Android Developer Fundamentals V2

User Interaction

Lesson 4

4.4 RecyclerView

Contents

- RecyclerView Components

- Implementing a RecyclerView

What is a RecyclerView?

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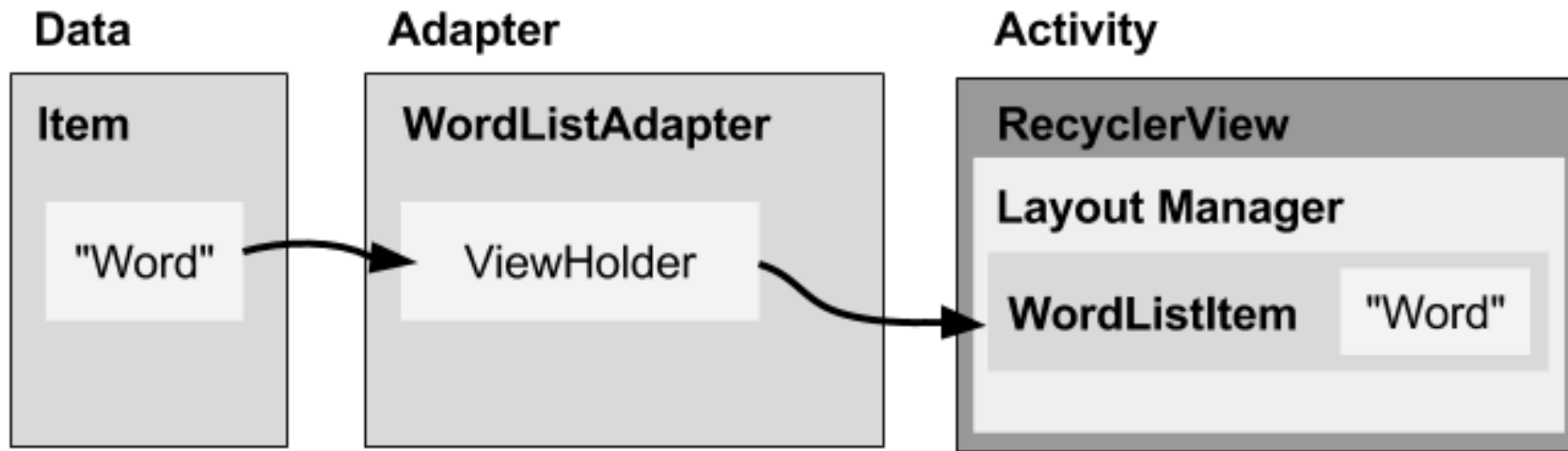 - Uses and reuses limited number of `View` elements
  - Updates changing data fast

4

RecyclerView Components

Overview

- **Data**

- **RecyclerView** scrolling list for list items—<u>RecyclerView</u>

- **Layout** for one item of data—XML file

- **Layout manage**r handles the organization of UI components in a View— <u>Recyclerview.LayoutManager</u>

- **Adapter** connects data to the RecyclerView—<u>RecyclerView.Adapter</u>

- **ViewHolder** has view information for displaying one item— <u>RecyclerView.ViewHolder</u>

How components fit together overview
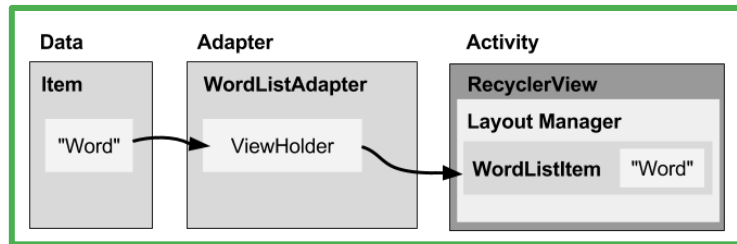
What is a layout manager?

- Each `ViewGroup` has 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`
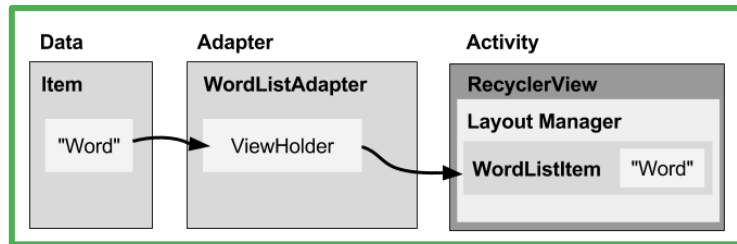
- Extend `RecyclerView.LayoutManager`

What is an adapter?

- Helps incompatible interfaces work together
  - Example: Takes data from database `Cursor` and prepares strings to put into a `View`

- Intermediary between data and `View`

- Manages creating, updating, adding, deleting `View` items as underlying data changes

- `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

- `RecyclerView.ViewHolder`

Implementing RecyclerView

Steps Summary

1. Add `RecyclerView` dependency to build.gradle if needed

2. Add `RecyclerView` to layout

3. Create XML layout for item

4. Extend `RecyclerView.Adapter`

5. Extend `RecyclerView.ViewHolder`

6. In `Activity onCreate()`, create `RecyclerView` with adapter and layout manager

Add dependency to app/build.gradle

Add `RecyclerView` dependency to build.gradle if needed:

```
dependencies {
  ...
  compile 'com.android.support:recyclerview-v7:26.1.0'
  ...
  }
```

13

Add RecyclerView to XML Layout

```xml
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</android.support.v7.widget.RecyclerView>
```

Create layout for 1 list item

```
<LinearLayout …>

  <TextView

      android:id="@+id/word"

      style="@style/word_title" />

</LinearLayout>
```

15

Implement the adapter

```java
public class WordListAdapter

    extends RecyclerView.Adapter<WordListAdapter.WordViewHolder> {


    public WordListAdapter(Context context,

                           LinkedList<String> wordList) {
        mInflater = LayoutInflater.from(context);
        this.mWordList = wordList;
    }
}
```

Adapter has 3 required methods

- `onCreateViewHolder()`

- `inBindViewHolder()`

- `getItemCount()`

Let's take a look!

onCreateViewHolder()

```
@Override
public WordViewHolder onCreateViewHolder(
        ViewGroup parent, int viewType) {
        // Create view from layout
        View mItemView = mInflater.inflate(
                R.layout.wordlist_item, parent, false);
        return new WordViewHolder(mItemView, this);
}
```

onBindViewHolder()

```java
@Override
public void onBindViewHolder(
        WordViewHolder holder, int position) {
        // Retrieve the data for that position
        String mCurrent = mWordList.get(position);
        // Add the data to the view
        holder.wordItemView.setText(mCurrent);
}
```

getItemCount()

```java
@Override
public int getItemCount() {
    // Return the number of data items to display
    return mWordList.size();
}
```

Create the view holder in adapter class

```
class WordViewHolder extends RecyclerView.ViewHolder { //.. }
```

If you want to handle mouse clicks:

```
class WordViewHolder extends RecyclerView.ViewHolder
                     implements View.OnClickListener { //.. }
```

View holder constructor

```java
public WordViewHolder(View itemView, WordListAdapter adapter) {
    super(itemView);
    // Get the layout
    wordItemView = itemView.findViewById(R.id.word);
    // Associate with this adapter
    this.mAdapter = adapter;
    // Add click listener, if desired
    itemView.setOnClickListener(this);
}
// Implement onClick() if desired
```

Create the RecyclerView in Activity onCreate()

```
mRecyclerView = findViewById(R.id.recyclerview);

mAdapter = new WordListAdapter(this, mWordList);

mRecyclerView.setAdapter(mAdapter);

mRecyclerView.setLayoutManager(new

                    LinearLayoutManager(this));
```

Practical: RecyclerView

● This is rather complex with many separate pieces. So, there is a whole practical where you implement a RecyclerView that displays a list of clickable words.

● Shows all the steps, one by one with a complete app

Learn more

- RecyclerView

- RecyclerView class

- RecyclerView.Adapter class

- RecyclerView.ViewHolder class

- RecyclerView.LayoutManager class

What's Next?

- Concept Chapter: 4.5 RecyclerView

- Practical: 4.5 RecyclerView

END