

Starting at 10am PST

Due to an All Hands at 9am PST, Day 3 of our Android Bootcamp will be starting an hour later at 10am PST

See you then 

Module Title in Title Case



Author Name

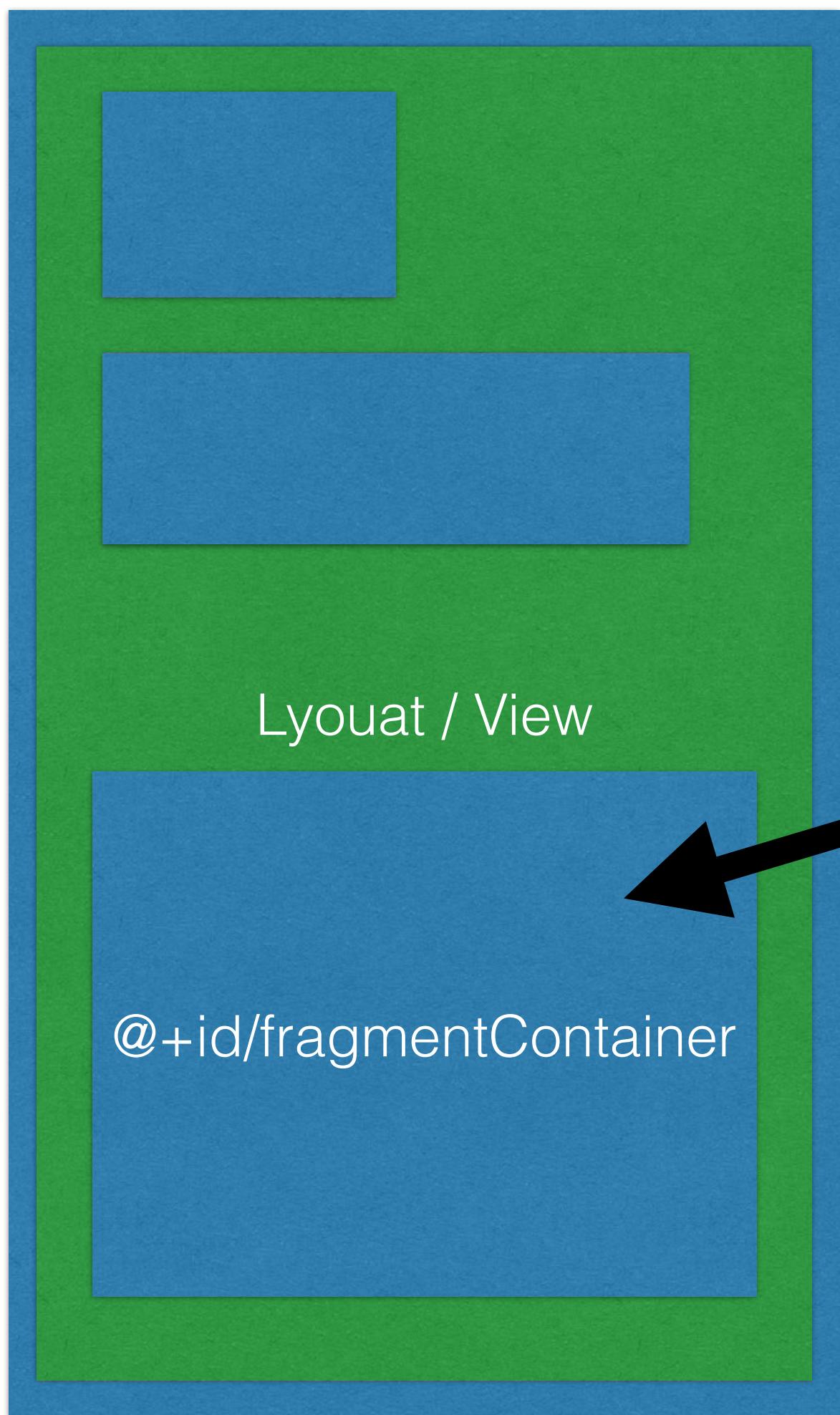
Author Title in Title Case

@authortwitter www.authorsite.com

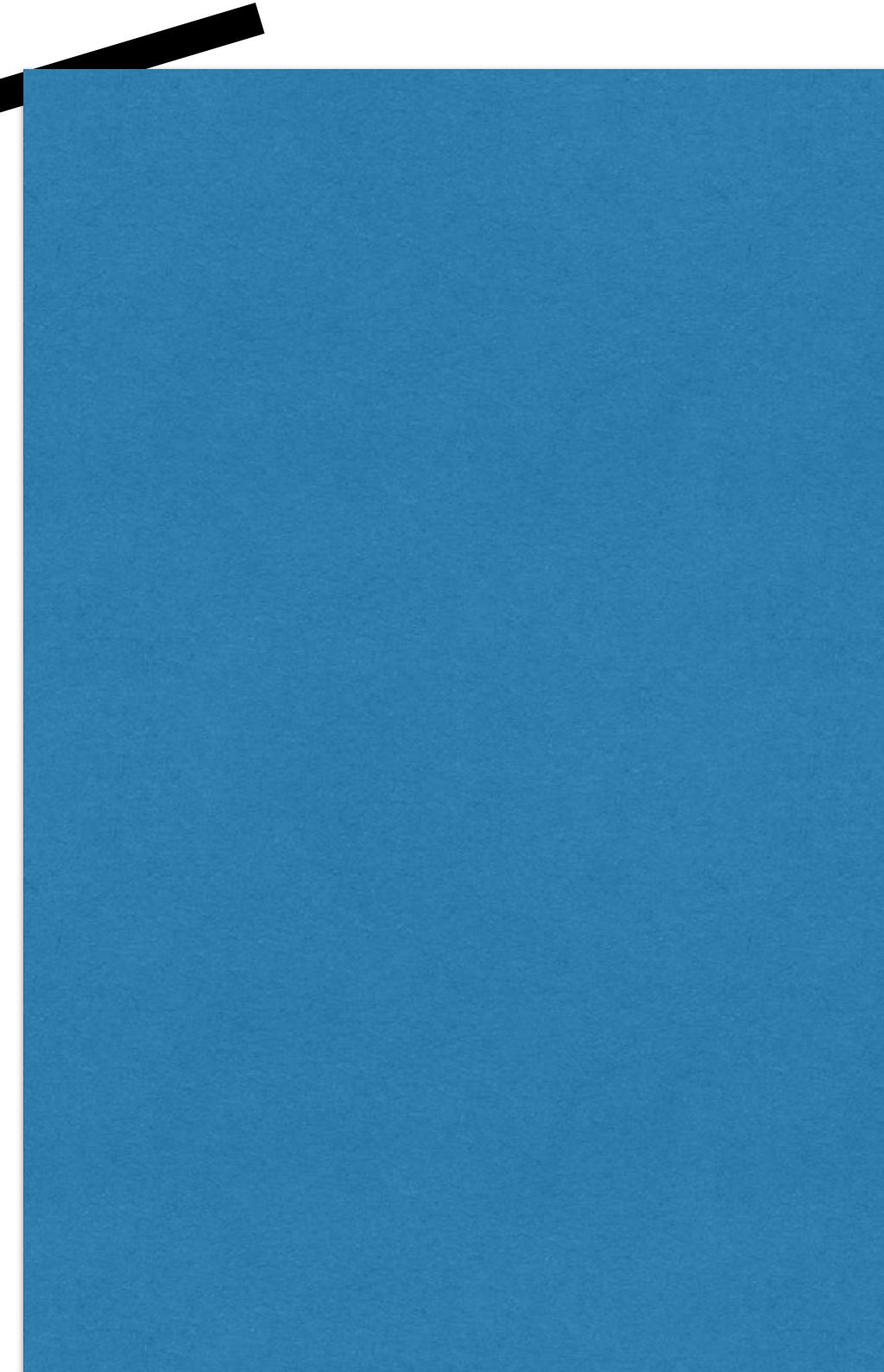
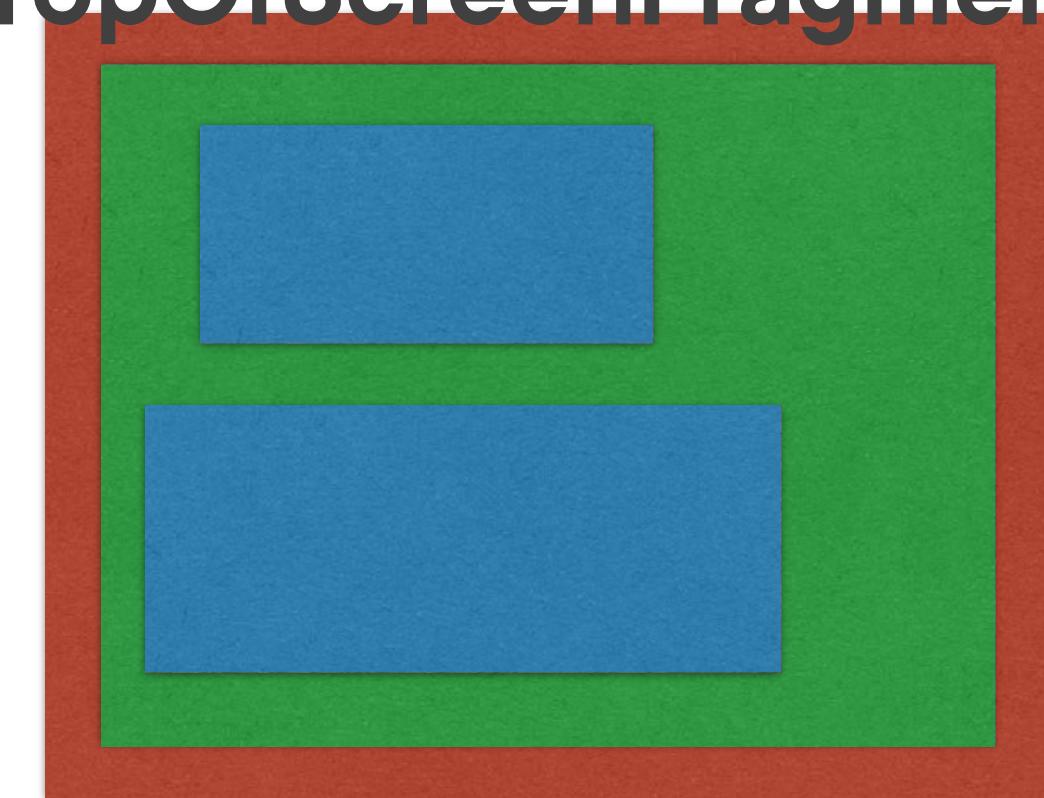


Any questions from yesterday?

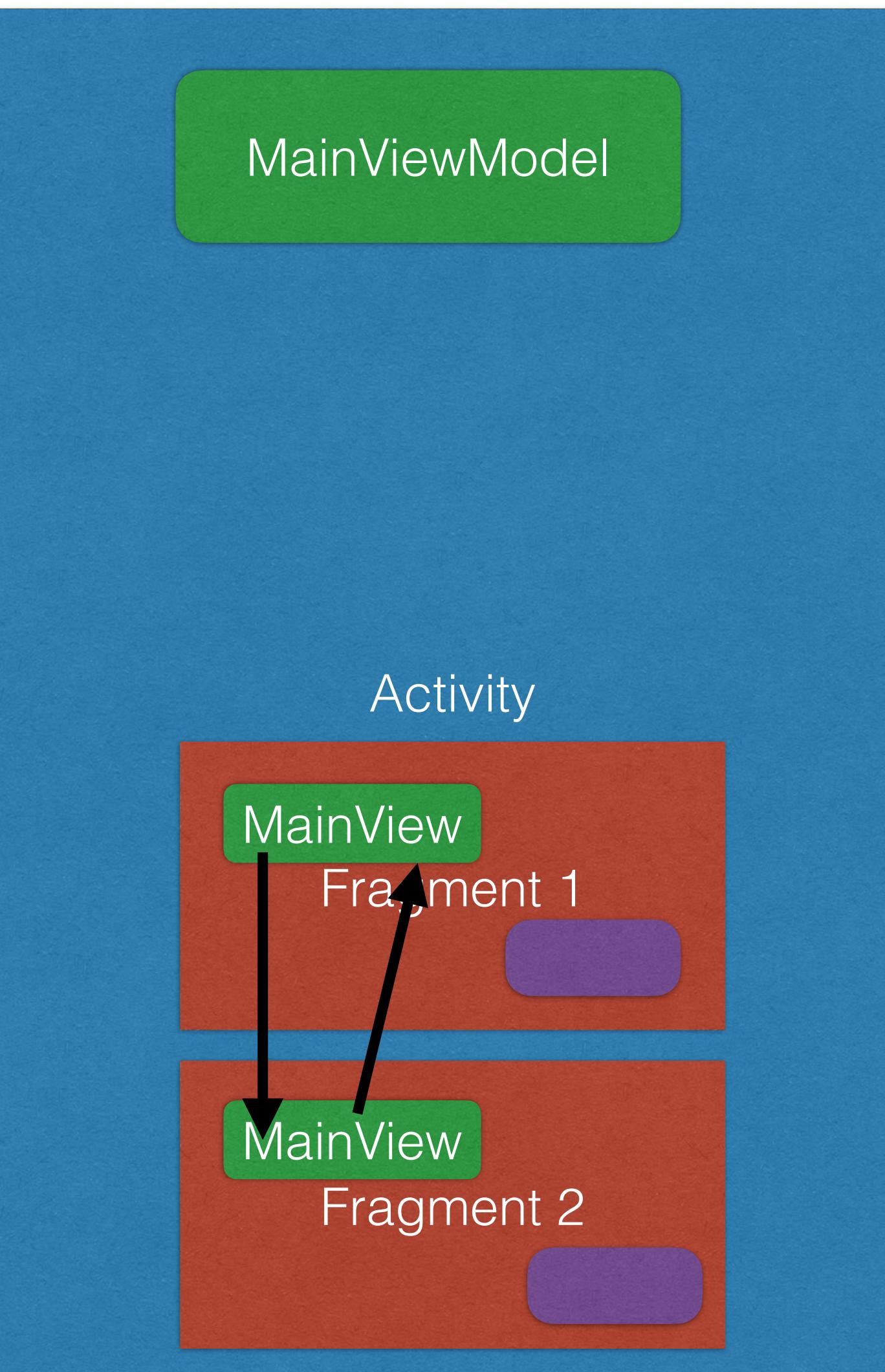
Any UI on the screen requires an Activity



TopOfScreenFragment



Any UI on the screen requires an Activity



What are we
learning today?

Explicit & Implicit Intents

RecyclerView

Local Storage Solutions

DataStore

Building for a global audience

MVVM & Android Architecture Components



Let's check in on our “want to learns”

Any Questions?

Working with Intents



Nate Ebel

Android Developer & Instructor

@n8ebel goobar.dev

Overview

Understand what Intents are used for

Understand Implicit vs Explicit Intents

Understand how to start an Activity with an Explicit Intent

Understand how to perform an action with an Implicit Intent

Intent

A messaging object you can use to request an action from another app component

<https://developer.android.com/guide/components/intents-filters>

Intent Types

Explicit Intents

Implicit Intents

Explicit Intent

Specify the specific application or component to start - typically within your own app.

ex. “start our MainActivity”

<https://developer.android.com/guide/components/intents-filters>

Explicit Intent

Always delivered to its target, regardless of any intent filters the component declares.

<https://developer.android.com/guide/components/intents-filters#Receiving>

Implicit Intent

Specify an action to perform - and let the OS and user choose which application to handle the request based on declared intent filters.

ex. “send a text”

<https://developer.android.com/guide/components/intents-filters>

Intent Filter

Specifies the type of intents it accepts based on the intent's action, data, and category. The system delivers an implicit intent to your app component only if the intent can pass through one of your intent filters

<https://developer.android.com/guide/components/intents-filters#Receiving>

What are some Implicit Intent
examples?

Displaying List Data with RecyclerView



Nate Ebel

Android Developer & Instructor

@n8ebel goobar.dev

Overview

Understand the need for efficient data display

Understand the efficiencies of RecyclerView

Understand how to display data with RecyclerView

Understand efficient list diffing

Why is efficient display of list
data so important?

List Display Widgets



ListView



RecyclerView

List Display Widgets



RecyclerView

RecyclerView

Makes it easy to efficiently display large sets of data. You provide the data and define how each item layout, and the RecyclerView library dynamically creates the elements when they're needed.

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

RecyclerView

RecyclerView “recycles” individual elements. When an item scrolls off the screen, RecyclerView doesn't destroy the item's view. Instead, RecyclerView reuses the view for new items that have scrolled on to the screen.

<https://developer.android.com/guide/topics/ui/layout/recyclerview>

RecyclerView Components

RecyclerView

RecyclerView.ViewHolder

RecyclerView.Adapter

LayoutManger

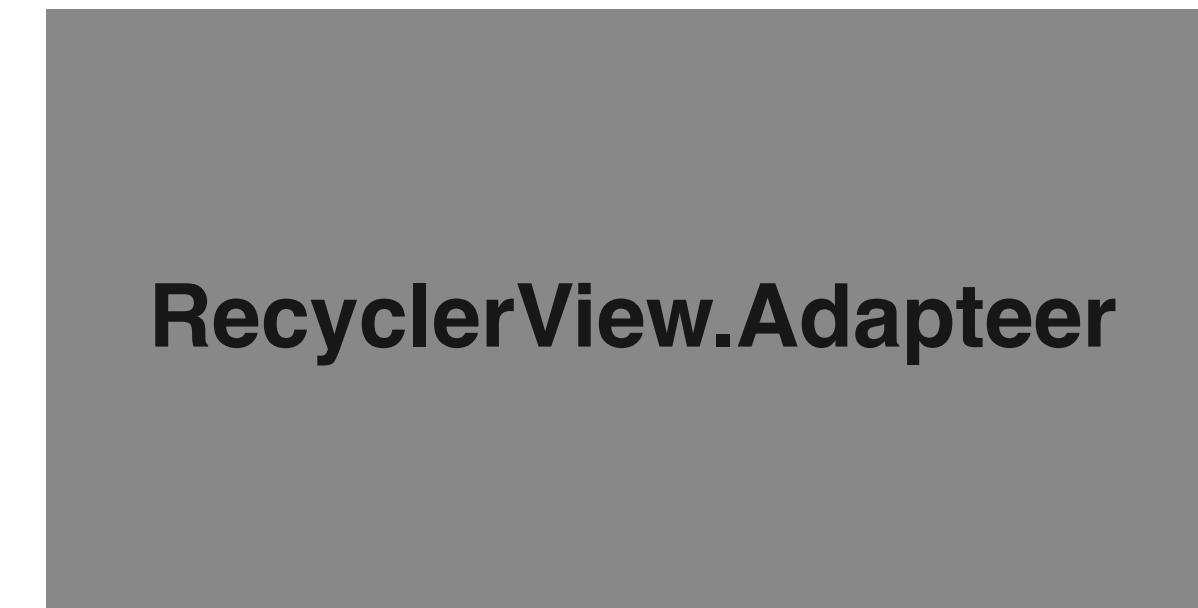
RecyclerView



RecyclerView



RecyclerView.Adapter



Kotlin

Java

C++

Rust

Java

Go



Kotlin

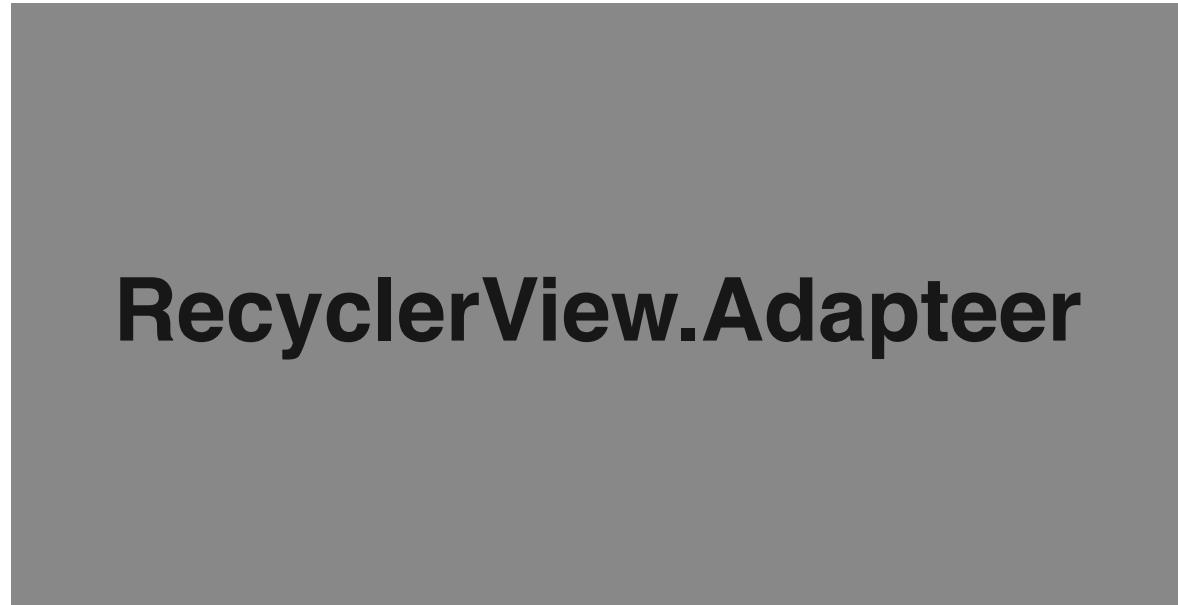
Java

C++

Rust

Java

Go



RecyclerView.Adapteer

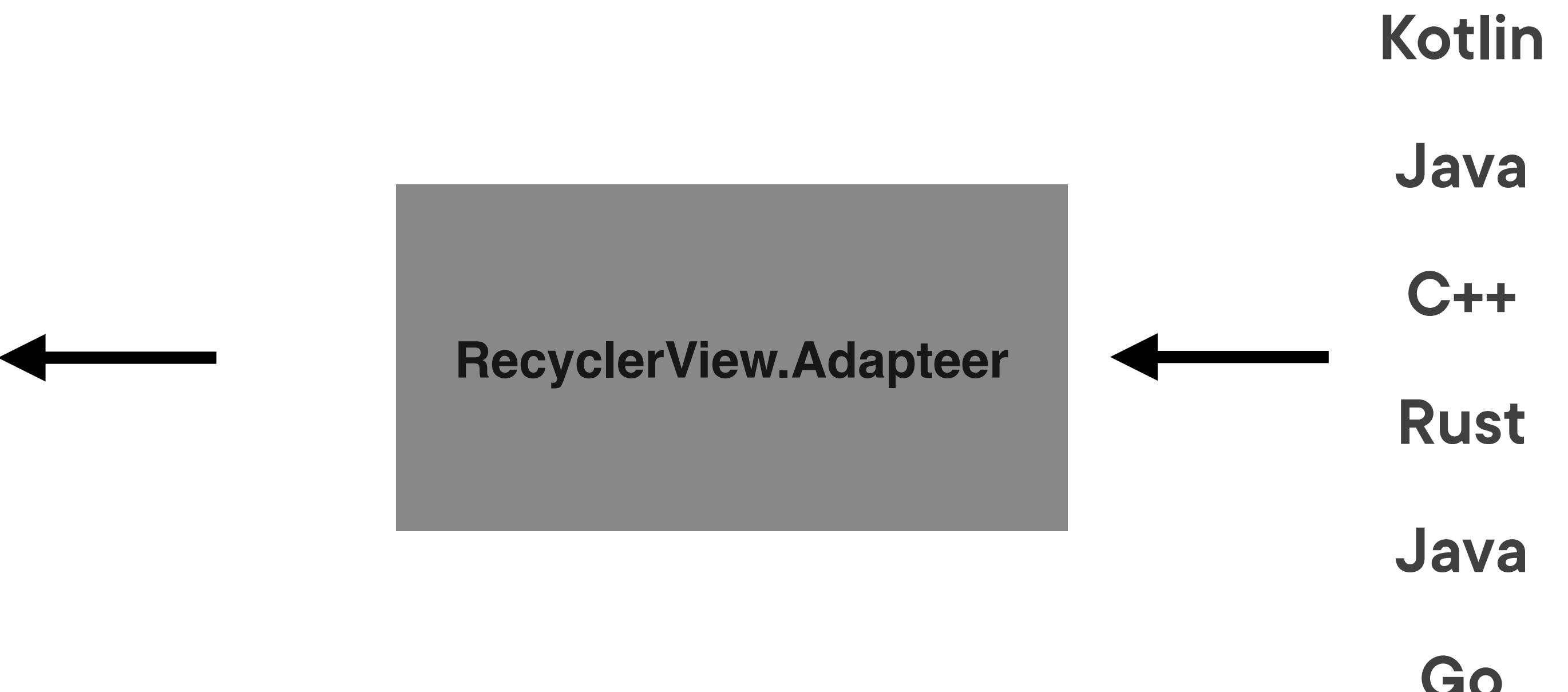
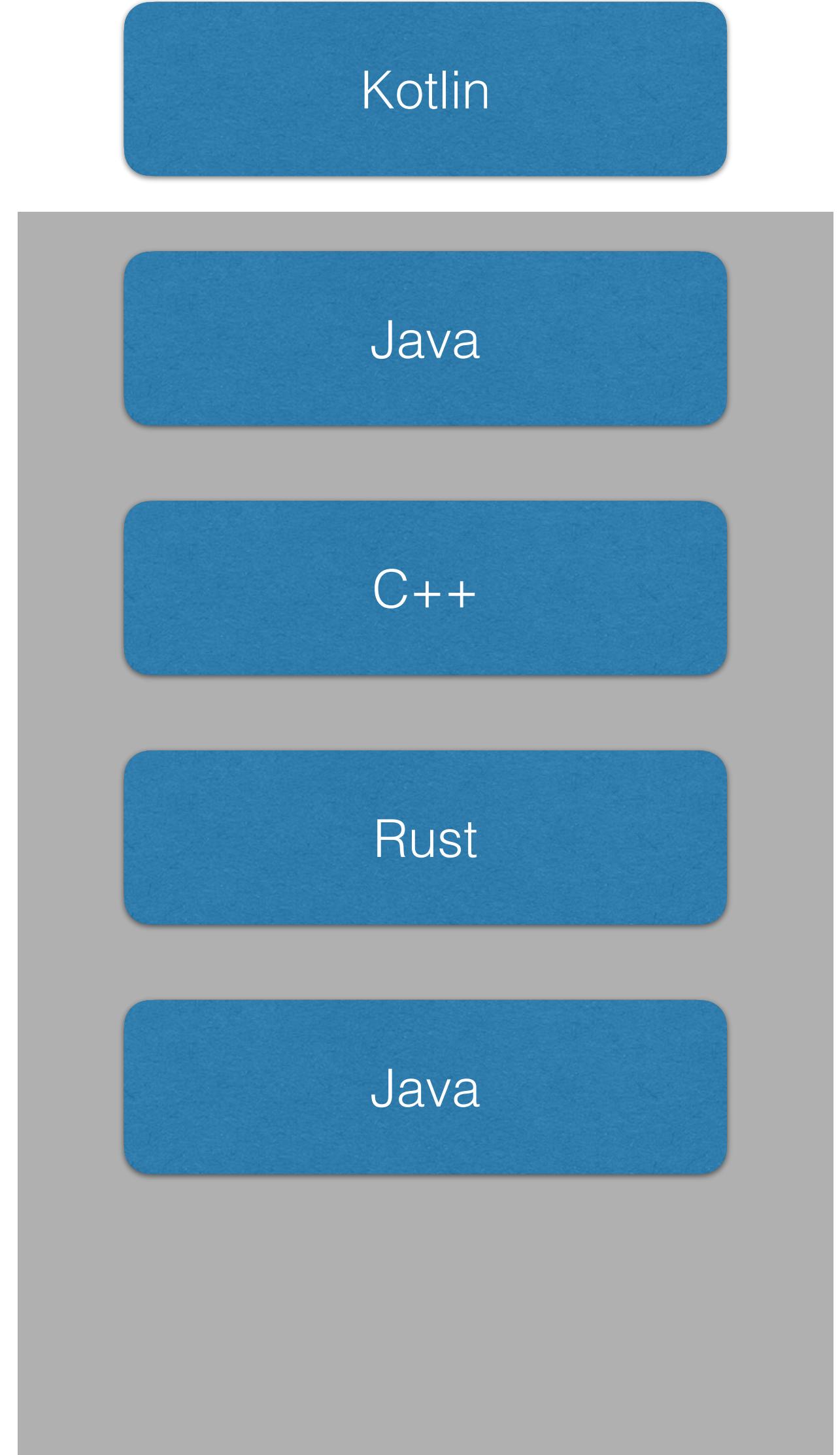


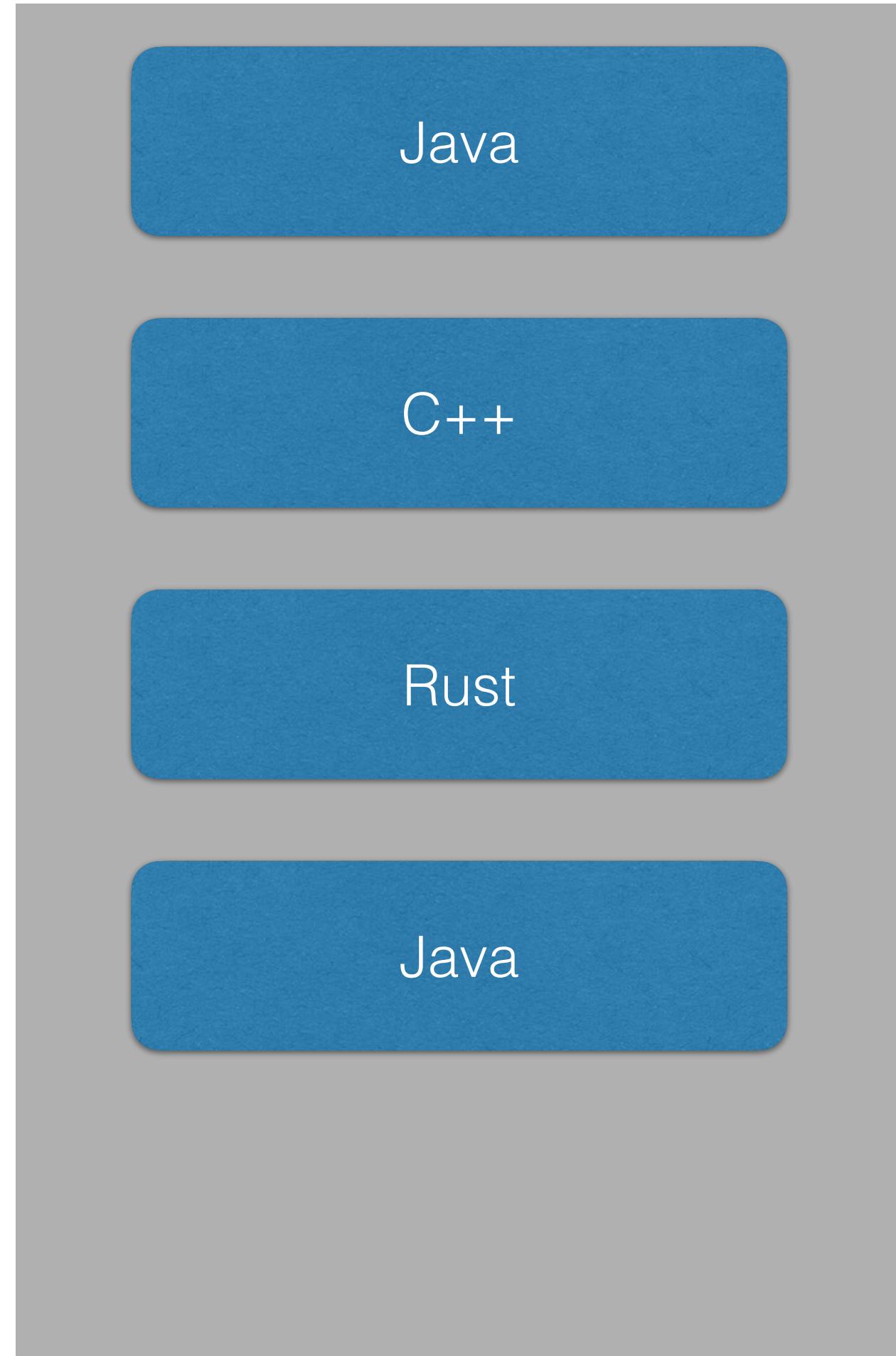


RecyclerView.Adapteer



Kotlin
Java
C++
Rust
Java
Go





RecyclerView.Adapteer



Kotlin

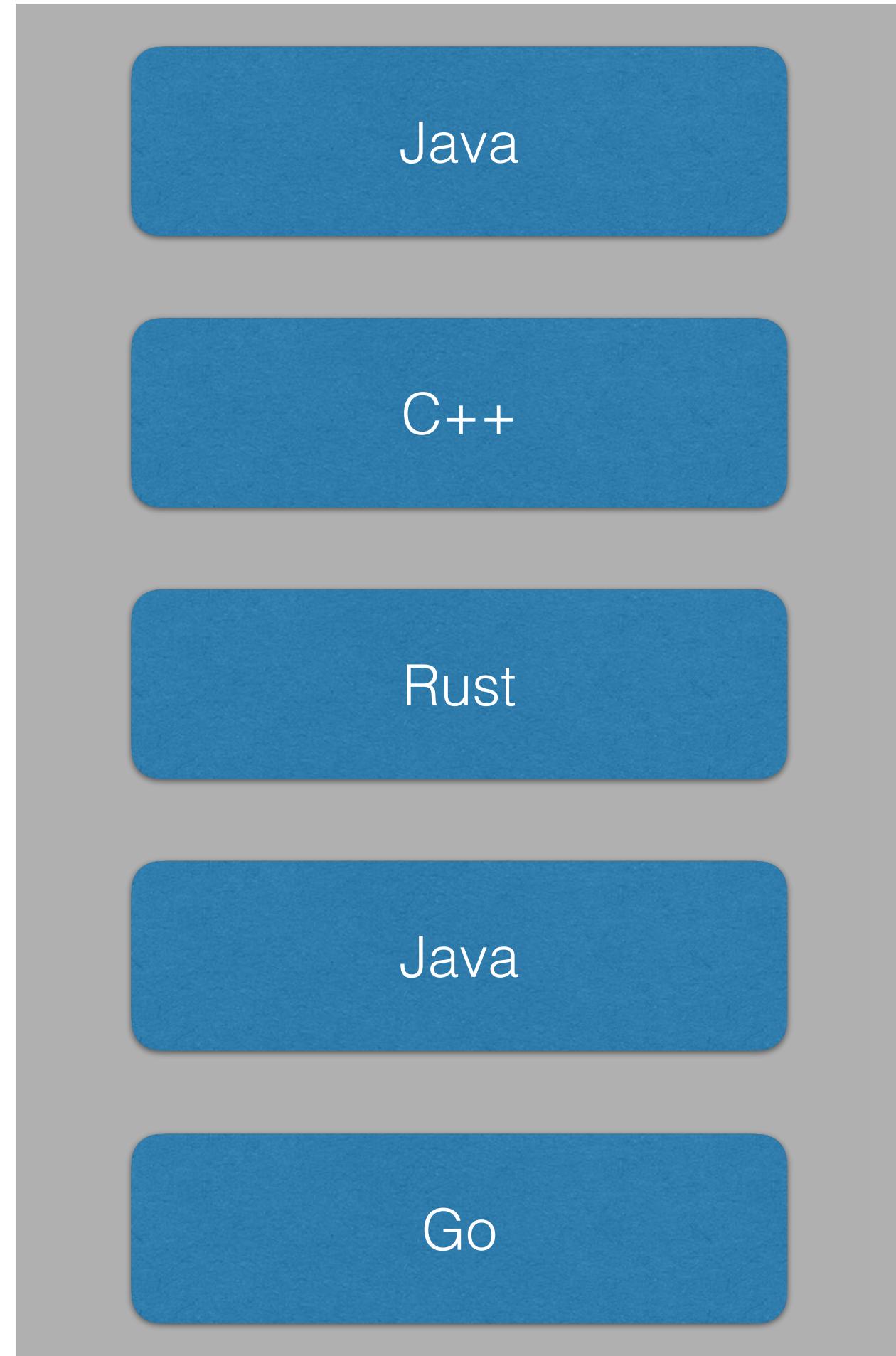
Java

C++

Rust

Java

Go



RecyclerView.Adapteer



Kotlin
Java
C++
Rust
Java
Go

RecyclerView

RecyclerView is the ViewGroup

RecyclerView.ViewHolder controls item layout

**RecyclerView.Adapter converts data into
ViewHolders**

**LayoutManager controls how item views are
arranged**

ListAdapter

Base class for presenting List data in a RecyclerView, including computing diffs between Lists on a background thread.

<https://developer.android.com/reference/androidx/recyclerview/widgetListAdapter>

Exploring Local Storage Solutions



Nate Ebel

Android Developer & Instructor

@n8ebel goobar.dev

Overview

Understand the need for local data storage

Understand SharedPreferences

Understand DataStore

Understand Room

Why do we need to save local
data?

What data are we
saving?

Application settings/preferences

Profile data

User generated content

Data fetched from network

Downloaded files

Local Storage Options

Filesystem

SharedPreferences

Jetpack DataStore

SQLite

Jetpack Room

ORMs

SharedPreferences

Abstraction interface for saving/accessing key/value data to a local file

<https://developer.android.com/reference/android/content.SharedPreferences>

SharedPreferences

Great for key/value pairs of simple data types

Data saved to local files

Edits are made through an Editor

DefaultSharedPreferences

Named preferences

PreferenceScreen for preference-generated UI for settings

DataStore

Allows you to store key-value pairs or typed objects with protocol buffers. DataStore uses Kotlin coroutines and Flow to store data asynchronously, consistently, and transactionally.

<https://developer.android.com/topic/libraries/architecture/datastore>

DataStore



- Modern replacement for SharedPreferences**
- Key/value data**
- Typed data via protocol buffers**
- Efficient asynchronous edits**

Protocol Buffers

Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data – think XML, but smaller, faster, and simpler

<https://developers.google.com/protocol-buffers>

```
// example.proto

message Person {
    string name = 1;
    int32 id = 2; // Unique ID number for this person.
    string email = 3;

    enum PhoneType {
        MOBILE = 0;
        HOME = 1;
        WORK = 2;
    }

    message PhoneNumber {
        string number = 1;
        PhoneType type = 2;
    }
}
```

- ◀ **Defined in a .proto file**
- ◀ **Platform/language agnostic syntax**
- ◀ **Tooling translates this into source code for desired language such as C++, Java, Kotlin, etc**

Room

Provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite.

<https://developer.android.com/jetpack/androidx/releases/room>



Lunch

Using DataStore for Local Storage



Nate Ebel

Android Developer & Instructor

@n8ebel goobar.dev

Overview

- Understand how to use DataStore**
- Understand how to save data with DataStore**
- Understand how to retrieve data with DataStore**
- Understand how to display local data**

DataStore or
SharedPreferences?

Which DataStore to Use?

Preferences DataStore

Proto DataStore

Preference DataStore

Stores and accesses data using keys. This implementation does not require a predefined schema, and it does not provide type safety.

<https://developer.android.com/topic/libraries/architecture/dastore>

Proto DataStore

Stores data as instances of a custom data type. This implementation requires you to define a schema using protocol buffers, but it provides type safety.

<https://developer.android.com/topic/libraries/architecture/dastore>



Office Hours