FTP File Download

This document shows how to connect to a FTP server and download a file to local storage.

You must have a FTP server running and know the IP address, username, and password to login.

Edit AndroidManifest.xml

Give permission to use the Internet.

Add the **INTERNET** permission to the AndroidManifest.xml

Add the **WRITE_EXTERNAL_STORAGE** permission only if storing file to SDCard.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.mygadgets2.ftp">

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
```

Edit app's build.gradle (Module)

Use the https://commons.apache.org/proper/commons-net/ library.

Add the **commons-net:commons-net:3.9.0** dependency to the app's build.gradle (Module)

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.0'
    implementation 'commons-net:commons-net:3.9.0' // for the FTPClient
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/mTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ProgressBar
        android:id="@+id/mProgressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/mTextView"
        android:indeterminateTint="@color/teal_700"
        android:visibility="invisible"/>

    <Button
        android:id="@+id/mButton"
        android:text="Download"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/mProgressBar" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

Refer to the CodeLabs Lession 7.1

https://developer.android.com/codelabs/android-training-create-asynctask?index=..%2F..%2Fandroid-training#0

```
package com.example.ftpfiledownload;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

import java.io.File;

public class MainActivity extends AppCompatActivity {

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView mTextView = findViewById(R.id.mTextView);
        ProgressBar mProgressBar = findViewById(R.id.mProgressBar);
        Button mButton = findViewById(R.id.mButton);

        mButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String filename = "song.mp3";
                new AsyncFTP(MainActivity.this, filename, mTextView,
mProgressBar).execute();
            }
        });

    }


}
```

AsyncFTP.java

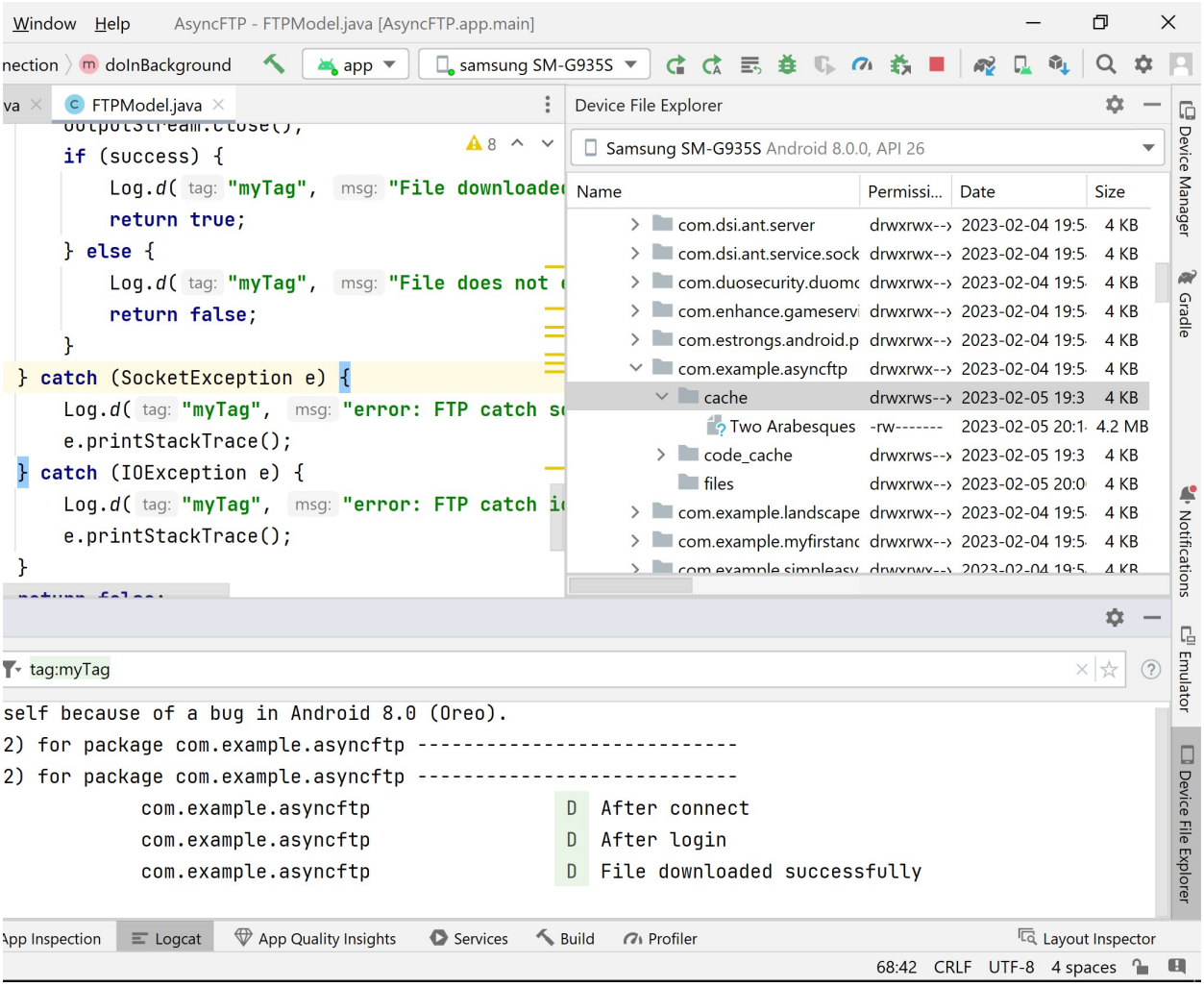
```
package com.example.ftpfiledownload;

import android.os.AsyncTask;
import android.util.Log;
import android.view.View;
import android.widget.ProgressBar;
import android.widget.TextView;

import org.apache.commons.net.ftp.FTP;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPFile;

import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStream;
import java.lang.ref.WeakReference;

// In the following three parameter types <Void, Integer, Boolean>
// the first is the value that is passed into here from the
AsyncFTP(...).execute(firstParam) call from MainActivity.

// the second is the value that is passed from the
publishProgress(secondParam) call to onProgressUpdate(Integer...secondParam)
for updating the progress

// the third is the return value from doInBackground()
// this value is sent to the onPostExecute(thirdParam) and can also be
accessed in AsyncFTP(...).execute().get()

// 1. Params: The data type of the parameters sent to the task upon executing
the doInBackground() override method.
// 2. Progress: The data type of the progress units published using the
```

```
onProgressUpdated() override method.
// 3. Result: The data type of the result delivered by the onPostExecute()
override method.
public class AsyncFTP  extends AsyncTask<Void, Integer, Boolean> {
    private final String host;
    private final int port;
    private final String username;
    private final String password;
    private final File localPath;
    private final String remotePath;
    private final String filename;

    private WeakReference<TextView> mTextView;
    private WeakReference<ProgressBar> mProgressBar;

    public AsyncFTP(MainActivity mainActivity, String filename, TextView
mTextView, ProgressBar mProgressBar) {
        // setup connection to the FTP server
        host = "10.15.15.15";    // hwang.lasierra.edu;
        port = 21;
        username  = "csmajor";
        password = "lsucs";
        // local file path
        localPath = mainActivity.getCacheDir();    // /data/data/<app's
package name>/cache
        // localPath = context.getFilesDir();      // /data/data/<app's
package name>/files
        // remote file path
        remotePath = "public_html";
        this.filename = filename;

        // UI
        this.mTextView = new WeakReference<>(mTextView);
        this.mProgressBar = new WeakReference<>(mProgressBar);
    }

    // this runs BEFORE the doInBackground() on the main UI thread
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        mTextView.get().setText("Going to download "+filename);
        mProgressBar.get().setVisibility(View.VISIBLE);
    }

    // this runs on an asynchronous background thread
    @Override
    protected Boolean doInBackground(Void... voids) {
        FTPClient ftpClient = new FTPClient();
        try {
            // setup local file and output stream
            File localFile = new File(localPath, filename);
            OutputStream outputStream = new BufferedOutputStream(new
FileOutputStream(localFile));
            // connect to remote FTP server
            ftpClient.setConnectTimeout(5000); // set timeout if cannot
connect
            Log.d("myTag", "Going to connect to "+host);
```

```
            ftpClient.connect(host, port);
            Log.d("myTag", "After connect");
            if (ftpClient.login(username, password)) {
                Log.d("myTag", "Login successful");
            } else {
                Log.d("myTag", "Login failed");
            }
            ftpClient.enterLocalPassiveMode();
            ftpClient.setFileType(FTP.ASCII_FILE_TYPE);
            if (ftpClient.changeWorkingDirectory(remotePath)) {   // cd to
remote directory
                Log.d("myTag", "Change directory to "+remotePath+ "
successful");
            } else {
                Log.d("myTag", "Change directory to "+remotePath+ " failed");
            }

            // optional list directory
            Log.d("myTag", "Files in "+remotePath+":");
            FTPFile[] files = ftpClient.listFiles();
            for (FTPFile aFile: files) {
                if (aFile.isFile()) {
                    Log.d("myTag", "  "+aFile.getName());
                } else {
                    Log.d("myTag", "  ["+aFile.getName()+ "]");
                }
            }

            // download file
            Log.d("myTag", "Going to download file "+filename);
            // ftpClient.getSize(filename);
            // publishProgress(i);    // this will call onProgressUpdate()
            boolean success = ftpClient.retrieveFile(filename, outputStream);
            outputStream.close();
            if (success) {
                Log.i("myTag", "File download successful");
                return true;
            } else {
                Log.i("myTag", "File download failed");
                return false;
            }
        } catch (Exception e) {
            Log.i("myTag", "Connection to "+host+" failed");
            e.printStackTrace();
        }
        return false;
    }

    // this runs on the main UI thread
    // this is called by publishProgress(i) in doInBackground()
    @Override
    protected void onProgressUpdate(Integer... values) {
        super.onProgressUpdate(values);
        mProgressBar.get().setProgress(values[0]);
    }

    @Override
```

```
    protected void onCancelled() {
        super.onCancelled();
        mProgressBar.get().setVisibility(View.INVISIBLE);
    }

    // this runs AFTER the doInBackground() on the main UI thread
    @Override
    protected void onPostExecute(Boolean success) {
        super.onPostExecute(success);
        if (success) {
            mTextView.get().setText("File download successful");
        } else {
            mTextView.get().setText("File download failed");
        }
        mProgressBar.get().setVisibility(View.INVISIBLE);
    }
}
```

To see the downloaded file, open the Device File Manager by clicking the tab on the bottom right side.



If you used

```
    localPath = mainActivity.getCacheDir();
```

for the local path then the file is in

/data/data/<app's package name>/cache

If you used

```
    localPath = mainActivity.getFilesDir();
```

for the local path then the file is in

/data/data/<app's package name>/files