

pre-TP2 – Service réseau

TP Cloud #2 / M2 CCI – JpGelas/UCB Lyon 1

PARTIE 1

Rappel: Le port d'écoute par défaut d'un serveur web est 80 (http) et/ou 443 (https).

Amusant: Remplacer la page d'accueil par défaut de votre serveur web Apache.

Cette page se trouve dans le fichier `/var/www/html/index.html`

Attention : Changer le contenu d'une page web sans autorisation s'appelle faire du "Defacing" jeune pirate !

Voici un exemple de contenu d'une page web simple. Remplacez le contenu du fichier `index.html` par ces lignes.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Pirate</title>
  <style type="text/css" media="screen">
    body, html {
      background-color: #D8DBE2;
      font-family: Verdana, sans-serif;
      font-size: 11pt;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>Pwned!</h1>
</body>
</html>
```

Facile : Avec la commande `wget` vous pouvez récupérer le contenu d'une page web (le fichier html) en ligne de commande. Récupérez la page `index.html` de votre serveur web Apache à partir de votre poste client.

```
wget adresseIPdevotreVM
```

Qu'est ce que vous obtenez ?

Bizarre : La commande `nc` permet (entre autre) d'établir une connexion simple à une machine distante via son adresse et un numéro de port.

```
nc adresseIPdevotreVM 80 (appuyez deux fois sur Entrée)
```

Qu'est ce que vous obtenez ?

Vous obtenez une erreur car on ne respecte pas le protocole `http`.
Maintenant lancez la commande ci-dessous.

```
echo -e "GET /index.html HTTP/1.0\r\n\r\n" | nc adresseIPdevotreVM 80
```

Cette commande permet d'envoyer une chaîne de caractères à la machine distante.
La chaîne de caractères donné ci-dessus respecte le protocole `http`.

Qu'est ce que vous obtenez ?

Quelle est la différence avec *wget* ?

Vous venez de comprendre que vous n'avez pas forcément besoin d'un navigateur et d'une interface graphique pour communiquer avec un serveur web. Il suffit de connaître le protocole de communication et de le respecter.

PARTIE 2

A présent nous allons créer notre propre serveur. Ce serveur respectera le protocole *http* et servira un petit message sympathique et probablement inutile :-) (au lieu de page web .html).

On pourrait le faire en Python ? Et pourquoi pas !

Voici le code source minimaliste d'un serveur Web en Python.

Ce source est à créer sur votre serveur dans le fichier ***server.py***

```
from http.server import SimpleHTTPRequestHandler
import socketserver
from http import HTTPStatus

class Handler(SimpleHTTPRequestHandler):
    def do_GET(self):
        self.send_response(HTTPStatus.OK)
        self.end_headers()
        self.wfile.write(b'Hello world')

httpd = socketserver.TCPServer(('', 8080), Handler)
httpd.serve_forever()
```

Pour l'exécuter saisissez la commande suivante.

```
python3 server.py
```

Vous voulez le tester avec un navigateur ? Vous pouvez par contre ne vous attendre pas à quelque chose d'épique et pensez à ajouter le numéro de port 8080 dans le champ URL.

adresseIPdevotreVM:8080

Sur votre VM il y a à présent deux serveur Web qui tourne.

Un vrai (Apache sur le port 80) et, hum... le votre sur le port 8080 ;-)

Une alternative à *wget* est la commande *curl* que je préfère personnellement.

```
curl adresseIPdevotreVM:8080
```

Si je tape : `curl adresseIPdevotreVM:8080/nimportenawak`

Cela fonctionne encore. Pourquoi ?

A présent, remplacez la chaîne 'Hello World' par '<h1>Hello Big World</h1>'

Relancez *server.py* et requêtez votre serveur minimaliste avec *curl* puis avec votre navigateur.

Pourquoi l'affichage est différent ?

PARTIE 3 - Quelques API Web

A présent nous allons faire des requêtes sur des serveurs disponibles sur Internet.

Exemple 1.1: `curl http://teredumili.eu/ip/`

Ce serveur vous renvoie l'adresse IP de votre machine (en tous cas l'adresse publique sous laquelle vous apparaissez sur internet).

Facultatif : Si vous souhaitez plus d'informations sur votre IP saisissez la sur un site comme <https://iplocation.com/> (ça fait peur hein !)

Exemple 1.2: `curl https://www.affirmations.dev/`

Renvoie des « affirmations » au format JSON. Essayez plusieurs fois. Oui on trouve des services étrange sur Internet.

Nous allons faire une requête cette fois-ci sur des serveurs qui accepte un paramètre et répondront en conséquence.

Exemple 2.1:

`curl http://wttr.in/1yon`

Ici le paramètre est le nom d'une ville. La résultat de la requête est la météo d'une ville avec un affichage tout en Ascii-Art. La classe ! Essayez avec d'autres noms de ville.

Exemple 2.1:

`curl http://numbersapi.com/42`

Renvoie une ligne de texte donnant une information sur un nombre passé en paramètre. Essayez plusieurs fois et avec d'autres nombres.

Enfin voici un dernier exemple avec une API un peu plus riche mais bon ok, pas franchement utile...

Exemple 3: Chuck Norris API

`curl https://api.chucknorris.io/jokes/random`

Renvoie au hasard une phrase « célèbre » dite par Chuck Norris.

`curl https://api.chucknorris.io/jokes/categories`

Renvoie la liste de toutes les catégories de ses citations.

`curl https://api.chucknorris.io/jokes/random?category=food`

Renvoie une phrase célèbre de Chuck Norris issue d'une catégorie particulière.

Remarque: Dans un navigateur les *backslash* ne sont pas nécessaire.

Conclusion

Il existe des milliers d'API web disponibles sur Internet. Beaucoup sont gratuites et probablement plus utile que celles présentée ici en exemple. A présent vous avez une idée de comment créer un service web simple. Pour exposer votre propre service web pas besoin d'un (gros) serveur web Apache !

Vous pouvez aussi imaginez que votre prochaine application tirera profit d'une ou plusieurs API disponibles sur le web. Remarquez que beaucoup renvoient une réponse au format JSON. Pas de soucis ! Ce format est très simple à exploiter avec la bibliothèque adaptée.

Pour vous inspirer, voici des pages qui recensent de nombreuses API qui ne demande qu'à être requêtée par votre futur applications.

<https://medium.com/better-programming/a-curated-list-of-100-cool-and-fun-public-apis-to-inspire-your-next-project-7600ce3e9b3>

<https://medium.com/@vicbergquist/18-fun-apis-for-your-next-project-8008841c7be9>

<https://any-api.com/>

<https://github.com/public-apis/public-apis>

<https://mixedanalytics.com/blog/list-actually-free-open-no-auth-needed-apis/>