

TP Android : Consultation de la météo sur Android (1ère partie)

Objectifs

Le but est de développer une application météo. Cette première partie consiste à développer une interface graphique avec plusieurs écrans et à interroger un service Web.

1 Affichage d'une liste

Au lancement de l'application, un premier écran (activité `MainActivity`) doit afficher une liste prédéfinie de villes dont on souhaite suivre la météo (Fig. 1).

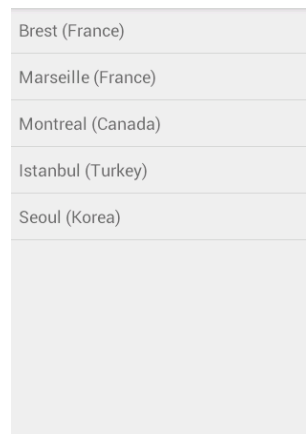


FIGURE 1 – Affichage de la liste des villes

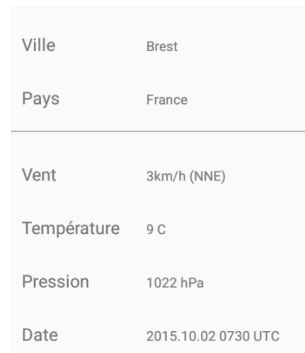
Définissez pour ce faire une `ListView` comme composant graphique et une `ArrayList` comme structure de données stockant les villes. Une ville devra correspondre à un objet `City` ayant les attributs suivants :

- nom de la ville,
- pays d'appartenance,
- date du dernier relevé météo,
- vitesse du vent (en km/h),
- direction du vent,
- pression (en hPa),
- température de l'air (en degrés Celsius).

2 Ajout des actions pour la liste

Ajoutez une première action dans l'activité `MainActivity` de façon à ce qu'un clic sur un item de la liste ouvre un nouvel écran `CityView` (Fig. 2).

Ajoutez une seconde action de façon à ce qu'un long clic sur un item de la liste ouvre un menu contextuel affichant « Supprimer » et permettant d'effacer la ville sélectionnée.

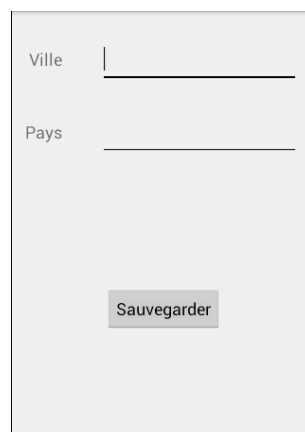


Ville	Brest
Pays	France
Vent	3km/h (NNE)
Température	9 C
Pression	1022 hPa
Date	2015.10.02 0730 UTC

FIGURE 2 – Affichage des informations collectées sur une ville

3 Ajout d'une ville

L'ajout d'une ville est à réaliser en plaçant un item « + » dans la barre d'actions (*Action bar*) de l'application. Cette nouvelle action devra afficher un nouvel écran (Fig. 3) permettant de saisir le nom de la nouvelle ville. Ce nouvel écran devra être géré par une activité nommée `AddCityActivity`.



Ville	<input type="text"/>
Pays	<input type="text"/>
<input type="button" value="Sauvegarder"/>	

FIGURE 3 – Affichage d'un formulaire de saisie d'une nouvelle ville

4 Tâche de rafraîchissement

Jusqu'à présent, toutes les données de l'application sont définies de manière statique. Ajoutez un item « Rafraîchir » dans la barre d'actions de l'application. Cette nouvelle action devra lancer une

tâche tournant en arrière-plan au moyen de la classe **AsyncTask**. Pour l'instant, vous pourrez simuler le téléchargement de données sur le réseau avec un **Thread.sleep()**. Pour indiquer à l'utilisateur que la tâche de rafraîchissement est terminée, lancez un **Toast**.

5 Interrogation d'un service Web

Les informations actualisées sur la météo sont à obtenir en interrogeant le service Web météo de Yahoo. Ce service vous permet d'afficher plusieurs informations sur la météo courante ainsi que des prévisions. La documentation de l'API peut être trouvée à l'adresse suivante : <https://developer.yahoo.com/weather/>.

Pour vous faciliter la mise en œuvre du traitement des réponses du service Web, la classe **JSONResponseHandler** est mise à disposition sur la page du cours.

Utilisez cette classe d'analyse dans l'**AsyncTask** défini à la tâche précédente. Mettez à jour le tableau de la liste de villes en fonction des informations recueillies.

6 Rendu

Il vous est demandé de déposer votre projet Android créé sur Android Studio sur le site du cours de la plateforme e-uapv la semaine du 16 octobre.