

# CMPS 312

Read Chapter 1 & 6



## RecyclerView

Dr. Abdelkarim Erradi  
CSE@QU

# Outline

1. RecyclerView Components
2. Implementing a RecyclerView

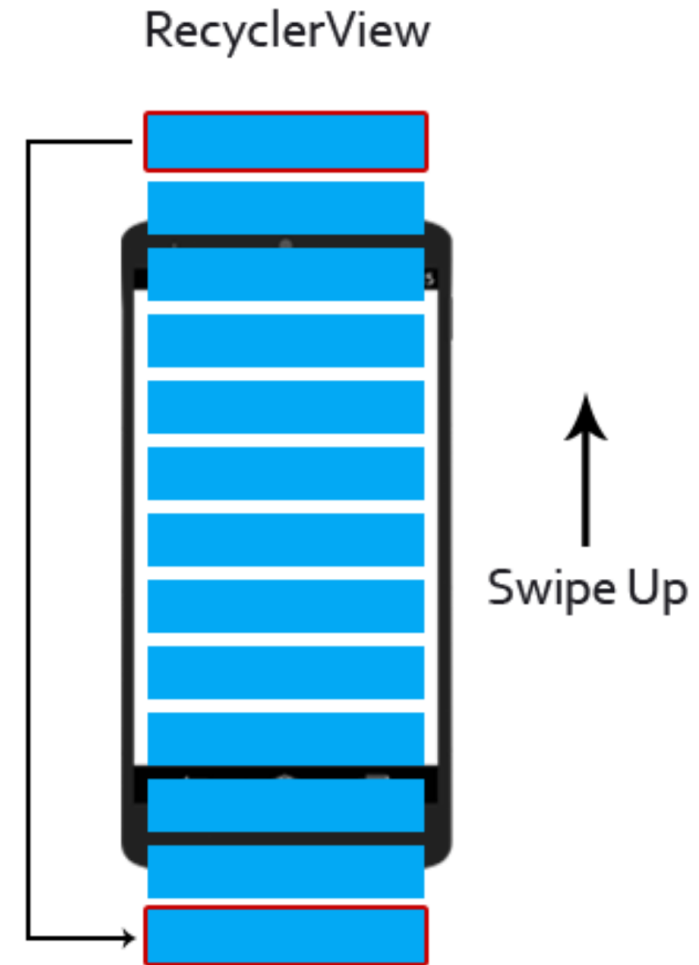
# RecyclerView

# Horizontal & Vertical Constraints

- To define a view's position you must add at least one horizontal and one vertical constraint for the view.
- Each constraint represents a connection or alignment to another view, the parent layout, or an invisible guideline.
- Each constraint defines the view's position along either the vertical or horizontal axis; so each view must have a minimum of one constraint for each axis, but often more are necessary

# What is a RecyclerView?

- RecyclerView is scrollable container for large data sets
- Efficient
  - Uses and reuses limited number of View elements
  - Updates changing data fast

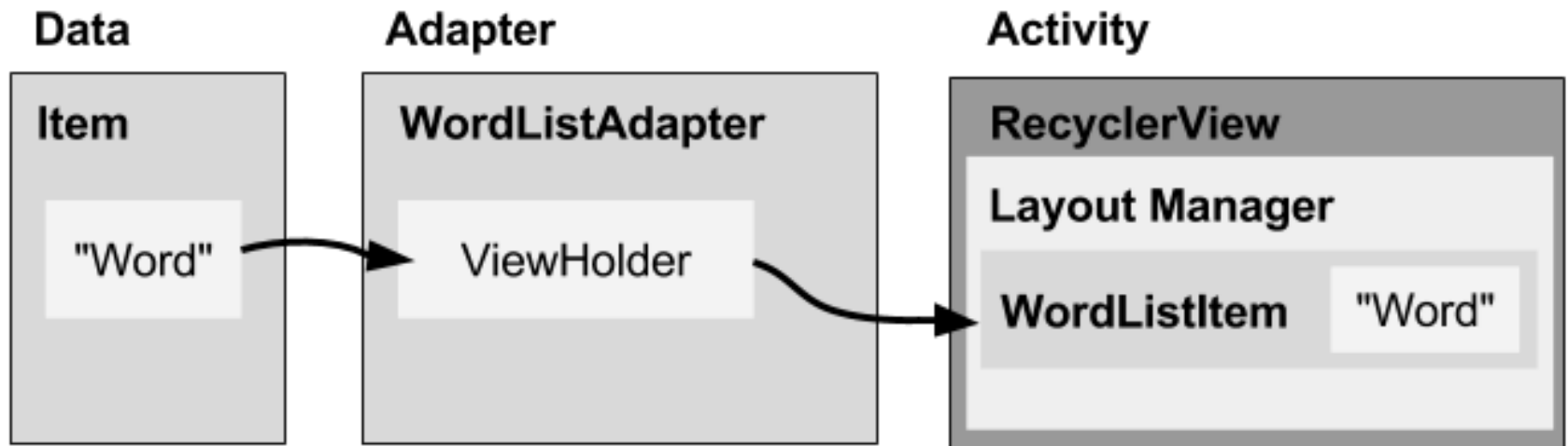


# RecyclerView Components

# Components Overview

- **List of data to display**
- **RecyclerView** scrolling list items
- **Layout** for one item of data - XML file
- **Layout manager** handles the organization of UI components in a View (Linear Layout or Grid Layout)
- **Adapter** connects data to the RecyclerView - extends [RecyclerView.Adapter](#)
- **ViewHolder** has view information for displaying one item - extends [RecyclerView.ViewHolder](#)

# How components fit together



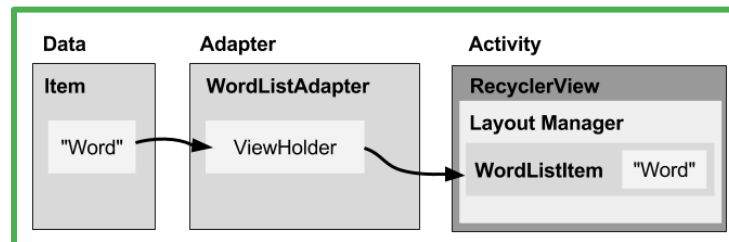


# What is a layout manager?

- Use to position View items inside a [RecyclerView](#)
- Reuses View items that are no longer visible to the user
- Built-in layout managers
  - [LinearLayoutManager](#)
  - [GridLayoutManager](#)
  - [StaggeredGridLayoutManager](#)
- Extend [RecyclerView.LayoutManager](#)

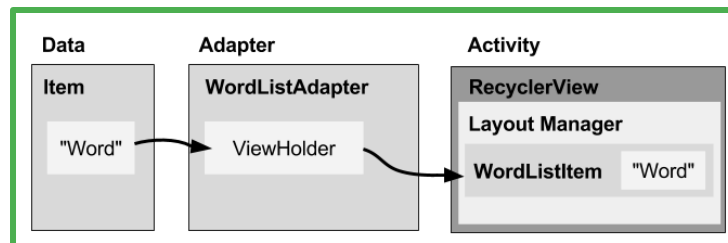
# What is an adapter?

- Helps incompatible interfaces work together
- Intermediary between data and View
- Manages creating, updating, adding, deleting View items as underlying data changes
- Extends [RecyclerView.Adapter](#)



# What is a ViewHolder?

- Used by the adapter to prepare one View with data for one list item
- Layout specified in an XML resource file
- Can have clickable elements
- Is placed by the layout manager
- Extends [RecyclerView.ViewHolder](#)

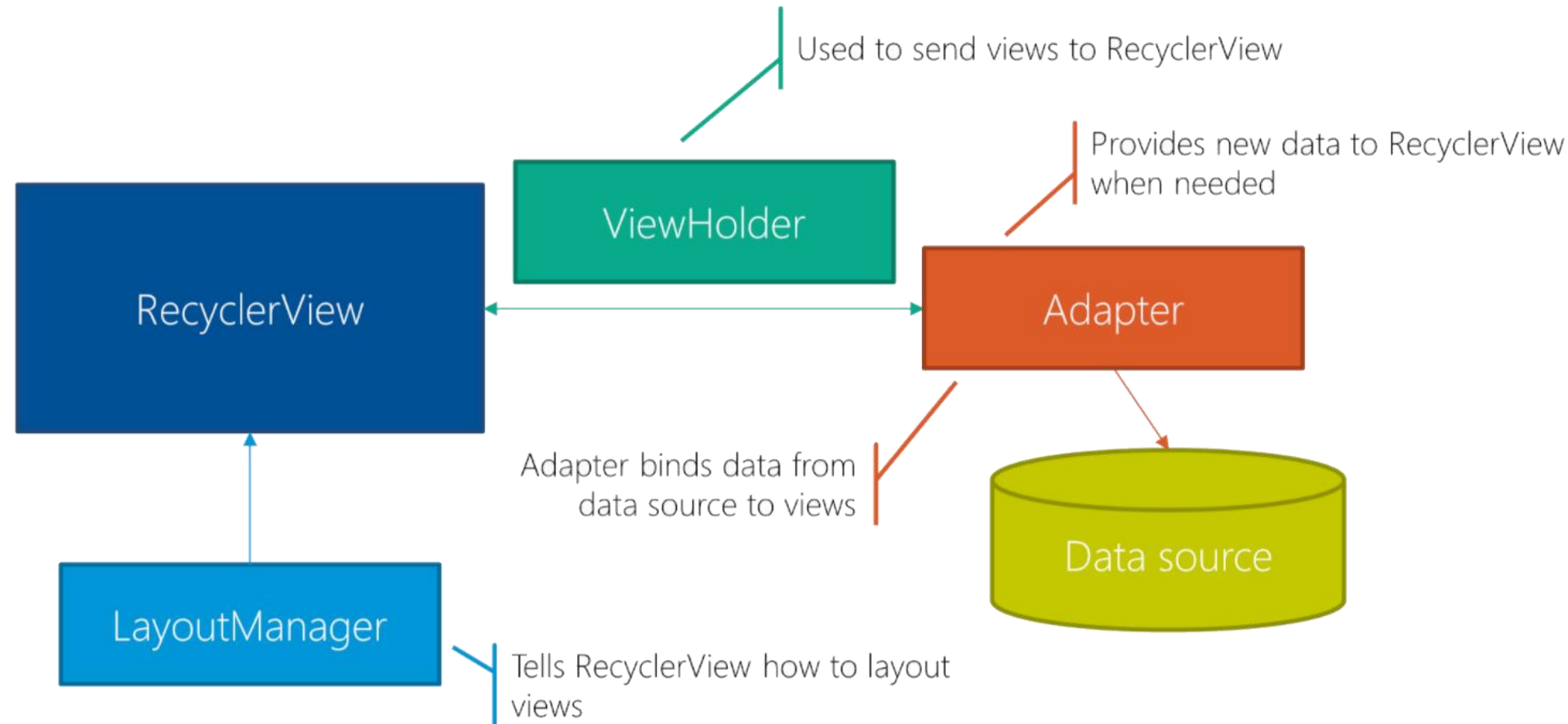


# Implementing RecyclerView

# Implementation Steps Summary

- Add `RecyclerView` to layout
- Create XML layout for 1 list item
- Extend `RecyclerView.Adapter`
- Extend `RecyclerView.ViewHolder`
- In `Activity onCreate()`, create `RecyclerView` with adapter and layout manager

# RecyclerView Flow



# Add RecyclerView to XML Layout

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/countriesRecyclerView"  
>
```

## Create layout for 1 list item

```
<LinearLayout ...>  
    <TextView  
        android:id="@+id/word"  
        style="@style/word_title" />  
</LinearLayout>
```

# Adapter has 3 required methods

- `onCreateViewHolder()`
- `inBindViewHolder()`
- `getItemCount()`

Let's take a look!



# Country Explorer

