



Exercice 1 : Rappel - développement mobile android

Développer une application mobile simple de "Gestion de rappels" :

1. Saisir un nouveau rappel avec un horaire, une date, un type, un titre et une description.
2. Afficher les rappels dans une liste pour pouvoir les consulter et pour faciliter la suite du TP.

Objectif: Créer une application de base pour faciliter le développement des autres exercices et revoir les bases du développement basé sur les états et les événements.

Documentation : <https://developer.android.com/reference/android/widget/ListView>

Exercice 2 : Notifications Android + Wearable

Reprendre l'application de l'exercice 1 et ajouter les fonctionnalités suivantes:

1. Permettre à l'utilisateur de pouvoir choisir ou non des notifications par rapport aux événements saisis. Cela peut être par sélection d'événements sur la liste de l'exercice précédent.
2. Ajouter des notifications en fonction des informations saisis par l'utilisateur.

Objectif: Avoir des notifications basiques pour le téléphone Android et la montre Android Wear pour avoir une première vue sur les notifications envoyés par l'application téléphone.

Documentation : <https://developer.android.com/training/notify-user/build-notification>

Vidéo : <https://classroom.udacity.com/courses/ud875A/lessons/4613198543/concepts/45790305480923>

Exercice 3 : Notifications pour Wearables

Tester les notifications précédentes pour des Wearables.

Exercice 4 : Notifications étendus

1. Ajouter des actions aux notifications en fonction du type de rappel. Par exemple, pour le rappel de type consulter un événement, ajouter l'action pour envoyer l'utilisateur



vers la page Facebook. Autre type d'actions à voir: "reply", "send to app" et "voice reply".

2. Ajouter différents style de notifications.
3. Ajouter des Pages pour des rappels avec des descriptions (technique obsolète donc optionnel).

Objectif: Étendre le potentiel des notifications sur les montres Wearables pour voir toutes les fonctionnalités.

Documentation : <https://developer.android.com/training/wearables/notifications/noti-styles.html>
<https://developer.android.com/training/wearables/notifications/creatingexpanded>
<http://www.androiddocs.com/training/wearables/notifications/pages.html>

Exercice 5 : Utilisation de la voix

Reprendre l'application précédente pour ajouter l'activation de certaines actions par la voix: ouvrir la saisie d'un nouveau rappel sur téléphone, description, titre...

Vidéo : vidéo sur l'utilisation de la voix pour répondre à une notification

<https://classroom.udacity.com/courses/ud875A/lessons/4613198543/concepts/45790305520923>

Exercice 6 : Synchronisation des données avec l'application mobile

Créer une application Android avec une application compagnon pour Wear OS (Figure 1):

1. Reprendre l'application précédente.
2. A l'ajout de rappels, transférer des rappels entre l'application Android et l'application Wear en utilisant des "data items".
3. Ajouter le transfert de données avec les messages.

Objectif : Utiliser les méthodes de transfert de données. Comprendre quand utiliser les messages et quand les "data items" sont utiles (c'est un choix entre vitesse de transfère, préservation de la batterie, taille des données, et garantie de réception).

Documentation : <https://developer.android.com/training/wearables/data-layer>

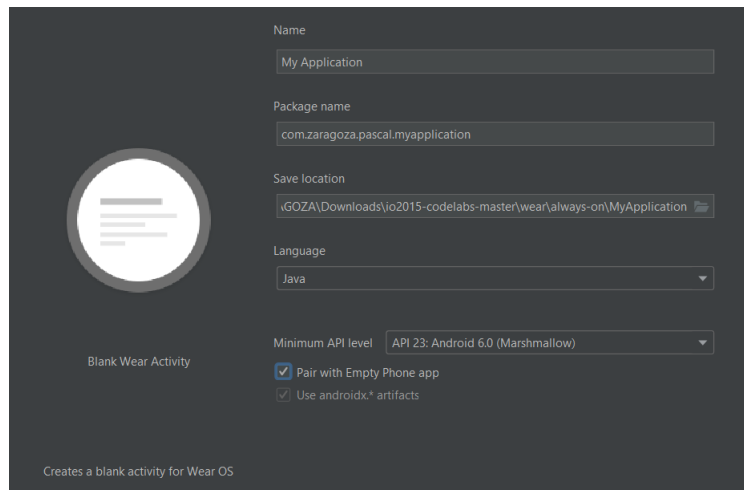


Figure 1: Il faut sélectionner le couplage d'application.

Android Samples : Voir l'exemple de Google qui s'appelle 'DataLayer' en allant dans 'File.New.Import Sample'.

Vidéo : [Data Items](#)

Exercice 7 : Application sur Wearables

Développer une application Wear OS qui affiche une liste de rappel et qui permet de cocher/-valider les rappels dans une liste.

1. Permettre à l'utilisateur de faire des gestuels pour quitter l'application.
2. Lorsqu'il valide un rappel, la montre garde les coordonnées GPS du lieu où le rappel a été validé.
3. Gérer le cas où l'utilisateur n'interagit pas avec l'application pendant plus de 5 secondes. Il faut utiliser AmbiActive Support.

Android Samples : Voir 'Speed Tracker' pour l'utilisation du GPS. Voir l'exemple 'Always On' pour la gestion de la batterie et de l'écran. **Documentation :** [Android Wear GPS](#)