

# Exercice 8 Développement Mobile

## Introduction

Le projet a été fait sous IntelliJ en utilisant une VM android pour les exécuteurs. Le téléphone utilisé pour faire fonctionner l'application est un Pixel 5 avec comme résolution  $1080 \times 2340 pixels$ .

Une videos est fournit pour indiquer toutes les fonctionnalités de l'applications

## Quelques class et fonction importante

Cette fonction prend en paramètre 2 String qui ont été écrite dans les EditText de l'application et retourne une liste de d'Itinéraire dont les premières lettres correspondes au caractères donner en paramètres(donc écrites dans les EditText).

```
1 public ArrayList<Itinéraire> firtcharactereequal(String villedepart ,  
    String villearriver){  
2     ArrayList<Itinéraire> retour = new ArrayList<>();  
3     for(Itinéraire i : listItinéraire ){  
4         if(i.equalfirstcaractere(villedepart , villearriver)){  
5             retour.add(i);  
6         }  
7     }  
8     return retour;  
9 }
```

elle utilise la fonction equalfirstcaractere qui retourne simplement un boolean à true si les premières lettres correspondes des villes correspondes au String passé en paramètres et False sinon.

```
1 public boolean equalfirstcaractere(String villedepart , String villearriver  
    ) {  
2     return this.villedepart.matches(villedepart + "(.*)") && this.  
        villearriver.matches(villearriver + "(.*)");  
3 }
```

Cette classe est un adaptateur qui permet d'afficher de façon jolie dans la ListView la liste des itinéraires.

Cette classe va utiliser la layout `intineraire_adaptateur_layout.xml` pour afficher la liste des itinéraires.

```
1 public class ItinéraireAdaptater extends BaseAdapter {
2     private Context context;
3     private List<Itinéraire> list;
4     private LayoutInflater inflater;
5     public ItinéraireAdaptater(Context context, List<Itinéraire> list) {
6         this.context = context;
7         this.list = list;
8         this.inflater = LayoutInflater.from(context);
9     }
10    @Override
11    public int getCount() {return list.size();}
12    @Override
13    public Itinéraire getItem(int i) {return list.get(i);}
14    @Override
15    public long getItemId(int i) {return 0;}
16    @Override
17    public View getView(int i, View view, ViewGroup viewGroup) {
18        view = inflater.inflate(R.layout.itineraire_adaptateur_layout, null)
19        ;
20        Itinéraire e = getItem(i);
21        TextView villed = view.findViewById(R.id.Villedepart);
22        TextView villea = view.findViewById(R.id.villearrivee);
23        TextView date = view.findViewById(R.id.date);
24        TextView heured = view.findViewById(R.id.heuredepart);
25        TextView heurea = view.findViewById(R.id.heurearrivee);
26        villed.setText(e.getVilledepart());
27        villea.setText(e.getVillearrivee());
28        date.setText(e.getDatedepart());
29        heured.setText(e.getHeuredepart());
30        heurea.setText(e.getHeurearrivee());
31        return view;
32    }
```

Cette parti de code permet est appelé directement dès qu'il y a une modification détecté dans l'EditText. Donc à chaque modification l'application va chercher la liste des itinéraire qui correspond au premier caractère écrit dans les EditText.

La liste d'itinéraire sera fournit à l'adaptateur qui permettra ensuite d'afficher cette liste

dans la ListView

```
1 Villedepart.setTextChangeListener(new TextWatcher() {
2     @Override
3     public void onTextChanged(CharSequence s, int start, int before, int
count) {}
4     @Override
5     public void beforeTextChanged(CharSequence s, int start, int count,
int after) {}
6     @Override
7     public void afterTextChanged(Editable editable) {
8         if (!Villedepart.getText().toString().isEmpty()) {
9             ArrayList<Itinéraire> o = list.firtcharactereequal(
Villedepart.getText().toString(), Villearriveur.getText().toString());
10             o.add(0, new Itinéraire(getString(R.string.depart), getString
(R.string.arriver), getString(R.string.Date), getSring(R.string.Datea),
getString(R.string.Heuredepart), getString(R.string.Heurearriveur), 0));
11             adapter = new IntinéraireAdaptater(MainActivity.this, o);
12             listView.setAdapter(adapter);
13         }
14     }
15 });
```

Fonction qui permet d'initialiser l'application avec quelque exemple d'itinéraire.

```
1 public void init(){
2
3     list = new IntinéraireList();
4
5     list.add(new Itinéraire("Paris", "Montpellier", "03/03/2015", "
03/03/2015", "14H30", "15H30", 50));
6     list.add(new Itinéraire("Paris", "Montpellier", "23/03/2015", "
23/03/2015", "19H30", "21H30", 70));
7     list.add(new Itinéraire("Paris", "Montpellier", "14/03/2015", "
14/03/2015", "11H30", "13H30", 60));
8     list.add(new Itinéraire("Paris", "Montpellier", "28/11/2015", "
28/11/2015", "14H30", "15H30", 55));
9     list.add(new Itinéraire("Paris", "Montpellier", "21/03/2015", "
21/03/2015", "7H30", "9H30", 45));
10    list.add(new Itinéraire("Paris", "Montpellier", "03/03/2015", "
03/03/2015", "14H30", "15H30", 50));
```

```
03/03/2015", "14H30", "15H30",50));  
11 list.add(new Itinéraire("Paris","Montpellier","23/03/2015", "23/03/2015", "19H30", "21H30",70));  
12 list.add(new Itinéraire("Paris","Montpellier","14/03/2015", "14/03/2015", "11H30", "13H30",60));  
13 list.add(new Itinéraire("Paris","Montpellier","28/11/2015", "28/11/2015", "14H30", "15H30",55));  
14 list.add(new Itinéraire("Paris","Montpellier","21/03/2015", "21/03/2015", "7H30", "9H30",45));  
15 .....
```

---