WEB EXERCICES

[**Curl**](#_f8dl04crzd8) **3**

[Fonctionnement](#_r0p95cduxti) 3

[Exercice](#_22zd1laahird) 4

[**/etc/hosts**](#_u8lydgjqv97o) **5**

[**Node**](#_e3bl76qqytb8) **6**

[Prérequis, installation](#_3xhz2hwn6nx0) 6

[Exercices](#_d8s8b1r8dm7o) 6

[**Nginx**](#_7jc031qfq4wz) **6**

[**Curl + node + nginx**](#_aus5iq73lte9) **6**

# Curl

cURL est sans doute l’un des utilitaires les plus utilisé au monde pour débugger des services webs, il permet d’envoyer une requête http / https sur un serveur via la console. Il est possible d’envoyer les datas que l’on veut, avec les headers que l’on veut.

## Fonctionnement

On l’installe de la façon suivante :

**sudo npm install curl**

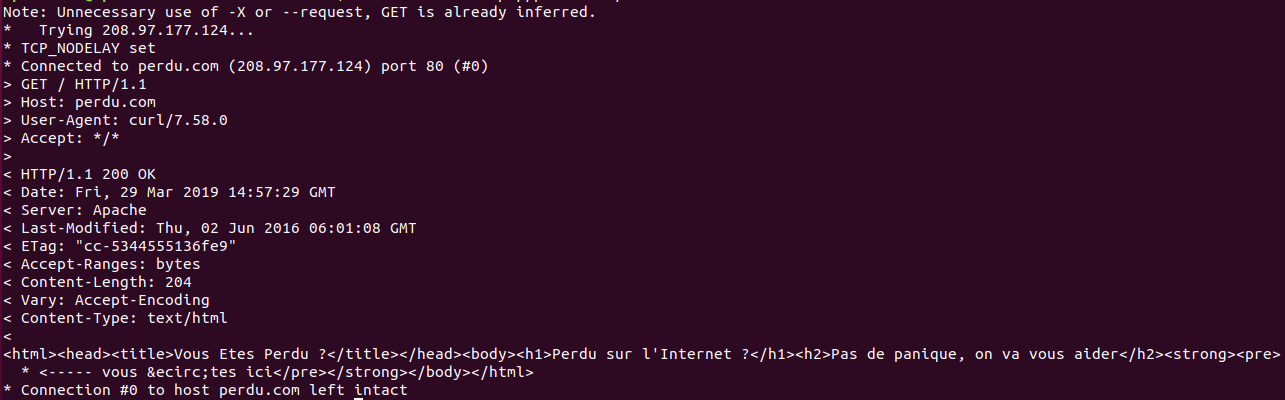
On le lance de la façon suivante :

**curl -X TYPE\_REQUETE “URL” [options]**

* TYPE\_REQUETE: GET / POST / PATCH / PUT / …
* URL : <https://www.lemonde.fr/page1> (contenant le protocol, le domaine et le path)
* Options :
  + **--compressed** : compression et décompression des requêtes
  + **-v** : affiche les informations https
  + **--data** : envoi un formulaire post html, exemple : --data “champs1=1000&champs2=2000”
  + **-H** : header, exemple -H “Cookie: v1=1” -H “Content-type: application/json;charset=UTF-8”
  + **-L** suivre les redirections
  + -**I** effectue une requête head (pas de données, pas de redirection, etc…)

Exemple récupération de la page <http://perdu.com/> avec les informations de connexion :

curl -X GET “<http://perdu.com/>” -v



## Exercice

**Exercice 0.curl.1** : Avec curl, Récupérer le contenu HTML de la page présente à l’adresse suivante : <https://stage.optionizr.com/html>

**Exercice 0.curl.2** : Avec curl, Récupérer la première url de redirection (il y en a plusieurs) de l’adresse <https://stage.optionizr.com/redirections1> vous pouvez utiliser la console chrome pour vous aider à voir le nombre de redirection. Quel est le code HTTP retourné lors de la redirection ?

**Exercice 0.curl.3** : Avec curl, Récupérer le contenu de la page <https://stage.optionizr.com/redirection> en suivant les redirections

**Exercice 0.curl.4** : Avec curl, ajoutez obligatoirement le header *-H "Content-Type: application/x-form-www-urlencoded"*  Envoyez les données form (form html) param1=3 et param2=5 à l’adresse suivante <https://stage.optionizr.com/api/formpost> , cette adresse fait la somme des paramètres et doit vous renvoyer : *<span>Resultat : </span><span class="result">8</span>*

**Exercice 0.curl.5** : Avec curl, ajoutez obligatoirement le header -H *"Content-Type: application/json"* Envoyez un objet json contenant param1: 3 et param2: 5 à l’adresse suivante <https://stage.optionizr.com/api/formjson> cette adresse fait la somme des paramètres et doit vous renvoyer : *{"result":8}*

Vous avez finis ? Bravo vous savez utiliser curl et vous êtes maintenant un expert des requêtes http !

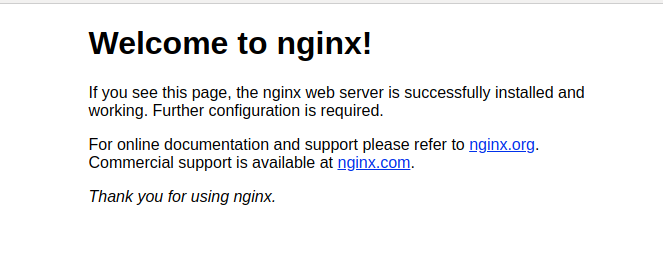
# 

# 

# /etc/hosts

**Exercice 0.host.1** : Récupérez l’adresse IP associée à *optionizr.com* puis essayez de rediriger *jaimelessamossas.com* vers cette adresse ip. Actuellement une requête curl GET sur <http://jaimelessamossas.com/> m’affiche le résutat suivant:



Si tout fonctionne une requête curl ou un accès navigateur doit rediriger sur la page d’accueil de NGINX : 

**Exercice 0.host.2 :** Bloquez l’accès aux urls suivante : <https://api.optionizr.com/health> <https://www.optionizr.com> et <https://tiny.optionizr.com/health> Est-il possible d’utiliser des wildcards ou regex ?

Bravo ! Vous savez maintenant manipulez les résolutions de noms, vous êtes dorénavant un vrai hacker

# 

# Node

## Prérequis, installation

Nous allons installer une simple application nodejs qui permet d’afficher “Bonjour : XXX” ou XXX est le numéro du port sur lequel tourne l’application.

## Exercices

# Nginx

# 

# Curl + node + nginx