

Notice d'utilisation de l'application :

L'application se décompose en plusieurs parties :

- Une base de données (MySQL) disposé sur le serveur
- Une API (Python avec Flask) disposé sur le serveur
- Une application mobile (React native) disposé des téléphones, représente vélos et relais GPS
- Une application web (Angular) disposé sur le serveur, appli utilisée par l'entraîneur
- Un serveur de websockets disposé sur le serveur

Pour l'instant le traitement des données se fait depuis une autre appli python. Ce traitement peut être modifié et fait à votre façon en récupérant les données de la BDD.

Pour que l'application fonctionne correctement il faut donc que toutes parties citées ci-dessus soit démarrer :

Pour la BDD : il faut qu'une instance de BDD MySQL soit lancé sur le serveur avec les schémas que l'on peut trouver dans les fichiers de l'API dans SQL.

Pour l'API : un simple *python3 app.py* pour la lancer, ne fonctionne pas sans BDD.

Pour l'application web : la commande *npm install* pour installer les dépendances nécessaires au fonctionnement de l'appli Puis on peut lancer avec *ng serve*

Il faut également une clé de l'API de Google Map pour faire fonctionner l'application.

Pour l'application mobile : *npm install* puis *npm start*. En mode développement utiliser l'application Expo Go ou un émulateur. Pour le déploiement, la commande : *expo build:android* puis télécharger l'apk à partir de l'application Expo Go (il faut un compte).

Les applis mobile et web ne fonctionnent pas sans la BDD et API.

Enfin, pour permettre de faire communiquer le tout vous pouvez lancer le serveur des websockets avec *python3 server.py* dans le dossier des websockets.

Toutes les applications sont donc lancées et peuvent commencer à communiquer entre elle.

Extraction de données :

Pour l'instant il n'existe pas de solution permettant de récupérer les données qui viennent d'être extraites d'une course.

Pour extraire les données il faut alors directement interagir avec la BDD grâce à la CLI.

Il faut donc extraire les données en CSV en exécutant la commande :

```
SELECT * FROM T_ENREGISTREMENT INTO OUTFILE  
'chemin_d'enregistrement_du_fichier';
```

Ensuite avant de refaire un enregistrement vous devez supprimer tous les enregistrements présents dans la table en réalisant la commande :

```
TRUNCATE TABLE T_ENREGISTREMENT;
```

Vous obtenez alors un CSV que vous pouvez analyser.

Justification de choix des technologies :

Nous manquons de temps nous avons donc décider d'utiliser les technologies que nous connaissions le mieux et qui nous permettait d'implémenter le plus rapidement possible nos applications. Python avec flask pour sa facilité d'implémentation. MySQL pour notre connaissance de la technologie et le côté relationnel ainsi que la petite quantité de données. React Native nous permettant de déployer l'application sur Android et IOS. Angular pour les connaissances de la technologie.