

# CSc 317

# Adapters and ListViews

Benjamin Dicken

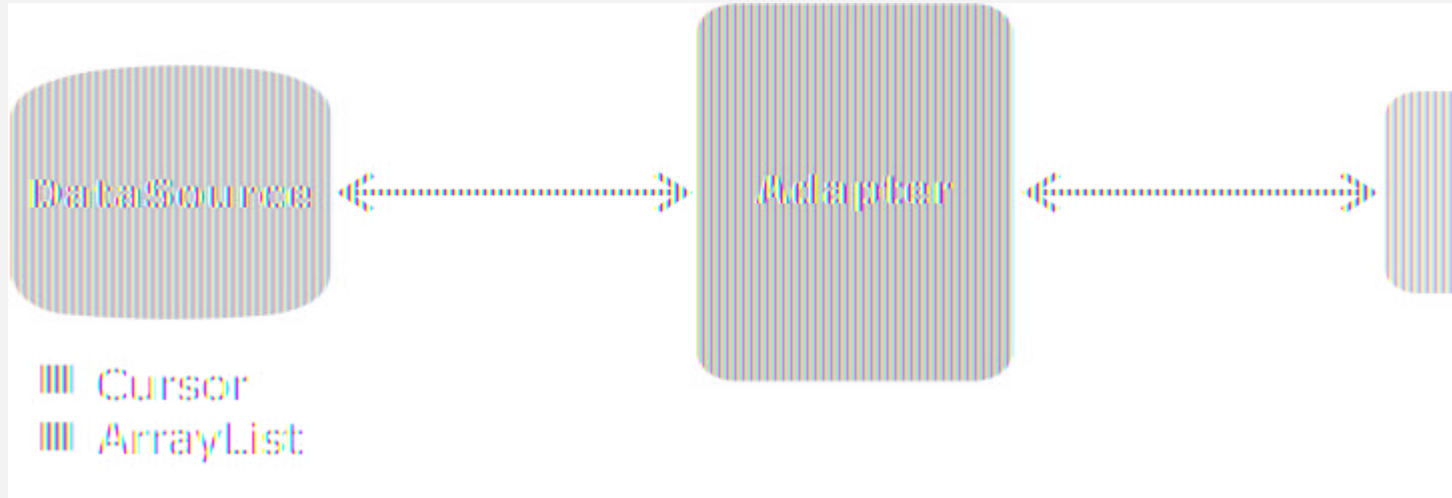


# List View

- A **List View** is a type of view, designed to display lists!
  - Does not know details of what it is displaying
  - Generates rows on-demand
- <https://developer.android.com/reference/android/widget/ListView>

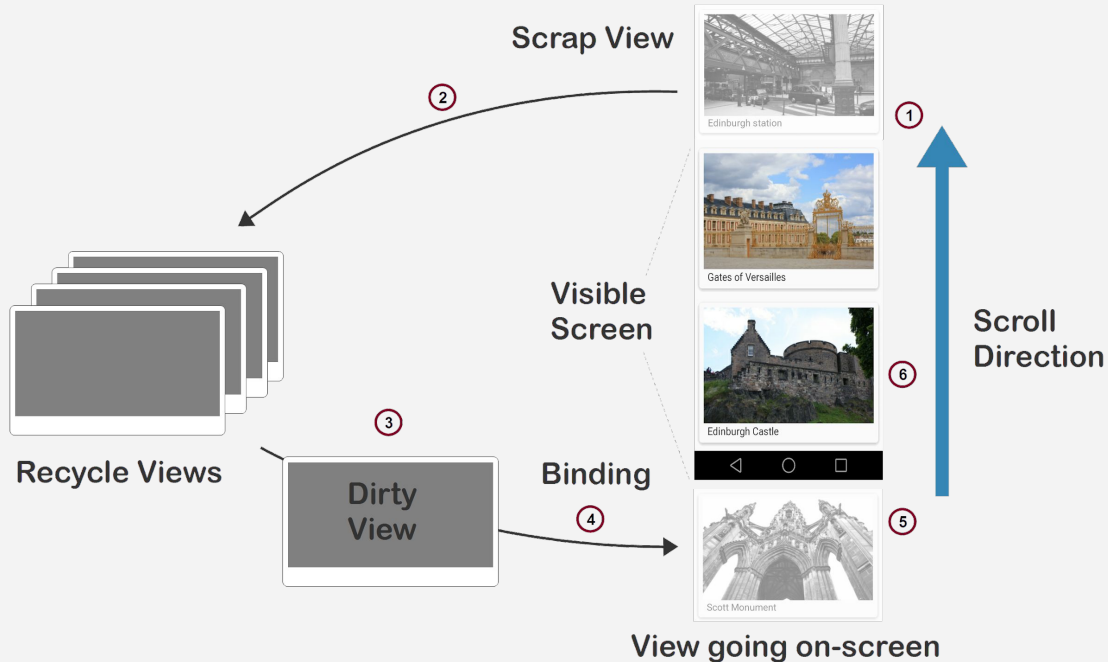
# Data, Adapters, and ListViews

<https://guides.codepath.com/android/Using-an-ArrayAdapter-with-ListView>



# Data, Adapters, and ListViews

<https://guides.codepath.com/android/Using-an-ArrayAdapter-with-ListView>

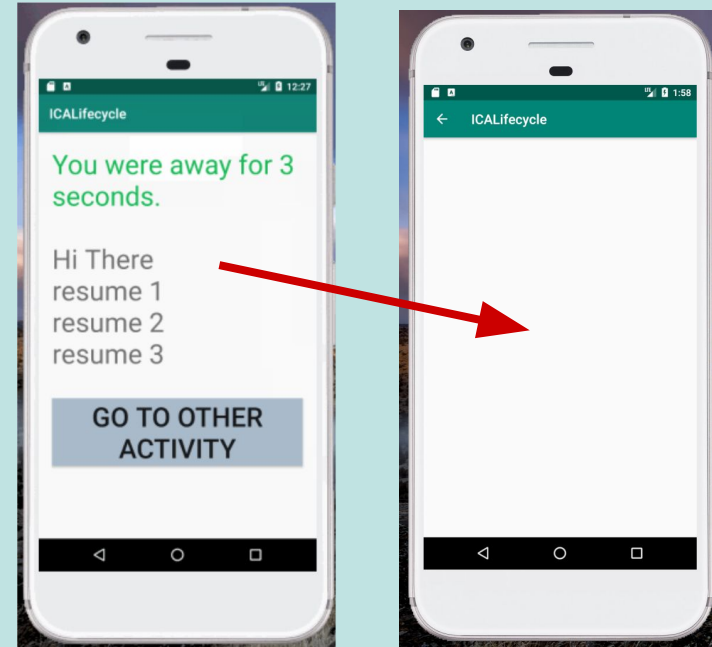


# ArrayAdapter

- Adapter that maps simple data in an array to being displayed via a view
- What if there are multiple elements to be displayed per list item?

# Open up the app

- Open up the ICALifecycle application
- Open up the xml file for the **OtherView**



# Add a LinearLayout

- Ensure that you are using a LinearLayout
- Add a Listview, as shown
- Run it - What does it look like?

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    ...
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".OtherActivity">

    <ListView
        android:id="@+id/words_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />

</LinearLayout>
```

# Create a new View for Row

- Right click on the **layouts** directory, then choose **New** -> **XML** -> **XML Layout File**
- Name it **shopping\_list\_row**
- Use a linearLayout with a textview

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    . . .
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/shopping_list_row_item"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        . . .
        android:textSize="55dp" />

</LinearLayout>
```



# Editing the Code

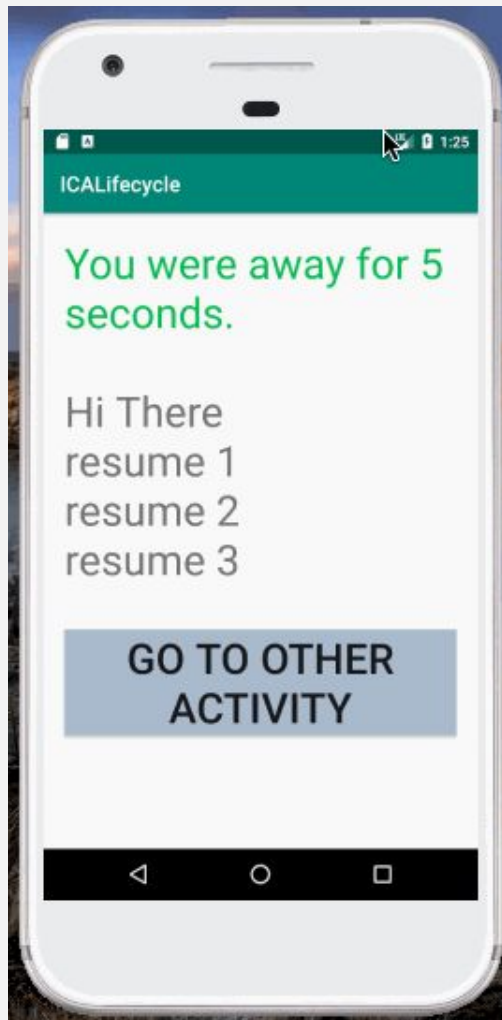
- Go to **OtherView.java**
- Add a ListView member variable
- Create an array with 10+ strings (shopping list)

# Create Adapter

```
// Get a reference to the ListView
shoppingListView =
    (ListView)findViewById(R.id.shopping_list_view);

// Create a new Array Adapter
// Specify which layout and view to use for a row
// and the data (array) to use
ArrayAdapter<String> arrayAdapter = new
    ArrayAdapter<String>(this, R.layout.shopping_list_row,
        R.id.shopping_list_row, shoppingList);

// Link the ListView and the Adapter
shoppingListView.setAdapter(arrayAdapter);
```



ICALifecycle

You were away for 5  
seconds.

Hi There  
resume 1  
resume 2  
resume 3

**GO TO OTHER  
ACTIVITY**

# SimpleAdapter

- A **SimpleAdapter** can be used to associate multiple values with each row of a listview
- The steps are slightly more complex than for an **ArrayAdapter**

# Update Row View

- Update **shopping\_list\_row.xml**
- Should be an **ImageView** and **TextView** for each row

```
<ImageView  
  
    android:id="@+id/shopping_list_row_image"  
    android:layout_width="150dp"  
    android:layout_height="150dp"  
    android:layout_gravity="center" />  
  
<TextView  
  
    android:id="@+id/shopping_list_row_text"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_gravity="center"  
    android:text=""  
    android:textColor="#2196F3"  
    android:textSize="55dp" />
```

# Editing the Code

- Change the shopping list to only have 4 items
- Search for (low-res) images of those 4 items
- Add those as image resources
- Create another list of the IDs of those images

# Create Lists

```
String[] shoppingList = new String[]{  
    "Steak", "Chicken", "Lettuce", "Arugula"  
};
```

```
int[] shoppingListImages = new int[]{  
    R.drawable.steak, R.drawable.chicken,  
    R.drawable.coffee, R.drawable.arugula  
};
```

# Map View IDs to elements

```
List<HashMap<String, String>> aList =  
    new ArrayList<HashMap<String, String>>();  
  
for (int i = 0; i < shoppingListImages.length; i++) {  
    HashMap<String, String> hm = new HashMap<String, String>();  
    hm.put("shopping_list_row_image", Integer.toString(shoppingListImages[i]));  
    hm.put("shopping_list_row_text", shoppingList[i]);  
    aList.add(hm);  
}
```

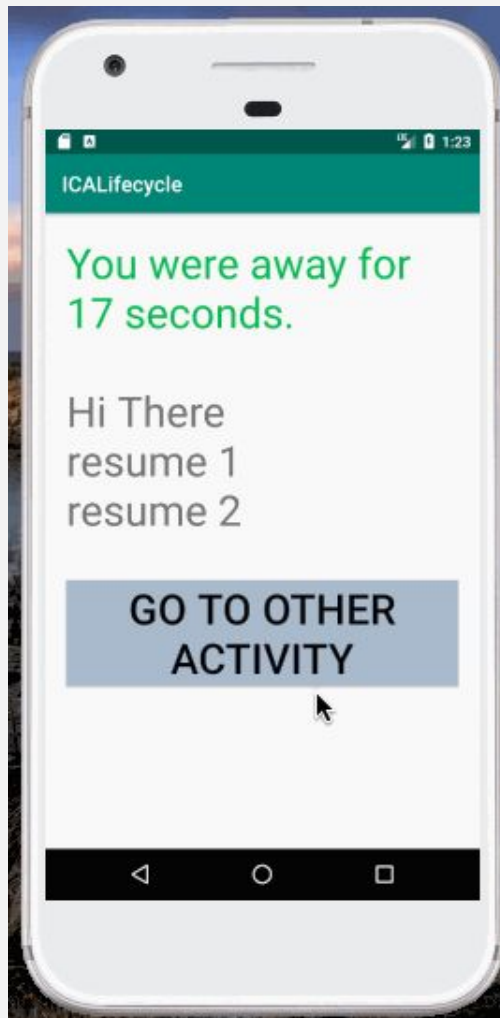


# Configure Simple Adapter

```
String[] from = {"shopping_list_row_image", "shopping_list_row_text"};
int[] to = {R.id.shopping_list_row_image, R.id.shopping_list_row_text};

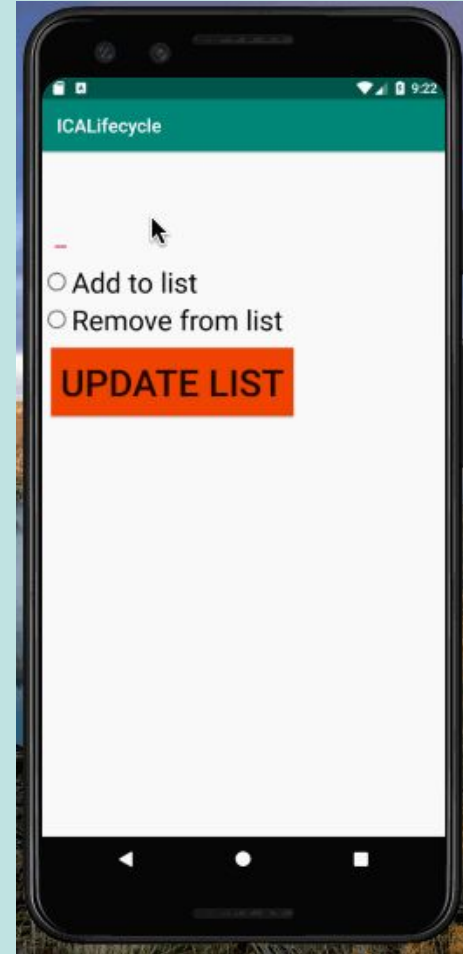
SimpleAdapter simpleAdapter =
    new SimpleAdapter(getBaseContext(), aList,
        R.layout.shopping_list_row, from, to);

shoppingListView = (ListView) findViewById(R.id.shopping_list_view);
shoppingListView.setAdapter(simpleAdapter);
```



# Return to shopping list

- Replace the other text on the MainActivity with:
  - An EnterText
  - A RadioButton
  - An “Update List” button
- As shown in the screenshot
- User should be able to add or remove elements from the shopping list



# Radio Buttons

```
<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="100dp">
    <RadioButton
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="add to list"
        android:onClick="addOption"/>
    <RadioButton
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:text="remove from list"
        android:onClick="removeOption"/>
</RadioGroup>
```

# Code

```
private static String addOrRemove = "";
```

```
public void goToOtherActivity(View view) {  
    Intent intent = new Intent(this, OtherActivity.class);  
    EditText editText = (EditText) findViewById(R.id.editText);  
    String message = editText.getText().toString();  
    intent.putExtra("DATA", message);  
    intent.putExtra("ACTION", addOrRemove);  
    startActivity(intent);  
}
```

```
public void addOption(View view) {  
    addOrRemove = "ADD";  
}
```

```
public void removeOption(View view) {  
    addOrRemove = "REMOVE";  
}
```

# If time permits

Show off a custom subclass of an **ArrayAdapter** (in app  
VideoGameCharacters)