

StatusEr, Outil de monitoring simple

Mansouri Yanis - 21003060¹

Intentions. Ce projet se veut utile bien que destiné au cours de Méthodologie de la programmation. Il est nécessaire de noter que le but principal de cet outil est le suivi d'état d'un parc informatique hors-ligne. En effet, l'interface de cet outil n'a pas vocation (et n'est pas faite) pour être exposée. Il m'a été nécessaire de me documenter sur le module Flask ainsi que Turbo puis de réfléchir à la conception de l'application afin qu'elle soit la plus optimisée et facilement maintenable

1 Introduction

La problématique principale du projet était de permettre le suivi du status d'adresses cibles en étant à la fois rapide, afin de ne pas surcharger le réseau et explicite afin qu'au premier coup d'oeil l'utilisateur ait une vision large de la situation.

Dans un but de maximiser les performances de l'outil, il n'a pas été jugé nécessaire de recréer certains outils déjà existants en python. Notamment : le module de ping (pythonping) et de modification de l'affichage (turbo).

Le choix de Flask s'est fait parce qu'il permet de mettre en place un serveur web avec le minimum, ce qui, dans notre quête de performance était un choix sans équivoque.

2 Fonctionnement

La page s'actualise (ce qui induit les données également) toutes les 5 secondes.

La liste des cibles est stockée dans une base de données interfacée à Python par sqlite, puis cette même liste est utilisée pour construire une liste de status avec les informations :

- Surnom associé à l'adresse
- La disponibilité (sous forme d'un booléen)
- Le temps de réponse moyen

Cette actualisation est initiée et tourne sur un thread parallèle à celui de l'application Flask afin de permettre l'ajout / retrait d'une adresse sans se heurter aux limitations d'une pile simple.

3 Modifications possibles

Bien que le fonctionnement pur de l'application soit invariable, il est possible de changer l'intervalle d'actualisation des données (directement dans le code).

Au delà de ça, changer la forme de la base de données et/ou l'affichage induira forcément des changements plus importants voir la nécessité de repenser le code.

¹yanis.mansouri02@etud.univ-paris8.fr

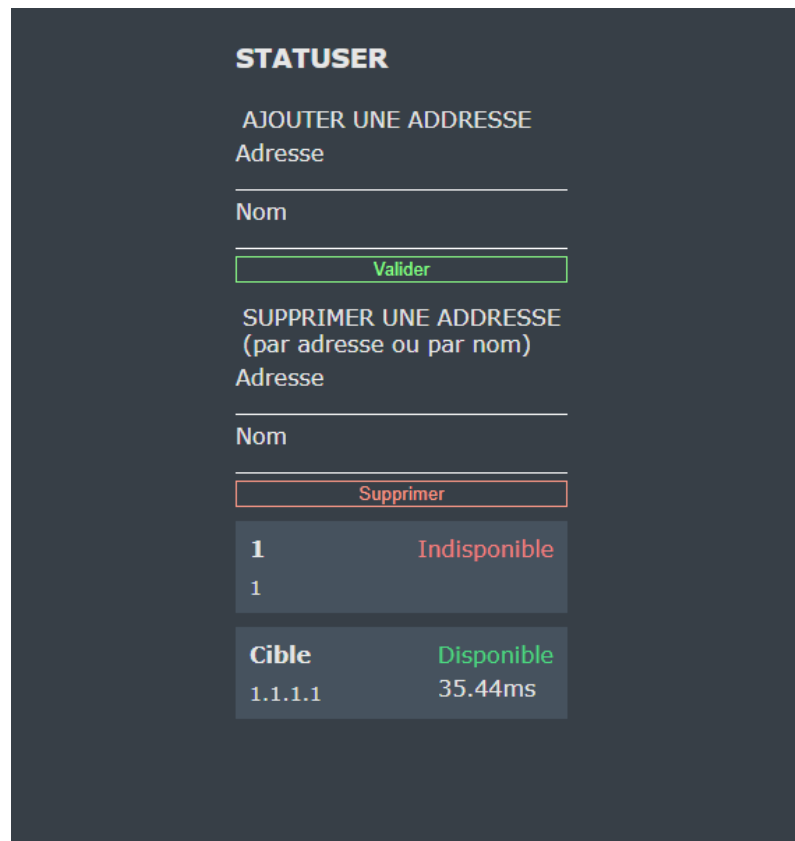


FIGURE 1 – Aperçu de l'application

4 Installation et mise en route

4.1 Installation des dépendances

Le projet nécessite l'installation de quelques dépendances, ceci peut-être fait en se plaçant dans le dossier du projet et en effectuant la commande :

```
pip install -r requirements.txt
```

4.2 Mise en route

Après l'installation des dépendances l'application se met en route en lançant simplement le fichier principal :

```
main.py
```