

## Exercice 2 : ListView et compagnie !

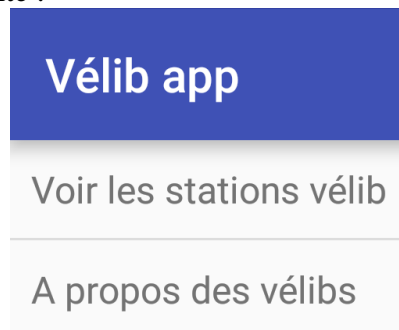
Vous allez, dans ce TP, construire une application s'exécutant sous Android. L'application présente tout d'abord une fenêtre avec deux items dans une liste , "Voir les stations Vélib" et "A propos des Velib". L'item "A propos des Vélib" amène un nouvel écran de renseignements sur les Vélibs, l'item "Voir les stations Vélib" amène un écran proposant la liste des stations Vélib. On interrogera pour cela l'URL <http://www.velib.paris.fr/service/carto>.

### L'interface graphique

**Première étape : Ecrire une Activity qui présente une ListView avec 2 items "Voir les stations Vélib" et "A propos des Vélib"**

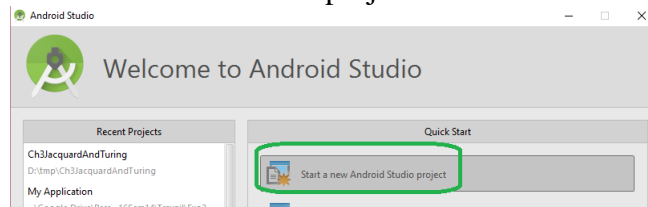
**Première étape : Construire une IHM affichant une ListView**

On veut construire l'IHM suivante :

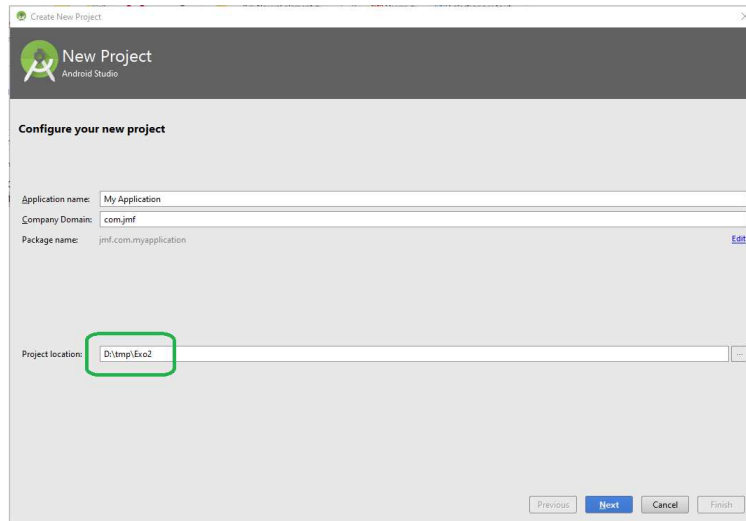


Les principales étapes de développement sont décrites ci-dessous.

1°) Lancer Android Studio et créer un nouveau projet :



Préciser, dans la fenêtre suivante, un nom de nouveau répertoire :



Cliquez le bouton Next.

Dans la fenêtre suivante, cliquez sur l'onglet 1: Project



Vous allez suivre les étapes du cours pour créer la `ListView` avec ses 2 items.

2°) Créer le fichier `res/layout/list_item.xml` représentant l'IHM des items de la liste (voir le cours).

3°) Dans le fichier `res/layout/activity_main.xml` représentant le premier écran de lancement de l'application, indiquer que vous devez avoir une `ListView`. Ce fichier `res/layout/activity_main.xml` doit donc avoir pour contenu :

-----  
fichier res/layout/activity\_main.xml  
-----

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ListView android:layout_height="wrap_content"
        android:layout_width="match_parent" android:id="@+id/listMenu">
    </ListView>
</LinearLayout>
```

Remarque : sous Android Studio, pour indenter correctement (un fichier XML, ou un code Java), utiliser CTRL+ALT+I. Voir aussi le menu Code.

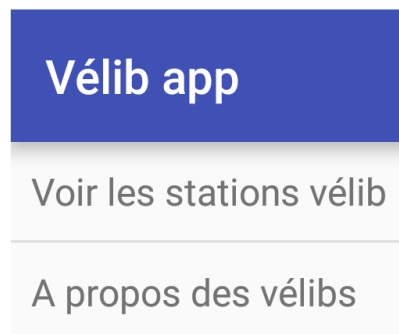
Par quel identifiant dans votre code Java, pourrez-vous manipuler cette `ListView` ?

4°) Ecrire, dans l'Activity de lancement de votre application, le code Java qui récupère la `ListView` et qui l'alimente des deux items "Voir les stations vélib" et "A propos des vélib". On repèrera ces deux items par les identifiants `seeAllStations` et `aboutUs`.  
Donne le titre "Vélib app" à votre application.

Remarque : sous Android Studio, pour importer les bonnes classes et interfaces Java, utiliser la combinaison de touche ALT+Return

5°) Lancer l'exécution de votre application.

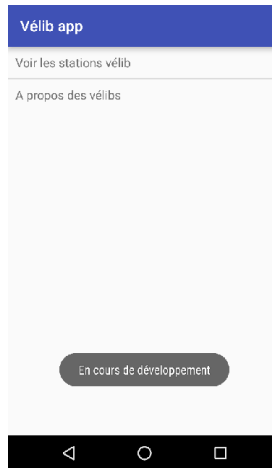
Vous devez obtenir :



## Seconde étape : Prendre en compte les interactions utilisateurs

1°) On veut désormais que :

- lorsque l'utilisateur choisit l'item "Voir les stations vélib", un `Toast` affiche "En cours de développement"



- lorsque l'utilisateur choisit l'item "A propos des vélib", une nouvelle activité contenant un `TextView` qui affiche des renseignements sur les Vélib est lancée.



Ecrire le code nécessaire au traitement des événements. Vérifier que la nouvelle activity est bien déclarée dans `AndroidManifest.xml`.