## Activities and Intents

### Activity

An activity in Android represents a single, focused screen within your application. It is the fundamental building block that provides the user interface (UI) and the logic behind what users see and interact with on your app.

#### Key Responsibilities:

* **UI Definition:** An activity defines the visual structure of the screen using an XML layout file. This layout file combines various UI elements like buttons, text views, images, and more, to create the user interface the user interacts with.
* **User Interaction Handling:** The activity class written in Java or Kotlin handles user interactions with the UI elements defined in the layout. This code defines what happens when a button is clicked, a text field is edited, or any other user action is performed on the screen.
* **Lifecycle Management:** An activity goes through various stages in its lifetime, managed by the Android system's Activity Manager. These stages include creation, starting, resuming, pausing, stopping, and destruction. Your activity code can implement methods specific to each stage to perform necessary actions (e.g., fetching data when the activity starts or saving data when it is paused).

**Core Components of an Activity:**

* **Layout (XML):** Defines the visual structure of the screen using a declarative approach. It specifies the arrangement of UI elements and their properties.
* **Activity Class (Java/Kotlin):** Written code that handles user interactions, manages the activity's lifecycle, and interacts with other components of your app (services, content providers, etc.). This code connects the UI elements defined in the layout to actions and functionalities.
* **Intent (Optional):** An intent is a message used to launch an activity from another activity, service, or even from outside your app. Intents specify what activity to launch and can optionally carry data along with the request.

**Activity Lifecycle:**

1. **onCreate():** Called when the activity is first created. This is where you typically perform initial setup tasks like inflating the layout and initializing UI elements.
2. **onStart():** Called when the activity is becoming visible to the user. You can start any animations or data fetching operations here.
3. **onResume():** Called when the activity has become fully active and interactive in the foreground. This is where you place most of your UI interaction logic.
4. **onPause():** Called when the activity is about to lose foreground focus. You can pause animations or stop ongoing tasks here.
5. **onStop():** Called when the activity is no longer visible to the user. You can stop background tasks or release resources here.
6. **onDestroy():** Called when the activity is destroyed and removed from memory. You can clean up resources and unregister listeners here.

A diagram of a process

Description automatically generated

### Creating a new Activity

* **Launch Android Studio:** Open Android Studio on your computer.
* **Start a new project (or open an existing one):** If you are starting fresh, choose "New Project" and follow the prompts. If you are working on an existing project, open it in Android Studio.
* **Create a new Activity:**
  + Go to **File > New > Activity (or Class)**.
  + Choose an activity template (e.g., Empty Views Activity, Blank Activity). These templates provide basic structures to get you started.
  + Customize the Activity name (e.g., MainActivity) and click "Finish."

#### Define the layout.

* **Open the layout file:** Android Studio automatically creates a layout file (e.g., activity\_main.xml) associated with your activity.
* **Design the UI:**
  + Use the layout editor (visual drag-and-drop interface) or write XML code to define the layout of your screen.
  + Add UI elements like buttons, text views, and images from the Palette.
  + Arrange elements using layout managers (e.g., ConstraintLayout, LinearLayout) to position them within the layout.
  + Set attributes for each element to define its appearance and behavior (size, color, text content, etc.).

### Intents

Intents are a fundamental messaging system in Android that allows communication between various components of your application, or even between different applications. They function as messengers, carrying information about an action to be performed and potentially some data to be used during that action.

**Types of Intents:**

* **Explicit Intents:** These explicitly specify the target component (activity, service, broadcast receiver) to interact with. You provide the class name of the target component within the Intent object.
* **Implicit Intents:** These do not specify a specific component but rather describe an action to be performed (e.g., view an image, play audio). The Android system finds an appropriate app that can handle the action and delivers the Intent to that app's component.

**Common Use Cases for Intents:**

* **Starting Activities:** Launching new activities within your app or from other applications.
* **Starting Services:** Initiating background services to perform long-running tasks.
* **Sending Broadcast Messages:** Notifying components about system events or custom app events.
* **Opening URLs:** Launching the web browser to display a specific web page.
* **Sharing Data:** Sharing text, images, or other data between applications.

**Creating Intents:** You can create Intent objects using the Intent class constructor, specifying the action (optional) and the target component (for explicit intents).

#### Intent Examples

* Explicit Intent
  + Sharing values from one activity to another when opening an activity from another activity.
  + **First Activity Code**

A screen shot of a computer code

Description automatically generated

* + Inside the lambda, we create an Intent specifying the target activity class (SecondActivity::class.java).
  + We can optionally add data to the Intent using putExtra with a key-value pair (here, "message" with a string value).
  + Finally, we call startActivity(intent) to launch the SecondActivity.
  + **Second Activity code**

A screenshot of a computer

Description automatically generated

* Implicit Intent

A screen shot of a computer

Description automatically generated