

# TP N°7

## Présentation

Dans ce TP nous allons utiliser notre service Redis. Nos clients/objets seront des capteurs de température. Ils transmettront la température qu'ils relèvent à notre plateforme en ajoutant des données de localisation (ville de la mesure).

Nous mettrons à disposition ces informations via une API Rest à quiconque souhaite cette information (approche OpenData).

## Schéma d'architecture

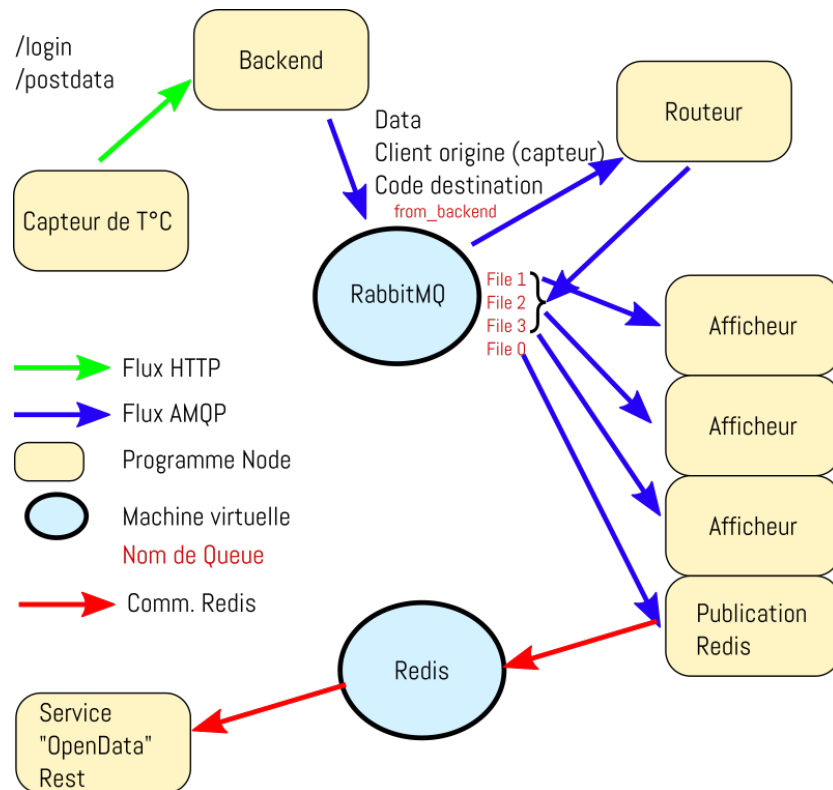


Figure 1:

## Modification du client

### Étape 1 : Modification du client

Notre client transmet aujourd'hui deux informations « `completdata` » et « `withnumber` ». Modifier notre outil client pour qu'il puisse transmettre les deux informations suivantes :

- Valeur de la température (qu'on pourra simuler à partir de la seconde courante et étaler sur une plage 0-30°C)
- Ville de prise de la mesure (par exemple, Angers, Paris...)

Notre client devra transmettre à la destination N°0 sa valeur pour la bonne prise en charge.

## Publication dans REDIS

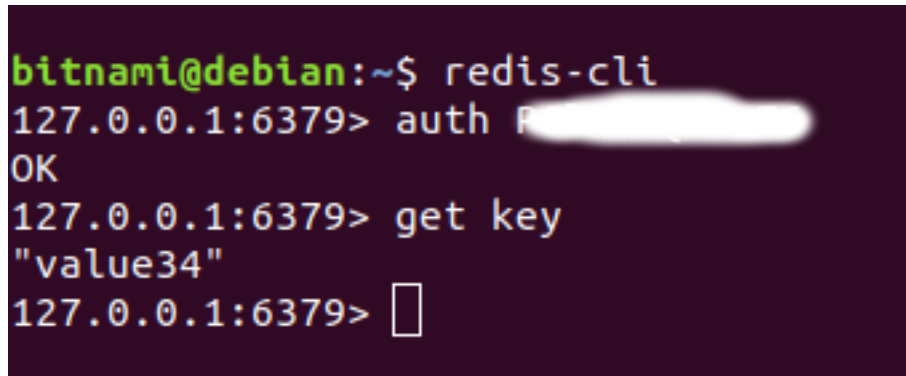
### Étape 1 : Création du service « PublishToRedis »

En vous basant sur le service « display », créer un nouveau service qui récupère les valeurs de température (depuis la file d'attente 0) et qui stocke cette valeur dans la base redis. Ainsi, nous stockerons une association clé/valeur :

- La clé sera le nom de la ville (en minuscule)
- La valeur sera la température courante

### Étape 2 : tester

Tester cette mise en œuvre en vérifiant (grâce au client redis d'exemple du TP N°6) que la valeur est correctement écrite. Vous pouvez aussi vérifier cela avec le client en ligne de commande dans la VM :



```
bitnami@debian:~$ redis-cli
127.0.0.1:6379> auth P
OK
127.0.0.1:6379> get key
"value34"
127.0.0.1:6379> 
```

Figure 2:

## Création d'un service REST

### Étape 1 : Création du service « REST\_TEMP »

En vous basant sur le code du backend, créer un nouveau service REST qui aura un unique point d'entrée :  
GET /temp/ville

Ce service retournera juste un json contenant la température (valeur en « dur » dans un premier temps) :

Code d'erreur 200, données json : « {temperature : 27.6 } »

Il n'y aura pas d'authentification pour les utilisateurs de ce service.

Attention : pensez à utiliser un numéro de port différent de celui du backend

### Étape 2 : Ajouter à votre service la récupération dans Redis

Votre nouveau service doit se connecter à la base Redis pour récupérer la température de la ville demandée.

- Si la valeur existe dans redis, le code d'erreur de retour sera « 200 » avec la valeur de la température (en json)
- Si la valeur n'existe pas, le code retour sera 404 (les données de retour pourront être un message d'erreur JSON)

### **Étape 3 : Vérifier avec votre navigateur**

Avec votre navigateur, vous devriez pouvoir tester cela sans difficulté. Publiez des données de températures dans plusieurs villes et tester les cas d'erreur.