



Table des matières

1) Utiliser la méthode REST.....	1
2)MQTT.....	1
3) Serveur REST.....	1

Lien du repo git : <https://github.com/CooKiiieS/TP-REST-LAFORGE>

1) Utiliser la méthode REST

1) Je crée un dépôt git (graphiquement) et dans les paramètres de mon dépôt j'ajoute l'utilisateur SebDruon comme collaborateur (dans manage access)

Mon Git est le suivant : CooKiiieS/TP-REST-LAFORGE

2) Avec mon téléphone je me connecte au réseau wifi de ma prise

Ensuite dans une interface web je tape l'ip de ma prise (192.168.33.1) et j'arrive sur la page de configuration. Sur cette page je peux voir un bouton power pour allumer ou éteindre ma prise et à côté voir la puissance consommée

3) A l'aide de la commande curl je récupère différentes informations :

La consommation électrique actuelle :

```
test@202-15:~/TP-REST-LAFORGE$ curl http://192.168.33.1/meter/0 |  
json_pp -json_opt pretty,canonical
```

La commande après le pipe permet de faire l'indentation automatiquement et ainsi ne pas avoir tout sur une seule ligne

Allumer ou éteindre la prise :

```
curl http://192.168.33.1/relay/0?turn=on | json_pp -json_opt  
pretty,canonical
```

```
curl http://192.168.33.1/relay/0?turn=off | json_pp -json_opt  
pretty,canonical
```

4)

Quand on envoi une commande à la prise, elle nous répond dans un certain format. Elle met tout sur une seule ligne

J'installe donc jq (sudo apt install jq) pour indenter automatiquement

Avec la commande jq je récupère les informations renvoyées par ma prise et je peux les filtrer en ajoutant .[champs_voulu]

```
test@202-15:~/TP-REST-LAFORGE$ curl http://10.202.255.252/meter/0  
| jq  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 119 100 119 0 0 5666 0 --:--:-- --:--:-- --:--:-- 5666  
{  
  "power": 49.02,  
  "overpower": 0,  
  "is_valid": true,  
  "timestamp": 1606215557,  
  "counters": [  
    30.002,  
    10.594,  
    35.931  
  ],  
  "total": 150  
}
```

Ici la commande jq permet d'indenter la réponse

```
test@202-15:~/TP-REST-LAFORGE$ curl http://10.202.255.252/meter/0  
| jq .power  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 119 100 119 0 0 7437 0 --:--:-- --:--:-- --:--:-- 7437  
48.78
```

Maintenant en ajoutant .power je ne récupère que l'information de la consommation actuelle en électricité

5) Je crée un jeu de scripts me permettant de piloter un ou plusieurs plugs simplement :

Script allumer.sh :

```
#!/bin/bash  
  
curl --silent http://10.202.255.252/relay/0?turn=on |jq # Allume  
la prise  
sleep 5 # Délai de 5s  
echo "La consommation est de : `curl --silent  
http://10.202.255.252/meter/0 |jq .power` W" # Voir sa  
consommation actuelle  
curl --silent http://10.202.255.252/relay/0 |jq # Voir si la prise  
est allumée ou non (ison=true ou false)
```

Script eteindre.sh :

```
#!/bin/bash  
  
curl --silent http://10.202.255.252/relay/0?turn=off |jq # Allume  
la prise  
sleep 5 # Délai de 5s  
echo "La consommation est de : `curl --silent  
http://10.202.255.252/meter/0 |jq .power` W" # Voir sa  
consommation actuelle  
curl --silent http://10.202.255.252/relay/0 |jq # Voir si la prise  
est allumée ou non (ison=true ou false)
```

Ces scripts me permettent d'allumer ou éteindre la prise et de vérifier certains états pour vérifier que cela a bien fonctionné

6) Les trames suivantes reprennent mes deux scripts créés précédemment

1) allumer la prise, afficher la consommation, vérifier l'état de la prise

2) éteindre la prise, afficher la consommation, vérifier l'état de la prise

A chaque fois on a deux requêtes (1 du client pour la requête et 1 du plug pour répondre)

On a 147 octets d'envoyés pour allumer un plug (voir trame 148)

No.	Time	Source	Destination
Protocol	Length	Info	
148	23.240490529	10.202.15.1	10.202.255.252
HTTP	147	GET /relay/0?turn=on	HTTP/1.1

No.	Time	Source	Destination
Protocol	Length	Info	
150	23.260611141	10.202.255.252	10.202.15.1
HTTP	290	HTTP/1.1 200 OK (application/json)	

Ces deux trames montrent que la prise est bien allumée (1 requête par le client et 1 réponse par la prise)

No.	Time	Source	Destination
Protocol	Length	Info	
161	23.776944661	10.202.15.1	10.202.255.252
HTTP	372	GET /status	HTTP/1.1

No.	Time	Source	Destination
Protocol	Length	Info	
164	23.849864264	10.202.255.252	10.202.15.1
HTTP	502	HTTP/1.1 200 OK (application/json)	

No.	Time	Source	Destination
Protocol	Length	Info	
219	28.307515215	10.202.15.1	10.202.255.252
HTTP	139	GET /meter/0	HTTP/1.1

No.	Time	Source	Destination
Protocol	Length	Info	
220	28.323815280	10.202.255.252	10.202.15.1
HTTP	287	HTTP/1.1 200 OK (application/json)	

Ces deux trames vérifient la consommation actuelle en électricité (1 requête par le client et 1 réponse par la prise)

No.	Time	Source	Destination
229	28.343322590	10.202.15.1	10.202.255.252
HTTP	139	GET /relay/0 HTTP/1.1	

No.	Time	Source	Destination
230	28.360642371	10.202.255.252	10.202.15.1
HTTP	290	HTTP/1.1 200 OK (application/json)	

Ces deux trames vérifient l'état de la prise (allumée ou non :true ou false)

No.	Time	Source	Destination
239	28.772526982	10.202.15.1	10.202.255.252
HTTP	372	GET /status HTTP/1.1	

No.	Time	Source	Destination
242	28.837785597	10.202.255.252	10.202.15.1
HTTP	503	HTTP/1.1 200 OK (application/json)	

No.	Time	Source	Destination
260	32.728463630	10.202.15.1	10.202.255.252
HTTP	148	GET /relay/0?turn=off HTTP/1.1	

No.	Time	Source	Destination
261	32.747408097	10.202.255.252	10.202.15.1
HTTP	291	HTTP/1.1 200 OK (application/json)	

No.	Time	Source	Destination
274	33.776246359	10.202.15.1	10.202.255.252
HTTP	372	GET /status HTTP/1.1	

No.	Time	Source	Destination
277	33.839348327	10.202.255.252	10.202.15.1
HTTP	503	HTTP/1.1 200 OK (application/json)	

No.	Time	Source	Destination

```
294 37.767391634 10.202.15.1 10.202.255.252
HTTP 139 GET /meter/0 HTTP/1.1

No.      Time      Source      Destination
Protocol Length Info
295 37.784283258 10.202.255.252 10.202.15.1
HTTP 286 HTTP/1.1 200 OK (application/json)

No.      Time      Source      Destination
Protocol Length Info
304 37.801180638 10.202.15.1 10.202.255.252
HTTP 139 GET /relay/0 HTTP/1.1

No.      Time      Source      Destination
Protocol Length Info
305 37.818862105 10.202.255.252 10.202.15.1
HTTP 291 HTTP/1.1 200 OK (application/json)
```

7) Je fais un tag nommé partie1 et je l'envoi sur mon git :

```
git tag partie_1
git push --tags
```

2)MQTT

On peut ouvrir une VM et installer un serveur mqtt dedans. Ensuite on retourne sur la page web de la prise et on la connecte au serveur mqtt

Maintenant on a plus qu'à passer les commandes pour avoir les infos. Je modifie donc mes deux scripts pour les passer en mqtt :

Script allumer-mqtt.sh :

```
#!/bin/bash

mosquitto_pub -h 10.202.0.107 -t
shellies/shellyplug-s-6A6534/relay/0/command -m "on" # Envoi
l'ordre pour allumer la prise
sleep 5
# On attend 5 secondes
echo "La consommation actuelle est de : `mosquitto_sub -h
10.202.0.107 -t shellies/shellyplug-s-6A6534/relay/0/power -C 1`
W" # On récupère le consommation actuelle
```

Script eteindre-mqtt.sh :

```
#!/bin/bash

mosquitto_pub -h 10.202.0.107 -t
shellies/shellyplug-s-6A6534/relay/0/command -m "off" # Envoi
l'ordre pour allumer la prise
sleep 5
# On attend 5 secondes
echo "La consommation actuelle est de : `mosquitto_sub -h
10.202.0.107 -t shellies/shellyplug-s-6A6534/relay/0/power -C 1`
W" # On récupère la consommation actuelle
```

Je refais un tag nommé partie_2 et je le push

3) Serveur REST

Pour cette partie on commence par installer nginx :

```
sudo apt install nginx
```

Si on a des erreurs c'est sûrement à cause d'apche2. On peut donc le supprimer

Il faut maintenant modifier le fichier de configuration /etc/nginx/sites-enabled/default :

```
server {
    listen 80;
    listen [::]:80;

    root /var/www/html;
    index index.html index.htm index.nginx-debian.html;

    server_name local;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Il faut aussi passer à 0 le pathinfo dans /etc/php/7.3/fpm/php.ini :

```
cgi.fix_pathinfo=0
```

Dès qu'on a fini toutes nos modifications il faut relancer le service nginx