**RecyclerView**

**What is a RecyclerView?**

* A **RecyclerView** is a flexible view for providing a limited window into a large data set. It is essentially a **ViewGroup** of containers called **ViewHolders**which populate a particular item.
* [RecyclerView is an extensive Android class to provide a flexible UI](https://developer.android.com/reference/kotlin/androidx/recyclerview/widget/RecyclerView" \t "_blank). A great benefit of using RecyclerViews is that you’re able to efficiently reuse views instead of managing items that aren’t even visible to a user.

## Why we use RecyclerView?

## In the past, Android was using ListView or GridView classes for displaying lists. The RecyclerView can be described as a combination of a ListView and a GridView.

## However, in a RecyclerView, there are features that separate your code into maintainable components while also enforcing memory-efficient design patterns.

**How is it better than ListViews and GridViews?**

* ListViews and GridViews are comparatively slower than RecyclerViews.

Diagram

Description automatically generated

## Layouts in RecyclerView?

* The **RecyclerView** changed everything which was used in the ListView and GridView.
* RecyclerView still uses an **Adapter** to act as a data source. However, you have to create **ViewHolders** to keep references in memory.
* To provide a new view, **RecyclerView** either creates a new **ViewHolder** object to inflate the layout and hold those references, or it recycles one from the existing stack.
* This is it why **it’s called a RecyclerView!**
* RecyclerView requires a ViewHolder, the **RecyclerView** knows which animation to apply to which item and adds them as required. You can also create your own animations and apply them as needed.
* The most important and interesting component of a RecyclerView is its **LayoutManager**. This object positions the RecyclerView’s items and tells them when to recycle items that have transitioned off-screen. The ListView used to do this work by itself.
* The RecyclerView has broken out of this functionality to allow for different kinds of layouts: Vertical, Horizontal, Grid, Staggered, or your own!

**There are three types of Layout Managers by default:**

1. **LinearLayoutManager:** Positions items to look like a standard ListView.
2. **GridLayoutManager: P**ositions items in a grid format similar to a GridView.
3. **StaggeredGridLayoutManager:** Positions terms in a staggered grid format.

**RecyclerView implementations requires:**

1. A list of **data objects** to work with.
2. An **XML file** of the view item.
3. An **adapter** to bind that data to the views.
4. A **ViewHolder** to populate the UI view from the XML item file.