Android Advance
Lesson 1
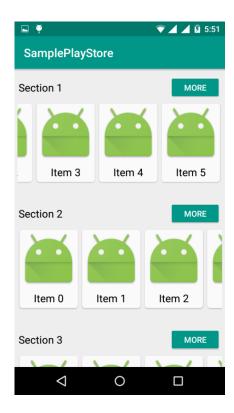# Recyclerview

Outline

I. **Introduction to recyclerView**

II. **LayoutManager**

III. **Adapter**

I. Introduction to RecyclerView

RecyclerView?

Recyclerview use for?

- Use to display a list of data

- Usually views have the same layout layout

- Used to display more data of the same type

- The number of rows in the list is fixed by the data passed in and the type of information presented

- Have the basic properties of a View

3. Implement RecyclerView

- Add this line into file **build.gradle**

compile 'com.android.support:recyclerview-v7:xx.xx.xx'

II. LayoutManager

LayoutManager

➢ A LayoutManager is responsible for measuring and positioning item views within a RecyclerView

➢ Have 2 LayoutManager:
  ○ LinearLayoutManager: A LayoutManager which provides similar functionality to ListView.
  ○ GridLayoutManager: A LayoutManager that lays out items in a grid.
  ○ StaggeredGridLayoutManager: A LayoutManager that lays out children in a staggered grid formation
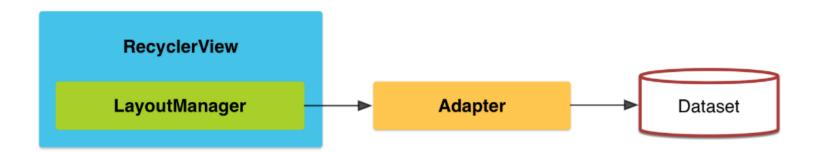
LayoutManager (cont)

Linear

Staggered

Grid

III. Adapter

1. What is Adapter?

➢ Adapters provide a binding from an app-specific data set to views that are displayed within a <u>RecyclerView</u>.

2. ViewHolder

➢ Describes an item view and metadata about its place within the RecyclerView.

➢ **Sample code**:

```java
public class RecyclerAdapter extends RecyclerView.Adapter<RecyclerAdapter.ViewHolder>{
 // ViewHolder là class quyết định hình dạng của từng item trong view
   public class ViewHolder extends RecyclerView.ViewHolder {
      CardView cv;
      TextView personName;
      TextView personAge;
      ImageView personPhoto;

      ViewHolder(View itemView) {
         super(itemView);
         cv = (CardView)itemView.findViewById(R.id.cv);
         personName = (TextView)itemView.findViewById(R.id.person_name);
         personAge = (TextView)itemView.findViewById(R.id.person_age);
         personPhoto = (ImageView)itemView.findViewById(R.id.person_photo);
      }
   }
}
```

3. More important methods

➢ **Method: "getItemCount()"**
  ○ Returns the number of items in the recyclerView if you want an array to be the size of the array
➢ **Sample code:**

```java
@Override
public int getItemCount() {
    return persons.size();
}
```

4. More important methods (cont)

➤ **Method "onCreateViewHolder()"**
  ○ This method is called when the custom ViewHolder needs to be initialized.

➤ **Sample code:**

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {
    View v = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.item, viewGroup, false);
    ViewHolder vh = new ViewHolder(v);
    return pvh;
}
```

4. More important methods

➢ **Method "onBindViewHolder()"**

○ To determine the contents of each element of the RecyclerView. This method is very similar to the getView method of a ListView adapter.

➢ **Sample code:**

```java
@Override
public void onBindViewHolder(ViewHolder viewHolder, int i) {
    viewHolder.personName.setText(persons.get(i).name);
    viewHolder.personAge.setText(persons.get(i).age);
    viewHolder.personPhoto.setImageResource(persons.get(i).photoId);
}
```

Q&A

Exercise

- Using RecyclerView create List Hero like image bellow: