Scan Barcode or QR code Using Camera

Tutorial followed: https://www.youtube.com/watch?v=A2PqYkLb_-A

Implement barcode scanner using the ZXing library.

Step 1: Add following dependencies to Gradle file and sync.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    //noinspection GradleCompatible
    implementation 'com.google.android.gms:play-services-auth:16.0.0'
    implementation 'me.dm7.barcodescanner:zxing:1.9'
}
```

Step 2: Add camera permissions in the Android manifest file.

```
<uses-permission android:name="android.permission.CAMERA"></uses-permission>
```

Step 3: Add a button and text view to xml layout file. Button will scan the QR code or barcode. And text view will display the results.

Step 4: Bind UI elements in MainActivity. Set an onClickListener for the scan button, this will start activity for the camera to scan the code, therefore another activity should be called ScanCodeActivity etc.

Step 5: Inside the ScanCodeActivity.class you must implement ZXingScannerView.ResultHandler. Implement the appropriate methods. Make new ZXingScannerView object, set it as the content view.

```
public class ScanCodeActivity extends AppCompatActivity implements ZXingScannerView.ResultHandler{

    ZXingScannerView ScannerView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ScannerView =  new ZXingScannerView( context: this);
        setContentView(ScannerView);

    }
```

Step 6: Implement handleResult, make text view from Main Activity public static, and set display to result of scanner. Implement onPause(), and onResume().

```
@Override
public void handleResult(Result result) {

    MainActivity.resultTExtView.setText(result.getText());
    onBackPressed();

}

@Override
protected void onPause() {
    super.onPause();

    ScannerView.stopCamera();
}

@Override
protected void onResume() {
    super.onResume();

    ScannerView.setResultHandler(this);
    ScannerView.startCamera();
}
```

Our implementation varies:

- Created and passed information through intent.
- Result returns either QR code or barcode.
- Follow link from QR code to server where json file is
- Barcode uses API to find information about item, returns json

Scan Barcode from Barcode Lookup API

Tutorial followed: https://www.barcodelookup.com/api

Authentication

Every user account is assigned one unique API key upon account creation.
Authenticate your account by including your API Key in all requests.

```
https://api.barcodelookup.com/v2/products?key=your_api_key
```

"The Barcode Lookup API documentation describes how to use our RESTful API service. All API responses are returned in JSON format."

Request URL Example

```
https://api.barcodelookup.com/v2/products?
barcode=3614272049529&formatted=y&key=your_api_key
```