

AQROPOL

Charte des communications entre les sous- groupes

Participants au projet :

Jean-Baptiste	Bordier
Manon	Derocles
Mélian	Ferrachat
Julien	Garnier
Alan	Gaubert
Adrien	Leblanc
Thibault	Leclair
Antoine	Leval
Pierre	Miola
Bashar	Nemeh
Noureddine	Kadri
Antoine	Posnic
Fanny	Prieur

AQROPOL

Hantavolaniaina Sylvia Rabe

Enseignant chercheur à l'université de Rennes 1(IRISA)

*Mr François Bodin
Enseignant référent du projet*

Table des matières

Partie Capteur	3
Vers l'Android	3
Communication Capteur → Android en détail	3
Vers le Serveur	4
Partie Android	5
Vers le Capteur	5
Vers le Serveur	5
Communication Android → Serveur en détail	5
Partie Serveur	6
Vers le Capteur	6
Vers le smartphone Android	6
Communication Android ← Serveur en détail	6
Vers la Visualisation	6
Partie Visualisation	7
Vers le Serveur	7

Partie Capteur

Vers l'Android

Étapes avant communication entre Capteur et Android :

1. Android se connecte au Wifi du NUC.
2. Android interroge le service REST du NUC

La communication entre le capteur et le smartphone Android est basé sur le protocole REST.

Communication Capteur → Android en détail

Android doit créer son propre fichier JSON.

Android récupère l'identifiant unique du NUC à l'adresse : 10.42.0.1:8080/nucs
qu'il ajoute à son fichier.

Format JSON du fichier :

```
{
  "ID_NUC": <idNuc>,
  "dateenvoi": "now()";
}
```

Android récupère la liste des sensors à l'adresse : 10.42.0.1:8080/sensors
qu'il ajoute à son fichier à la suite.

Format JSON du fichier :

```
"sensors" : [
  {
    "name" : "nomDuCapteur",
    "id" : 1,
    "type" : "typeDuCapteur",
    "unity" : "unitéDuCapteur",
  }
]
```

Android récupère les mesures à l'adresse : 10.42.0.1:8080/measures
qu'il ajoute à son fichier à la suite.

Format JSON du fichier :

```
"measures" : [ {
  "value" : "",
  "id" : 1,
  "hash" : "PmubPzdFE/FKYI/MBpODg0LHM/y9tX9D17k+ARFuOmM=",
  "timestamp" : "2018-04-11T13:11:08.010+0000",
  "nuc" : {
    "token" : "token_du_nuc"
  },
  "sensor" : {
    "id" : 1,
    "name" : "nomDuCapteur",
  }
}
```

```
"type" : "typeDuCapteur",  
"unity" : "unitéDuCapteur"  
}
```

Améliorations possibles :

1. Chiffrement du fichier JSON.
2. Chargement asynchrone des données via un stream. (Spring WebFlux, incompatible avec un SGBD relationnel).

Vers le Serveur

Le capteur et le serveur ne communiquent jamais directement. Cependant ils se sont mis d'accord sur les données qu'ils doivent échanger par l'intermédiaire du terminal Android. Le NUC doit exporter l'ensemble de sa base de données qui contient les données mesurées de chaque capteur qui lui est rattaché. Cet export se fera sous la forme d'un fichier JSON.

Ce fichier doit contenir les données sous la forme :

- La date (en millisecondes)
- Le nom du capteur (ex: "CapteurAir")
- L'identifiant unique du NUC
- Le type de la mesure (ex: "pm10")
- La valeur de la mesure (ex: "0.42")
- Coordonnées GPS avec redondance GPS (ex: "lat: 48.154548, long: -1.45784")
- Hash sur les données en SHA-1 ou SHA256 :
 - Date
 - Type de mesure
 - Valeur de la mesure

L'une des améliorations possibles est que toutes les données soient chiffrées avec une clé symétrique dont ne dispose que les NUC et le serveur (et éventuellement une application mobile "administrateur" non disponible pour le public), qui permet de vérifier du côté capteur ou bien du côté serveur si les données sont toujours intègres.

Partie Android

Vers le Capteur

Android envoie un tableau de hash du serveur pour la suppression des données (JSON) et récupère les mesures du capteur.

Vers le Serveur

Android envoie les données sous un format JSON avec l'enregistrement de date de la réception du fichier lors de la communication avec le NUC, ainsi que la date actuelle au serveur lors de la communication.

Les données sont envoyées au serveur par un POST sur l'URL correspondante de l'API.

L'envoi de données est composé de 3 paramètres :

1. Fichier JSON (contenant les données)
2. Date d'enregistrement du fichier (début des réceptions des données via le NUC)
3. Date d'envoi du fichier (début d'émission des données vers le serveur)

Communication Android → Serveur en détail

En mode écoute(GET): Fichier JSON formaté comme suite :

```
[
  {
    "id_hub": 1,
    "hash": "291873b4e297ed30c25583689452d90c"
  },
  {
    "id_hub": 2,
    "hash": "4d94bf3f3a436420c31fb4d65a3a6d79"
  },
  {
    "id_hub": 5,
    "hash": "52a4bafd92da0ed96a0c3986b40f22eb"
  }
]
```

Fig.1: fichier JSON contenant le dernier Hash de chaque NUC

Partie Serveur

Vers le Capteur

Voici la forme du tableau de hash qui sera constitué tel que :

ID du NUC(unique)	Dernier hash du NUC reçu
-------------------	--------------------------

Vers le smartphone Android

_____En mode écoute(GET): Fichier JSON (contenant un tableau de Hash) pour signaler la bonne réception du fichier.

Sinon envoie une erreur.

En mode écoute(POST): Recevoir un fichier JSON (contenant les données récoltées par les capteurs).

Communication Android ← Serveur en détail

En mode écoute(POST): Recevoir un fichier JSON formaté comme ci-dessous :

```
{
  "nuc": 50,
  "dateenvoi": "2018-04-10 00:00:00",
  "mesures": [
    {
      "capteur": "capteurAir1",
      "type": "pm30",
      "unite": "Q/m3",
      "valeur": 2.3,
      "date": "2018-04-09 00:00:00",
      "lat": 48.1151495,
      "long": -1.74707547,
      "hash": "291873b4e297ed30c25583689452d90c"
    }
  ]
}
```

Fig.2: Fichier JSON contenant les données récoltées par les capteurs

Vers la Visualisation

Le serveur envoie un fichier JSON avec un format spécial pour la visualisation toujours en clair.

Améliorations possibles : API avec des paramètres (pour filtrer la visualisation)

Partie Visualisation

Vers le Serveur

Ne transmet rien, ne fait que prendre des données collectées par la base de données.