RecyclerView

As the name implies, RecyclerView is the ViewGroup *recycles* the individual elements.

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.

RecyclerView improves performance and your app's responsiveness, and it reduces power consumption.

what is adapter in android?

An Adapter object acts as a bridge between an AdapterView(UI) and the data for that view

Imp Classes in RV :

1)RecyclerView is the ViewGroup that contains the views corresponding to your data.

2)Each individual element in the list is defined by a *view holder* object, after the view holder is created, the RecyclerView *binds* it to its data.You define the view holder by extending RecyclerView.ViewHolder.

3) The RecyclerView requests views, and binds the views to their data, by calling methods in the *adapter.* You define the adapter by extending RecyclerView.Adapter.

4)The *layout manager* arranges the individual elements in your list.

Importnat methos in recyclerview adapter :

•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 contents—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. For example, if the RecyclerView displays a list of names, the method might find the appropriate name in the list and fill in the view holder's TextView widget.

- getItemCount(): RecyclerView calls this method to get the size of the dataset. For example, in an address book app, this might be the total number of addresses. RecyclerView uses this to determine when there are no more items that can be displayed.

  Now to add up the divider item between items of recycleview , we can use code:

  recyclerView.addItemDecoration(

```
    DividerItemDecoration(
      baseContext,
      layoutManager.orientation
    )
)
```