oneRagtime

Technical Test
Backend Developer

1. Technical test

For the technical test we want you to solve an example of a business problem we had.

You can use any language, or framework of your choice, but it should not contain any frontend.

We don't need you to worry about users and permissions.

If you feel the need to adapt the data structure of the giving example data do so.

We choose any of the following data at random and does not represent an actual use case.

2. Problem

The sending of invoices is currently done manually with excel.

We need an automated way to send the cash calls to the investor. A Cash call is an invoice we send to the user (in this example an investor) it groups a set of bills.

There are many types of bills that can be created. One investor can have multiple bills. These bills need to be generated by a script/command.

Those generated bills will be stored to be checked by a human eye. The Human has the option to group them by investor. Once validated we need to generate an invoice and email the investor, this is the cash call.(the generation of the invoice and sending of email are out of scope)

We need to check the status of the cash call if it has been: validated, sent, paid or is overdue. (receiving feedback from email servers or payment providers is out of scope)

2.1 Bills

Underneath you find some possible bills. Know that the way we calculate bills is always changing over time.

- **Membership**

  Active members need to pay a yearly subscription for using the platform. This is 3000 euro per year. When an investor invests that year over 50 000 euro he does not need to pay any of the yearly subscription.

- **Upfront fees**

  The investor has the choice to pay all his fees upfront for an investment.

  So we can only bill him once for it

  fee percentage x amount invested x 5 years

- **Yearly fees**

  An investor can also pay his fees per year for an investment.

Yearly fees before 2019/04

First year:

date of the investment bought / 365 x fee percentage x amount invested

other years:

fee percentage x amount invested

Yearly fees after 2019/04

First year:

date of the investment bought / amount days in that year x fee percentage x amount invested

Second year:

fee percentage x amount invested

Third year:

(fee percentage - 0.20 %) x amount invested

Fourth year:

(fee percentage - 0.50 %) x amount invested

Following years:

(fee percentage - 1 %) x amount invested

3. Deliverables

- production ready code
- if needed: a brief description of the assumption made.
- a git repo of your choosing where we can clone the code.