

Formation Android

Épisode II



Xavier MARIN



@XavMarin

<https://github.com/Giwi>

CTO chez @qaobee



#OneDayWithoutInternet

Objectifs de l'atelier

Maîtriser et être autonome sur les éléments suivants:

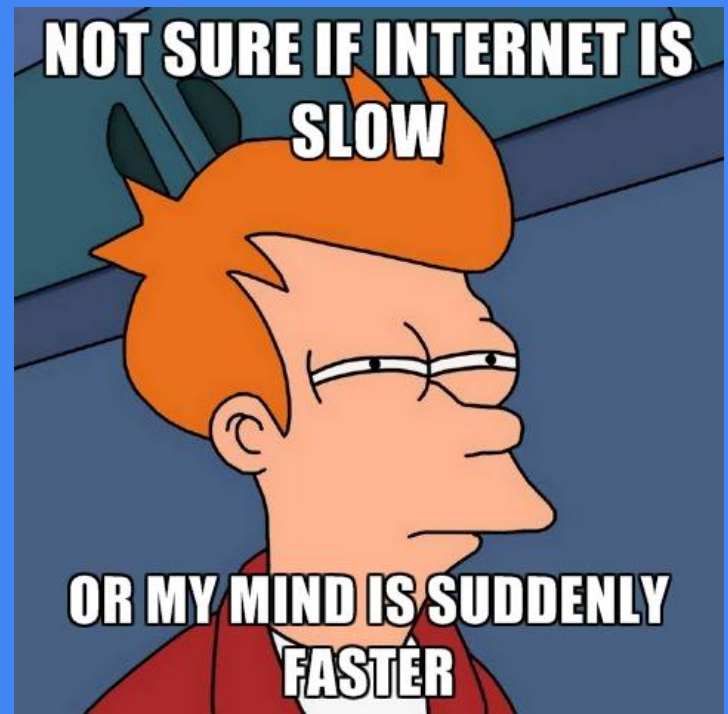
- AsyncTask
- Requête HTTP
- Permissions



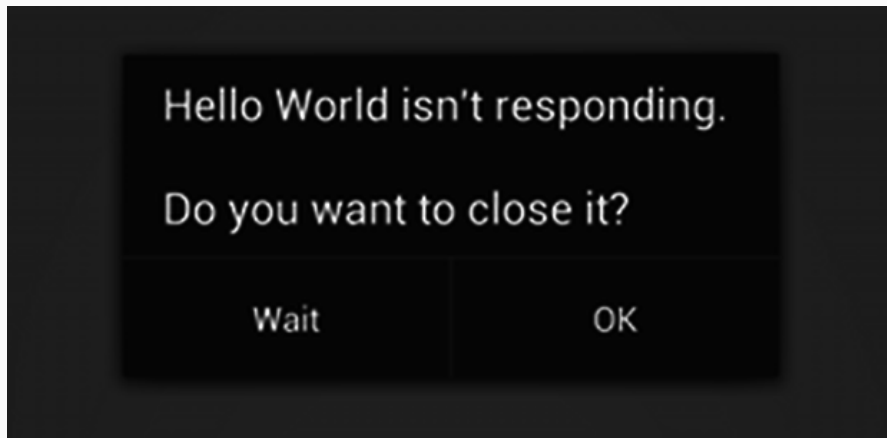
Attention

Utilisation de Http

org.apache.http.*



Vous risquez de finir comme ça



Comment y remédier?

- Les services
- Les threads
- Les AsyncTasks

Les services

- Composant qui tourne en tâche de fond
- Pas d'UI
- Cycle de vie similaire à l'activity
- Contrôlable depuis des activités, d'autres services...

C'est un composant essentiel sur Android

- Complexe
- Interaction avec l'UI d'une activité non triviale

D'autres solutions?

- Les threads
 - Identique au Java
 - Standard
 - MaJ de l'UI par message
- Les AsyncTask
 - Surcouche de gestion du multi threading sur Android
 - Simplicité & efficacité

Nous allons utiliser les AsyncTask

AsyncTask

Trois paramètres <X, Y, Z>:

- Le type de paramètre envoyé à l'exécution
- Le type d'objet pour notifier la progression
- Le type de résultat de l'exécution

Des méthodes à surcharger:

- `doInBackground(Object... params)`
- `onPostExecute(Result)`

```
private class DownloadFilesTask extends AsyncTask<URL, Integer, Long> {  
    protected Long doInBackground(URL... urls) {  
        int count = urls.length;  
        long totalSize = 0;  
        for (int i = 0; i < count; i++) {  
            totalSize += Downloader.downloadFile(urls[i]);  
            publishProgress((int) ((i / (float) count) * 100));  
            // Escape early if cancel() is called  
            if (isCancelled()) break;  
        }  
        return totalSize;  
    }  
  
    protected void onProgressUpdate(Integer... progress) {  
        setProgressPercent(progress[0]);  
    }  
  
    protected void onPostExecute(Long result) {  
        showDialog("Downloaded " + result + " bytes");  
    }  
}
```

Les permissions

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="cesi.com.helloworld">

    <application android:allowBackup="true" android:label="HelloWorld"
        android:icon="@mipmap/ic_launcher" android:theme="@style/AppTheme">
        <activity
            android:name=".MyActivity"
            android:label="HelloWorld"
            android:theme="@style/AppTheme"
            android:windowSoftInputMode="stateHidden"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

Bon, on code?



Atelier 1

Objectifs:

- 1 activités
- 1 vues
- 2 appels Http
 - POST Ping
 - GET hello

Une dernière petite aide?

```
@Override
protected String doInBackground(String... params) {
    if(!NetworkHelper.isInternetAvailable(context)){
        return "Internet not available";
    }
    try {
        //then create an http client.
        HttpClient client = new DefaultHttpClient();
        HttpGet request = new HttpGet();
        request.setURI(URI.create(context.getString(R.string.url_hello)+"?name="+params[0]));

        // do request.
        HttpResponse httpResponse = client.execute(request);
        String response = null;
        //Store response
        if (httpResponse.getEntity() != null) {
            response = EntityUtils.toString(httpResponse.getEntity());
        }

        Log.d("HelloWorld", "received for url: " + request.getURI() + " return code: " + httpResponse
            .getStatusLine()
            .getStatusCode());
        return response;
    } catch (Exception e){
        Log.d("HelloWorld", "Error occurred in your AsyncTask : ", e);
        return "an error occurred";
    }
}
```

A votre tour !

<https://cesi-giwisoft.rhcloud.com/apidoc/>

