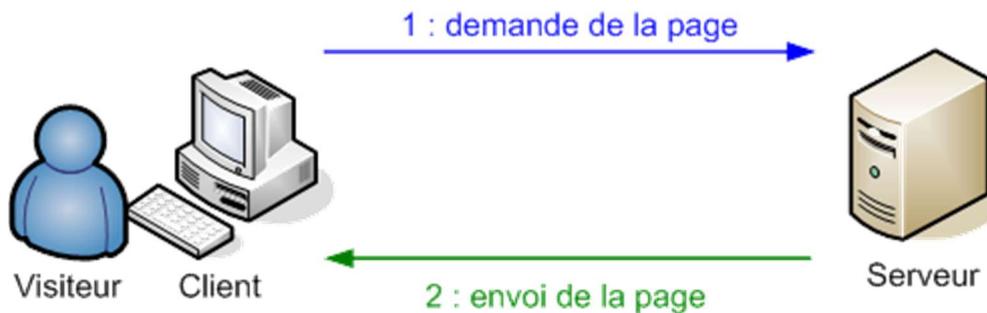


On utilisera le logiciel curl, normalement installé par défaut sous windows 10. Il faudra le télécharger et l'installer si ce n'est pas le cas.

Le logiciel curl va permettre de de faire des requêtes http.

1 – Présentation du protocole HTTP

Le Hypertext Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web. Il est utilisé pour échanger toute sorte de données entre client HTTP et serveur HTTP.



Principales méthodes de requête client :

GET

C'est la méthode la plus courante pour demander une ressource. Une requête **GET** est sans effet sur la ressource, il doit être possible de répéter la requête sans effet.

HEAD

Cette méthode ne demande que des informations sur la ressource, sans demander la ressource elle-même.

POST

Cette méthode doit être utilisée lorsqu'une requête modifie la ressource.

OPTIONS

Cette méthode permet d'obtenir les options de communication d'une ressource ou du serveur en général.

CONNECT

Cette méthode permet d'utiliser un proxy comme un tunnel de communication.

TRACE

Cette méthode demande au serveur de retourner ce qu'il a reçu, dans le but de tester et d'effectuer un diagnostic sur la connexion.

PUT

Cette méthode permet d'ajouter une ressource sur le serveur.

DELETE

Cette méthode permet de supprimer une ressource du serveur.

Les plus courantes : **HEAD**, **GET** et **POST**.

Exemple 1 :

⇒ Sous l'invite de commande, taper la ligne suivante :

Sans proxy :

```
curl "http://mede.fr/tpweb/p81.html"
```

Avec proxy :

```
curl -x 172.16.0.253:3128 "http:// mede.fr/tpweb /p81.html"
```

La réponse du serveur doit être la page HTML

Sous WireShark :

172.16.6.19	212.227.247.63	HTTP	139 GET /tpweb/p81.html HTTP/1.1
212.227.247.63	172.16.6.19	HTTP	455 HTTP/1.1 200 OK (text/html)

On retrouve dans la requête :

- L'hôte : mede.fr
- L'URI : tpweb/p81.html
- La méthode (GET par défaut)
- La version http (1.1).

```
▼ Hypertext Transfer Protocol
  ▼ GET /tpweb/p81.html HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /tpweb/p81.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /tpweb/p81.html
      Request Version: HTTP/1.1
      Host: mede.fr\r\n
      User-Agent: curl/7.55.1\r\n
      Accept: */*\r\n
      \r\n
```

Dans la réponse :

- Le code 200 : OK
- La page p81.html

Exemple 2 :

⇒ Sous l'invite de commande, taper la ligne suivante (on met l'option TRACE, qui demande au client de retourner ce qu'il a reçu, l'échange des données est enregistré dans le fichier test.txt) :

Sans proxy :

```
curl --trace test.txt "http://mede.fr/tpweb/p81.html"
```

Avec proxy : ajouter l'option **-x 172.16.0.253:3128**

⇒ Sous l'invite de commande, taper la ligne suivante pour afficher le contenu de test.txt :

```
notepad test.txt
```

Retrouver alors l'échange des données.

Remarque : la commande **curl --help** permet de visualiser l'aide curl.

Exemple 3 :

⇒ Sous l'invite de commande, taper la ligne :

```
curl --head "http://mede.fr/tpweb/p81.html"  
Avec proxy : ajouter l'option -x 172.16.0.253:3128
```

Le serveur ne renvoie que l'entête (sans le corps, le fichier html ici).

2 – Echange de données

La méthode GET transmet les données dans l'entête et la méthode POST dans le corps.

On utilise des fonctions en php pour récupérer les données transmises.

Méthode GET :

⇒ Sous l'invite de commande, taper la ligne (--get n'est pas obligatoire, car c'est la méthode par défaut) :

```
curl --get --trace test.txt "http://mede.fr/tpweb/p82.php?nom=colin&prenom=guy"
```

Avec proxy : ajouter l'option -x 172.16.0.253:3128

Le fichier p82.php est le suivant :

```
<!DOCTYPE html>  
<html>  
  <body>  
  
    <?php  
      echo "Le nom est : " . $_GET["nom"] . "<BR>";  
      echo "Le prenom est : " . $_GET["prenom"] ;  
    ?>  
  </body>  
</html>
```

Méthode POST :

⇒ Sous l'invite de commande, taper la ligne

```
curl --data "nom=colin&prenom=guy" --trace test2.txt "http://mede.fr/tpweb/p83.php"
```

Avec proxy : ajouter l'option -x 172.16.0.253:3128

Le fichier p83.php est le suivant :

```
<!DOCTYPE html>  
<html>  
  <body>  
  
    <?php  
      echo "Le nom est : " . $_POST["nom"] . "<BR>";  
      echo "Le prenom est : " . $_POST["prenom"] ;  
    ?>  
  </body>  
</html>
```

⇒ Comparer les 2 fichiers test.txt et test2.txt. Vérifier les commandes envoyées (GET ou POST) et indiquer dans quelle partie se trouve les données transmises (Entête http ou corps http).

3 – Utilisation du site " https://rbaskets.in"

Le site https://rbaskets.in va permettre de recevoir et afficher les requêtes http.

⇒ Aller sur le site https://rbaskets.in ⇒ Clic sur Create ⇒ Open Basket

⇒ Copier le lien (exemple : <https://rbaskets.in/ut9uixm>)

⇒ Sous l'invite de commande, taper la ligne en remplaçant le lien par celui obtenu précédemment :

```
curl --data "nom=colin&prenom=guy" "https://rbaskets.in/ut9uixm"
```

Avec proxy : ajouter l'option **-x 172.16.0.253:3128**

Si message d'erreur avec revocation : ajouter l'option `--ssl-no-revoke` (pour ne pas valider la révocation du certificat SSL). C'est dû au fait qu'on est en https.

⇒ Sous le site rbaskets.in observer les données reçues, et l'entête de la requête.

Envoi de donnée sous un format JSON

⇒ Sous l'invite de commande, taper la ligne en remplaçant le lien par celui obtenu précédemment :

```
curl --data "{\"nom\":\"colin\",\"prenom\":\"guy\",\"age\":36}" -H "Content-Type: application/json" "https://rbaskets.in/ut9uixm"
```

(L'option `-H` ou `--header` permet de modifier l'entête de la requête)

`\"` correspond à l'envoi du caractère `"`, pour distinguer le séparateur de champ et le caractère.

Options : **-x 172.16.0.253:3128** et `--ssl-no-revoke` à ajouter éventuellement.

⇒ Sous le site rbaskets.in observer les données brutes reçues (RAW), et l'entête de la requête, comme ci-dessous

The image shows a screenshot of a web browser's developer tools. On the left, the 'Headers' tab is active, displaying a list of response headers. The header 'Content-Type: application/json' is circled in red. On the right, the 'Body' tab is active, displaying a JSON object: { "nom": "colin", "prenom": "guy", "age": 36 }.

```
Headers
Accept: */*
Connection: close
Content-Length: 29
Content-Type: application/json
User-Agent: curl/7.55.1
X-City: Fruges
X-Country: FR
X-Forwarded-For: 86.192.211.237
X-Real-Ip: 86.192.211.237

Body
{
  "nom": "colin",
  "prenom": "guy",
  "age": 36
}
```

Envoi d'un fichier

⇒ Sous l'invite de commande, entrer la commande notepad test.json pour éditer et sauvegarder le fichier test.json ci-dessous (faire un copier/coller bien évidemment):

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "landline",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

⇒ Sous l'invite de commande, taper la ligne en remplaçant le lien par celui obtenu précédemment :

```
curl --data @test.json -H "Content-Type: application/json" "https://rbaskets.in/ut9uixm"
@ permet d'indiquer qu'il s'agit d'un fichier.
```

Options : **-x 172.16.0.253:3128** et **--ssl-no-revoke** à ajouter éventuellement.

⇒ Sous le site rbaskets.in observer les données reçues.