Networking in Flutter

TJMAD

📌 Announcements!

→ No announcements! (maybe)

Networking in Mobile Apps

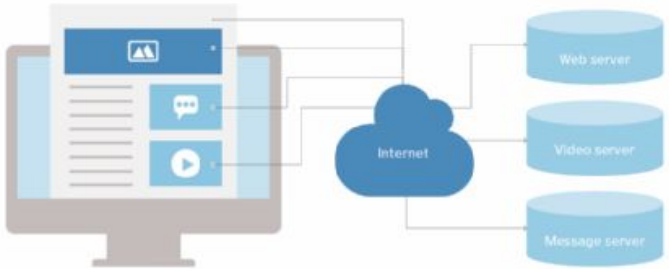
- → Networking = devices that can share data with each other
- → Protocols -> HTTP
- → Used in mobile apps frequently
- → Use cases:
  - Fetching data from servers
  - Interacting with APIs
  - Authentication (OAuth)

TJMAD

More HTTP

- **You** connect to server to fetch data = Request
- **Server** sends data to you = Response

- In Http:
  - Methods = GET, POST
  - Headers (e.g User-Agent, Authorization)
  - Body (Usually response or content you send with POST)



Response

→ Can be in different forms: JSON, plaintext, xml, pdf, etc.

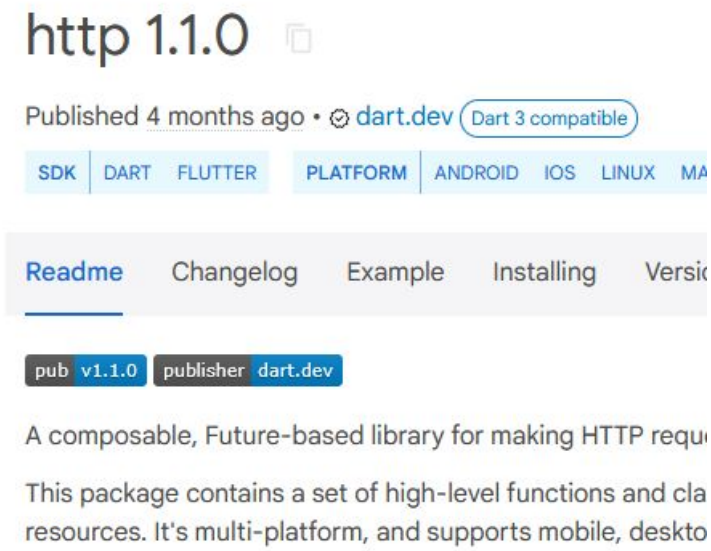
⟶ Error/status codes:

- **200** - **Success** (everything worked)
- **401** - **Unauthorized** (the client isn't allowed to view the url, possibly due to insufficient authorization)
- **403** - **Forbidden** (the client isn't allowed to view the url, but instead of insufficient authorization, the server refuses the client, possibly due to rate limits)
- **404** - **Not found** (server didn't find the url requested)

Http in Flutter

- Flutter doesn't have a built-in way to do http requests/networking

- http package
  - `flutter pub add http`
- https://pub.dev/packages/http



Making Requests in Flutter

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

String request() {
  final resp = http.get(Uri.parse("https://example.com"));

  // ...
  //resp.statusCode;
  // resp.body;
  // etc
  // ...
```

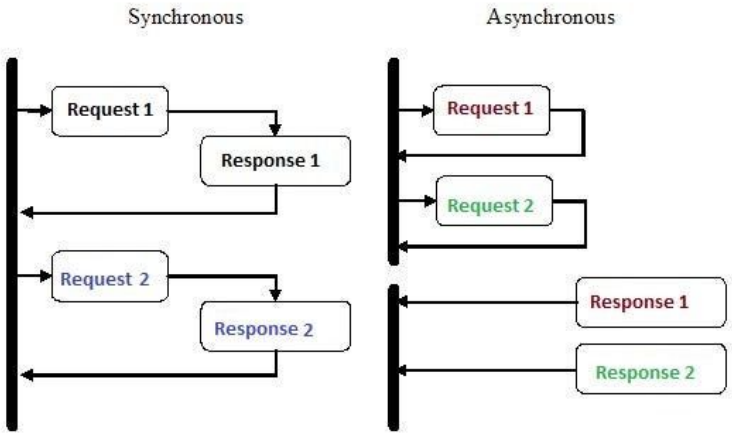
TJMAD

Asynchronous Programming

→ Asynchronous vs Synchronous

⟶ **Synchronous** - Tasks executed one after another, each task must complete before another begins

⟶ **Asynchronous** - Tasks can be run concurrently, tasks don't block the execution of other tasks



⟶ Network requests can take a while, and so should be done asynchronously.

Async Programming in Flutter

→ **async** - keyword to define asynchronous functions

→ **await** - indicates the function should pause and not block the execution of other code

→ **Future<T>** - The Future class represents a value that will be available sometime in the *future*.

→ **FutureBuilder<T>** - Flutter-provided widget to deal with Futures

TJMAD

Example

→ Go to **github.com/tjmadclub/lectures -> 2023-18-10 folder**

→ Copy the gallery code, unless you've made it yourself already

→ Now, instead of simply having a list of text details, **We will make a GET request to three urls to get the text details**

TJMAD