

Backend Developer Test Task

Introduction

With this test we want to get an idea of your proficiency in backend related web development technologies. We will give you a task specification below and you are free to use whatever PHP libraries, frameworks etc. you consider as useful to implement it together with SQL database (preferably MySQL or SQLite).

You shall provide us with a hosted git repository of your resulting work including a readme file to describe the required steps to deploy it locally.

Task

Implement a REST API that shall be used internally, so no authentication is needed. Format shall be JSON. No front-end or graphical interface is needed.

We need the following functionality/endpoints:

- Add new customer (gender, first name, last name, country, email)
 - each customer during creation should have assigned random bonus parameter between 5% and 20%
 - email shall be unique
- Edit customer details given on registration

For each customer the following operations shall be possible:

- Deposit Money
 - Every 3rd deposit of the customer should be awarded with bonus on the deposit amount according to his bonus parameter. For instance if a customer with 10% bonus is making deposit of 100 EUR, his balance shall increase by 110 EUR
 - Bonus balance needs to be kept separate from real money balance
- Withdraw Money
 - Customer balance can never go below 0 and bonus money cannot be withdrawn. For instance if customer balance is 110 EUR, but 10 EUR is bonus money, the maximum withdrawal amount is 100 EUR.
- Reporting
 - Endpoint generating a list of the total deposits and withdrawals (unique customers doing at least one deposit or withdrawal, number and total amount of both, deposits and withdrawals) per country and date for a given time frame (default: last 7 days). Example:

Date	Country	Unique Customers	No of Deposits	Total Deposit Amount	No of Withdrawals	Total Withdrawal Amount
2015-05-06	MT	32	45	456.34	24	-200.45
2015-05-06	DE	16	14	65.32	6	-456.34

Financial operations (deposit/withdrawal) needs to be implemented in a way that ensures data integrity also for situations where different transaction requests are made at the same moment.

Please implement tests you consider useful for a CI approach for at least part of the code, e.g. the functionality of one endpoint.