Assignment
for the position as a Backend developer
at Favendo

Introduction

This document defines some tasks that developers who would like to join Favendo will solve in their free time. The tasks are not supposed to take much time, but are rather designed to capture a general view on the programming skills of the candidate.

Topic: Banking service

A bank account has attributes like owners and their account balance. The owner should be able to withdraw money from cash machines or initiate money transfers to other users.
In this task, a little class hierarchy should be established that will consider different types of bank accounts. Each type of account stores an Owner (bank customer) and his Balance.

Here are some bank account types:

a) Savings Account: supports interest rates that are valid for all account holders. You may want to store the current interest rate and provide means to change it. From time to time, account holders would receive the interest that they desire.

b) Checking Account: the owner can withdraw more money than the size of his balance, but also restricted by a limit, which is different for each bank customer.
From a checking account, money transfers to other bank customers is possible.

Task

1) Implement a micro service that exposes a set of REST endpoints for each use case:
   - get balance for a customer
   - redraw money
   - transfer money from one customer to another
   - provide interest rates to saving account customer
2) Build a docker compose which contains all the components needed to run the service.

It is recommended implementing it using Spring boot framework and a in memory db, but you are free to choose otherwise if you desire.

When you are done, please send us a zip file containing your work.

Tips

- this is just an exercise so keep it simple, so security or other production ready features are out of scope
- even if it is just an exercise, design it as extensible as possible
- cover the use cases with unit tests
- add swagger or create a postman collection so we can easily test the endpoints