Server:

You will be writing a server that implements a status API and returns a result that is pending, completed or error. This is just simulating the video translation backend. It will return pending until a configurable time has passed.

GET /status

- Return result with {“result”: “pending” or “error” or “completed”}

Client Library:

You are writing a small client library to hit this server endpoint. Imagine you will be giving this library to a third party. They will be using it to get the status of the job.

In a trivial approach your library might just make a simple http call and wrap the errors and you ask the user of the library to call this repeatedly. If they call it very frequently then it has a cost, if they call it too slowly it might cause unnecessary delays in getting the status.

How can you do better than a trivial approach?

Demonstrate customer mindset while writing this library.

Deliverable

- A public git repository with your code
- Write a small integration test that spins up your server and uses your client library to demonstrate the usage and print logs.
- Write a small doc on how to use your client library.
- Demonstrate as many bells and whistles as you can