

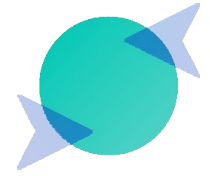


nodejs backend developer case study

2023

01

Hello

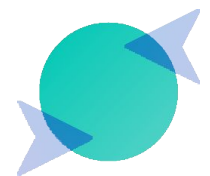


Welcome to this tech assessment.

You have to achieve a mission today, we hope you're ready for it!

Below are the team members you can interact with if you have any question.

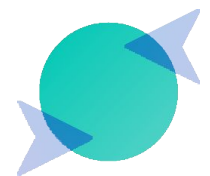
Good luck!



Timeboxing

You need to respond to email with the link on your personal github account.

You have 2 hours only to realise this exam, please respect the time



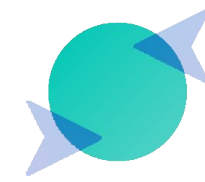
A question ?

Contacts:

antonin.b@bridgeapi.io

alexandre.k@bridgeapi.io

02 Mission 1 : Coding



Develop the application (context of application in annexes)

Pre Requirement

- Install node 16x on your machine
- Install your favorite IDE (example : VSCODE-)
- Clone this repository <https://github.com/bridgeapi-engineering/challenge-backend-connectors>
- You must have received an API Key to access the example bank API.
- Host your application code on your personal Github

Your mission: Build an application that consume the bank API <https://dsague.fr> to aggregate user's accounts and transactions.

Please, respect some constraints:

- Parse and aggregate the accounts and transactions with the format provided on next slide
- Remove duplicate accounts and transactions
- Handle pagination
- Your application must be launch with **npm start** command
- Optional: You heard about clean architecture, TDD, craftsmanship ? Show us please

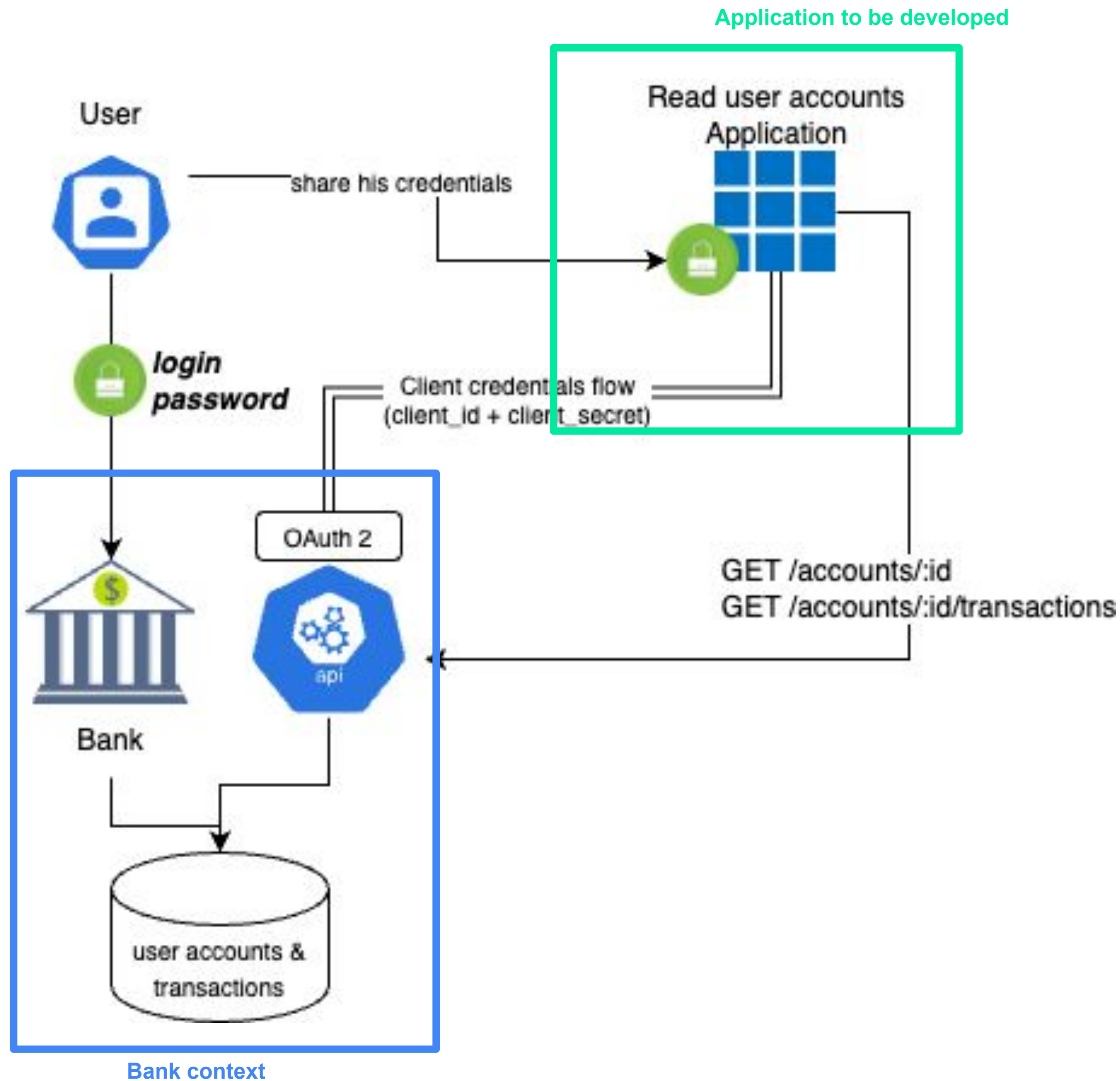
02 Mission 1 : Output format

```
[{
  "acc_number": "12345678",
  "amount": "123",
  "transactions": [
    {
      "label": "ZERTY",
      "amount": "-334",
      "currency": "EUR"
    }
  ],
  ...
},
{
  "acc_number": "12345688",
  "amount": "543",
  "transactions": [
    {
      "label": "label 1",
      "amount": "30",
      "currency": "EUR"
    }
  ],
  ...
}]
```



Annexes

System schema



Context:

- A user accesses his accounts using his credentials.
- A new application (called a [TPP](#)) asks for user's credentials to build a system on user's accounts and transactions.
- An API (PSD2 standard) is available. To access user accounts, the application needs to authenticate on API using his credentials (called client_id and client_secret) through the OAuth2 Client credentials flow