

Study Independent

Android Development (Day 6)

Ringkasan

Background Process dan Networking

- a. Background Thread
- b. Reactive Programming
- c. Kotlin Coroutine
- d. Rest API

Pembahasan

Background Thread

Pemrosesan latar belakang di Android mengacu pada eksekusi tugas secara asinkron di thread yang berbeda dari Thread utama yang juga dikenal sebagai Thread UI.

- a. Kenapa pemrosesan latar belakang diperlukan
 - Untuk menghindari aplikasi membeku/tidak merespon UI oleh peristiwa I/O.
 - Beberapa operasi tidak diizinkan untuk dijalankan di Thread utama, seperti salah satunya request jaringan HTTP.
 - Untuk meningkatkan kinerja.
- b. Beberapa contoh thread
 - Thread & Handler
 - IntentService
 - AsyncTask
 - Loader & Cursor Loader

Reactive Programming

Pemrograman Reaktif pada dasarnya adalah pemrograman asinkron. Keuntungan dari pendekatan asinkron adalah, karena setiap tugas berjalan pada threadnya sendiri, semua tugas dapat dimulai secara bersamaan dan jumlah waktu yang dibutuhkan untuk menyelesaikan semua tugas setara dengan tugas yang lebih panjang dalam daftar. Data yang dikirim berupa stream yang dikirimkan secara berurutan.

- a. Istilah dalam reactive programming
 - Observable
 - Observer
 - Subscription
 - Operator / Transformation
 - Schedulers
- b. Daftar schedulers
 - Schedulers.io()
 - AndroidSchedulers.mainThread()
 - Schedulers.newThread()
 - Schedulers.computation()
 - Schedulers.single()
 - Schedulers.immediate()
 - Schedulers.trampoline()
 - Schedulers.from()

Kotlin Coroutine

Coroutine adalah pola desain serentak yang dapat digunakan di pemrograman Android untuk menyederhanakan kode yang dieksekusi secara asinkron. Coroutine ditambahkan pada Kotlin dalam versi 1.3 dan didasarkan pada konsep yang telah ditetapkan dari bahasa lain. Di Android, coroutine berguna untuk mengelola tugas yang berjalan lama dan mungkin memblokir thread utama yang menyebabkan aplikasi tidak responsif. Berikut ini beberapa solusi atau fitur penting yang ditawarkan oleh coroutine:

- a. Ringan
- b. Lebih sedikit kebocoran memori
- c. Dukungan pembatalan bawaan (Cancellation dan Timeout)
- d. Integrasi Jetpack

Rest API

REST API merupakan salah satu dari desain arsitektur yang terdapat di dalam API itu sendiri. Dan cara kerja dari RESTful API yaitu REST client akan Melakukan akses pada data/resource pada REST server dimana masing-masing resource atau data/resource tersebut akan dibedakan oleh sebuah global ID atau URI (Universal Resource Identifiers).

- a. Berikut adalah beberapa sifat Rest API:
 - Client-Server
 - Stateless
 - Cacheable
 - Layered
- b. Library Rest API yang biasa digunakan di Android:
 - Retrofit
 - OkHttp
 - Volley
 - Fuel

Referensi

Berikut merupakan sumber referensi dari materi yang dibuat:

- a. https://medium.com/@julian_falcionelli/background-processing-in-android-575fd4ecf769
- b. <https://www.androidhive.info/RxJava/android-getting-started-with-reactive-programming/>
- c. <https://medium.com/@rohmanhakim/mengulik-reactive-programming-di-android-bagian-1-916b11c5597>
- d. <https://developer.android.com/kotlin/coroutines?hl=id>
- e. <https://www.dicoding.com/blog/apa-itu-rest-web-service/>
- f. <https://www.codepolitan.com/rest-api-client-sederhana-dengan-retrofit-pada-android-studio-58986d62c46ae/>

Penilaian

Penilaian akan dilakukan dengan memberikan tugas pada minggu terakhir (Jumat).