

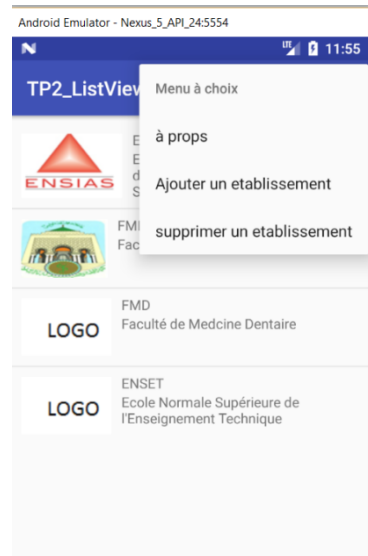
TP2 (RecyclerView)



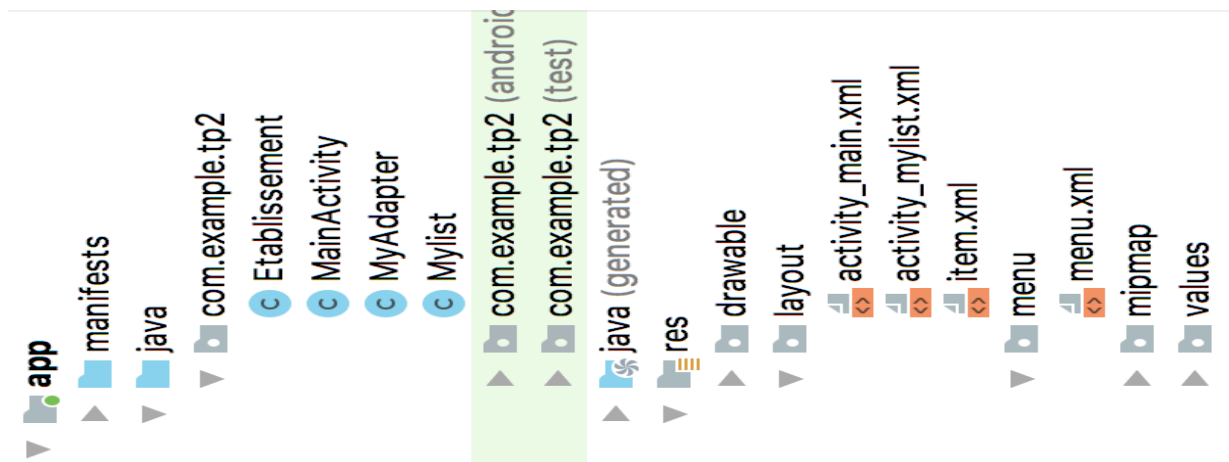
Interface 1



Interface : ListView



Interface 3 : Menu



L'objectif de ce TP est l'initiation à la création des interfaces graphiques en utilisant les layouts, vues, RecyclerView, Menu et l'adaptateur personnalisé.

La structure (ci-dessus) et les consignes (ci-dessous) sont présentées pour vous guider à réaliser ce TP.

NB : Vous pouvez procéder autrement pour avoir le même résultat.

Etape 1 :

1. Ajouter dans builder.gradle puis synchronisez votre gradle (la version est en relation avec le compileSdkVersion utilisé)
- 2.

```
implementation 'com.android.support:recyclerview-v7:29.1.1'
```

Etape 2 :

3. Insérer les différents contrôles graphiques (TextView, EditText... Button) dans le layout principal du projet voir interface 1:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Bienvenue"
        ...
    />
    ...
</>
```

4. Ajoutez une activité « Mylist » pour gérer le recyclerView.
5. Dans le layout « activity_mylist.xml ajoutez le RecyclerView

```
<android.support.v7.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/list"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
```

6. Ajouter un layout « item.xml » contenant la vue à afficher sur la liste d'établissements. Les informations à afficher sont (logo avec id 'img', abréviation avec id 'label', nom avec id 'name'). De préférence choisissez le type constraintLayout pour votre layout
NB : N'oubliez pas de mettre après la pause de vos vues la hauteur du layout à « wrap_content »
7. Dans l'activité principale « Main_activity » charger le layout « activity_main.xml » par le setContentView.
8. Créer la classe Etablissement.

```
public Etablissement(String label, String name, int img ){  
  
    this.label= label;  
    this.name= name;  
    this.imag=img;  
}
```

Ajouter les méthodes pour récupérer les informations relatives à l'établissement (getlabel, getdesc, getimage) qui retourne respectivement le label, la description et l'image et qui seront utilisées par l'adaptateur.

9. Ajouter un adaptateur personnalisé «MyAdapter » qui va jouer le rôle d'interface entre les données et les Vues. Utilisez la classe déposée sur Classroom.
10. Dans l'activité Mylist Récupérez le RecyclerView en lui associant le LayoutManager, les données et l'adapter. Cette activité sera lancée grâce au bouton « se connecter » de la classe Main_activit.

```
public class Mylist extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // get the reference of RecyclerView  
        RecyclerView rv = (RecyclerView) findViewById(R.id.list);  
        // set a LinearLayoutManager with default vertical orientation  
        rv.setLayoutManager(new LinearLayoutManager(this));  
        // call the constructor of MyAdapter to send the reference and data to Adapter  
        rv.setAdapter(new MyAdapter(this, Etablissements));  
    }  
}
```

11. Pour gérer le menu à choix, ajoutez dans l'activité « Mylist » la méthode
« onCreateOptionsMenu »

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // TODO Auto-generated method stub
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return super.onCreateOptionsMenu(menu);
}

```

NB : Ajoutez dans res, le dossier menu et le fichier menu.xml → (res/new/android resource File et donnez menu comme nom)

Allez dans menu.xml puis créez les items ('à propos' avec item1 comme id, 'Ajouter un établissement' avec item2 comme id, 'Supprimer un établissement' avec item3 comme id)

12. Pour gérer les interactions avec le menu, ajoutez aussi la méthode

« onOptionsItemSelected »

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch(item.getItemId()) {
        // On peut créer le menu via le code
        case R.id.item1: Toast.makeText(this, "à propos",
            Toast.LENGTH_LONG).show(); break;
        case R.id.item2: Toast.makeText(this, "Ajouter un établissement",
            Toast.LENGTH_LONG).show(); break;
        case R.id.item3: Toast.makeText(this, "supprimer un établissement",
            Toast.LENGTH_LONG).show(); break;
    }

    return super.onOptionsItemSelected(item);
}

```

13. A vous : Personnaliser votre Application :

- 13.1. Utiliser les fragments (pour votre application ex pour chaque ville ses établissements)
- 13.2. Séparer les lignes de votre liste RecyclerView: ItemDecoration
- 13.3. Ajouter une animation à votre liste : ItemAnimator
- 13.4. Revoir vos IHM (cf Material Design)
- 13.5. ...