

## Modul 4

# Getting Data from the Web (1)

Using the HTTP library to retrieve data from a web service

### Module Overview

Memprogram cara aplikasi dalam mengambil data dari Website menggunakan library HTTP dan bekerja dengan JSON

### Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

- Mengetahui bagaimana bekerja dengan data yang didapatkan dalam website
- Bekerja dengan data format JSON

Ada banyak aplikasi mobile yang datanya sepenuhnya diperoleh dari data eksternal, seperti aplikasi ramalan cuaca, aplikasi mendengarkan musik, aplikasi membaca buku, aplikasi berita, dan lainnya. Kebanyakan aplikasi tersebut memiliki kesamaan, yaitu aplikasi tersebut mengandalkan data yang diambil dari sumber eksternal, yang bersumber dari *web service* atau *web API*.

Proses pengambilan data dari *web service* atau *web API* dapat dilakukan melalui perangkat mobile. Aplikasi biasanya akan melakukan permintaan data ke dalam *web service* atau *web API*, dan selama permintaan tersebut sah, maka *web service* atau *web API* akan mengembalikan *service responds*, dengan mengirimkan data kembali ke aplikasi yang melakukan permintaan data tersebut. Sebenarnya pola ini (klien/server) bukanlah hal baru, tetapi sangat umum saat merancang aplikasi.

Secara umum, *Web service* dapat menyediakan data dengan 2 format teks, JSON atau XML. Akan tetapi, penggunaan JSON lebih disarankan karena JSON sedikit lebih kompak sehingga lebih sering anda temukan saat menggunakan *web service*.

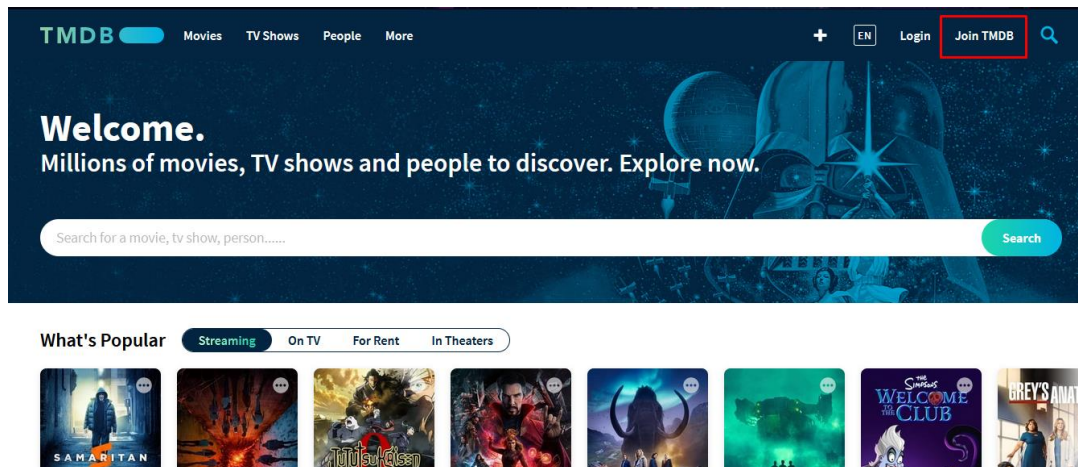
Web Services Data	
JSON	XML
<pre>{   "title" : "The Matrix",   "year" : "1999",   "genre" : "Science Fiction",   "actors": ["Keanu Reeves", "Laurence Fishburne", "Carrie Ann Moss"] }</pre>	<pre>&lt;movie&gt;   &lt;title&gt; The Matrix &lt;/title&gt;   &lt;year&gt; 1999 &lt;/year&gt;   &lt;genre&gt; Science Fiction&lt;/genre&gt;   &lt;actors&gt;     &lt;actor&gt;Keanu Reeves&lt;/actor&gt;     &lt;actor&gt;Laurence Fishburne&lt;/actor&gt;     &lt;actor&gt; Carrie Ann Moss &lt;/actor&gt;   &lt;/actors&gt; &lt;/movie&gt;</pre>

Gambar 1. Contoh data format JSON dan XML.

Kami tidak akan masuk ke detail format JSON dan XML. Yang perlu anda ketahui adalah bahwa kedua format tersebut sangat dimungkinkan untuk anda gunakan dalam mengekspresikan data yang kompleks di dalam Flutter. Pada Pertemuan ini, kita akan menggunakan JSON sebagai hasil data dari *web service*.

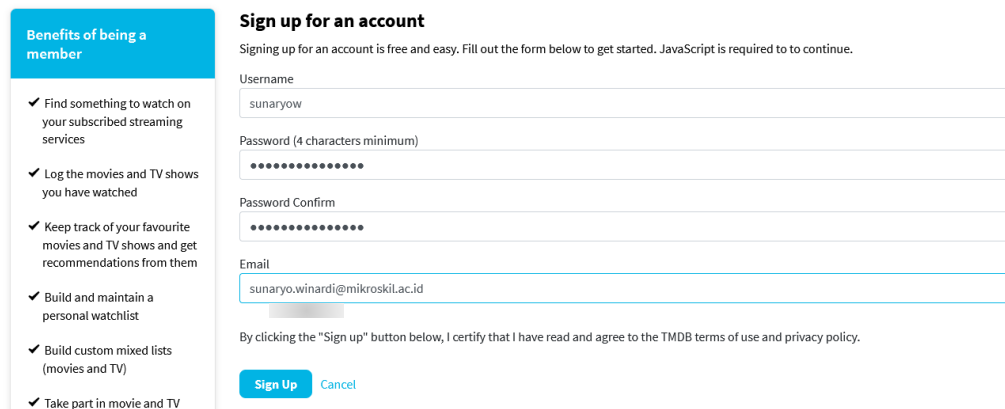
## Movie Database API Web Service

Pada contoh ini, Kita akan mencoba mengambil data dari salah satu penyedia data JSON untuk film dan serial tv dengan beberapa Bahasa yang tersedia, yaitu <https://www.themoviedb.org/>. Untuk melakukan koneksi ke dalam aplikasi, anda diwajibkan memiliki akun di dalam <https://www.themoviedb.org/> tanpa diperlukan biaya.



Gambar 2. Halaman themoviedb

Lakukan pendaftaran pada web service tersebut, dan pastikan anda menggunakan email yang aktif.



**Benefits of being a member**

- ✓ Find something to watch on your subscribed streaming services
- ✓ Log the movies and TV shows you have watched
- ✓ Keep track of your favourite movies and TV shows and get recommendations from them
- ✓ Build and maintain a personal watchlist
- ✓ Build custom mixed lists (movies and TV)
- ✓ Take part in movie and TV

**Sign up for an account**

Signing up for an account is free and easy. Fill out the form below to get started. JavaScript is required to continue.

Username  
sunaryow

Password (4 characters minimum)  
.....

Password Confirm  
.....

Email  
sunaryo.winardi@mikroskil.ac.id

By clicking the "Sign up" button below, I certify that I have read and agree to the TMDB terms of use and privacy policy.

**Sign Up** Cancel

Gambar 3. Halaman Registrasi

Lakukan aktivasi melalui email, dan masuklah ke dalam website, sehingga anda terdaftar sebagai pengguna *web service*

### Login to your account

In order to use the editing and rating capabilities of TMDb, as well as get personal recommendations you will need to login to your account. If you do not have an account, registering for an account is free and simple. [Click here](#) to get started.

If you signed up but didn't get your verification email, [click here](#) to have it resent.

**Account verification required**

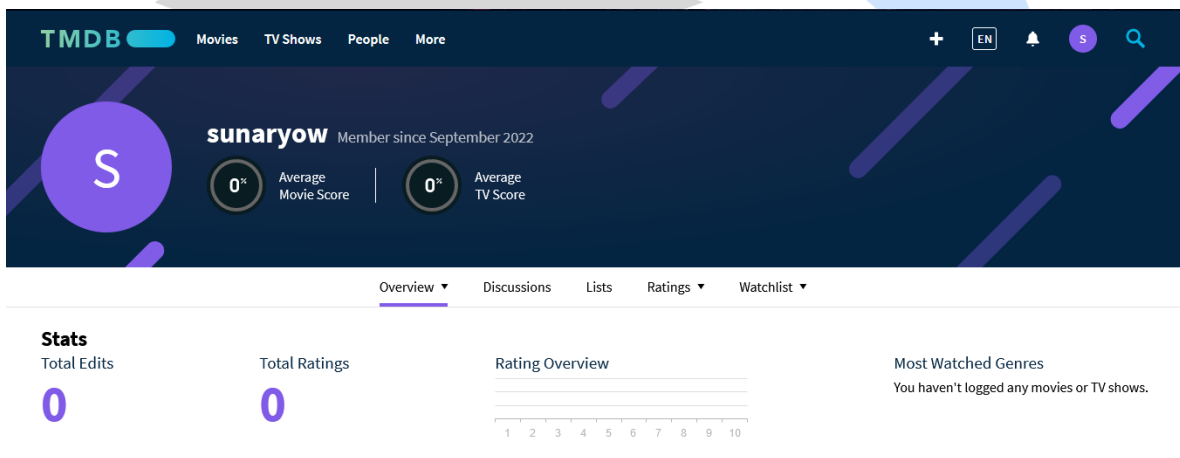
- Your email address hasn't been verified. Please click the verification link in the email that was sent to the address you signed up with. (Don't forget to check your spam folder.) You may [request the email be resent](#) if you are unable to locate your activation email.

Username

Password

Login
[Reset password](#)

Gambar 4. Halaman Login



Gambar 5. Halaman Utama pengguna

Setelah mendaftar, anda dapat membuat sebuah proyek Flutter baru, dengan nama `the_movie`. Untuk dapat bekerja dengan web service, pastikan anda menambahkan dependence http berikut:

```
http: ^0.13.5
```

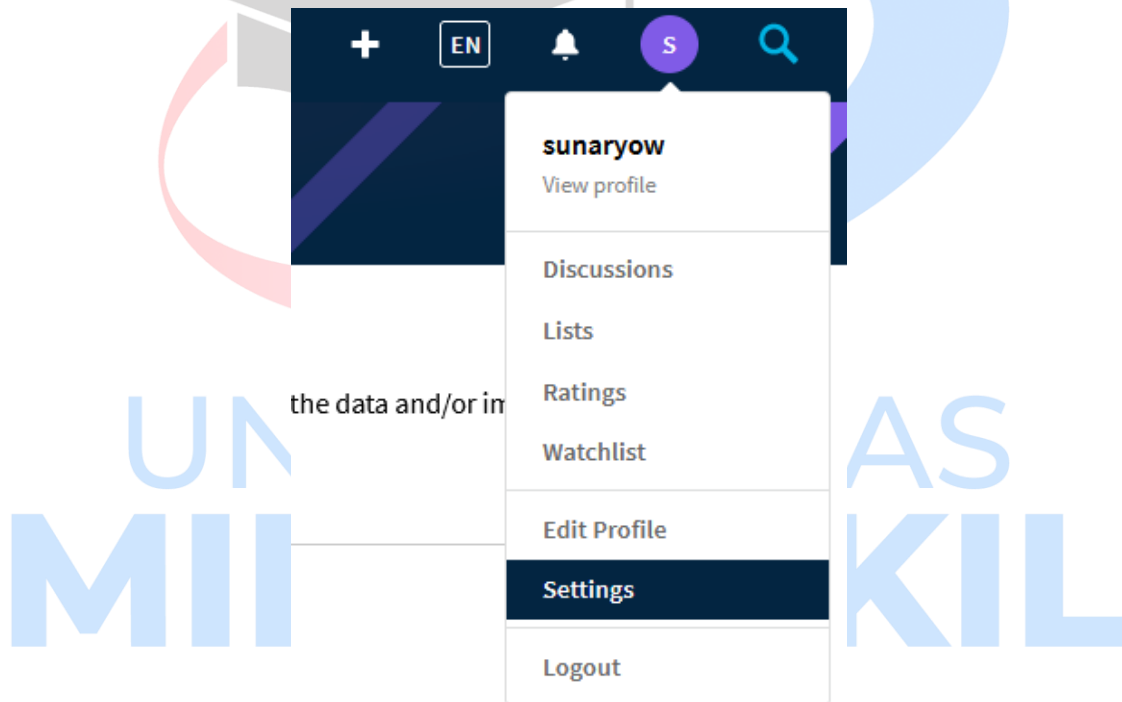
http merupakan Library yang memanfaatkan Future untuk membuat permintaan data di HTTP. Paket ini berisi serangkaian fungsi yang memudahkan penggunaan dalam mengolah sumber daya HTTP. Library ini juga mendukung untuk mobile, desktop, dan browser.

Untuk membantu anda dalam melakukan proses pengambilan data, anda dapat membuat sebuah kelas helper. Untuk itu buatlah sebuah file baru dengan nama `Http.Helper.dart` dan buat kelas baru seperti berikut ini:

```
class HttpHelper { }
```

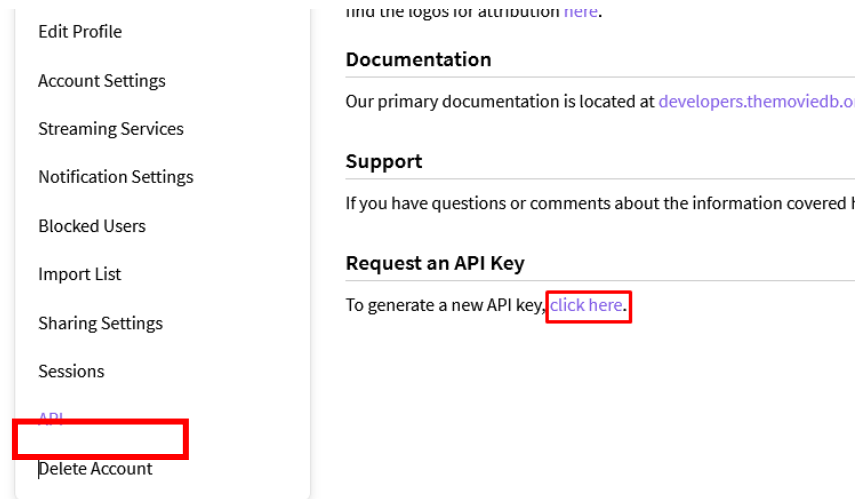
Untuk melakukan permintaan data ke *web service*, setiap proses pengambilan data memerlukan API KEY unik untuk aplikasi. API KEY adalah sebuah kode unik yang berfungsi untuk memberikan akses login dan menghubungkan kode dari developer satu sistem lain. Untuk mendapatkan API KEY <https://www.themoviedb.org/>. Adapun tahapannya adalah sebagai berikut:

1. Buka *Setting* di *profile picture* anda yang ada di <https://www.themoviedb.org/>.



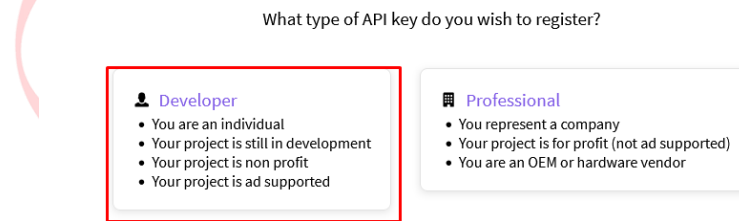
Gambar 6. User Settings

2. Pada menu bagian kiri layar, anda dapat memilih pilihan menu "API", dan melakukan Request API Key.



Gambar 6. Request API Key

- Pilih API Key untuk Developer yang merupakan API Key yang dapat anda gunakan selama proses Developer (belum menghasilkan profit)



Gambar 7. Request API Key untuk Developer

- Proses berikutnya adalah menerima *Agreement* dalam menggunakan API key.

5. **No Transfer.** The rights and obligations of these Terms of Use is personal to you and may not be transferred by you, either voluntarily or by operation of law.

6. **Notice.** Any notice to be sent to you under these Terms of Use may be sent via email, post, or any other reasonable means, at the contact information provided by you to TMDB from time to time. It is your obligation to insure that this information is current.

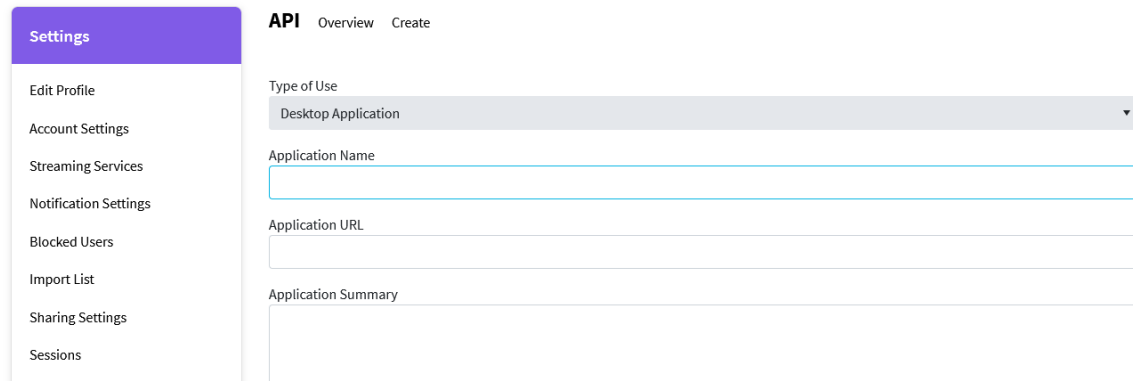
**Miscellaneous.** The section headings and subheadings contained in this agreement are included for convenience only, and shall not limit or otherwise affect the terms of the Terms of Use. Any construction or interpretation to be made of the Terms of Use shall not be construed against the drafter. The Terms of Use constitute the entire agreement between TMDB and you with respect to the subject matter hereof.

This Agreement was last updated on: July 28, 2014.



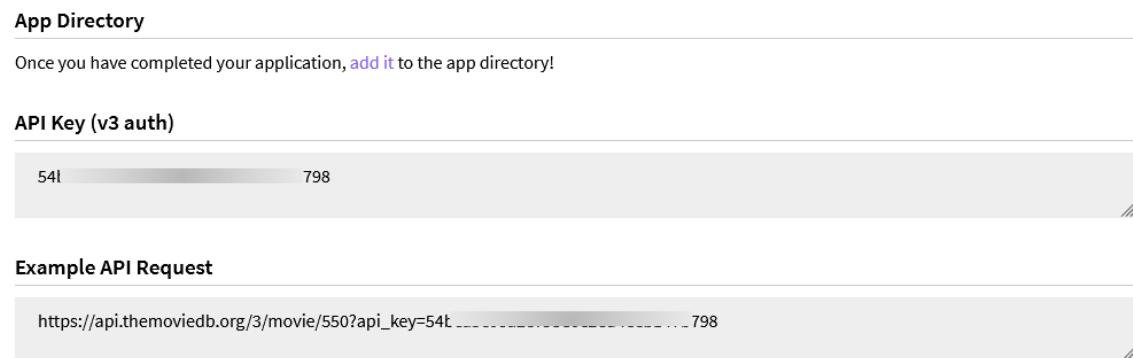
Gambar 8. Proses menyetujui *Term and Condition*

5. Data sesuai dengan isian pada form. Pada pengisian ini, anda hanya wajib mengisi semua isian pada form, tanpa harus menggunakan data yang aplikasi yang benar, seperti nama aplikasi, url aplikasi, dan summary aplikasi.



Gambar 9. Proses registrasi API Key

6. Setelah proses selesai, anda akan mendapatkan API Key yang dapat anda gunakan untuk mendapatkan data dari *web service*



Gambar 10. Tampilan Nilai API Key

Masukkan API Key yang anda dapatkan sebagai URL Key pada class `HttpHelper` anda, seperti berikut :

```

7  class HttpHelper {
8      final String _urlKey = "?api_key=54t...798";
9      final String _urlBase = "https://api.themoviedb.org";
10

```

Untuk dapat menggunakan paket http, pastikan anda melakukan *import* paket http dengan cara seperti berikut :

```
3  import 'package:http/http.dart' as http;
```

Sekarang anda siap untuk menulis metode pertama dari kelas ini, yang gunakan untuk ambil daftar film dengan menggunakan fungsi Future

```
18  Future<String> getMovie() async {  
19      var url = Uri.parse(_urlBase + '/3/movie/now_playing' + _urlKey);  
20      http.Response result = await http.get(url);  
21      if (result.statusCode == HttpStatus.ok) {  
22          String responseBody = result.body;  
23          return responseBody;  
24      }  
25      return result.statusCode.toString();  
26  }
```

Untuk dapat menggunakan enum HttpStatus.ok, anda perlu menambahkan sebuah library dengan mengimpor:

```
1  import 'dart:io';
```

Future digunakan untuk mewakili nilai yang akan dikembalikan, atau kesalahan, yang akan terjadi pada suatu waktu di masa depan. Dalam metode getMovie(), kita akan melakukan pengambilan data dari *web service*, dengan menentukan url tujuan pengambilan data. Proses pengambilan data dilakukan dengan fungsi http.get(Uri). Metode tersebut akan mengembalikan data yang berhasil diterima Ketika HTTP berhasil dijalankan.

Untuk Melihat hasilnya, anda dapat membuat sebuah tampilan home berikut ini:

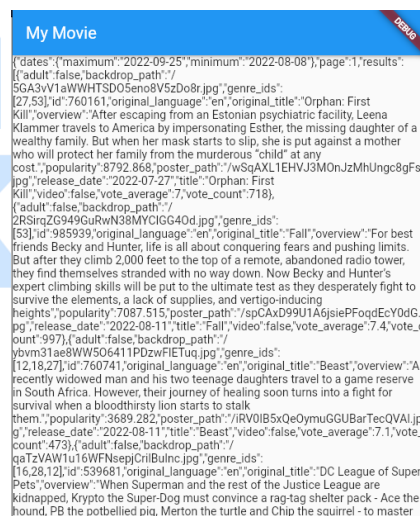


```

1 import 'package:flutter/material.dart';
2 import 'package:flutter_application_1/Praktek/HttpHelper.dart';
3
4 class MyHome extends StatefulWidget {
5   const MyHome({Key? key}) : super(key: key);
6
7   @override
8   State<MyHome> createState() => _MyHomeState();
9 }
10
11 class _MyHomeState extends State<MyHome> {
12   late String result;
13   late HttpHelper helper;
14
15   @override
16   void initState() {
17     super.initState();
18     helper = HttpHelper();
19     result = "";
20   }
21
22   @override
23   Widget build(BuildContext context) {
24     helper.getMovie().then((value) {
25       setState(() {
26         result = value;
27       });
28     });
29     return Scaffold(
30       appBar: AppBar(title: Text("My Movie")),
31       body: SingleChildScrollView(
32         child: Text("$result"),
33       ),
34     );
35   }
36 }
37

```

Adapun tampilan saat ini adalah sebagai berikut:



Gambar 11. Hasil Pembacaan API

Tampilan yang tersedia di layar saat ini hanya berupa teks dan berisi semua kode JSON diambil dari Movies API. Untuk menampilkan hasil pembacaan API yang lebih

baik, maka perlu dilakukan adalah melakukan Parsing JSON data menjadi objek data pada pertemuan kita berikutnya.

## Latihan

---

1. Kembangkanlah contoh pada pembacaan data, sehingga dapat membaca data film, berdasarkan : 'Latest', 'Now Playing', 'Popular', 'Top Rated', 'Upcoming', dengan menggunakan pilihan seperti Chips, Dropdownmenu, atau Radio Button.
2. Cari *web service* yang menyediakan data gratis yang dapat anda gunakan dalam menampilkan data yang disediakan oleh *web service* .



UNIVERSITAS  
MIKROSKIL