Recycler View

RecyclerView

- Widget for displaying lists of data
- Successor to ListView with improved performance and features
- "Recycles" (reuses) item views to make scrolling more performant
- Can specify a list item layout for each item in the dataset
- Supports animations and transitions

Why RecyclerView?

- **View Recycling**: Reuses view objects when they scroll off-screen
- **Performance**: More efficient than ListView for large datasets
- **Flexibility**: Supports multiple layout types (linear, grid, staggered grid)
- **Animation**: Built-in animation support for item changes
- **Decoration**: Easily add dividers, spacing, and other decorations

RecyclerView Component

RecyclerView: The ViewGroup that contains the list

Adapter: Provides data to the RecyclerView

ViewHolder: Holds and manages view references

LayoutManager: Arranges items in the RecyclerView

ItemDecoration: Adds visual decorations to items

ItemAnimator: Animates item changes

RecyclerView vs ListView

Feature	ListView	RecyclerView
View Recycling	Basic	Advanced
Layout Types	Vertical only	Multiple layouts
Item Animation	Not built-in	Built-in support
View Holder	Optional pattern	Required pattern
Customization	Limited	Highly customizable

Implementation Steps

- Add RecyclerView dependency
- Add RecyclerView to your layout XML
- Create an item layout XML
- Create a ViewHolder class
- Create an Adapter class
- Set up RecyclerView in your Activity/Fragment

Step 1: Add Dependency

In your app-level build.gradle file:

dependencies {

implementation 'androidx.recyclerview:recyclerview:1.3.2'

}

// Add RecyclerView dependency
implementation(libs.androidx.recyclerview)

Or already available in material library

Step 2: Add RecyclerView to Layout

```
<!-- activity_main.xml -->
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Step 3: Create Item Layout

```xml
<!-- item_user.xml -->
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
            android:id="@+id/textName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/textEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="14sp" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Step 4: Create Data Model

```kotlin
// Simple data class in Kotlin
data class User(
    val id: String = UUID.randomUUID().toString(),
    val name: String,
    val email: String
)
```

Step 5: Create ViewHolder

```
class UserViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
    private val textName: TextView = itemView.findViewById(R.id.textName)
    private val textEmail: TextView = itemView.findViewById(R.id.textEmail)

    fun bind(user: User) {
        textName.text = user.name
        textEmail.text = user.email
    }
}
```

Step 6: Create Adapter

```
class UserAdapter(
    private val userList: List<User>,
    private val onItemClick: (User) -> Unit = {}
) : RecyclerView.Adapter<UserViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Use
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.item_user, parent, false)
        return UserViewHolder(view)
    }

    override fun onBindViewHolder(holder: UserViewHolder, position: Int) {
        val user = userList[position]
        holder.bind(user)
        holder.itemView.setOnClickListener { onItemClick(user) }
    }

    override fun getItemCount() = userList.size
}
```

Thank you