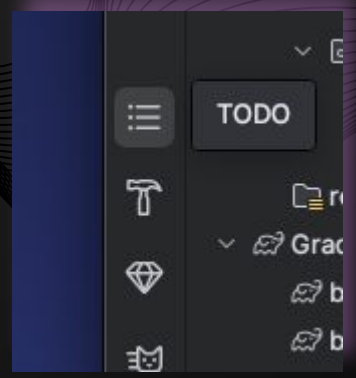
05 Dev Task

Develop an Android App that selects an Image from the phone’s camera roll and uploads the image, via an API to be processed by a Computer Vision model.

Getting Started

1. Download the example App from:
   https://github.com/wpburns/we-bridge-app
   a. Click on ‘Code’ then ‘Download Zip’
2. Unzip the downloaded file
3. Open Android Studio
4. Select ‘File’ > ‘Open’ and navigate to the location you extracted the zip file. Select ‘ImageClassifier’
5. On the left of the screen, click the ‘ToDo’ Icon

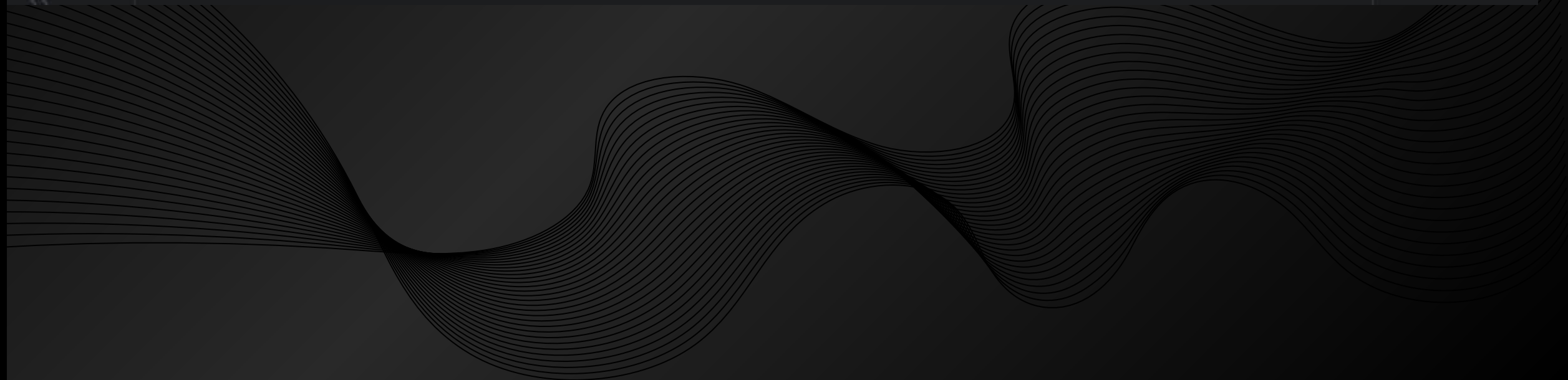


TODO 1 – Permissions

```
<!-- TODO: Include permissions to access the Internet and Read External Storage-->
<!-- For Android 12 and below -->
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
    android:maxSdkVersion="32" />

<!-- For Android 13+ (API 33+) -->
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
<uses-permission android:name="android.permission.INTERNET" />
```

TODO 2 – API Endpoint

```
30      private static final int PERMISSION_REQUEST_CODE = 100;
31      // TODO: Update the API endpoint url
            1 usage
32      private static final String API_ENDPOINT = "https://correct-optimal-marmoset.ngrok-free.app/classify";
```



TODO 3 – UI Logic

```
44        // TODO: Setup User Interface code. Connect buttons to the code.
45
46        client = new OkHttpClient();
47
48        selectImageBtn = findViewById(R.id.selectImageBtn);
49        uploadBtn = findViewById(R.id.uploadBtn);
50        responseText = findViewById(R.id.responseText);
51        imagePreview = findViewById(R.id.imagePreview);
52
53        selectImageBtn.setOnClickListener( View v -> checkPermissionAndSelectImage());
54        uploadBtn.setOnClickListener( View v -> uploadImage());
55
56        uploadBtn.setEnabled(false);
```

TODO 4 – Handle API Response

```
174        // TODO: Add code to display the response from the image classification API
175        try {
176            JSONObject json = new JSONObject(jsonString);
177            StringBuilder formatted = new StringBuilder();
178            formatted.append("API Response:\n\n");
179
180            // Check if response contains base64 image
181            if (json.has( name: "response_image")) {
182                String base64Image = json.getString( name: "response_image");
183                displayBase64Image(base64Image);
184            }
185
186            // Format JSON for display
187            formatJsonObject(json, formatted,  indent: 0);
188
189            responseText.setText(formatted.toString());
190        } catch (JSONException e) {
191            responseText.setText("Response: " + jsonString);
192        }
```

TODO 6 – Setup UI (1/2)

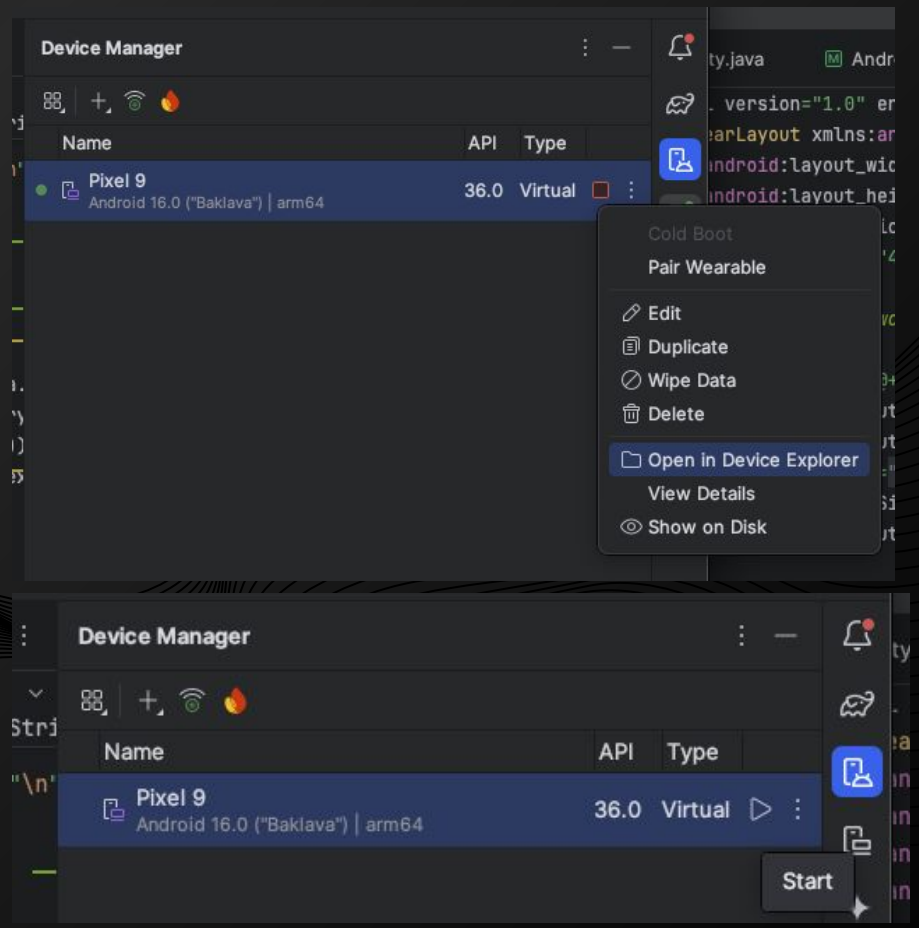
```
<!-- TODO: Add two Buttons, an ImageView, TextView and ScrollView -->
<Button
    android:id="@+id/selectImageBtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Select Image"
    android:textSize="16sp"
    android:layout_marginBottom="16dp" />

<ImageView
    android:id="@+id/imagePreview"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginBottom="16dp"
    android:background="#f0f0f0"
    android:scaleType="centerCrop"
    android:visibility="gone"
    android:contentDescription="Selected image preview" />
```

TODO 6 – Setup UI (2/2)

```
26
27      <Button
28          android:id="@+id/uploadBtn"
29          android:layout_width="match_parent"
30          android:layout_height="wrap_content"
31          android:text="Analyse Image"
32          android:textSize="16sp"
33          android:layout_marginBottom="16dp"
34          android:enabled="false" />
35
36      <ScrollView
37          android:layout_width="match_parent"
38          android:layout_height="0dp"
39          android:layout_weight="1">
40
41          <TextView
42              android:id="@+id/responseText"
43              android:layout_width="match_parent"
44              android:layout_height="wrap_content"
45              android:text="Select and image to get started"
46              android:textSize="14sp"
47              android:padding="16dp"
48              android:background="#f5f5f5"
49              android:fontFamily="monospace" />
50
51      </ScrollView>
```

TODO 7 – Add images and Test

1. Select 'Device Manager' from the icons on the right side of the screen
2. Click the 3 vertical dots, and select 'Open Device Explorer'
3. Navigate to `storage > emulated > 0 > Pictures`
4. Select the 'Pictures' folder, and press the 'Upload' icon
5. Select the test images from the ‘test_images’ folder (Or add some of your own from Google)
6. Start the Virtual device by clicking ‘Start’



Your Mobile App

Once you've got everything working, why not try modifying the code?

Add your own app icon?

Change the button colours and layout in 'activity_main.xml'

Format the API response to look better in ~line 159 in MainActivity.java