



The X Course: Android

Session 6

Recap Time

- First Version of GeoQuiz.
- Cheating Version of GeoQuiz.
- Cheat-Aware Version of GeoQuiz.
- The MVC GeoQuiz.



Agenda

- Building Lists in Android
 - a. ListView
 - b. RecyclerView
- Adding the List Activity to the GeoQuiz !



Building Lists: ListView

- Layout UI:

```
<ListView
```

```
    android:id="@+id/list_view"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent" />
```



Building Lists: ListView

- ListView is an `adapter view` that does not know the details, such as type and contents, of the views it contains.
- ListView requests views on demand from a `ListAdapter` as needed, such as to display new views as the user scrolls up or down.
- In order to display items in the list, we associate an adapter with list.

`setAdapter(android.widget.ListAdapter)`



Building Lists: Populating ListView

- We need to bind the `AdapterView` instance to an `Adapter`.
- `Adapter` retrieves data from an external source and creates a `View` that represents each data entry.
- Android provides several subclasses of `Adapter` that are useful for retrieving different kinds of data
- We will see the `ArrayAdapter` class.



Building Lists: ArrayAdapter & ListView

- `ArrayAdapter` creates a view for each array item.
- Placing the contents of each item in a `TextView`.

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1, myStringArray);
```

- Then,

```
ListView listView = (ListView) findViewById(R.id.listview);  
listView.setAdapter(adapter);
```



Building Lists: Customizing ArrayAdapter

- **Customize Objects:** You can override the `toString()` method for the objects in your array.
- **Customize Layouts:** Or, to create a view for each item that's something other than a `TextView` (for example, if you want an `ImageView` for each array item)
- **More Customization:** Extend `BaseAdapter` and create and configure the view for each data item in `getView(...)`

```
private class MyAdapter extends BaseAdapter {
```

```
    // override other abstract methods here
```

```
        @Override
        public View getView(int position, View convertView, ViewGroup
container) {
            if (convertView == null) {
                convertView =
getLayoutInflater().inflate(R.layout.list_item, container, false);
            }
            ((TextView) convertView.findViewById(android.R.id.text1))
                .setText(getItem(position));
            return convertView;
        }
    }
}
```

Let's build the List View Activity in GeoQuiz !





Better and Modern Lists: RecyclerView

- The `RecyclerView` widget is a more advanced and flexible version of `ListView`. Several different components work together to display data.
- The `RecyclerView` fills itself with views provided by a *layout manager* that you provide.
- This `RecyclerView` model does a lot of optimization work so you don't have to.



Better and Modern Lists: RecyclerView

- Views in the list are represented by *view holder* objects. Each view holder is in charge of displaying a single item with a view.
- The `RecyclerView` creates only as many view holders as are needed to display the on-screen portion of the dynamic content, plus a few extra.
- As the user scrolls through the list, the `RecyclerView` takes the off-screen views and rebinds them to the data which is scrolling onto the screen.



Better and Modern Lists: RecyclerView

- The view holder objects are managed by an *adapter*.
 - creates view holders as needed.
 - binds the view holders to their data.
- We use one of the standard layout managers ([LinearLayoutManager](#) or [GridLayoutManager](#)), or implement your own.



Better and Modern Lists: RecyclerView

Adapter Methods:

- `onCreateViewHolder()` method. needs to construct a `ViewHolder` and set the view it uses to display its contents.
- `onBindViewHolder()` method, binds the view holder to its data, and passing the view holder's position in the `RecyclerView`.



Better and Modern Lists: RecyclerView

Steps:

- Add the support library in the gradle file.
- Add the RecyclerView in the layout XML file.
- Create ViewHolder Class to inflate the layout with the object.
- Create ViewAdapter Class to create and bind ViewHolder objects.

**Let's build the RecyclerView Activity in
GeoQuiz !**





Further Readings

- <https://developer.android.com/guide/topics/ui/layout/recyclerview>
- <https://developer.android.com/reference/android/widget/ListView>
- <https://developer.android.com/guide/topics/ui/declaring-layout.html#FillingTheLayout>
- <https://guides.codepath.com/android/using-the-recyclerview>