

Networking

Ikrima Amanda



API - Parsing JSON

Postman

API

- API (Application Programming Interface) adalah sekumpulan aturan, protokol, dan alat pemrograman yang memungkinkan berbagai aplikasi atau komponen perangkat lunak berinteraksi dan saling berkomunikasi satu sama lain.
- Memiliki beberapa jenis
- Web API => membantu aplikasi untuk berkomunikasi melalui jaringan menggunakan protokol HTTP atau HTTPS. API Web dapat menyediakan data, fungsionalitas, atau integrasi dengan aplikasi atau layanan lain melalui permintaan dan respons HTTP.
- Jenis Web API:
 - REST API (Representational State Transfer API) => digunakan untuk mengakses dan memanipulasi sumber daya (resources) melalui permintaan HTTP seperti GET, POST, PUT, DELETE, dll.

Poin Penting REST API

- Prinsip REST: REST API mengikuti prinsip-prinsip arsitektur REST, yang mencakup:
 - Penggunaan metode HTTP yang sesuai: REST API menggunakan metode HTTP seperti GET (mendapatkan data), POST (membuat data baru), PUT (memperbarui data), DELETE (menghapus data), dan lainnya.
 - Penggunaan URI (Uniform Resource Identifier): Setiap sumber daya diidentifikasi oleh URI yang unik, seperti /users untuk koleksi pengguna atau /users/{id} untuk pengguna dengan ID tertentu.
 - Representasi sumber daya: Data sumber daya diwakili dalam format umum seperti JSON atau XML. Respons API REST seringkali mengembalikan data dalam format JSON.
 - Tidak ada status sesi di sisi server: Server tidak menyimpan status sesi pengguna di antara permintaan. Setiap permintaan dianggap mandiri dan berisi semua informasi yang diperlukan.
- Endpoint: REST API menyediakan serangkaian endpoint yang mewakili sumber daya yang dapat diakses melalui API. Setiap endpoint memiliki URI yang terkait dengan operasi yang spesifik, seperti /users untuk mendapatkan daftar pengguna atau /users/{id} untuk mendapatkan pengguna dengan ID tertentu.

Poin Penting REST API (2)

- Metode HTTP: REST API menggunakan metode HTTP seperti berikut:
 - GET: Mengambil data dari sumber daya yang ditentukan.
 - POST: Membuat sumber daya baru dengan data yang diberikan.
 - PUT: Memperbarui sumber daya yang ada dengan data yang diberikan.
 - DELETE: Menghapus sumber daya yang ditentukan.
- Format Respons: Respons dari REST API seringkali menggunakan format data yang umum seperti JSON (JavaScript Object Notation). JSON memungkinkan representasi data yang ringkas, mudah dibaca oleh mesin, dan mudah diuraikan oleh klien.
- Stateless: yang berarti setiap permintaan yang dikirim ke API berisi semua informasi yang diperlukan untuk memahami dan memproses permintaan tersebut. Server tidak menyimpan status sesi pengguna di antara permintaan.

JSON (JavaScript Object Notation)

- sebuah format standar yang digunakan untuk pertukaran data antara sistem yang berbeda. JSON menggunakan sintaksis yang mudah dipahami oleh manusia dan juga dapat dengan mudah diuraikan oleh mesin.
- Poin penting:
 - Struktur: JSON menggunakan struktur data dalam bentuk objek dan array. Objek JSON didefinisikan dalam pasangan "key-value" yang diapit oleh kurung kurawal ({}). Array JSON didefinisikan dalam tanda kurung siku ([]).
 - Tipe Data: JSON mendukung beberapa tipe data dasar, termasuk string, integer, boolean, null, objek, dan array. Contohnya, sebuah objek JSON bisa terdiri dari pasangan "key-value" di mana key adalah string dan nilainya bisa berupa string, integer, boolean, objek, atau array.
 - Representasi Nilai: Dalam JSON, nilai-nilai diwakili sesuai dengan tipe datanya. String diapit oleh tanda kutip ganda (misalnya "nama": "John"), integer ditulis tanpa tanda kutip (misalnya "usia": 25), boolean ditulis sebagai "true" atau "false", null ditulis sebagai "null", objek dan array ditulis dalam bentuk yang sesuai dengan kurung kurawal ({}) dan tanda kurung siku ([]).
 - Nested: JSON memungkinkan struktur data nested, yang berarti objek JSON dapat berisi objek atau array lain di dalamnya. Hal ini memungkinkan representasi data yang kompleks dan hirarkis.
 - Pemrosesan dan Implementasi: JSON dapat diuraikan dan dibangun dengan mudah menggunakan berbagai bahasa pemrograman. Hampir semua bahasa pemrograman modern memiliki dukungan untuk memarsing dan menghasilkan data dalam format JSON.

Parsing JSON

- Proses mengurai (mengubah) data dalam format JSON menjadi struktur data yang dapat diakses dan dimanipulasi oleh program atau aplikasi.
- Proses parsing JSON => mengambil data dalam format JSON dan mengonversinya menjadi objek atau struktur data yang sesuai dalam bahasa pemrograman yang digunakan.

Postman

Sebuah platform kolaboratif yang digunakan oleh para pengembang software untuk menguji, mengelola, dan mendokumentasikan API (Application Programming Interface).

Poin Penting Postman

- **Pengujian API:** pengembang dapat dengan mudah mengatur header permintaan, parameter, body, dan metode HTTP yang sesuai. Ini memungkinkan pengembang untuk menguji fungsionalitas dan mengevaluasi respons dari API secara interaktif.
- **Pengaturan Environments:** memungkinkan pengembang untuk mengatur dan mengelola environment yang berbeda untuk pengujian dan pengembangan. Environments memungkinkan pengembang untuk menyimpan dan mengubah nilai variabel seperti URL, kunci API, atau token otentikasi, sehingga memudahkan pengujian pada berbagai lingkungan, seperti pengembangan, pengujian, dan produksi.
- **Dokumentasi API:** Postman menyediakan fitur untuk membuat dan mendokumentasikan dokumentasi API. Pengembang dapat dengan mudah menghasilkan dokumentasi yang terstruktur dengan menjelaskan permintaan dan respons API dan menambahkan contoh permintaan dan respons yang dihasilkan.

Retrofit

Deskripsi

- Library HTTP client yang populer untuk pengembangan aplikasi Android menggunakan bahasa pemrograman Kotlin.
- Menyederhanakan proses komunikasi jaringan antara aplikasi Android dan server yang menyediakan API (Application Programming Interface).
- Developer dapat dengan mudah mengintegrasikan dan mengonsumsi API dengan menulis kode yang bersih dan terstruktur.

Contoh anotasi pada Retrofit

- **@GET** => mengirim permintaan GET ke endpoint API.
- **@POST** => mengirim permintaan POST ke endpoint API.
- **@PUT** => mengirim permintaan PUT ke endpoint API.
- **@DELETE** => mengirim permintaan DELETE ke endpoint API.
- **@Path** => mengganti nilai pada bagian dari URL yang merupakan parameter dinamis dalam permintaan HTTP.
- **@Query** => menambahkan parameter query pada permintaan HTTP.
- **@Body** => mengirim objek dalam bentuk body permintaan HTTP (biasanya dalam format JSON).
- **@Header** => menambahkan header kustom pada permintaan HTTP.
- **@FormUrlEncoded** dan **@Field** => mengirim data dalam format x-www-form-urlencoded dalam permintaan POST.

Logging Interceptor by OkHttp

- Komponen yang disediakan oleh library OkHttp untuk melacak, mencatat, dan menganalisis log dari permintaan HTTP dan respons yang dikirim dan diterima oleh aplikasi.
- Berfungsi sebagai perantara antara aplikasi dan koneksi jaringan, dan dapat digunakan untuk tujuan debugging, pemantauan, dan analisis.

Langkah penggunaan Retrofit

- Siapkan API yang dibutuhkan dan cek di Postman
- Tambahkan dependensi retrofit dan logging interceptor pada file build.gradle(Module) dan synchronize it

```
// retrofit
implementation 'com.squareup.retrofit2:retrofit:2.9.0'
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'

// logging interceptor by OkHttp
implementation("com.squareup.okhttp3:logging-interceptor:4.11.0")
```

- Buat data class sesuai hasil response GitHub API
- Buat interface sesuai URL GitHub API
- Buat file untuk mengkonfigurasi Api Service kita
- Persiapkan tampilan (recyclerview dan adapternya)
- Inisialisasikan client pada activity/fragment dan handle ketika response success dan failure
- Note : saat menggunakan logging interceptor pastikan hanya tampil ketika mode debug

Referensi

- <https://aws.amazon.com/id/what-is/api/>
- <https://www.json.org/json-en.html>
- <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- https://www.w3schools.com/whatis/whatis_json.asp
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/parse?retiredLocale=id
- <https://www.postman.com/product/what-is-postman/>
- <https://square.github.io/retrofit/>
- <https://github.com/square/okhttp/tree/master/okhttp-logging-interceptor>

Tugas

Aplikasi Pengguna GitHub

- Terdapat splash screen aplikasi
- Terdapat tampilan list pengguna GitHub (Photo Profile dan UserName)
- Berdasarkan API =>
<https://docs.github.com/en/rest/users/users?apiVersion=2022-11-28#list-users>
- Item pengguna dapat diklik dan berpindah ke halaman baru dengan detail data dari pengguna yang diklik/dipilih sebelumnya dan dibawah detail data pengguna terdapat 2 Tab (Followers dan Following dari pengguna tersebut)
- Dibuat menggunakan Tab Layout dan View Pager

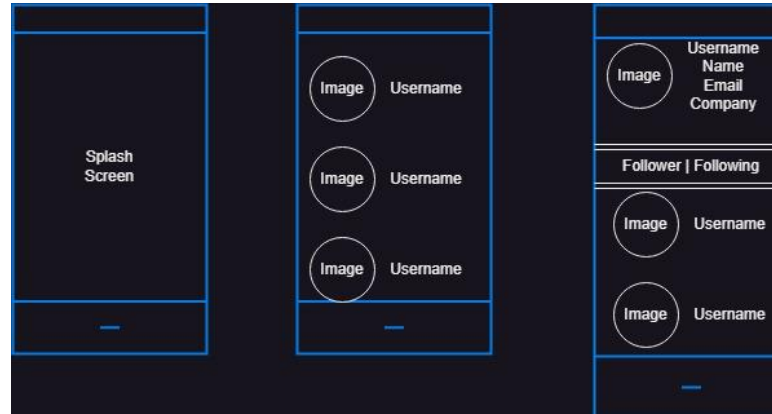
Tugas (2)

Aplikasi Pengguna GitHub

- Detail Data pengguna mencakup => Photo Profile, UserName, Name, Email, Company
- Tiap tab menampilkan follower dan following yang ditampilkan menggunakan recyclerview (data photo profile dan username)
- Total ada 3 tampilan => splash screen, list pengguna, dan detail pengguna (diikuti dengan tab follower dan following)
- Dikumpulkan dalam bentuk **Link Google Drive (Berisi link repo GitHub dan APK)**
- Deadline pengumpulan tugas => **10 Juli 2023 12.00 WIB**
- **Link pengumpulan tugas** => <https://forms.gle/nF35wCviJ6YH43Hu9>

Tugas (3)

- Perhatikan bahwa terdapat rate limit dalam mengakses data dari GitHub API
- Hanya diberi batasan 60 request per jam
- Untuk menambah akses, baca =>
<https://www.dicoding.com/blog/apa-itu-rate-limit-pada-github-api/>
- Contoh tampilan:



Stuck?

talk with ur best friend...
chatgpt, google, stack over flow, medium

The Most Important Thing...

<https://www.instagram.com/reel/CtoupCfADAv/?igshid=Y2lzZGU1MTFhOQ%3D%3D>

“Practice doesn’t make perfect. Perfect practice makes perfect”

- Vince Lombardi

Terima kasih!

telegram : @ikrimaamandaa

LinkedIn :

<https://www.linkedin.com/in/ikrimaamanda/>

