

## Dixa Backend Engineer test 💜

Thank you for taking your time and energy for this stage of our process.  
We hope you will enjoy the assignment! 🙏

Below you will find the description of the test and some guidelines on how to work on it. We're looking forward to seeing your best efforts even if you cannot solve the whole solution, so please also include your thinking & working process 💡

Time to begin... 🙌

Develop a set of 2 small services that work together to deliver a Fibonacci calculation of a given number.

### Description:

#### Proxy-service

The `proxy-service` acts as an entry point to the outside world.

It's main tasks are:

- expose a HTTP endpoint over REST responding to GET `/fibonacci/<number>` that continuously streams all Fibonacci numbers up to a given `<number>` e.g. `/fibonacci/17` should return `0,1,1,2,3,5,8,13`.
- delegates the actual calculation to the second microservice via a Finagle-Thrift OR GRPC RPC call
- handles wrong inputs in a proper way

#### Fibonacci-server

The `fibonacci-server` does the actual Fibonacci number calculation:  
it serves responses continuously over Finagle OR GRPC and  
uses proper abstractions to communicate failure

### Deliverables

There are three deliverables necessary to complete the task

proxy-service

fibonacci-server

thrift OR protobuf contracts used for communication between the two services

### Requirements

Language of implementation: Scala OR Java

Communication between - Finagle-Thrift OR GRPC

Basic scenario test cases

Proper commit history

README describing implementation choices and preferences