

TP noté à rendre

Consigne :

Développer une application Android capable d'afficher les prévisions météo d'une ou plusieurs villes à l'aide de l'API gratuite open-meteo.com. L'application devra afficher les informations de manière claire et intuitive, tout en intégrant des fonctionnalités supplémentaires pour enrichir l'expérience utilisateur.

Vous devrez héberger votre projet sur un dépôt Git (de votre choix).

Fonctionnalités obligatoires :

- Écran d'accueil comportant une barre de recherche ainsi que le résumé météorologique des villes ajoutées en favoris.
- Système de recherche de ville en utilisant l'API de geocoding open-meteo.com
- Une option devra être disponible pour rechercher la météo à partir de la géolocalisation de l'utilisateur
- Écran de détail des prévisions météorologiques de la ville sélectionnée. Afficher au moins les éléments suivants :
 - Température actuelle
 - Conditions météorologiques (ensoleillé, nuageux, pluie)
 - Températures minimale et maximale
 - Vitesse du vent
- L'utilisateur devra être capable d'ajouter et supprimer une ville de ses favoris
- Gestion du cache :
 - mise en cache des résultats de requête, des favoris etc
 - l'application devra pouvoir fonctionner en mode hors connexion
- Gestion des erreurs API et réseau (timeout ou hors connexion)
- Gestion de la rotation de l'écran

Endpoints de l'API :

- Geocoding (récupération des coordonnées GPS par le nom de la ville) :
[https://geocoding-api.open-meteo.com/v1/search?name=\[VILLE\]](https://geocoding-api.open-meteo.com/v1/search?name=[VILLE])
- Récupération des données : voir documentation
<https://open-meteo.com/en/docs/meteofrance-api>.

Exemple de requête :

https://api.open-meteo.com/v1/forecast?latitude=42.7028&longitude=9.45&hourly=temperature_2m,relative_humidity_2m,apparent_temperature,rain,wind_speed_10m&models=meteofrance_seamless

Note : veillez à bien utiliser le modèle *meteofrance_seamless*.

Critères d'évaluation :

- **Fonctionnalités obligatoires** (voir ci-dessus)
- **Qualité du code** : respect des bonnes pratiques de développement Android (architecture, modularité, lisibilité)
- **Tests et robustesse de l'application** : éviter les bugs et les crashes
- **Interface et expérience utilisateur** : L'UI est-elle intuitive ? Est-ce fluide ?
- **Gestion des erreurs** : affichage de messages clairs à l'utilisateur en cas d'erreur

Notes additionnelles :

- Toute fonctionnalité ajoutée de votre propre initiative sera évidemment fortement appréciée lors de la notation
- L'API open-meteo autorise 10 000 appels par jour. C'est conséquent mais attention à ne pas faire d'appel réseau dans une boucle non maîtrisée ;)