

## R409 TP1 : 11/02/2025

### MQTT

**Objectif :** MQTT



## RÉPONSE AU QUESTION DU TP

### 1.Contexte

Dans la salle un broker MQTT est installé à une adresse fournie par l'enseignant. Ce broker contient deux branches de topics : public et private La branche public autorise l'abonnement et la publication à tous les utilisateurs, authentifiés ou non. La branche private nécessite un mot de passe et un login. Des publishers (invisibles pour vous) sont programmés pour générer en continue des données dans les deux branches.

### 2 Client

#### Subscriber

a) afin de s'abonner au au topic public welcome on va utiliser la commande <mosquitto\_sub -h 10.216.0.117 -t public/welcome -q 2 -v >

on utilise -v afin d'avoir la verbose sur le résultat de la commande.

```
[[$] mosquitto_sub -h 10.216.0.117 -t public/welcome -q 2 -v  
public/welcome Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA
```

#### Screen de la commande et du resultat du sub au broker de la D216

En résultat de la commande on obtient la valeur de public/welcome ici « Bonjour et bienvenue ... » et ceux de maniere periodique.

```
public/welcome Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA  
public/welcome Hello, world!  
public/welcome Test retain MQTT  
public/welcome (null)
```

#### Screen des messages envoyer au cours du temps

```
[ssea2@Nitro-AN515-54][~]  
[[$] mosquitto_sub -h 10.216.0.117 -t public/welcome -q 2 -F " retain:%r : topic:%t : message:%p"  
retain:0 : topic:public/welcome : message:Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA  
^C  
[ssea2@Nitro-AN515-54][~]  
[[$] mosquitto_sub -h 10.216.0.117 -t public/welcome -q 2 -F " retain:%r : topic:%t : message:%p"  
retain:1 : topic:public/welcome : message:Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA  
^C
```

#### Screen des valeurs des retains pour le topic public/welcome

11/02/25

On voit dans le screen si dessus que le Message Retained a le flag retain est a 0 puis il passe a 1 en relançant la requête. La raison est que pour le 1<sup>er</sup> la commande a été lancée avant que le flag retain soit prit en compte.

```
[ssea2@Nitro-AN515-54][~]
[$] mosquitto_sub -h 10.216.0.117 -t public/welcome -q 1 -F " retain:%r : topic:%t : message:%p, qos:%q"
retain:1 : topic:public/welcome : message:Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA, qos:0
retain:0 : topic:public/welcome : message:HI, qos:1
retain:0 : topic:public/welcome : message:HI, qos:0
retain:0 : topic:public/welcome : message:HI, qos:1
```

```
[ssea2@Nitro-AN515-54][~]
[$] mosquitto_pub -h 10.216.0.117 -t public/welcome -q 1 -m HI

[ssea2@Nitro-AN515-54][~]
[$] mosquitto_pub -h 10.216.0.117 -t public/welcome -q 0 -m HI

[ssea2@Nitro-AN515-54][~]
[$] mosquitto_pub -h 10.216.0.117 -t public/welcome -q 2 -m HI
```

Screens des valeurs de Qos selon la qualité du sub et du publisher

On observe dans les screens si dessus que la valeur du Qos correspond a celui du publisher a condition que le subscriber demande une Qos  $\geq$  a celle du publisher. Exemple si le publisher envoie un message avec un qos de 2 et le subscriber veut le message mais avec un Qos de 1 il le recevra avec se Qos et non celui de l'émetteur.

b) On va maintenant tester le resultat de requête MQTT avec les signe jokers (#, +), pour cela on utilise la même commande pour lire les topics qu'au dessus.

```
[ssea2@Nitro-AN515-54][~]
[$] mosquitto_sub -h 10.216.0.117 -t public/# -q 1 -F " retain:%r : topic:%t : message:%p, qos:%q"
retain:1 : topic:public/capteur/1/statut : message:inactif, qos:0
retain:1 : topic:public/welcome : message:Bonjour et bienvenue sur le Broker du TP de réseau Rob&IA, qos:0
```

Screen des resultat de la commande avec le joker #

On observe que l'en mettant le # a la suite de public/ on obtient les message de chaque topic de public. C'est l'équivalent de \* sous linux.

```
[ssea2@Nitro-AN515-54][~]
[$] mosquitto_sub -h 10.216.0.117 -t public/capteur+/statut -q 1 -F " retain:%r : topic:%t : message:%p, qos:%q"
```

Screen des resultat de la commande avec le joker +

On observe que le plus nous permet de passer sur un des embranchements ainsi on peut récupérer, par exemple, les status des differents capteurs sans devoir faire une requête pour chaque capteur.

c) Pour s'abonner au 3 topic de public/capteur/1/ on utilisera :

```
<mosquitto_sub -h 10.216.0.117 -t public/capteur/1/mesures/# -t public/capteur/1/statut -F " retain:
%r : topic:%t : message:%p, qos:%q">
```

On utilise le # pour avoir les mesures car dans ce topic il n'y a que les mesures de pression et de températures, qui sont les topics que l'on souhaite afficher.

```
[ssea2@Nitro-AN515-54] [~]
[$] mosquitto_sub -h 10.216.0.117 -t public/capteur/1/mesures/# -t public/capteur/1/statut -F " retain:
:r : topic:%t : message:%p, qos:%q"
retain:1 : topic:public/capteur/1/mesures : message:Température: 22°C, qos:0
retain:1 : topic:public/capteur/1/statut : message:actif, qos:0
retain:0 : topic:public/capteur/1/mesures/pression : message:902, qos:0
retain:0 : topic:public/capteur/1/mesures/température : message:13.5, qos:0
retain:0 : topic:public/capteur/1/mesures/pression : message:901, qos:0
retain:0 : topic:public/capteur/1/mesures/température : message:14.817, qos:0
retain:0 : topic:public/capteur/1/mesures/pression : message:915, qos:0
retain:0 : topic:public/capteur/1/mesures/température : message:15.20378, qos:0
retain:0 : topic:public/capteur/1/mesures/pression : message:970, qos:0
retain:0 : topic:public/capteur/1/mesures/température : message:16.17386, qos:0
^C
```

Screen des valeurs de pression/temperature/statut

d)

```
[ssea2@Nitro-AN515-54] [~]
[$] mosquitto_sub -h 10.216.0.117 -t public/capteur/1/mesures/# -t public/capteur/1/statut -F " retain:
:r : topic:%t : message:%p, qos:%q"
retain:1 : topic:public/capteur/1/mesures : message:Température: 22°C, qos:0
retain:1 : topic:public/capteur/1/statut : message:actif, qos:0
retain:0 : topic:public/capteur/1/mesures/pression : message:940, qos:0
retain:0 : topic:public/capteur/1/mesures/température : message:12.16472, qos:0
```

Screen des valeurs de pression/temperature/statut

Afin de vérifier l'état de ses topic on a relancé la commande.

On observe que les flags retain on les mêmes valeurs qu'avant, le statut est retenue mais pas la temperature et la pression. On voit aussi que mesures contient un message de retenue avec une valeur de temperature a 22°. De plus tous ses messages sont envoyer au plus une fois Qos=0.

e) pour forcer avoir un qos a 1 on fait <mosquitto\_sub -h 10.216.0.117 -t public/capteur/1/mesures/# -t public/capteur/1/statut -q 1 -F " retain:%r : topic:%t : message:%p, qos:%q">

En passant le process en background pendant un petit moment puis en utilisant la commandes <fg> afin de le reactiver, on observe que plein de valeur se sont afficher d'un coup. Le Qos étant a 1 le broker a émit jusqu'à se qu'il reçoive un ack, c'est pourquoi on a reçue plein de valeur quand on c'est connecté.

On observe que le Qos du subscriber établit la qualité de la connection avec le broker mais le qos que l'on obtient en réponse correspond à celui du message du coté du publisher vers le broker.

f) Maintenant on va voir comment laisser un testament au broker. Pour cela on va utiliser la commande suivante `<mosquitto_sub -h 10.216.0.117 -t public/capteur/1/mesures/# -t public/capteur/1/statut -q 1 -F "retain:%r : topic:%t : message:%p, qos:%q" --will-topic public/alerte --will-payload "Network Error">`

```
[ssea2@Nitro-AN515-54][~]  
[ ] mosquitto_sub -h 10.216.0.117 -t public/alerte  
Capteur déconnecté !  
Network Error
```

screen du testament suite a la perte de ma connection

On voit qu'après avoir été time out suite au process qui passe en background le topic public/alerte affiche le testament et il disparaît quand l'appareil se reconnect.

q) L'inconvénient de ne pas retenir le testament est que si l'on redémarre le broker mais que le réseau ne fonctionne pas, il n'affichera pas le testament et donc on ne saura pas si la connection est fonctionnel ou non.

h) si on retenir le testament il faut prévoir qu'à la reconnection de l'appareil il faut supprimer son testament.

i)

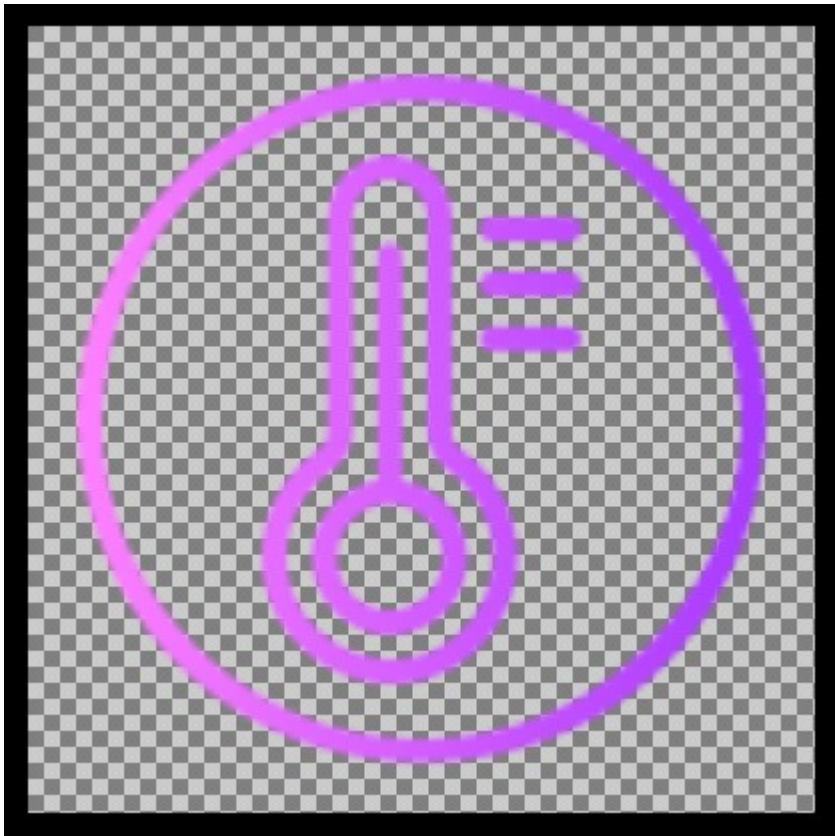


photo stocké dans le topic public/capteur/1/data

Pour récupérer cette image on utilise la commande suivante `<mosquitto_sub -h 10.216.0.117 -t 11/02/25`

public/capteur/1/data/# > Downloads/TP\_MQTT.png>. C'est une commande pour lire les topics suivie d'une redirection vers un fichier .png

en affichant l'image dans le terminal dans une des lignes on voit que l'extension est PNG. De plus on met le joker # à la fin de data car sinon il ne veut pas afficher le contenu de l'image.