Documentation Android

Imaginons que nous avons une liste à faire :

On va dans le manifiest.xml et on reprend la ligne qui autorise internet :

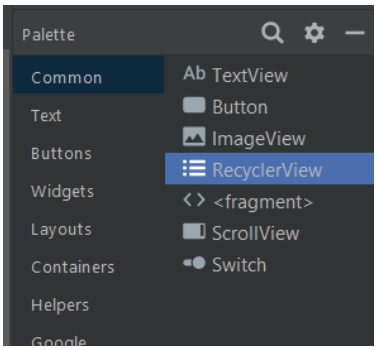
```
<uses-permission android:name="android.permission.INTERNET" /> <!--Donne l'acces à internet-->
```

Ensuite, on va dans le deuxième build.gradle et on met ces deux lignes pour avoir retrofit (ce qui permet de faire des appels API (comme Ajax en js) :

```
implementation 'com.squareup.retrofit2:retrofit:2.6.0' //implementation de retrofit
implementation 'com.squareup.retrofit2:converter-gson:2.6.0' //implementation d'un module de retrofit qui permet de traduire les json
```

Ensuite on crée un layout (on crée une page) en créant une nouvelle activité (click droit fichier racine new activity, empty activity, en ayant bien coché create ressource file)

Dans le layout de la nouvelle activité, prendre la recyclerView :



Ensuite on crée un nouveau layout qu'on va appeler x_cell

Aller dans le code xml de ce layout et dans la balise « constraint_layout » (la 1ere) on va mettre :

android:layout_width="match_parent"

android:layout_height="150dp">

Ensuite dans cette cellule on met ce que doit contenir la cellule (ex si nom mettre textView, avec un text view par élément (Deux textViews si nom prenom))

Tout d'abord on va préparer retrofit en créant une class APIClient :

```
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class ApiClient {

    private static final String BASE_URL = "http://serveur1.arras-sio.com/symfony4-4063/projetppeb2/public/api/";
```

```
    private static Retrofit getRetrofit(){ //Fonction pour faire des appels(call)
        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();

        return retrofit;
    }

    public static WebServiceInterface getInnovAnglaisService(){ //Appel
l'interface WebServiceInterface pour choisir l'API qu'on appel.
        WebServiceInterface webServiceInterface =
getRetrofit().create(WebServiceInterface.class);
        return webServiceInterface;
    }
}
```

Url à modifier si besoin

Changer le nom du getInnovAnglaisService par ce qu'il faut.

Ensuite on crée la classe de la table qu'on va appeler (ex : niveau), mettre dans cette classe :

Instancier les string, int etc…

Au-dessus de chaque instanciation mettre :

```
@SerializedName("id")
int id;
```

(On est obligé de mettre le même nom que le nom de champs de l'api)

Faire le constructeur et le getteur

Ensuite on crée l'interface WebServiceInterface.

```
@GET("niveaux.json") //Complete le "base url" dans APIClient pour avoir l'url
complet de l'API
Call<Niveaux[]> getListNiveaux(); //Comme le json est sous forme de tableau, on
appel un tableau([])
```

Attention ! Si on appelle une liste, on devra mettre tableau ([]), si on appel qu'un élément (ex qu'un utilisateur), il ne faudra pas les mettre.

On va ensuite créer l'adapteur

Par exemple pour niveau,

```
public class NiveauListAdapter extends
RecyclerView.Adapter<NiveauListAdapter.ViewHolder> {
```

On copie colle cette ligne, elle sera soulignée en rouge, il faudra passer la souris dessus et faire « implement method »

Ensuite on remplit les méthodes générées en se basant sur :

```
List<Niveaux> listNiveaux; //On instancie (ici on crée une liste des niveaux)
OnNiveauListClickListener listener; //On appel l'interface pour le OnClick

public NiveauListAdapter (List<Niveaux> listNiveaux, OnNiveauListClickListener
listener){
    this.listNiveaux = listNiveaux;
    this.listener = listener;
```

```
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
//Ici on a la vue (ce qui va s'afficher)
    View listView =
LayoutInflater.from(parent.getContext()).inflate(R.layout.niveau_cell,parent,false
);
    ViewHolder viewHolder = new ViewHolder(listView);
    return viewHolder;
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) { //Reprend
les objets niveaux et les inserent dans les cellules
    holder.difficulteTextView.setText(listNiveaux.get(position).getLibelle());

Picasso.get().load(listNiveaux.get(position).getImage_url()).into(holder.imageView
);
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) { //Nous renvoie les informations dans la
cellule
            listener.OnNiveauListClickListener(listNiveaux.get(position));
        }
    });
}

@Override
public int getItemCount() {
    return listNiveaux.size();
}  //Renvoie le nbr de cellule(item) qu'il y a dans la liste

public class ViewHolder extends RecyclerView.ViewHolder { //Permet d'appeler les
views dans la cellule
    TextView difficulteTextView;
    ImageView imageView;

    public ViewHolder(@NonNull View listNiveaux){
        super(listNiveaux);
        this.difficulteTextView =
listNiveaux.findViewById(R.id.difficulteTextView);
        this.imageView = listNiveaux.findViewById(R.id.niveauImageView);
    }

}
```

(Ce qui est écrit en rouge est en cas de OnClick)

La classe viewHolder ne sera pas générée, elle sera écrite

C'est normal qu'il y ait bcp d'erreurs.

Si jamais on veut faire un onClick :

On crée l'interface OnListClickListener :

```
public interface OnNiveauListClickListener {
    void OnNiveauListClickListener(Niveaux niveaux); //Interface qui permet le
```

```
onClick
}
```

S'il y a un onclick, lors de l'appel de la classe, alors on fait class nomDeLaClass(toute première ligne de l'activity) implements et le nom de l'interface du onclick et faire passer la souris avec un implement method. (Dans l'activité dans laquelle on affiche la recyclerView)

Ensuite, on va dans la vue (le java de l'activité), on instancie une RecyclerView et une List<Niveaux>

Ensuite dans le publicView onCreateView :

```
recyclerView = findViewById(R.id.choixTestRecyclerView); //On appel la vue
recyclerView.setLayoutManager(new LinearLayoutManager(this)); //Dispose la liste
en plusieurs lignes
```

On crée une fonction void pour faire l'appel

On n'oublie pas d'appeler dans le OnCreateView

Dans la fonction void, on fait le call :

```
Call<Niveaux[]> niveauCall =
ApiClient.getInnovAnglaisService().getListNiveaux();//On appel l'API Niveau grace
à ApiClient et WebServiceInterface
niveauCall.enqueue(new Callback<Niveaux[]>() {
    @Override
    public void onResponse(Call<Niveaux[]> call, Response<Niveaux[]> response) {
        for(int i = 0; i < response.body().length; i++){ //Permet de parcourir
toutes les cases du tableau json
            listNiveau.add(new Niveaux(response.body()[i].getIdDuNiveau(),
response.body()[i].getLibelle(), response.body()[i].getImage_url()));//rempli la
liste des niveaux avec les informations qu'on souhaite
        }
        recyclerView.setAdapter(new NiveauListAdapter(listNiveau,
ChoixTestFragment.this::OnNiveauListClickListener)); //Va inserer les données de
la liste dans la recyclerView
    }

    @Override
    public void onFailure(Call<Niveaux[]> call, Throwable t) {
        System.out.println("Fail");
    }
});
```