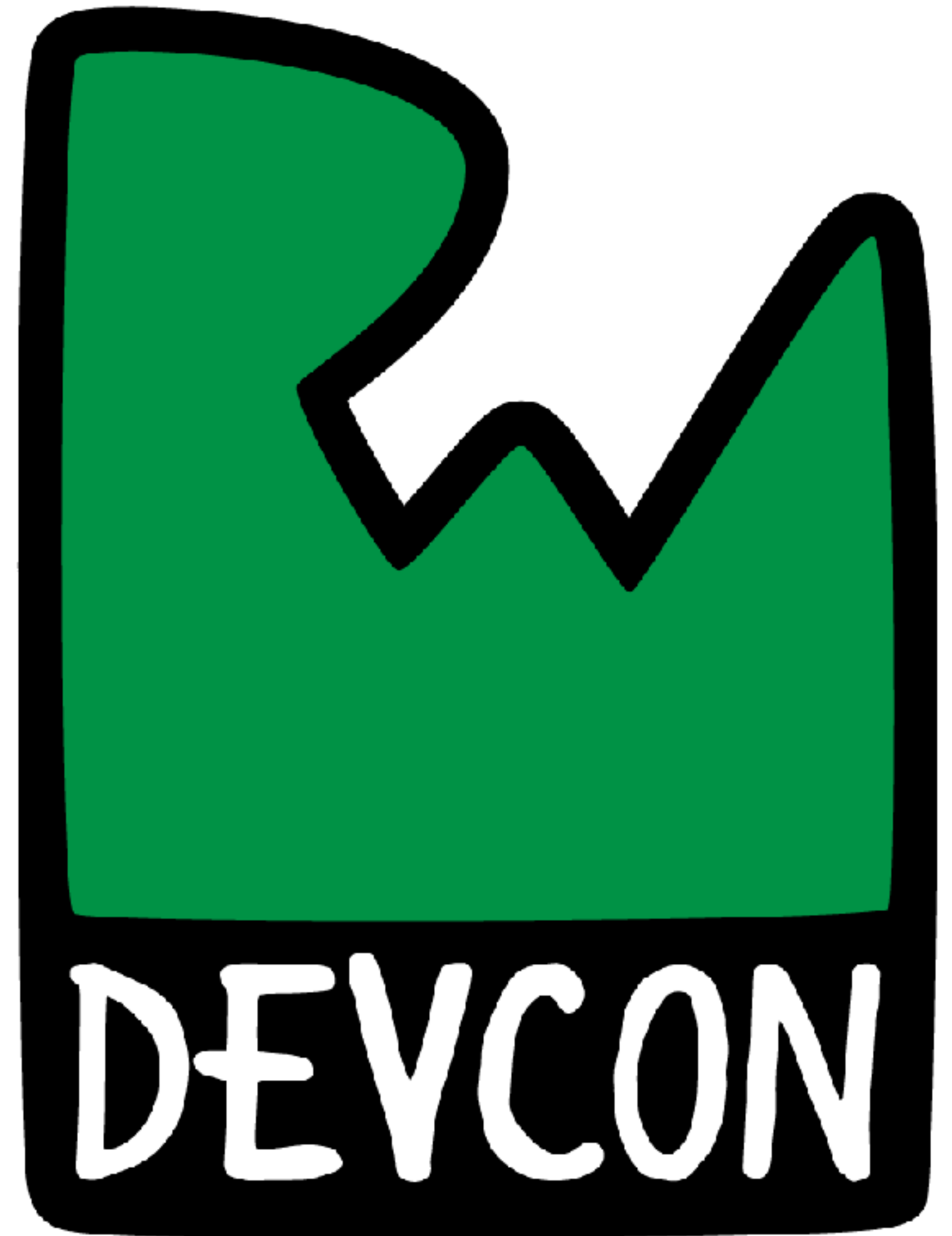


# Session 7: Android for iOS Developers



CREATING THE BASIC UI

# BEFORE YOU START

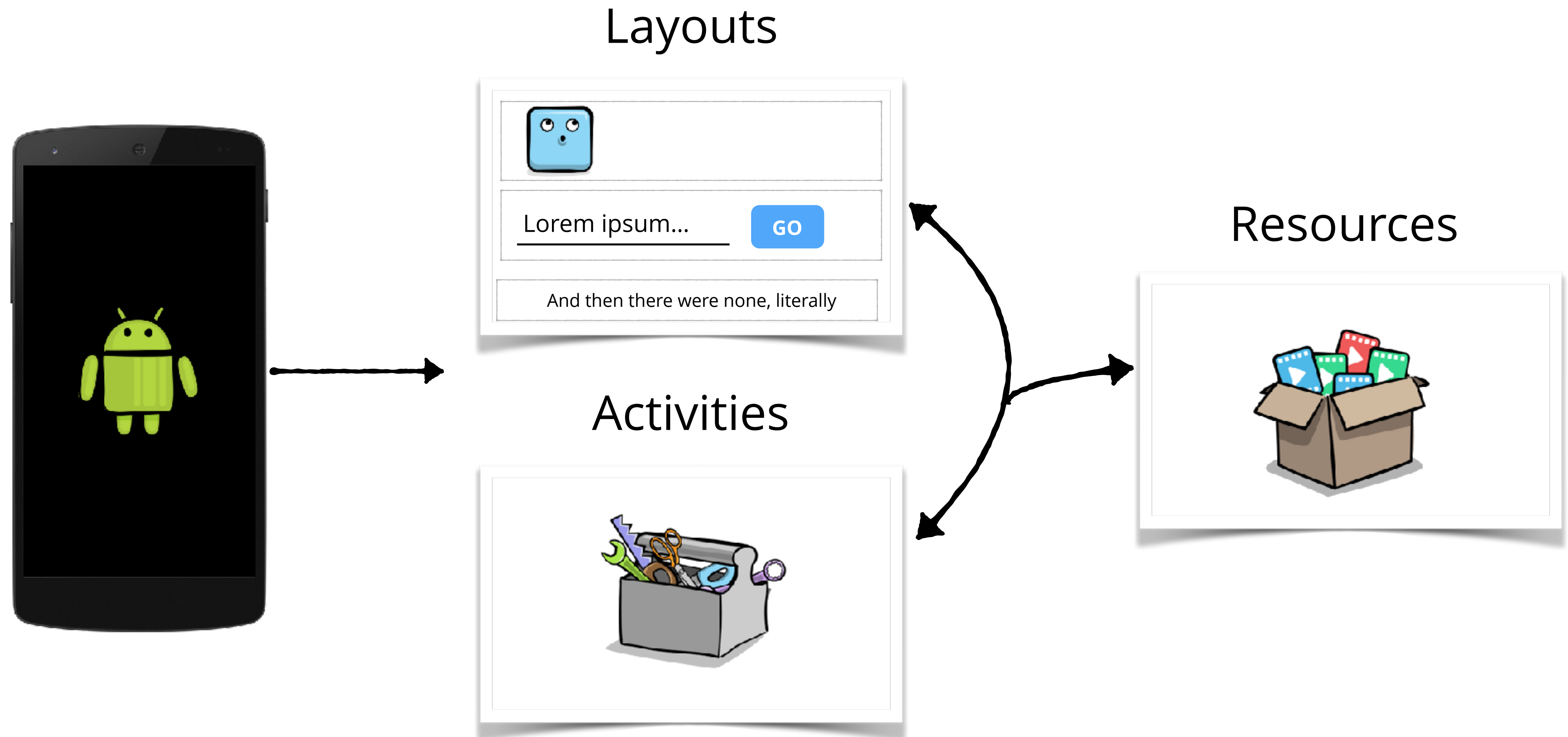
---

- ⚙ **Install:** Android Studio 3.0.1 & Android SDK
- ⚙ **Install:** Android Emulator
- ⚙ **Build & Run:** Demo1 Starter Project



# ANDROID APP

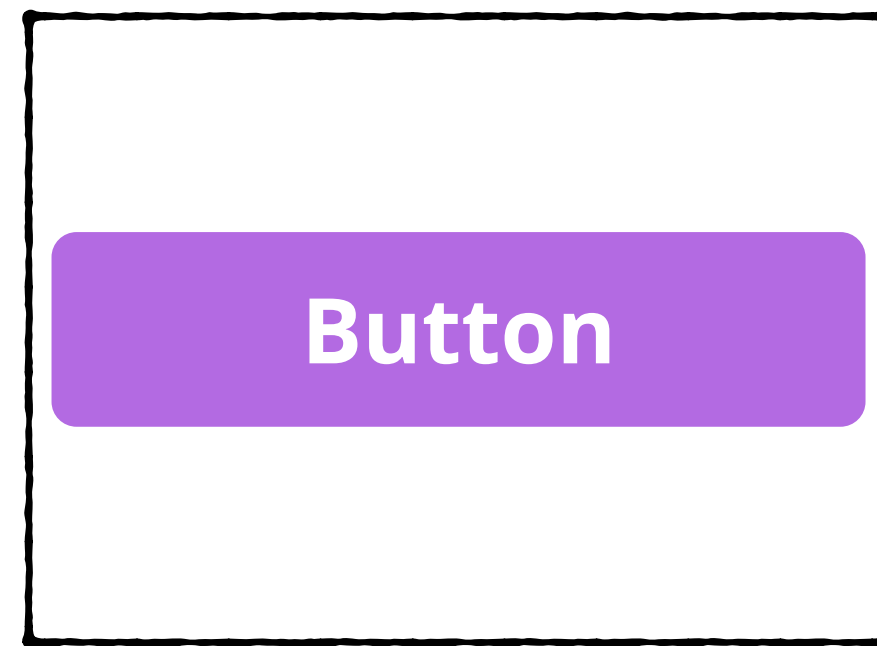
---



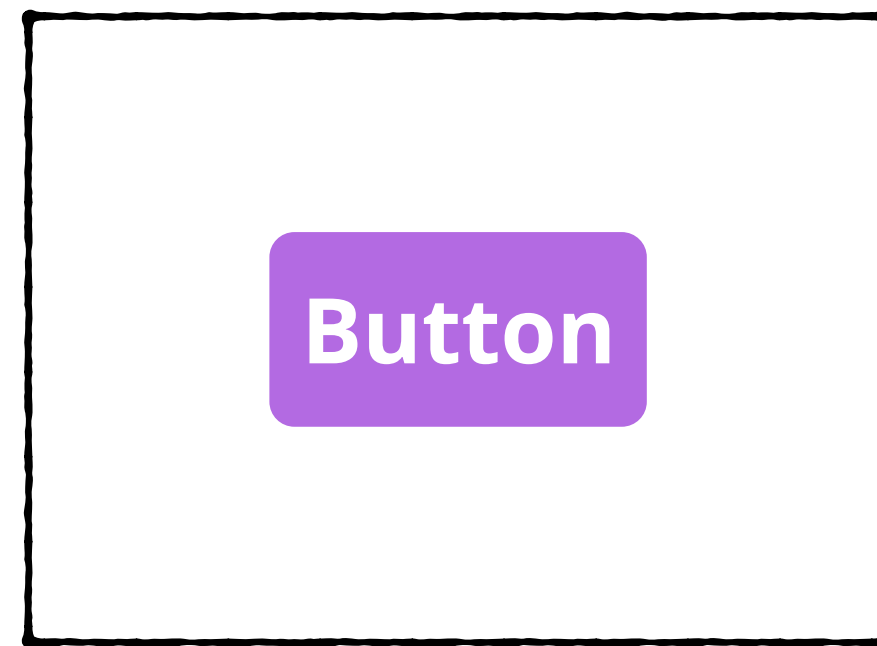
# LAYOUT

---

`android:layout_height="wrap_content"`  
`android:layout_width="??"`

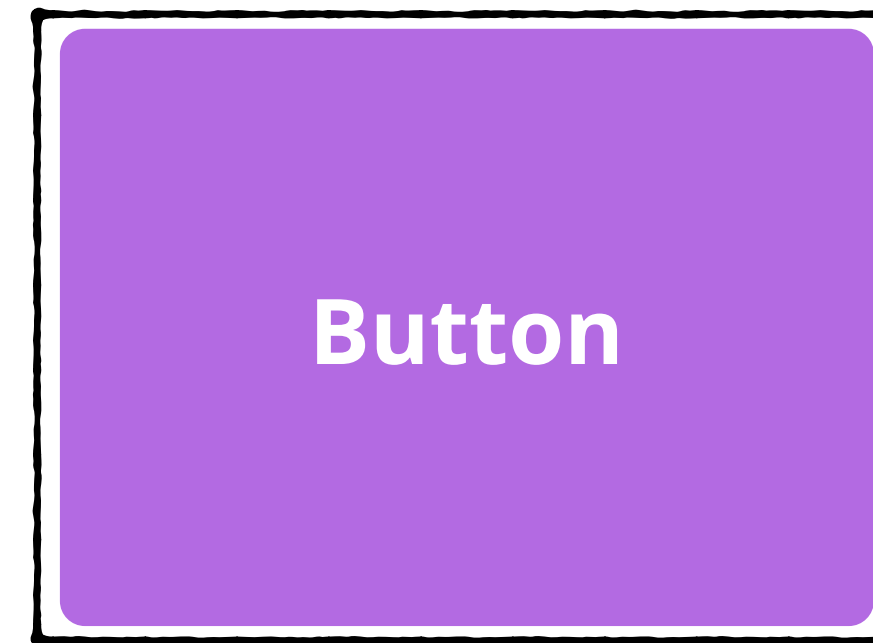


match\_parent

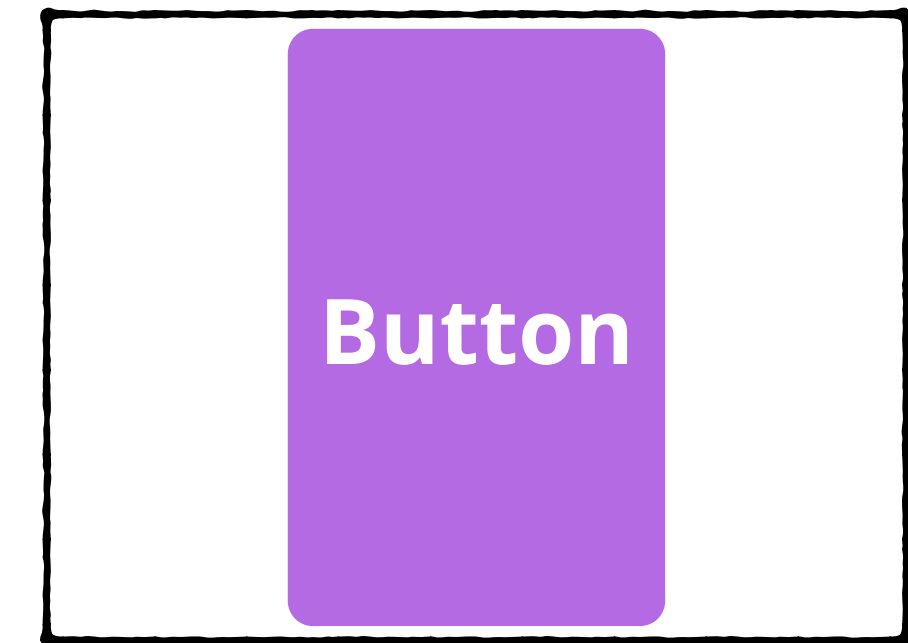


wrap\_content

`android:layout_height="match_parent"`  
`android:layout_width="??"`



match\_parent



wrap\_content



# LAYOUT

---

## Relative Positioning - ConstraintLayout

`app:layout_constraint<ThisElementEdge>_to<OtherElementEdge>Of="<other_element>"`

⚙ Start

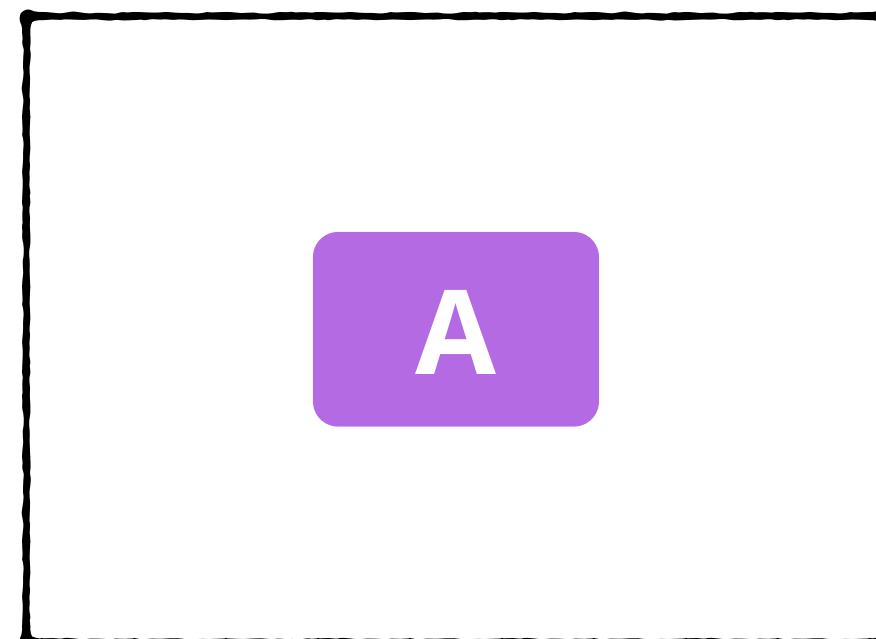
⚙ Left

⚙ Top

⚙ End

⚙ Right

⚙ Bottom



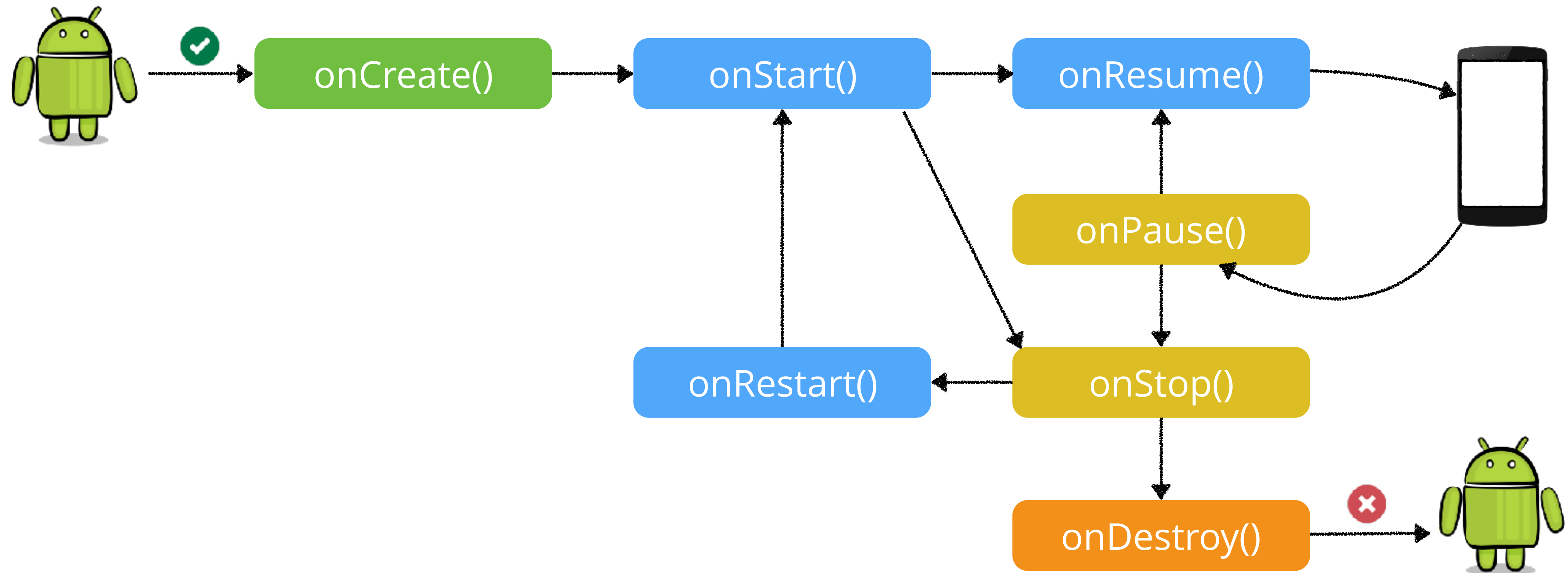
`app:layout_constraintStart_toStartOf="parent"`

`app:layout_constraintTop_toTopOf="parent"`

`app:layout_constraintEnd_toEndOf="parent"`

`app:layout_constraintBottom_toBottomOf="parent"`

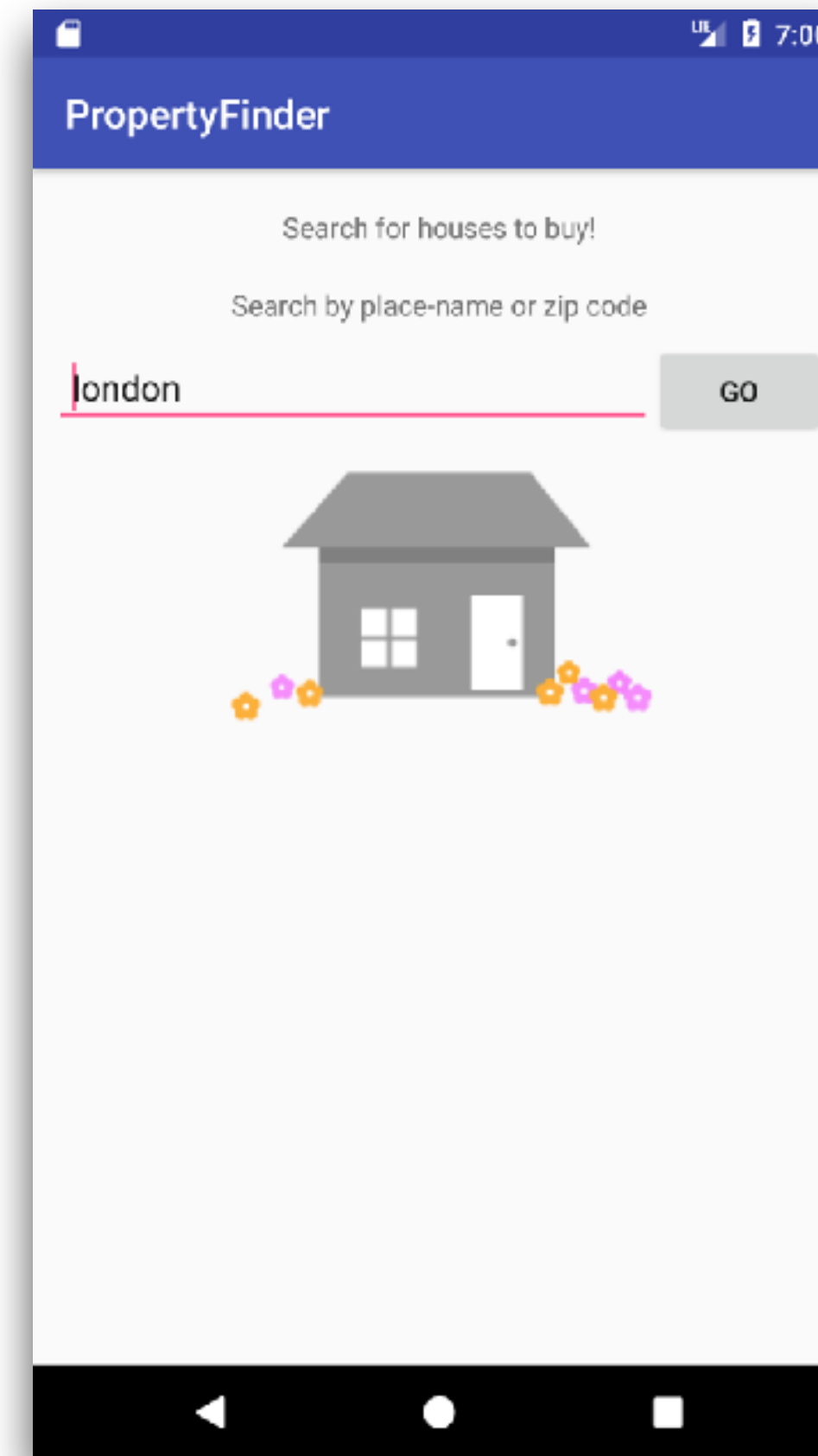
# ACTIVITY



# WHAT'S IN DEMO 1

---

- ▶ **Android Studio:** Tour
- ▶ **Layout:** Constraints, parameters, resources, editors
- ▶ **Activity:** Lifecycle, handling events

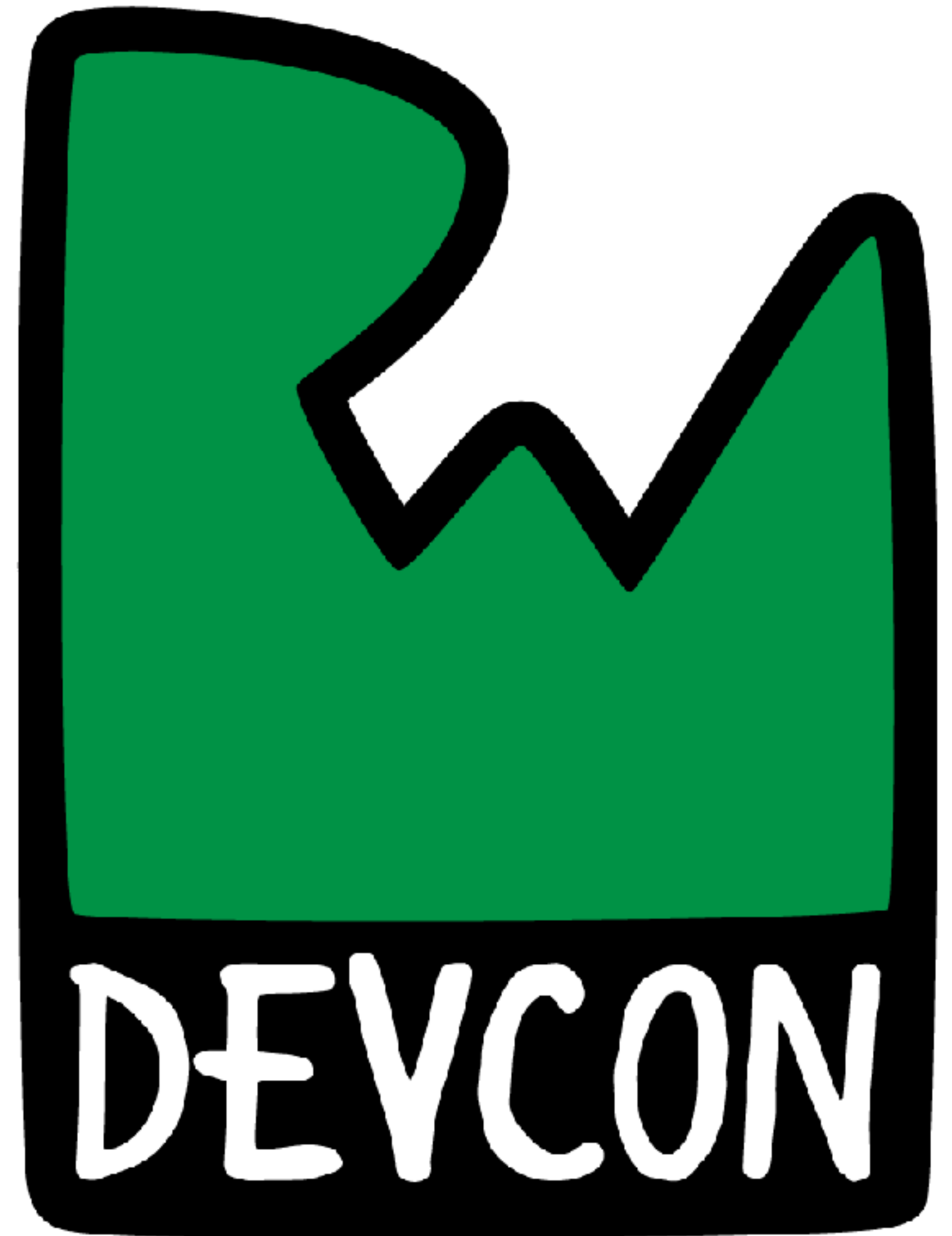


DEMO 1





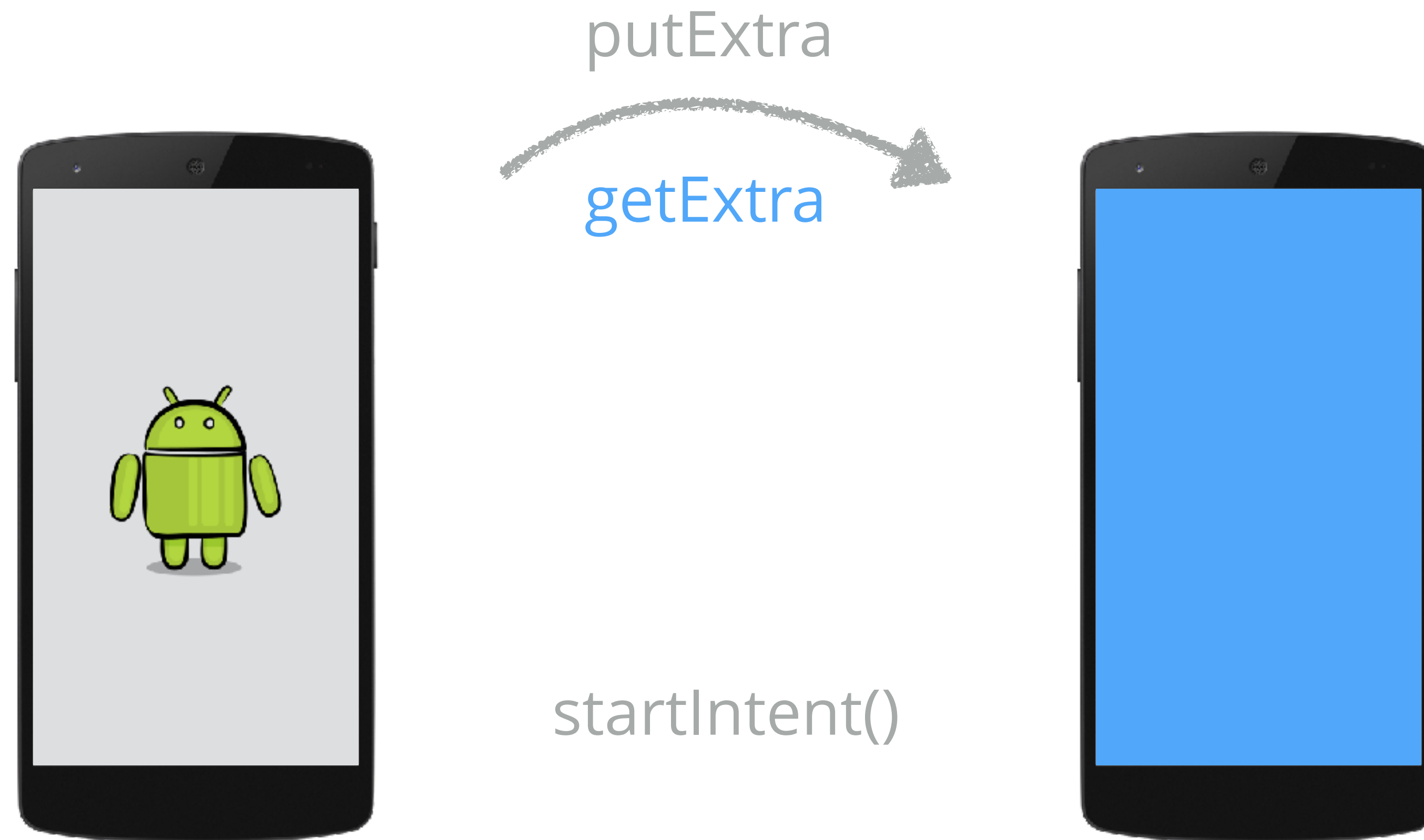
# Session 7: Android for iOS Developers



NAVIGATION, RECYCLER VIEW

# NAVIGATION: INTENT

---



# RECYCLER VIEW

---

## iOS

- ▶ UITableView
- ▶ UITableViewDataSource
  - ▶ tableView(\_:cellForRowAt:)
  - ▶ tableView(\_:numberOfRowsInSection:)
- ▶ UITableViewCell

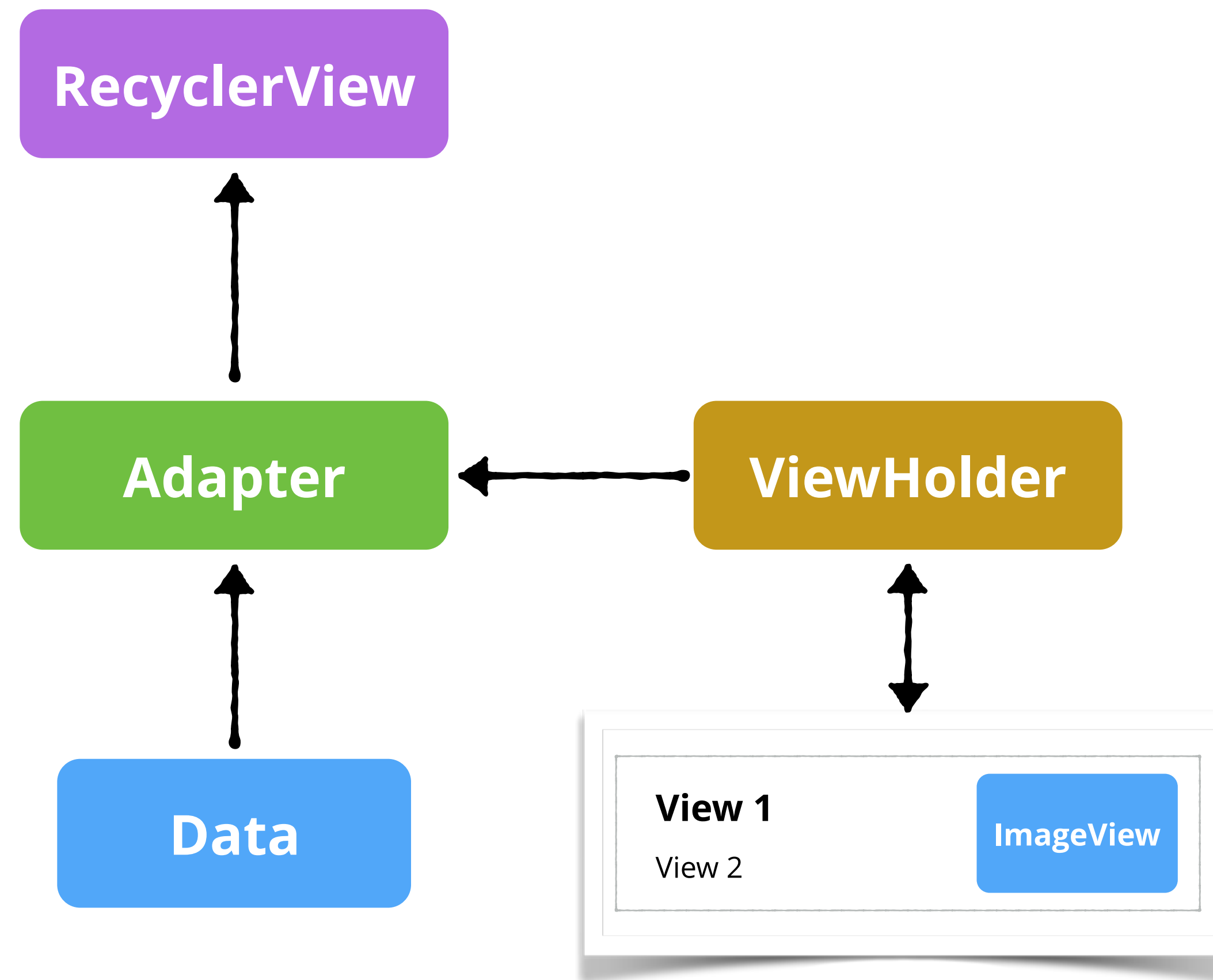
## Android

- ▶ RecyclerView
- ▶ Adapter & ViewHolder
  - ▶ onCreateViewHolder(), onBindViewHolder()
  - ▶ getItemCount()
- ▶ Layout (item row)



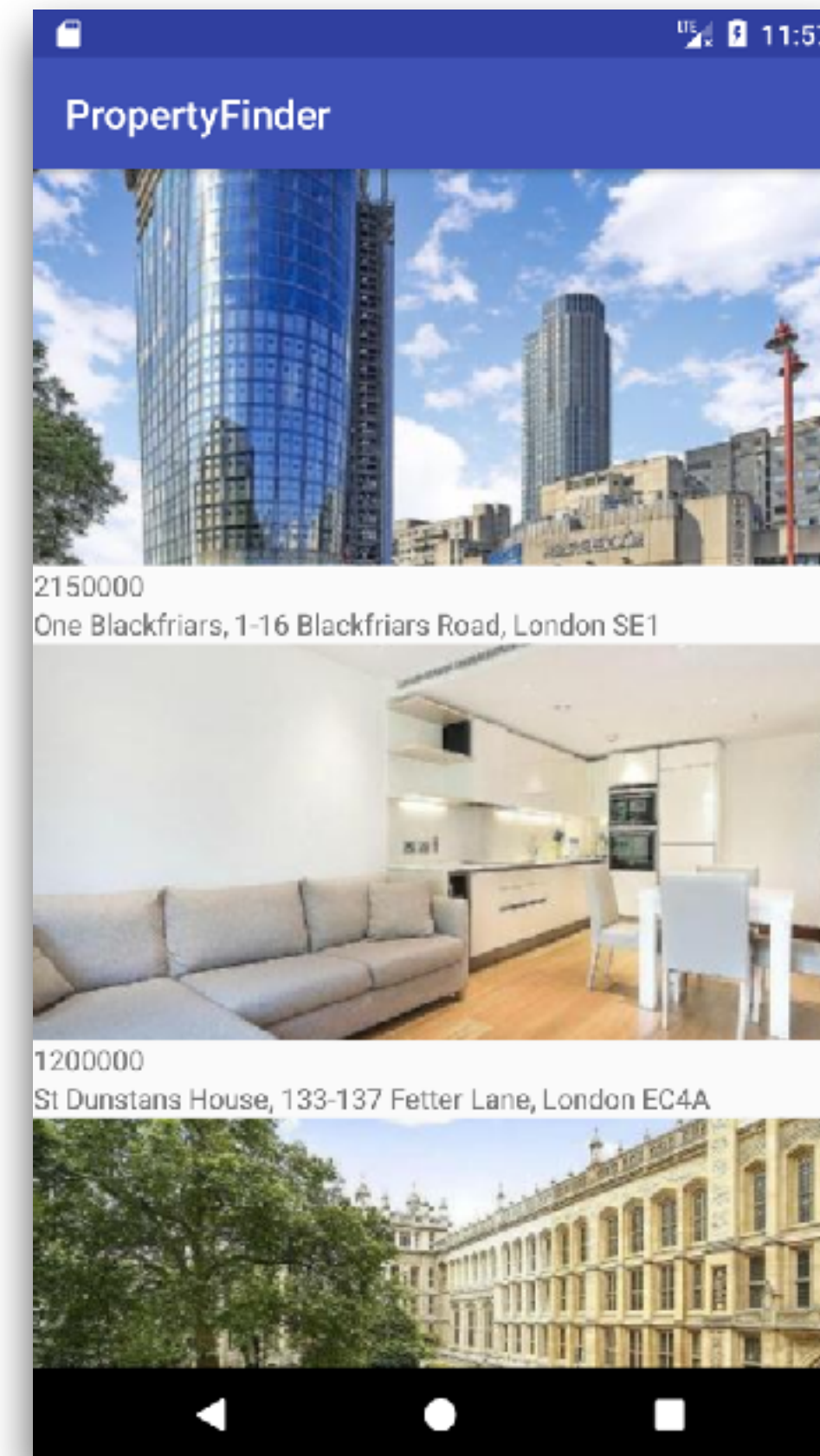
# RECYCLER VIEW

---



# WHAT'S IN DEMO 2

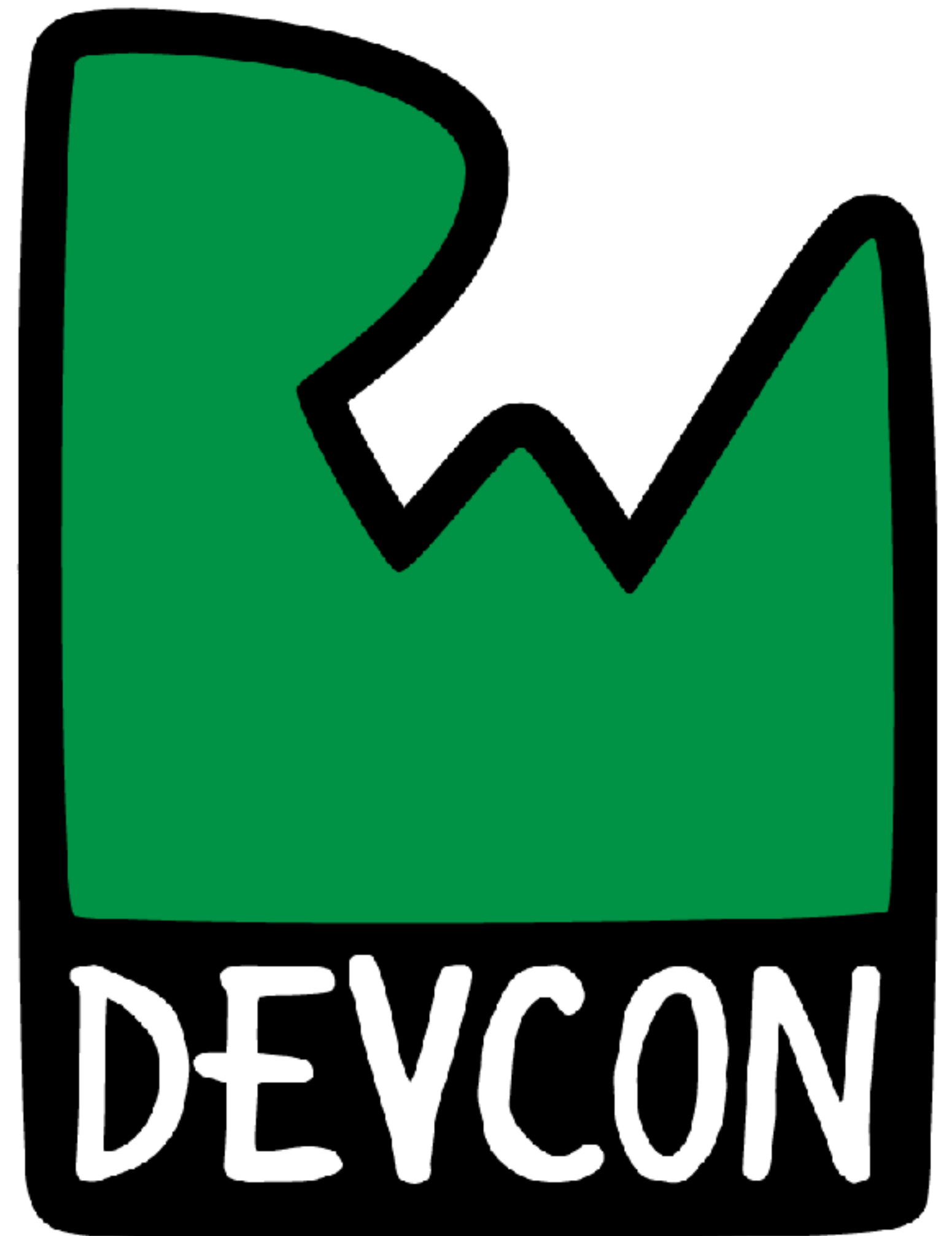
- ▶ **Intent:** Starting activities, passing data
- ▶ **RecyclerView:** Layout, view holder, adapter
- ▶ **Libraries:** GSON for working with JSON, Fresco for handling images



DEMO 2



# Session 7: Android for iOS Developers



MATERIAL DESIGN, NETWORKING



# MATERIAL DESIGN

---



Material is the metaphor



Bold, graphic, intentional

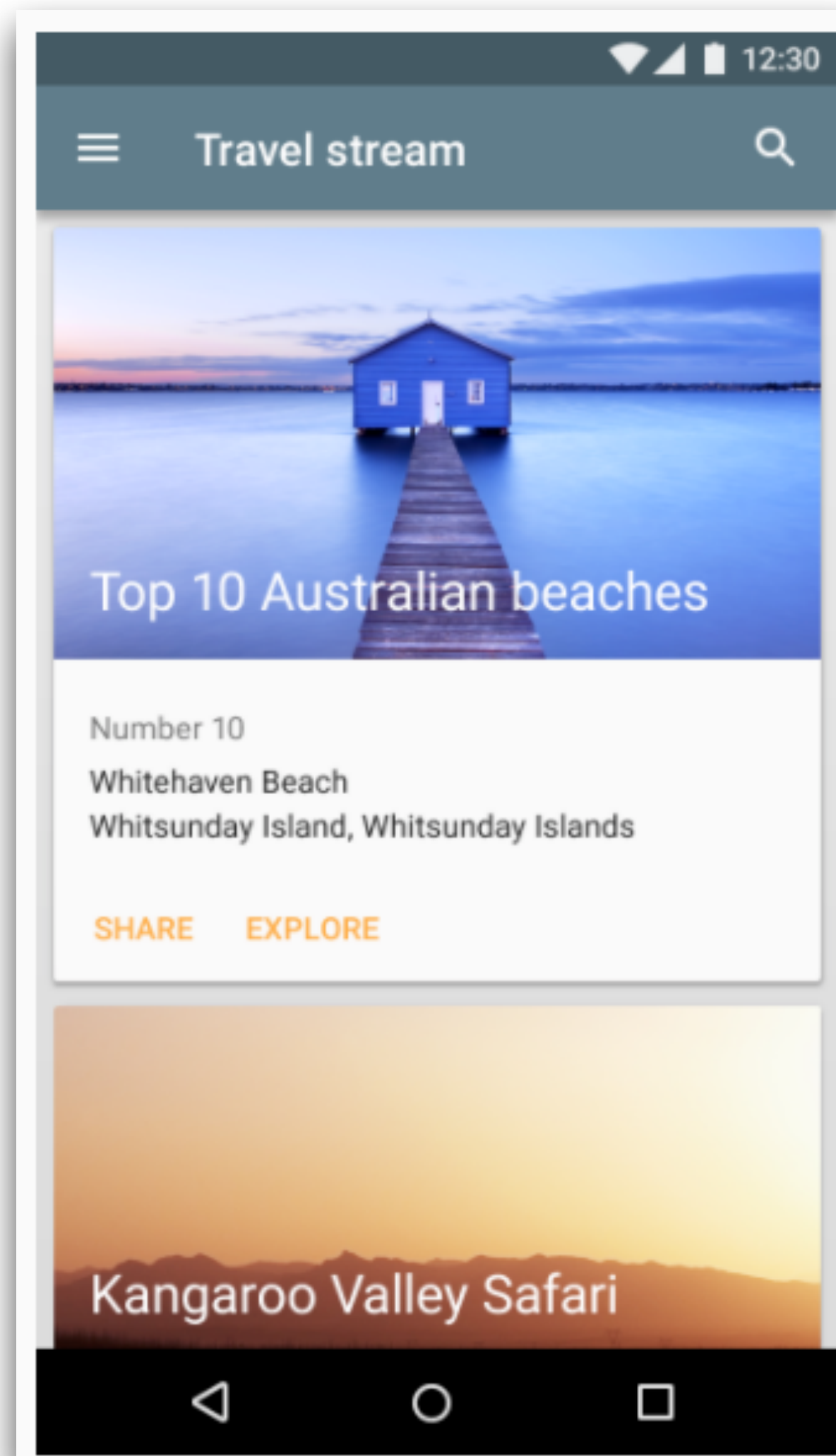


Motion provides meaning





# MATERIAL DESIGN - CARDS



# NETWORKING

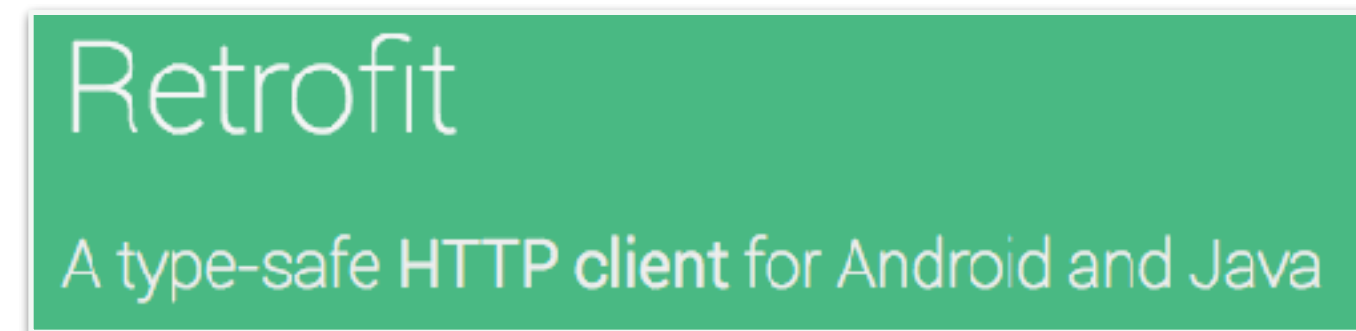
---

## iOS



- ▶ URLSession under the hood
- ▶ Response deserialization (JSON)
- ▶ Asynchronous requests

## Android



- ▶ OkHttp under the hood
- ▶ Response deserialization (JSON, XML, Protocol Buffers)
- ▶ Asynchronous or synchronous requests

# NETWORKING - RETROFIT

---

Create an interface based  
on your HTTP API

```
interface MyService {  
    @GET("users")  
    fun listUsers(@Query("id") id: String)  
    Call<List<Users>>  
}
```

Generate an implementation  
for your service

```
val retrofit = Retrofit.Builder()  
    .baseUrl("https://api.example.com/")  
    .build()  
  
val service = retrofit.create(MyService::class.java)
```

Call the service and handle  
the response / error

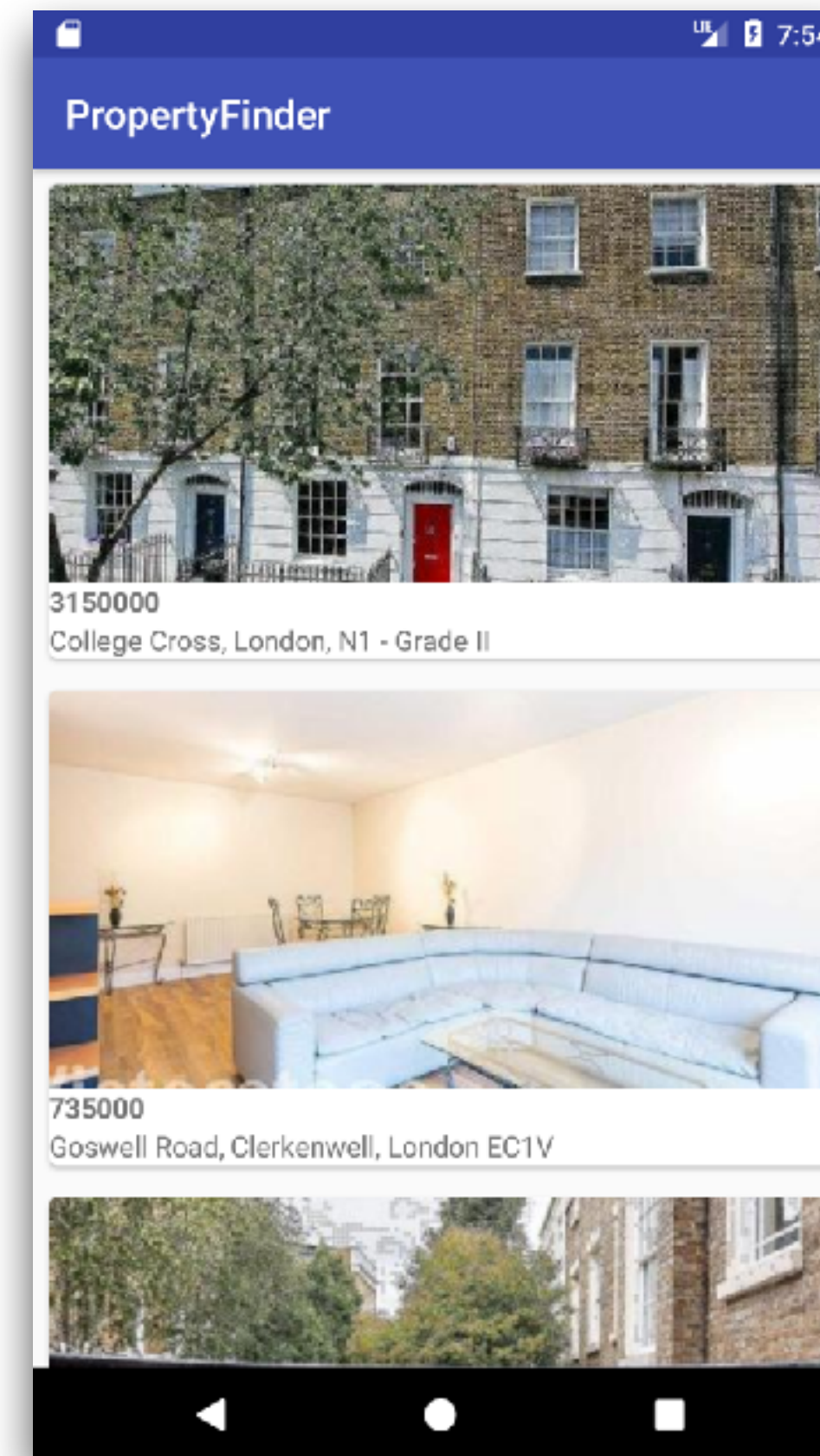
```
val call = service.listUsers("123")  
call.enqueue(object: Callback<List<Users>> {  
    override fun onResponse(...) {} // handle response  
    override fun onFailure(...) {} // handle error  
})
```



# WHAT'S IN DEMO 3

---

- ▶ **Material Design:** CardView Example
- ▶ **Networking:** Retrofit for making HTTP requests, Nestoria API, showing progress

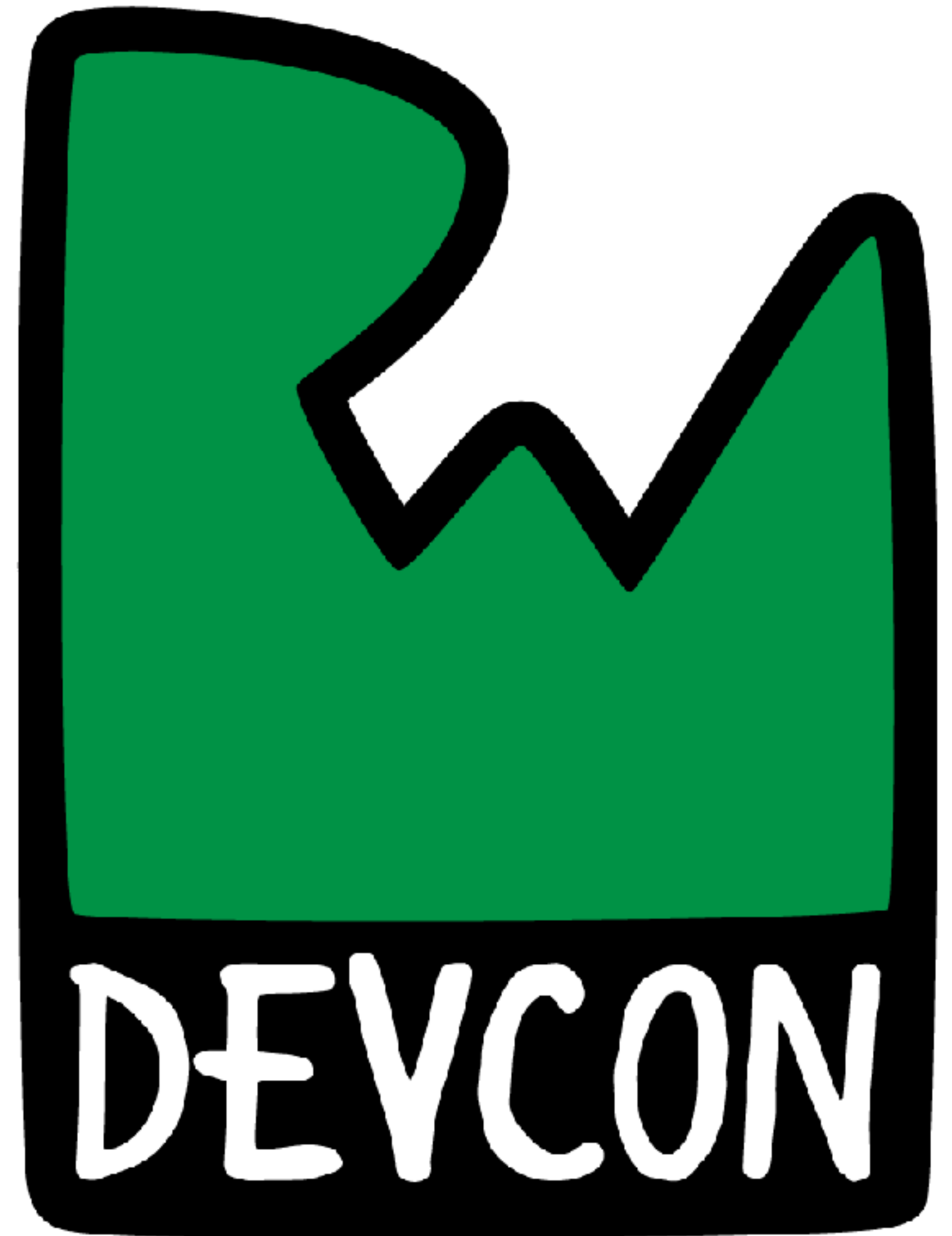


DEMO 3





# Session 7: Android for iOS Developers



CONCLUSION

# WHAT YOU LEARNED

---

- ⚙️ **Demo 1:** Creating the basic UI (Layout, Activity)
- ⚙️ **Demo 2:** Navigation (Intent), RecyclerView, Handling Data (GSON, Fresco)
- ⚙️ **Demo 3:** Material Design (CardView) and Networking (Retrofit)



# WHERE TO GO FROM HERE?

---

- ⚙ Learn more about Android Development
  - ⚙ <https://www.raywenderlich.com/category/android>
  - ⚙ <http://androidweekly.net/>
- ⚙ Material Design
  - ⚙ <https://material.io/guidelines>
  - ⚙ <https://www.materialpalette.com>
  - ⚙ <https://github.com/chrisbanes/cheesesquare>
- ⚙ Twitter: [@abernathyca](https://twitter.com/abernathyca)

