

Développement Mobile

Cours : RecyclerView

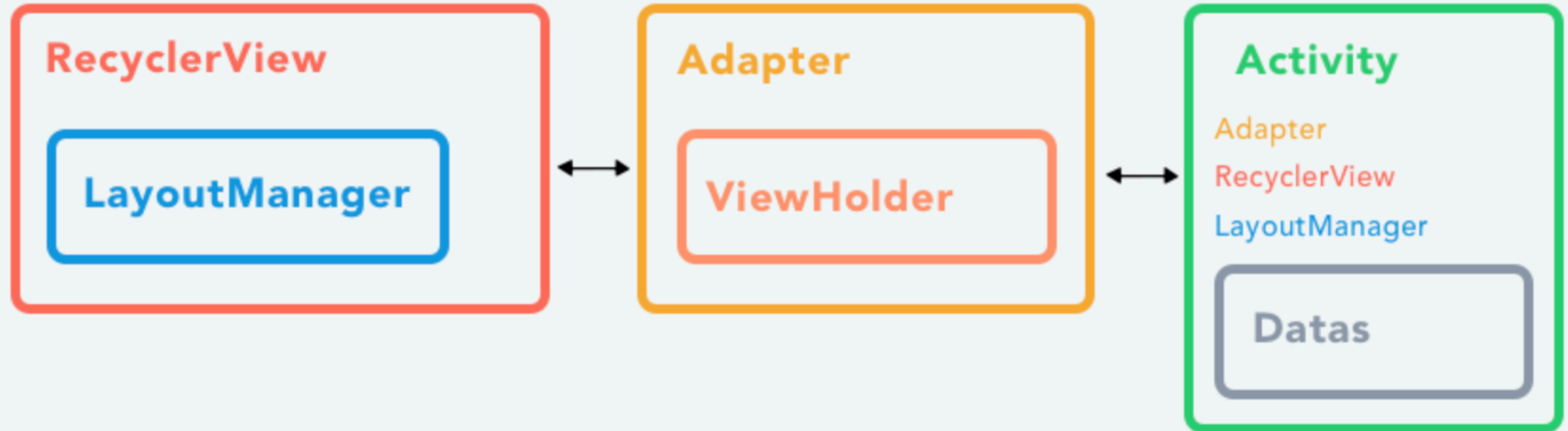
Année universitaire 2019/2020



Qu'est-Ce qu'une RecyclerView ?



- Digne successeur de la [ListView](#) et de la [GridView](#), la [RecyclerView](#) s'impose en étant un pattern plus performant et permettant une personnalisation plus poussée que ces prédécesseurs.
- Pour fonctionner, cette dernière a besoin de trois composants obligatoires lors de son implémentation
 - Un Adapter ([RecyclerView.Adapter](#)) : Permet de faire la liaison (Bind) entre la vue RecyclerView et une liste de données.
 - Un LayoutManager ([RecyclerView.LayoutManager](#)) : Permet de positionner correctement l'ensemble des données de la liste.
 - Un ViewHolder ([RecyclerView.ViewHolder](#)) : Permet de représenter visuellement un élément de la liste de données dans le RecyclerView (Une ligne).



Architecture d'une RecyclerView



Implémenter une RecyclerView



- Créer un nouveau projet intitulé RecyclerViewNomPrenom
- Ajoutez la bibliothèque de support à la dependencies section

implementation 'com.google.android.material:material:1.0.0'

- Commencez par créer une classe Product.java avec les champs suivant:

Un `String id`, un `String name` et un `int image`.

- Définissez un constructeur avec des paramètres, puis définissez les paramètres Getters and Setters.



- Créez le fichier *items.xml* avec un parent *CardView* :

```
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">
</androidx.cardview.widget.CardView>
```

- Ajouter ensuite une *ImageView* et un *TextView* dans cette *CardView* pour ressembler à cette figure



Word 1



- Ensuite Créez une nouvelle classe ***MyAdapter.java***

```
public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {  
  
    @Override  
    public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
        return null;  
    }  
  
    @Override  
    public void onBindViewHolder(MyAdapter.ViewHolder holder, int position) {  
  
    }  
  
    @Override  
    public int getItemCount() {  
        return 0;  
    }  
  
    public class ViewHolder extends RecyclerView.ViewHolder {  
        public ViewHolder(View itemView) {  
            super(itemView);  
        }  
    }  
}
```



- Créer 2 paramètre

```
private Context mContext;  
private List<Product> productList;
```

- Modifier les méthodes de l'adapter comme suit:

```
@Override  
public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    View itemView = LayoutInflater.from(mContext).inflate(R.layout.items, parent, false);  
    return new ViewHolder(itemView);  
}  
  
...  
  
@Override  
public int getItemCount() {  
    return productList.size();  
}
```



- Pour la classe ViewHolder :

```
public class ViewHolder extends RecyclerView.ViewHolder {  
  
    private ImageView imgProduct;  
    private TextView txtProduct;  
  
    public ViewHolder(View itemView) {  
        super(itemView);  
  
        imgProduct = itemView.findViewById(R.id.imgProduct);  
        txtProduct = itemView.findViewById(R.id.txtProduct);  
    }  
}
```


- 
- 
- Pour finir avec l'adapter, modifier la méthode

```
@Override
public void onBindViewHolder(MyAdapter.ViewHolder holder, int position) {
    Product singleItem = productList.get(position);
    holder.txtProduct.setText(singleItem.getName());
    holder.imgProduct.setBackgroundResource(singleItem.getImage());
}
```

- Et créer un constructeur avec paramètres

```
public MyAdapter(Context mContext, List<Product> productList) {
    this.mContext = mContext;
    this.productList = productList;
}
```



- Maintenant dans activity_main.xml, créez un <RecyclerView /> avec un id : list

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/list"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

- Dans la classe MainActivity.java :
 - Créer un paramètre

```
private List<Product> productList = new ArrayList<>();
```



- Dans la méthode `onCreate` :
- Ajouter quelques produits

```
productList.add(new Product("1","Product1",R.mipmap.ic_launcher));  
productList.add(new Product("2","Product2",R.mipmap.ic_launcher));  
productList.add(new Product("3","Product3",R.mipmap.ic_launcher));  
productList.add(new Product("4","Product4",R.mipmap.ic_launcher));  
productList.add(new Product("5","Product5",R.mipmap.ic_launcher));
```

- Ensuite instancier votre `recyclerView` comme suit

```
RecyclerView recyclerView = findViewById(R.id.list);  
recyclerView.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false));  
recyclerView.setAdapter(new MyAdapter(this , productList));
```