Ynov Bordeaux
Development Mobile 2022-2023

LABARRE Arthur
MARCHAND Yohann
ABDOUL-WAHAB Akram

Lien GitHub: https://github.com/ArthyTheFox/vcubMonitor

VCubMonitor

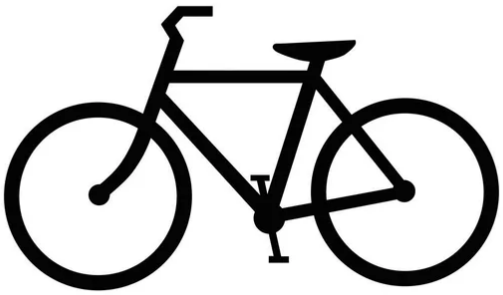


Table des matières

1. Introduction

Nous avons décidé de créer une application permettant de trouver les différentes stations VCub sur Bordeaux ainsi que de consulter les informations les concernant. Nous avons choisi ce projet, car il permet de pouvoir développer plusieurs aspects du développement mobile Kotlin.

C'est notamment un produit qui peut être utile en ces temps de mauvaises circulations sur Bordeaux pour les automobilistes et ainsi les permettre de pouvoir facilement trouver des vélos disponibles rapidement et à proximité.

2. Présentation de l'application

1. Librairies utilisées

1.1. Gson

- Gson est une bibliothèque Java qui peut être utilisée pour convertir des objets Java en leur représentation JSON. Il peut également être utilisé pour convertir une chaîne JSON en un objet Java équivalent.

1.2. Volley

- Volley est une bibliothèque HTTP qui rend le réseautage pour les applications Android plus facile et surtout, plus rapide.

1.3. Play_service_map

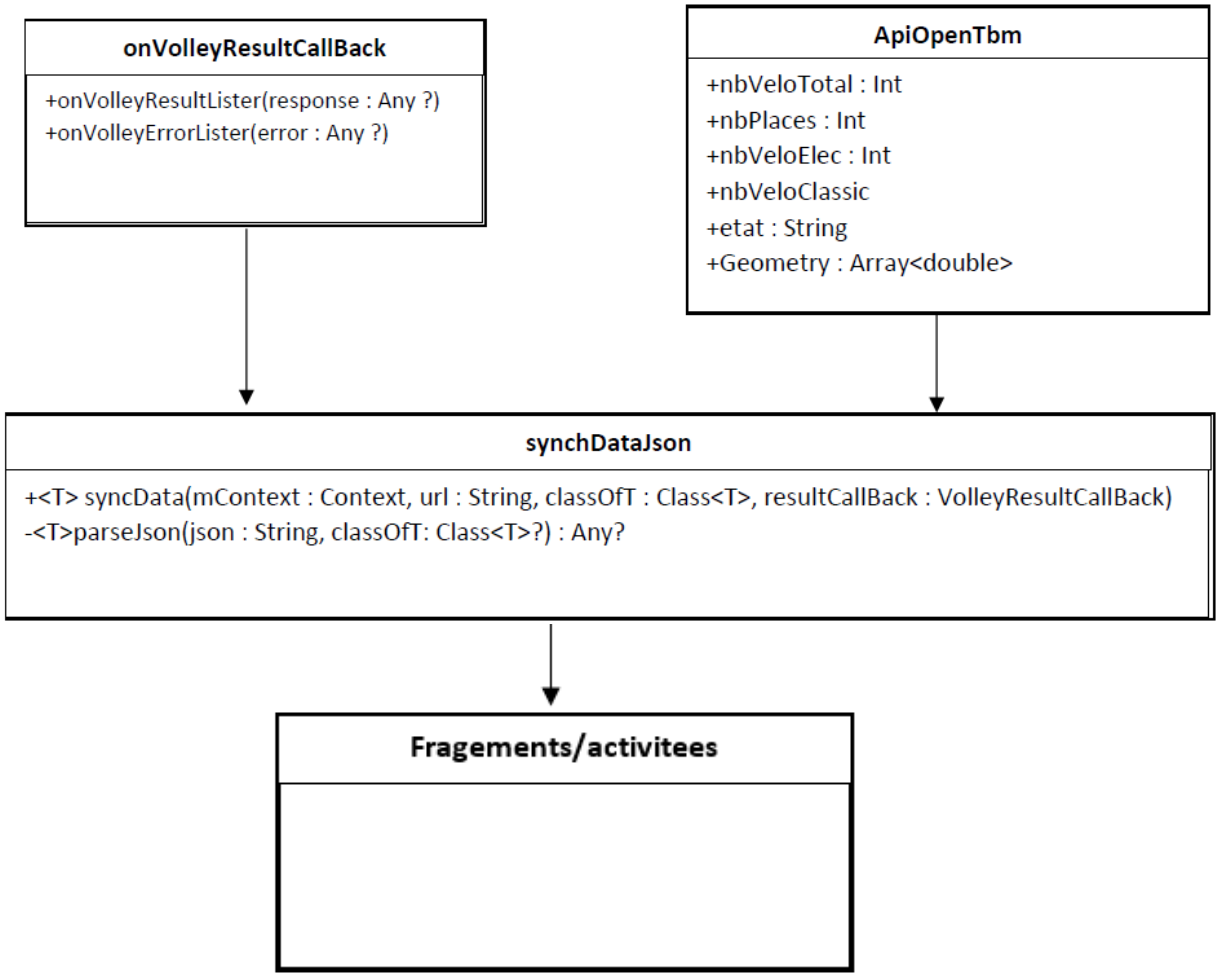
-Play_service_map est une bibliothèque google qui permet de créer et gérer une carte Google map.

2. Diagramme de classe

synchDataJson est pour le service Web. C'est une classe générique pour toutes les autres classes utilisant Volley et parseJson.

onVolleyResultCallBack verifie si le json est bien traité sinon onVolleyErrorLister.

ApiOpenTbm est une classe pour les paramètres du json tbm.



3. Les écrans

2.1. Splashscreen

Page de lancement de l'application affichant le nom de l'application ainsi qu'un logo

VCub Monitor

2.2. Liste des stations

Page listant les différentes stations récupérées par l'API avec les nombres de vélos disponible, l'état "EN MAINTENANCE / CONNECTE/ DÉCONNECTÉ" ainsi que le nom de la station

2.3. Détails d'une station

Cette page est affiché lorsque que l'utilisateur clique sur une des stations dans la liste, de ce fait, les informations récupérés sont transférés sur la liste détails et affiches les mêmes infos que la page liste avec des données supplémentaires comme le nombre de vélos classiques et électriques.

2.4. Map

Cette page affiche les différentes stations récupérées par l'API ainsi que leur position par le biais de marqueur. Pour développer cela, nous avons utilisé l'API google map permettant d'afficher une map avec des points définie par des coordonnées géographiques.

On note notamment que la couleur des marqueurs diffère en fonction de l'état de la station (déconnecté = rouge, maintenance = orange, connecté = vert)

Quand on clique sur un marqueur, la caméra zoom sur la position et une bulle d'information s'ouvre afin de pouvoir y retrouver des informations sur la station telle que le nombre de vélos disponible et le nombre de places disponibles afin d'y déposer son vélos.

4. API

Pour le choix de l'API nous avons opté pour une API open data bordeaux métropole donnant en temps réel les informations des différentes stations ainsi que les vélos disponibles. Cette API ne requiert pas d'authentification spécifique et est donc facile à manipuler.

Voici un exemple d'objet JSON d'une station renvoyée par l'API, en rouge sont les objets, tableaux et paramètres utilisés :

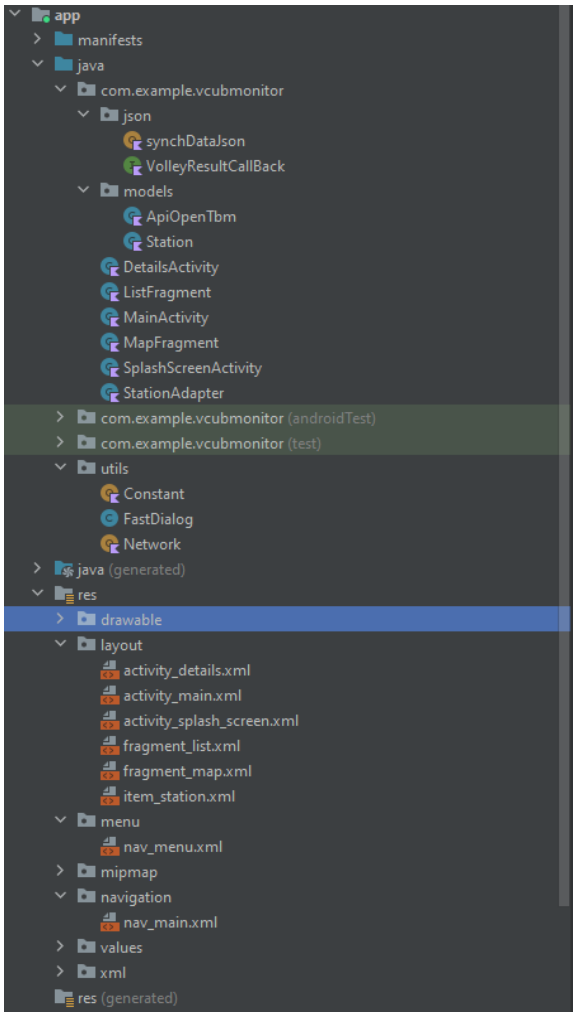

```
{
"datasetid": "ci_vcub_p@scnbdx",
"recordid": "621534d86aba6b0fad794500655fdc50180cbe94",
"fields": {
"gml_id": "CI_VCUB_P.1",
"commune": "Bordeaux",
"geo_point_2d": [
44.83803,
-0.58437
],
```

```
"nbvelos": 23,
"geo_shape": {
"coordinates": [
-0.58437,
44.83803
],
"type": "Point"
},
"cdate": "2011-01-01T00:00:00+00:00",
"mdate": "2022-12-02T12:39:44+00:00",
"nbplaces": 3,
"ident": 1,
"gid": 1,
"nom": "Meriadeck",
"type": "VLS",
"nbelec": "14",
"etat": "CONNECTEE",
"code_commune": "33063",
"nbclassiq": "9",
"geom_o": "0"
},
"geometry": {
"type": "Point",
"coordinates": [
-0.58437,
44.83803
]
},
"record_timestamp": "2022-12-02T11:42:26.961Z"
},
```


5. Activity / Fragment

Activity	Fragment

SplashScreenActivity	MapFragment
MainActivity	ListFragment
DetailsActivity	



3. Améliorations possibles

- Un système d'itinéraire permettant de trouver le chemin de la station la plus proche
- Un système de gestion d'utilisateur permettant de pouvoir mettre en "favoris" les stations de notre choix et les retrouver plus facilement.
- Un système d'alerte lorsque l'une des stations favoris change d'état ex: "maintenance"
- Ajout d'un bouton retour quand on est dans une liste

4. Conclusion

Ce projet était très intéressant, d'un côté, nous avons pu nous familiariser davantage avec notre environnement de développement ainsi que les fonctionnalités d'Android Studio, il nous a permis de collaborer avec GitHub ainsi que d'approfondir nos connaissances sur le développement mobile et plus particulièrement le langage Kotlin.