Software for Mobile Devices

BS (CS) \_Spring\_2025

**Lectures\_13 Manual**



Learning Objectives:  
**ASynTask**

**Input stream and internet connection**

**1. Download Image Example**

This example shows how to download an image from a URL and display it in an ImageView using an AsyncTask in Android.

**Step-by-Step Breakdown:**

1. **Layout Setup**:
   * The layout file (activity\_download\_image\_example.xml) contains:
     + ImageView (imageDown): Displays the downloaded image.
     + Button (btnDown): The user clicks this button to initiate the download.
2. **MainActivity** (DownloadImageExample.java):
   * The DownloadImageExample class extends AppCompatActivity, the base activity class in Android.
   * We initialize the ImageView and Button in the onCreate() method.
   * When the button is clicked, the AsyncTask class ImageDownload is instantiated and executed to download the image in the background.
   * After the image is downloaded, we set the Bitmap to the ImageView using imageView.setImageBitmap(bitmap).
3. **ImageDownload Class**:
   * This class extends AsyncTask<String, Void, Bitmap>, where:
     + The String parameter represents the URL of the image.
     + The Void parameter represents progress updates (not used in this example).
     + The Bitmap parameter represents the result of the background task (the downloaded image).
   * In the doInBackground() method:
     + A connection to the URL is opened.
     + The input stream is obtained from the URL connection, and the image is decoded into a Bitmap object using BitmapFactory.decodeStream().
   * The Bitmap object is returned to the onPostExecute() method (not shown, but it's implicitly handled by AsyncTask).
4. **Button Click Handler**:
   * When the button is clicked, the AsyncTask is executed with the URL of the image to be downloaded.
   * The image is set to the ImageView once the download is complete.

**Key Concepts Explained:**

* AsyncTask: Helps perform background operations (like downloading) and updates the UI thread without blocking the main thread.
* BitmapFactory.decodeStream(): Converts the downloaded stream into a bitmap.
* HttpsURLConnection: Used to open an HTTPS connection to download the image.

**2. Web Content Download Example**

This example demonstrates how to download the content of a webpage and display it in a EditText widget using an AsyncTask in Android.

**Step-by-Step Breakdown:**

1. **Layout Setup**:
   * The layout file (activity\_web\_content\_download\_main.xml) contains:
     + EditText (tbtMulltiLineWeb): Displays the downloaded web content.
     + Button (button8): The user clicks this button to initiate the download of web content.
2. **MainActivity** (WebContentDownloadMainActivity.java):
   * The WebContentDownloadMainActivity class extends AppCompatActivity.
   * We initialize the EditText and Button in the onCreate() method.
   * When the button is clicked, the AsyncTask class DownloadWebContent is instantiated and executed to download the webpage content in the background.
   * Once the content is downloaded, we set the retrieved content as the text of the EditText.
3. **DownloadWebContent Class**:
   * This class extends AsyncTask<String, Void, String>, where:
     + The String parameter represents the URL of the webpage.
     + The Void parameter represents progress updates (not used in this example).
     + The String parameter represents the result of the background task (the webpage content).
   * In the doInBackground() method:
     + A connection to the URL is opened using HttpURLConnection.
     + The input stream is obtained from the URL connection.
     + An InputStreamReader is used to read the stream.
     + The content is read character by character and appended to a String that is returned once the content is fully fetched.
4. **Button Click Handler**:
   * When the button is clicked, the AsyncTask is executed with the URL of the web page to be downloaded.
   * Once the content is downloaded, it's set as the text of the EditText widget.

**Key Concepts Explained:**

* AsyncTask: Used to perform background tasks and update the UI thread without blocking the main thread.
* HttpURLConnection: Allows communication with an HTTP server to download the webpage content.
* InputStreamReader: Converts byte streams into character streams to read the web content.

**General Explanation of AsyncTask**

AsyncTask is an Android class that allows you to perform background operations and publish results on the main UI thread. It is helpful for tasks that might take a long time to process, such as downloading data from the internet.

**Basic Steps of AsyncTask**:

1. **doInBackground()**: Runs in the background thread. This is where the heavy task (like downloading data) is performed.
2. **onPostExecute()**: Runs on the main UI thread once the background task completes. This is where you update the UI with the results.

Code :

**Download Image Example:**

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.os.AsyncTask;

import android.os.Bundle;

import android.widget.Button;

import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

import java.io.IOException;

import java.io.InputStream;

import java.net.URL;

import java.util.concurrent.ExecutionException;

import javax.net.ssl.HttpsURLConnection;

public class DownloadImageExample extends AppCompatActivity {

ImageView imageView;

Button button;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_download\_image\_example);

imageView = findViewById(R.id.imageDown);

button = findViewById(R.id.btnDown);

// Set click listener for the button

button.setOnClickListener(v -> {

// Create and execute the AsyncTask

ImageDownload download = new ImageDownload();

Bitmap bitmap = null;

try {

// Execute AsyncTask and get the result (image)

bitmap = download.execute("https://images.pexels.com/photos/414612/pexels-photo-414612.jpeg").get();

imageView.setImageBitmap(bitmap); // Set the downloaded image in the ImageView

} catch (ExecutionException | InterruptedException e) {

throw new RuntimeException(e);

}

});

}

// AsyncTask to download the image in the background

class ImageDownload extends AsyncTask<String, Void, Bitmap> {

@Override

protected Bitmap doInBackground(String... strings) {

try {

URL url = new URL(strings[0]);

HttpsURLConnection connection = (HttpsURLConnection) url.openConnection();

connection.connect();

InputStream inputStream = connection.getInputStream();

return BitmapFactory.decodeStream(inputStream); // Convert InputStream to Bitmap

} catch (IOException e) {

throw new RuntimeException(e);

}

}

}

}

2. Web Content Download Example

mport android.os.AsyncTask;

import android.os.Bundle;

import android.util.Log;

import android.widget.Button;

import android.widget.EditText;

import androidx.activity.EdgeToEdge;

import androidx.appcompat.app.AppCompatActivity;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.concurrent.ExecutionException;

public class WebContentDownloadMainActivity extends AppCompatActivity {

EditText editText;

Button button;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

EdgeToEdge.enable(this);

setContentView(R.layout.activity\_web\_content\_download\_main);

button = findViewById(R.id.button8);

editText = findViewById(R.id.tbtMulltiLineWeb);

// Button click listener to start the download

button.setOnClickListener(v -> {

try {

// Execute AsyncTask to download web content

String content = new DownloadWebContent().execute("http://www.google.com").get();

Log.d("TAG", "Back in MainActivity");

editText.setText(content); // Set the downloaded content in the EditText

} catch (ExecutionException | InterruptedException e) {

throw new RuntimeException(e);

}

});

}

// AsyncTask to download the web content in the background

class DownloadWebContent extends AsyncTask<String, Void, String>{

@Override

protected String doInBackground(String... strings) {

Log.d("Tag", "doInBackground in Progress");

try {

URL url = new URL(strings[0]);

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

connection.connect();

InputStream stream = connection.getInputStream();

InputStreamReader reader = new InputStreamReader(stream);

int data = reader.read();

StringBuilder webContent = new StringBuilder();

// Read the content byte by byte

while (data != -1){

webContent.append((char) data);

data = reader.read();

}

return webContent.toString(); // Return the web content as a String

} catch (IOException e) {

throw new RuntimeException(e);

}

}

}

}

Xml code:

### ****1. Layout for Download Image Example (****activity\_download\_image\_example.xml****)****

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<!-- Button to start the image download -->

<Button

android:id="@+id/btnDown"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Download Image"

android:layout\_gravity="center\_horizontal"

android:padding="10dp"

android:textSize="18sp"/>

<!-- ImageView to display the downloaded image -->

<ImageView

android:id="@+id/imageDown"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

android:layout\_marginTop="20dp"

android:scaleType="centerCrop"

android:contentDescription="Downloaded Image"

android:layout\_gravity="center\_horizontal"/>

</LinearLayout>

### ****2. Layout for Web Content Download Example (****activity\_web\_content\_download\_main.xml****)****

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<!-- Button to start the web content download -->

<Button

android:id="@+id/button8"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Download Web Content"

android:layout\_gravity="center\_horizontal"

android:padding="10dp"

android:textSize="18sp"/>

<!-- EditText to display the downloaded web content -->

<EditText

android:id="@+id/tbtMulltiLineWeb"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_marginTop="20dp"

android:layout\_weight="1"

android:padding="10dp"

android:scrollbars="vertical"

android:gravity="top"

android:textSize="16sp"

android:background="@android:drawable/edit\_text"

android:hint="Web content will appear here..."/>

</LinearLayout>