RecyclerView is a view group representing a large data set that was introduced since Android 5.0. it has a technique that increases the performance of the view. In addition, it is efficient in memory and stores the data that are only present on the screen and on scrolling it will recycle the data. It is more efficient than ListView and GridView.

RecylerView permits us to set the different user interfaces dynamically at run time and have different classes that make it highly customizable. You have provided data and layout RecyclerView will provide you with elements that are needed.

Definition

In RecyclerView Viewhloder will hold the view and helps in recycling. And the Adapter will adapt our data so that it can be displayed on the screen. The layout manager sets up each component in your list individually. You can create your own layout manager or utilize one that the RecyclerView library offers. All layout managers are derived from the LayoutManager abstract class in the library.

If you want to utilize RecyclerView in your application, you will need these component as given below:

RecyclerView.Adapter: it is the main code in a class that is responsible for extending ViewHolder and scrolling view options.

Layout: it is a part of user interface that helps to position the item.

ItemAnimator: As the name suggests it is responsible for animating the item.

How to implement RecyclerView:

To implement ReCyclerView these steps should be follow:

* Create a model class that will serve as the data source.
* To display the items, add a RecyclerView to your activity.
* To see the item, make a custom row layout XML file.
* Make a RecyclerView first to render the object, use ViewHolder and Adapter.
* In order to fill the RecyclerView, bind the adapter to the data source.

CardView:

The CardView extension of Framelayout is used to display objects in the card format. We may increase the radius and elevation of our RecyclerView elements with the aid of CardView. Our list of data has a rich appearance and feel thanks to CardView.

RecyclerView is Further classified into 3 sections:

* Card Layout.
* Modal Class.
* ViewHolder class.

Card Layout:

A layout manager for a container is a CardLayout object. Each element of the container is treated as a card. The container serves as a stack of cards, and only one card is ever visible at once. The component that is initially visible when the container is displayed is the first component added to a CardLayout object.

Modal Class:

The class interval with the most data points, or what we may refer to as the highest frequency, inside a set of data is called the modal class.

ViewHolder class:

he layout for each item in the list is contained in the ViewHolder, which is a wrapper around a View. The Adapter sets the data for those views and builds ViewHolder objects as necessary. Binding is the action of connecting views to their underlying data.

Conclusion:

RecyclerView is the successor of ListView and GridView. It uses the ViewHolder design pattern to function. We can give our list of data a lot more characteristics with the aid of RecyclerView. Prior to beginning our demonstration on how CardView is implemented in RecyclerView. CardView and RecyclerView should be clear to us.