

Mobil megoldások

Kocsis Gergely
2019.10.21.

AsyncTask

A felhasználó egyik rémálma olyan felhasználói felülettel találkozni, ami egy gombnyomás hatására gyakorlatilag lefagy és semmit nem lehet tenni, amíg egy költséges (időigényes) folyamat le nem játszódik. Tipikusan ilyen például egy fájl letöltése.

A helyzet kezelésére az Android SDK az AsyncTask absztrakt osztályt hívja segítségül, melynek segítségével háttérben futtathatunk külön szálon folyamatokat, míg a felhasználói felületet tovább használhatjuk.



AsyncTask

A felhasználó egyik rémálma olyan felhasználói felülettel találkozni, ami egy gombnyomás hatására gyakorlatilag lefagy és semmit nem lehet tenni, amíg egy költséges (időigényes) folyamat le nem játszódik. Tipikusan ilyen például egy fájl letöltése.

A helyzet kezelésére az Android SDK az AsyncTask absztrakt osztályt hívja segítségül, melynek segítségével háttérben futtathatunk külön szálon folyamatokat, míg a felhasználói felületet tovább használhatjuk.

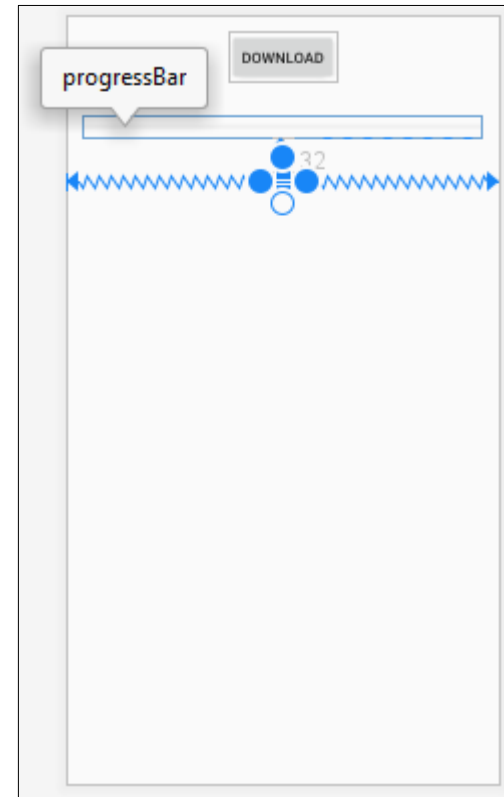
Készítsünk olyan Android alkalmazást, mely egy gomb megnyomására egy ProgressBar-ban mutatja, hogy halad egy fájl letöltése. Egyelőre letöltés helyett használjunk egy adott ideig tartó alvást.



AsyncTask

Hozzuk létre egy új alkalmazást Celebration néven és készítsük el hozzá az alábbi layout-ot.

A gomb megnyomására elindul a “letöltés”, amit százalékos formában a TextView-ban is látni lehet.



AsyncTask

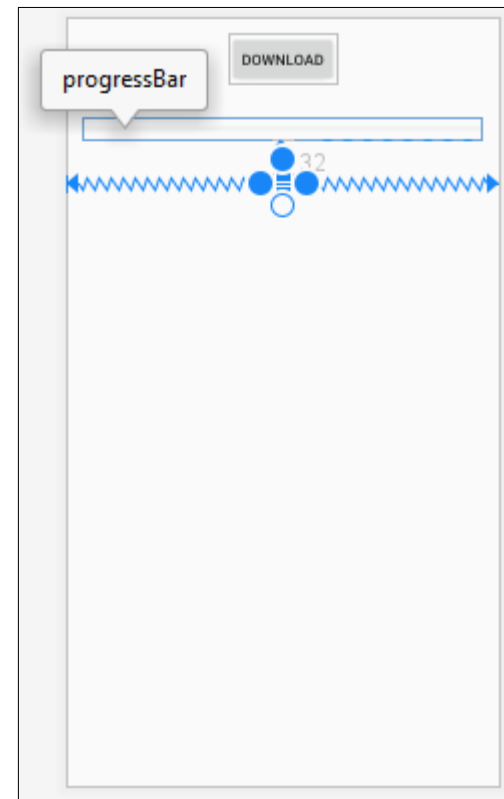
Hozzuk létre egy új alkalmazást Celebration néven és készítsük el hozzá az alábbi layout-ot.

A gomb megnyomására elindul a “letöltés”, amit százalékos formában a TextView-ban is látni lehet.

A gomb megnyomására a `startDownload` metódus hívódik meg.

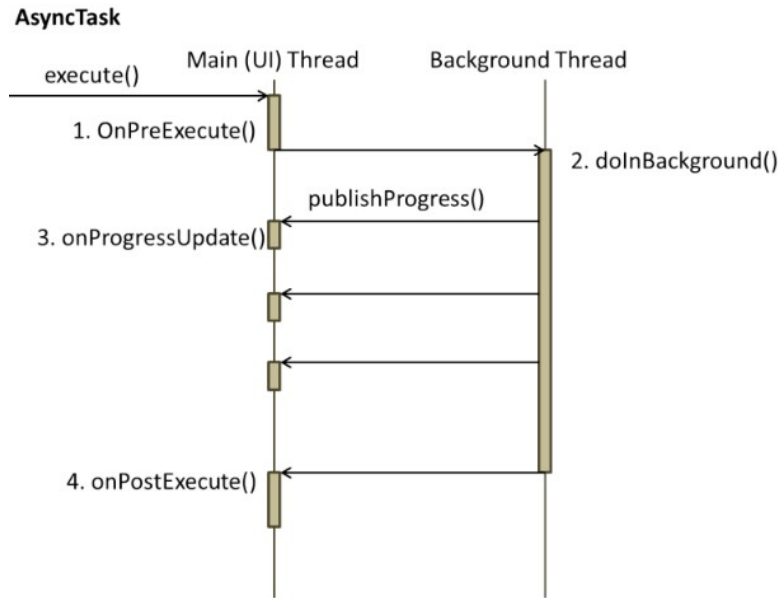
```
public void startDownload(View view) {  
}
```

Mindhárom komponenshez rendeljünk változót a Java oldalon és inicializáljuk is őket az `onCreate`-ben.



AsyncTask

Két kép, ami segít megérteni az AsyncTask működését.



Forrás: <https://corochann.com/async-task-usage-summary-341.html>

```
public class AsyncTaskTestActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        ...

        new MyTask().execute("my string parameter");
    }

    private class MyTask extends AsyncTask<String, Integer, String> {

        @Override
        protected void onPreExecute() {
        }

        @Override
        protected String doInBackground(String... params) {
            String myString = params[0];

            int i = 0;
            publishProgress(i);

            return "some string";
        }

        @Override
        protected void onProgressUpdate(Integer... values) {
        }

        @Override
        protected void onPostExecute(String result) {
            super.onPostExecute(result);
        }
    }
}
```

The diagram includes several annotations: a green circle around "my string parameter" in the onCreate method, a green arrow pointing from it to the parameter in the doInBackground method, a blue circle around "i" in the publishProgress call, a blue arrow pointing from it to the parameter in the onProgressUpdate method, a red circle around "some string" in the return statement, and a red arrow pointing from it to the parameter in the onPostExecute method. A red line also highlights the entire MyTask class definition.



Forrás: <https://stackoverflow.com/questions/9671546/async-task-android-example>

AsyncTask

Hozzuk létre három WeakReference változót, melyekkel elérjük a MainActivity három megfelelő elemét. Az összerendelést a konstruktorban végezzük el.

```
private WeakReference<TextView> weakTextView;  
private WeakReference<ProgressBar> weakProgressBar;  
private WeakReference<Button> weakButton;
```

```
DownloadAsyncTask(TextView tv, ProgressBar pb, Button b) {  
    weakTextView = new WeakReference<>(tv);  
    weakProgressBar = new WeakReference<>(pb);  
    weakButton = new WeakReference<>(b);  
}
```



AsyncTask

Hozzuk létre három WeakReference változót, melyekkel elérjük a MainActivity három megfelelő elemét. Az összerendelést a konstruktorban végezzük el.

```
private WeakReference<TextView> weakTextView;  
private WeakReference<ProgressBar> weakProgressBar;  
private WeakReference<Button> weakButton;
```

```
DownloadAsyncTask(TextView tv, ProgressBar pb, Button b) {  
    weakTextView = new WeakReference<>(tv);  
    weakProgressBar = new WeakReference<>(pb);  
    weakButton = new WeakReference<>(b);  
}
```

A letöltés indulásakor inaktiváljuk a DOWNLOAD gombot:

```
@Override  
protected void onPreExecute() {  
    super.onPreExecute();  
    weakButton.get().setEnabled(false);  
}
```



AsyncTask

A doInBackground metódusban aludjunk 1 másodpercig. 10 milisecundumonként küldjünk visszajelzést.

```
@Override
protected String doInBackground(Void... voids) {
    try {
        for (int i = 0; i < 100; i++) {
            Thread.sleep(10);
            publishProgress(i);
        }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    return "Download complete";
}
```



AsyncTask

Az előrehaladást jelezzük az `onProgressUpdate` metódusban mind a `ProgressBar`-n, mint a `TextView`-n.

Az `AsyncTask` zárultakor írjuk ki, hogy sikeres letöltés és állítsuk újra megnyomhatóra a gombot.

```
@Override
protected void onProgressUpdate(Integer... values) {
    super.onProgressUpdate(values);
    weakTextView.get().setText("" + values[0] + "%");
    weakProgressBar.get().setProgress(values[0]);
}
```

```
@Override
protected void onPostExecute(String result) {
    weakTextView.get().setText(result);
    weakButton.get().setEnabled(true);
}
```



AsyncTask

Indítsuk el az új AsyncTaskot a gomb megnyomásakor.

```
//MainActivity  
public void startDownload(View view) {  
    new DownloadAsyncTask(progressTextView, progressBar,  
downloadButton).execute();  
}
```

Ezzel kész vagyunk. Futtassuk az alkalmazást!



AsyncTask

Fejlesztjük tovább az alkalmazást úgy, hogy tényleg letöltsön egy mp3 fájlt. A letöltés után adjon lehetőséget a lejátszásra.

Adjunk a felülethez egy PLAY gombot, melyet egyelőre inaktívrá állítunk. Megnyomásakor a playSong metódus hívódik meg. Hozzunk létre hozzá változót a Java oldalon és inicializáljuk.

```
private Button playButton;  
...  
playButton = findViewById(R.id.playButton);  
...  
public void playSong(View view) {  
}
```



AsyncTask

Alakítsuk át a doInBackground törzst az alábbiak szerint:

```
int count;
try {
    URL url = new URL("https://irh.inf.unideb.hu/~kocsisg/song.mp3");
    URLConnection connection = url.openConnection();
    connection.connect();
    int lenghtOfFile = connection.getContentLength();
    // input stream to read file - with 8k buffer
    InputStream input = new BufferedInputStream(url.openStream(), 10*1024);
    // Output stream to write file
    OutputStream output = new FileOutputStream("/data/user/0/
        hu.unideb.inf.mobil.celebration/files/song.mp3");
    byte data[] = new byte[1024];
    long total = 0;
    while ((count = input.read(data)) != -1) {
        total += count;
        publishProgress((int) ((total * 100) / lenghtOfFile));
        output.write(data, 0, count);
    }
    output.flush();
    output.close();
    input.close();
} catch (Exception e) {
    Log.e("Error: ", e.getMessage());
}
return "Download complete";
```



AsyncTask

Szükség lesz az AndroidManifest átalakítására is:

```
<!-- Permission: Allow application to connect to Internet -->  
<uses-permission android:name="android.permission.INTERNET" />  
<!-- Permission: Allow application to write to SDCard -->  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
>  
  
<application  
    android:usesCleartextTraffic="true"
```

Adjuk az alábbi sort az onPostExecute-hoz. Ehhez szükségünk lesz az új gomb felvezetéére a korábbiak szerint...

```
weakPlayButton.get().setEnabled(true);
```



AsyncTask

A zene lejátszása:

```
private MediaPlayer mPlayer;

public void playSong(View view) {
    mPlayer = new MediaPlayer();
    try {
        mPlayer.setDataSource(
            "/data/user/0/hu.unideb.inf.mobil.celebration/files/song.mp3");
        mPlayer.prepare();
        // Start playing the Music file
        mPlayer.start();
    } catch (Exception e) {
        Log.e("IOEX", e.getMessage());
    }
}
```

