

M4104C - Développement d'applications mobiles

TP2 – To-do List 1

Le but de ce TP est de créer une version minimaliste d'un gestionnaire de tâches quotidiennes.

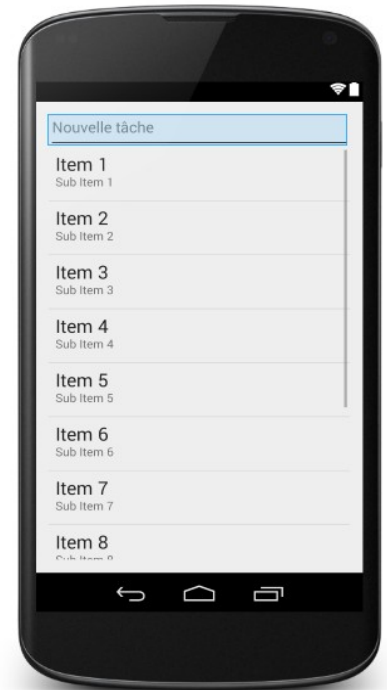
Compétences visées :

- création d'un projet Android
- layout simple
- événement attaché à une view (edittext et bouton)
- listview avec liaison à un tableau de données

1. **Projet To-do List : Ne pas hésiter à utiliser les références du cours dont le site des développeurs principalement (mais Qwant est ton ami ...)**
Inspirez-vous également du projet du tutorial « Building your first app » que vous avez faits dans le TP1

Créer un nouveau projet Android Studio : File->New Project...

- Application name : To-do List
- Company Domain : au choix
- Project location : ce qui vous convient
- Next
- Phone and tablet
- Minimum SDK (API) : plus il est haut plus de nouvelles fonctionnalités sont accessibles mais moins d'appareils pourront utiliser l'application : choisissez API 14 Android 4.0.
- Choisir Basic Activity – Next
- Activity Name : ToDoList
- Title : To-do List
- laisser le Menu ressource name par défaut
- ne pas cocher Use a fragment
- Finish



2. Mettre en place l'interface de l'activité ToDoList en modifiant son layout (activity_to_do_list.xml)
 - Le layout par défaut contient un Textview (zone de text non éditable) à l'intérieur d'un ConstraintLayout (conteneur permettant de disposer des view en respectant des contraintes – à gauche, au dessus, avant, après ... etc.)
 - Les paramètres des vues (éléments d'interface) et des conteneurs de vues sont précisés par chaque « android:... »
 - Modifier ce ConstraintLayout afin qu'il contienne :
 - un EditText prenant toute la largeur de son parent mais juste la hauteur nécessaire et dont le paramètre *hint* sera une chaîne définie dans res/values/string.xml de valeur « Nouvelle tâche »
 - une ListView sous l'EditText prenant toute la largeur de son parent mais juste la hauteur nécessaire
 - Tester votre application (pour l'instant la listview est invisible)
3. Dans le code Java de l'activité, la méthode *onCreate* est déjà présente et met en place l'interface définie dans le layout. Le menu d'option créé dans *onCreateOptionsMenu* peut être laissé mais ne sera pas utilisé pour l'instant.
Ajouter la récupération des références à l'EditText et la ListView de l'interface afin de

pouvoir les utiliser dans le code :

- ajouter d'abord dans le layout un identifiant pour ces 2 vues à l'aide du paramètre « id » (exemple : `android:id="@+id/identifiant_de_la_vue"`)
- dans la classe de l'activité, récupérer les références aux 2 vues grâce à la méthode « `findViewById` » (voir le tutoriel « Building your first app »)
Dans le code qui suit la variable correspondant à l'edittext s'appelle *mavariabaleEditText* et celle de la listview s'appelle *mavariabaleListView*.
- Il faudra ajouter les imports à ces 2 classes en en-tête (cliquer sur le nom de la classe en rouge puis faire Alt+Entrée sous Android Studio – import classe).
L'ajout des imports devra être fait pour chaque nouvelle classe utilisée par la suite.

4. Toujours dans *onCreate*, ajouter un écouteur de touche à l'*EditText* par un appel à *setOnKeyListener* qui prend en paramètre une nouvelle instance de *OnKeyListener* redéfinissant la méthode *onKey* :

```
mavariabaleEditText.setOnKeyListener(new OnKeyListener() {  
    public boolean onKey(View v, int keyCode, KeyEvent event) {}  
});
```

Dans *onKey*, nous allons déclencher l'ajout de la tâche à la liste par un appui sur la touche volume bas :

```
if (event.getAction() == KeyEvent.ACTION_DOWN)  
    if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN)  
    {  
        mavariabaleEditText.setText(""); // 3 - remise à vide de l'EditText  
        return true;  
    }  
return false;
```

Tester l'application

5. Définir un attribut privé *todoitems* de type *ArrayList<String>* pour stocker les éléments de la liste de tâches au fur et à mesure de leur saisie.

L'instancier dans *onCreate*.

Ajouter dans *onKey* avant la ligne // 3 :

```
todoItems.add(0, mavariabaleEditText.getText().toString()); // 1
```

Cet appel déclenche l'ajout dans la liste de chaînes du texte récupéré de l'*EditText*

6. Pour faire apparaître la liste de chaînes dans l'interface, il faut la relier à la *ListView* .

Pour cela, définir un nouvel attribut privé *aa* de type *ArrayAdapter* :

```
private ArrayAdapter<String> aa;
```

Ajouter dans *onCreate* l'instanciation de cet adapter qui va relier *todoitems* à une liste de chaînes simple définie par la constante android *simple_list_item_1* :

```
aa = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,  
                             todoItems);
```

Il reste à mettre en place l'adapter sur la *ListView* :

```
mavariabaleListView.setAdapter(aa);
```

7. Maintenant que la liste de chaînes est reliée à la listview, il faudra notifier tout changement (ajout, suppression, modification de la liste) à l'*arrayadapter* afin que l'affichage soit mis à jour dans la listview. Pour cela, ajouter dans *onKey* entre // 1 et // 3 :

```
aa.notifyDataSetChanged(); // 2
```

Tester l'Application

8. A vous de jouer : ajouter un bouton à droite de l'EditText qui déclenche l'ajout dans la liste des tâches.

Pour bien organiser le layout linéaire, revoir Add a Button dans le premier tutorial fait lors du TP 1 : <http://developer.android.com/training/basics/firstapp/building-ui.html>

