La commande ci-dessus n'exécutera pas l'instruction qu'on lui demande car `googel.fr` ne donnera pas de réponse et que le mode strict est activé.

3. ngrok

Ngrok permet d'exposer publiquement une application serveur en local sans configuration et en toute sécurité.

Cet outil crée un tunnel entre votre machine et les serveurs de ngrok. Il redirige ensuite ce qui est envoyé sur l'URL publique vers votre machine. Pour comprendre le fonctionnement de cet outil en détaille je vous redirige vers son [site web](#) .

4. *pandoc*

L'utilisation de ngrok est très simple.

- Il faut télécharger l'application (pour cela il faut se connecter)
- Utiliser la commande `ngrok` sous la forme `ngrok <protocole> <port>` (ex : `ngrok http 8080`)
- Partager le lien public donné

4. **pandoc**

Pandoc est un convertisseur de document.

Il est très pratique pour par exemple écrire du contenu en markdown pour ensuite l'exporter au format PDF, HTML, DOCX, ...

Dans un cas comme celui-ci il permet de se concentrer sur le contenu plutôt que la forme. Pour réaliser la mise en page voulue il est possible de donner des arguments à la ligne de commande afin de configurer en fonction du choix voulu.

Son usage se fait très simplement en ligne de commande :

```
1 pandoc soutenance_fin_etude.md -s -o soutenance_fin_etude.pdf
```

Ici, `soutenance_fin_etude.md` est mon document source, ma soutenance de fin d'études écrite en markdown. Je convertis ce fichier au format PDF avec l'argument `-o soutenance_fin_etude.pdf`. L'argument `-s` permet de préciser que le document produit (celui au format PDF) doit se suffire à lui-même (donc, par exemple, inclure si besoins les images).

C'est un outil assez répandu qui possède de multiples options en fonction de la cible de la conversions. Pour avoir plus d'informations, je vous redirige vers son [site web](#) .

Conclusion

Ces outils peuvent s'intégrer à votre manière de travaillé de la manière suivante par exemple :

- Développement de votre app avec `modd` pour profiter du *hot reload*
- Démarrage des applications dans l'ordre avec `wait-for-it` pour tester en local
- Test de l'application à distance avec `ngrok`
- Ecriture de documents liée à l'application avec `pandoc`

Les outils présentés ici sont adaptés à un usage en local pour du développement. Pour un usage en production, il existe d'autres outils plus complexes et bien mieux adapté.