Programmation Android
TP2 –ListView

Dans ce TP, vous créez et construisez une activité qui contient une zone de texte, un bouton et une ListView. A la suite d'un clic sur ce bouton, on va ajouter l'élément saisi à la ListView (la ListView est remplie par les éléments saisis dans la zone de texte).

Partie 1 : ListView simple

Etape 1 : Créer un projet nommé **Liste_view_simple** avec **Empty activity**,

Etape 2 : Créer le fichier xml initial (**activity_main.xml**) dans lequel vous déclarez la liste. L'interface ressemble à celle-ci:

Voici quelques attributs les plus utilisées pour assurer la bonne représentation des ListViews.

```
<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@id/list"
    android:divider = "#00ffff"
    android:dividerHeight = "1dp"
    android:listSelector="#00ffff "
    />
```

Avec:
-divider: séparateur entre les item,
-dividerHeight : l'épaisseur d'un séparateur,
-listSelector: la couleur attribuée à l'item sélectionné.

Etape 3 :

Dans le MainActivity.java , vous créez le code java pour :

1. Définir la source de données (données saisies dans la zone de texte)

```
ArrayList<String> sr=new ArrayList<String>();
```

2. Créer L'objet ArrayAdapter :
   ArrayAdapter est l'adaptateur adéquat pour les listes simples dont les éléments sont des chaines de caractères, il dispose par défaut un layout contient un seul textview. ArrayAdapter est créé comme suite :

```
adp=new ArrayAdapter<String>( MainActivity.this, android.R.layout.simple_list_item_1,sr);
```

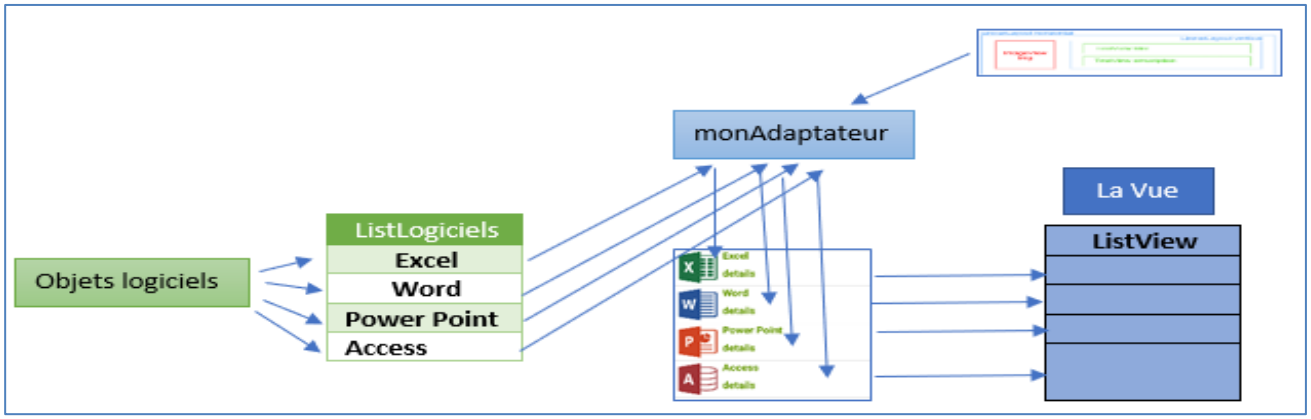
3. Attribuer l'adaptateur à votre ListView définie.
```
listV.setAdapter(adp);
```

La ListView est remplie par les éléments saisis dans la zone de texte à la suite d'un clic sur le bouton « Ajouter ».

4. Récupérer la donnée sélectionnée à la suite d'un clic sur item d'une ListeView et l'afficher dans un Toast. Vous utilisez la méthode **setOnItemClickListener**. Elle permet d'associer une action lors de la clique sur une ligne de ListView.

Partie 2 : ListView personnalisée

On crée une ListView dont les données sont les logiciels.



Etapes à suivre :

1. Créer une activité qui contient une ListView ;
2. Créer un "**Layout ressources file**" pour définir la vue d'un élément de la ListView (modèle de la listItem) :

Ajouter 2 TextView + une image

3. Création de la classe des éléments de la liste : vous créez une classe **logiciels** pour définir les setters et les getters, et un constructeur d'initialisation.
4. Créer une nouvelle **classe adaptateur héritant** de ArrayAdapter<logiciels> :

- Générer un constructeur pour initialiser les champs ;
- Redéfinir la méthode **getView** et utiliser la méthode **inflate(…);**

```
public class monAdapteur extends ArrayAdapter {
    ArrayList<logiciels> listlog;
// inf: un objet permettant de convertir les éléments d'un fichier
layout XML à un nouvel objet de type View .
    LayoutInflater inf;
//Constructeur
    public monAdapteur(@NonNull Context context, int resource,
@NonNull  ArrayList<logiciels> lslog) {
```

```
            super(context, resource, lslog);
            this.listlog = lslog;
            this.inf = LayoutInflater.from(context);
        }
        @NonNull
        @Override
    //méthode getView
        public View getView(int position, @Nullable View convertView,
    @NonNull ViewGroup parent) {
    //accéder au fichier XML (design de item) et d'en récupérer la vue:
            convertView = inf.inflate(R.layout.item_modele, null);
    //récupérer les composants de layout
            TextView titre = convertView.findViewById(R.id.titre);
            TextView desp = convertView.findViewById(R.id.description);
            ImageView img = convertView.findViewById(R.id.logo);
            desp.setText(listlog.get(position).getDesscription());
            titre.setText(listlog.get(position).getTitre());
            img.setImageResource(listlog.get(position).getLogiciel());
            return convertView;
        }
    }
```

5. Ecrire le code Java dans l'activité, dans laquelle vous déclarez la listView, utilisant une instance de **monAdapteteur** pour afficher les données dans la ListView.
    - Etape 1: préparer la source de données (Les données sont stockées dans un ArrayList<logiciels >)
    - Etape 2: créer l'adaptateur personnalisé
    - Etape 3: association entre listeView et adaptateur