

Methods of the AsyncTask class

We have studied about multi-threading in the previous topics. Multi-threading is a concept that allows us to use two threads to perform different tasks. We use the worker thread to perform background operations that require more memory, and the main thread that handles all UI related operations. This is done so that the screen does not freeze for the user, so as to ensure a good user experience.

All Database operations, like read, write, are expensive, not in terms of money, but in terms of the memory usage. Hence, we will use the worker thread to perform these operations.

We used Volley while sending GET and POST network requests, but this time we are already using the Room library for dealing with database operations. Hence for database operations, we will use the AsyncTask class of the SDK. AsyncTask stands for asynchronous task. This class allows us to perform background operations on a worker thread and then publish the results on the main/UI thread.

When an asynchronous task is executed, it goes through 4 phases:

onPreExecute()- This method is invoked on the UI thread before the task is executed. This step is mostly used to set up the task, such as showing the progress bar as we did while using Volley.

doInBackground(Params...)- This method is invoked on the background thread immediately after the onPreExecute() finishes executing. This step is used to perform background operations that can take a long time. The parameters of the asynchronous task are passed in this step. The result of the computation must be returned by this step and will be passed back to the last step, that is onPostExecute(Result). This step can also use publishProgress(Progress...) to publish one or more units of progress. These values are shown on the UI thread, in the onProgressUpdate(Progress...) step.

onProgressUpdate(Progress...)- This method is invoked on the UI thread after a call to publishProgress(Progress...) has been made. The timing of the execution is undefined. This method is used to display any form of progress in the user interface while the background computation is still executing. For example, it can be used to display the progress of a file download.

onPostExecute(Result)- This method is invoked on the UI thread after the background computation finishes. The result of the background computation is passed in this step as a parameter.

The 4 steps mentioned above are the 4 different methods of the AsyncTask class. It is not mandatory to implement them all for every task. Only the doInBackground() method is mandatory to use.

The params, progress, and result, you see inside the parentheses are used to perform any asynchronous operation.

Params- These are the parameters sent to the task upon execution.

Progress- These are the progress units published during the background computation.

Result- These are the results of the background computation.