

# Application de gestion d'horaires de lignes de trains

## Introduction

Le projet a été fait sous Android Studio en utilisant mon Honor 9 comme appareil Android pour exécuter les projets.

## Description du projet

Ce projet est constitué de trois activités (MainActivity, RoutesList et ResultRoutes) et d'une classe (Trajet). Il nous permet d'avoir une application de gestion des horaires sur les lignes de trains. On peut donc avoir accès à une liste de lignes correspondant à notre recherche.

## MainActivity

La première activité est la page d'accueil de l'application. C'est sur celle ci que l'on peut rentrer les informations que l'on souhaite comme la ville de départ et la ville d'arrivée. Ensuite, elle contient aussi deux boutons (Button). Ces boutons servent l'un à accéder à l'activité ResultRoutes qui liste les lignes de trains disponibles par rapport à nos informations et l'autre, à accéder à l'activité RoutesList qui liste tous les trajets de trains possibles. Il faut savoir que dans l'application les trajets sont générés aléatoirement à partir d'une liste de villes composées de 5 villes (Montpellier, Marseille, Nice, Toulouse et Perpignan).

```
1 public ArrayList<Trajet> makeListTrajet() {
2     ArrayList<String> ville = new ArrayList<String>();
3     ville.add("Montpellier");
4     ville.add("Toulouse");
5     ville.add("Marseille");
6     ville.add("Perpignan");
7     ville.add("Nice");
8
9     ArrayList<Trajet> trajets = new ArrayList<Trajet>();
10    int random = (int)(Math.random() * 80);
11    for(int i = 0; i <= random; i++){
12        int rand1 = (int)(Math.random() * 5);
13        int rand2 = (int)(Math.random() * 5);
14        while(rand2 == rand1){
15            rand2 = (int)(Math.random() * 5);
```

```

16     }
17     int hour = (int)(Math.random() * 24);
18     int minute = (int)(Math.random() * 60);
19     trajets.add(new Trajet(ville.get(rand1), ville.get(rand2), "" +
hour + "h" + minute));
20 }
21 return trajets;
22
23 }

```

La classe Trajet utilisée ici a été créée afin d'avoir accès à des objets ayant une ville de départ et une ville d'arrivée. C'est à dire des objets qui correspondent aux lignes de trains. De cette manière, il était bien plus simple de manipuler ces objets et ensuite d'en faire une liste, comme vu au-dessus.

Suite à des problèmes d'envoi d'objets par le biais des intents, la classe Trajet a été rendue "Parcelable" afin de pouvoir envoyer des instances de celle ci d'une activité à une autre.

```

1 public class Trajet implements Parcelable {
2
3     private String cityFrom, cityTo, time;
4
5     public Trajet(String cityFrom, String cityTo, String time){
6         this.cityFrom = cityFrom;
7         this.cityTo = cityTo;
8         this.time = time;
9     }
10
11     protected Trajet(Parcel in) {
12         cityFrom = in.readString();
13         cityTo = in.readString();
14         time = in.readString();
15     }

```

## AddEvent et ResultRoutes

Ces deux activités ont un fonctionnement assez similaire dans le sens où elles affichent toutes les deux la liste de trajet. Seulement, il y a un petit test avant dans ResultRoutes

car, dans celle-ci, on ne doit afficher que les trajets correspondant à notre recherche. Des test supplémentaires sont réalisés afin de pouvoir accéder à la recherche de l'utilisateur si il veut par exemple avoir tous les trajets en provenance de Montpellier ou tous les trajets en direction de Marseille....

```
1  for (Trajet tr : listTrajets){
2      String routeCityFrom = tr.getCityFrom().toString();
3      String routeCityTo = tr.getCityTo().toString();
4      if (routeCityFrom.equals(userCityFrom)) {
5          if (routeCityTo.equals(userCityTo)) {
6              TextView newRoute = new TextView(ResultRoutes.this);
7              newRoute.setText(routeCityFrom + "→" + routeCityTo + " at " +
tr.getTime());
8              ll.addView(newRoute);
9          }
10         else if (userCityTo.equals("")){
11             TextView newRoute = new TextView(ResultRoutes.this);
12             newRoute.setText(routeCityFrom + "→" + routeCityTo + " at " +
tr.getTime());
13             ll.addView(newRoute);
14         }
15     }
16     else if (userCityFrom.equals("")){
17         if (routeCityTo.equals(userCityTo)){
18             TextView newRoute = new TextView(ResultRoutes.this);
19             newRoute.setText(routeCityFrom + "→" + routeCityTo + " at " +
tr.getTime());
20             ll.addView(newRoute);
21         }
22         else if (userCityTo.equals("")){
23             TextView newRoute = new TextView(ResultRoutes.this);
24             newRoute.setText(routeCityFrom + "→" + routeCityTo + " at " +
tr.getTime());
25             ll.addView(newRoute);
26         }
27     }
28 }
```

## **A Noter**

Pour le bon fonctionnement de l'application, quand on rentre les villes, il faut s'assurer qu'il n'y ait pas d'espace à la fin et que la première lettre soit en majuscule.