RecyclerView

Jigisha Patel

RecyclerView

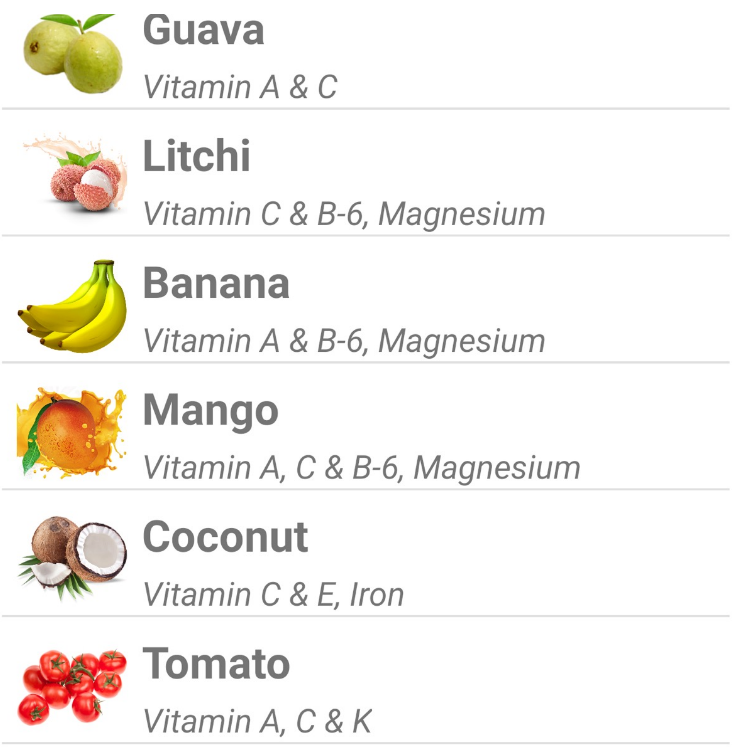
- RecyclerView makes it easy to efficiently display large sets of data.
- The RecyclerView library dynamically creates the elements when they're needed with supplied data and layout.
- When an item scrolls off the screen, RecyclerView doesn't destroy its view.
- Instead, RecyclerView reuses the view for new items that have scrolled onscreen.
- This reuse vastly improves performance, improving your app's responsiveness and reducing power consumption.

RecyclerView cont…

- RecyclerView class is the ViewGroup that contains the views corresponding to your data.
- Each individual element in the list is defined by a view holder (RecyclerView.ViewHolder) object.
- When the view holder is created, it doesn't have any data associated with it. After it is created, the RecyclerView binds it to its data.
- The RecyclerView uses RecyclerView.Adapter methods to request those views and bind the views to their data.
- The layout manager arranges the individual elements in your list which can be customized by defining the layout.

Steps to create RecyclerView

- ✓ Add the RecyclerView widget to the layout
- ✓ Create a Layout that will represent an item in the List
- ✓ Create an adapter and ViewHolder
- ✓ Set a LayoutManager and an instance of the adapter to the RecyclerView



Guava
Vitamin A & C

Litchi
Vitamin C & B-6, Magnesium

Banana
Vitamin A & B-6, Magnesium

Mango
Vitamin A, C & B-6, Magnesium

Coconut
Vitamin C & E, Iron

Tomato
Vitamin A, C & K

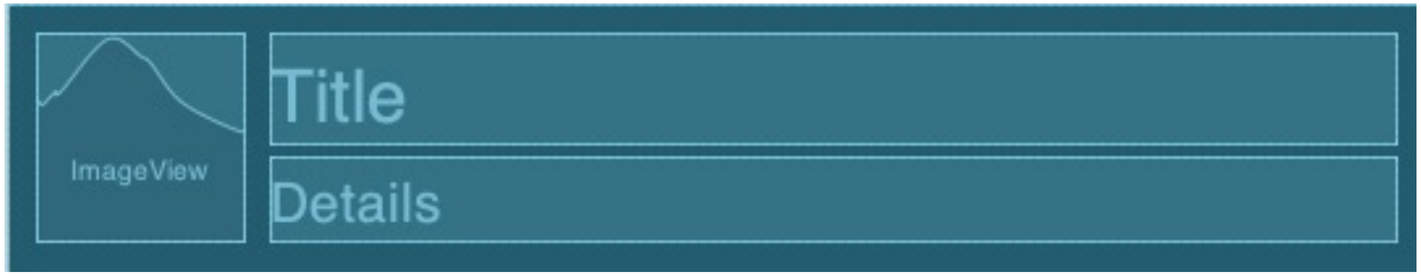
Add the RecyclerView widget

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvFruits"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_margin="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Create a Layout

- The items in your RecyclerView are arranged by a LayoutManager class.
- The RecyclerView library provides three layout managers, which handle the most common layout situations:
  - LinearLayoutManager arranges the items in a one-dimensional list.
  - GridLayoutManager arranges all items in a two-dimensional grid
  - StaggeredGridLayoutManager is similar to GridLayoutManager, but it does not require that items in a row have the same height or items in the same column have the same width.

Example:



Adapter

- An Adapter object acts as a bridge between an AdapterView and the underlying data for that view.
- The Adapter provides access to the data items.
- The Adapter is also responsible for making a View for each item in the data set.

- Types of Adapter classes:
  - ArrayAdapter
  - CursorAdapter
  - SimpleCursorAdapter
  - RecyclerView.Adapter

RecyclerView Adapter

- RecyclerView.Adapter provide a binding from an app-specific data set to views that are displayed within a RecyclerView.
- The Adapter creates ViewHolder objects as needed and sets the data for those views.
- The adapter needs to override onCreateViewholder(), onBindViewHolder() and getItemCount() methods.

ViewHolder

- The ViewHolder is a wrapper around a View that contains the layout for an individual item in the list.
- The process of associating views to their data is called binding.
- The bind method in the ViewHolder is responsible for binding the data with its Views.

onCreateViewHolder()

- RecyclerView calls this method whenever it needs to create a new ViewHolder.
- The method creates and initializes the ViewHolder and its associated View but does not fill in the view's content.

- In this method, the ViewHolder has not yet been bound to specific data.

onBindViewHolder()

- RecyclerView calls this method to associate a ViewHolder with data.

- The method fetches the appropriate data and uses the data to fill in the view holder's layout.

getItemCount()

- RecyclerView calls this method to get the size of the data set.

- RecyclerView uses this to determine when there are no more items that can be displayed.

References

- https://developer.android.com/guide/topics/ui/layout/recyclerview