ProjetAndroid4A

Projet et fonctionnalités :

Le projet consiste en une application censée affichant la météo à l’endroit où se trouve l’utilisateur. Les informations affichées sont la localisation, le temps (soleil, pluie, etc.), ainsi que d’autres informations secondaires telles que la vitesse du vent ou la pression. L’application est développée en Kotlin.

Les fonctionnalités visées sont :

* Choix d’affichage de la météo actuelle ou des prévisions sur les 7 prochains jours
* Géolocalisation de l’utilisateur

Ce que j’ai appris/utilisé :

J’ai suivi le tutoriel à l’adresse suivante pour réaliser ce projet :

<https://www.youtube.com/watch?v=yDaaM3u389I&list=PLB6lc7nQ1n4jTLDyU2muTBo8xk0dg0D_w&index=1>

J’ai cependant assez vite rencontré un problème limitant : l’API utilisée par le développeur dans le tutoriel n’existe plus et a été remplacée par une autre qui envoie les données sous un format différent. Le tutoriel étant très sophistiqué en particulier sur la gestion des données (Coroutine, Room, SQLite, MVVM), je n’ai pas réussi à adapter les instructions aux données sous leur forme récente, et j’ai donc été très rapidement bloqué dans le développement du projet.

Je n’ai accompli que les étapes suivantes :

* Mise en place des fondations de l’application : gradle et fragments
* Configuration de la navigation entre les fragments via un menu fixe en bas de page et avec un bouton de retour à la page d’accueil.
* Appel à l’API, conversion des données JSON en une classe Kotlin
* Affichage brut des données, dans le premier fragment, de la météo actuelle à la localisation passée en argument directement dans le code (cette étape était dans le tutoriel un test pour vérifier le bon fonctionnement de l’API et des fondations de l’application)

La partie UI du tutoriel arrivant après toute la phase de mise en place du système de données, je n’ai pas été capable de mettre en forme les données brutes simplement à partir des classes Kotlin par moi-même. Dans le tutoriel, des données sont présentes dans un répository dans lequel elles sont injectées via Kodein après un stockage en local dans une base SQLite (si j’ai bien compris).

L’affichage simple des données brutes tel qu’il est réalisé dans mon application se fait avec un GlobalScope via Coroutine.

Les deuxième et troisième fragments sont vides.

Captures d’écran de l’application :



 