

Extension 5: Overdraft

By default the balance of a bank account should never be lower than 0 euros and therefore any action that attempts to lower this balance under this level should be rejected. However there are cases where a customer might want to overdraft his account to make some necessary payments. Therefore this extensions requires that you make the over-drafting of a bank account possible.

Not every user will want to use this service and even those who do want to use this service should have an overdraft limit for security reasons.

Your task in this extension is to add an overdraft limit to each bank account. The default limit should be 0 euro's but the user can alter this limit by logging in. At it's max this limit can be 5000 euros.

Users should be able to change this limit at any time, even if they are currently over the new limit. E.g. if a users with a negative balance of 1500 changes his limit to 1000. In this case the user will not be able pay/transfer money from this account anymore.

A bank account cannot be closed with a non zero balance. So any overdraft should be payed back before the account can be closed.

Your banking system should also have a negative interest rate of 10% in the case that the balance of a bank account is below zero. At the first day of each month you should calculate the combined interest of the previous month and retrieve this from the bank account.

Some specific details. Retrieving the interest from a bank account should be performed no matter the circumstances. Even if the bank account is already at or over it's current limit.

In addition. The interest should be calculated per day. You should use the lowest balance reached on a day as the value you use to calculate interest over that day.

This method should work together with the time simulation method in that on the first day of each month all interest of all bank accounts should be calculated and withdrawn from the accounts.

Finally, be aware that you should correct for compound interest, i.e. the interest on interest. If a balance is negative 1000 at the beginning of the year, it should be negative 1100 at the end of the year, even though you withdraw interest each month.

See the next page for additional methods that you will need to implement. See the third page of this extension for more information about calculating the interest.

setOverdraftLimit method

Method setOverdraftLimit

Description Method that sets the overdraft limit for bank account.

Authentication REQUIRED (account)

Parameters This method accepts the following parameters:

authToken The authentication token, obtained with getAuthToken

iBAN The number of the bank account

overdraftLimit The new overdraft limit for this bank account.

Returns An empty dictionary if successful

Errors InvalidParamValueError: One or more parameter has an invalid value. See message.
NotAuthorizedError: The authenticated user is not authorized to perform this action.

Example

```
--> {"method":"setOverdraftLimit",
      "params":{"
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
        "overdraftLimit": 1234,
      }}
<-- {"result": {}}
```

getOverdraftLimit method

Method getOverdraftLimit

Description Method that asks the server for the overdraft limit of a bank account.

Authentication REQUIRED (account)

Parameters This method accepts the following parameters:

authToken The authentication token, obtained with getAuthToken

iBAN The number of the bank account

Returns A dictionary containing the following members:

overdraftLimit The overdraft limit of this bank account.

Errors (none)

Example

```
--> {"method":"getOverdraftLimit",
      "params":{"
        "authToken": "some.long.auth.token!%&...",
        "iBAN": "NL69INGB0123456789",
      }}
```

```
    }}  
<-- {"result": {"overdraftLimit": 1234}}
```

Detail on interest calculation.

A large part of this extension is to implement the interest calculation each month. You however should not spend too much time on figuring out how to calculate the interest in theory. Therefore we give an example of how this can be done below. But be warned that we use rounded numbers, so this method produces some rounding errors. Try to minimize these rounding errors in your implementation.

We take our initial example where you have a negative 1000 dollars and a 10% annual interest rate.

If you calculate this on a yearly base, this would simply be $1000 * 1.1 = 1100$ dollars. But we want to calculate and charge this interest on a monthly base while still obtaining the same result after one year. Therefore we have to change the annual percentage rate to a monthly one that corrects for compound interest. This can be done as following:

$$\text{Effective rate for period} = (1 + \text{annualrate})^{(1/\#ofperiods)} - 1$$

or

$$\text{Monthly rate} = (1 + 0.10)^{1/12} - 1 \approx 0.00797$$

If use and charge this interest rate each month, the result would be 1100 once gain at the end of the year. I.e. $1000 * (1.00797)^{12} \approx 1100$.

However, because the value on the bank account can change everyday, we cannot calculate the monthly interest using the value of the account at the last day of the month. We want to calculate it individually for each day's lowest negative amount. But since we don't add this interest each day, we do not have to correct for compound interest. So we can simply calculate the daily rate by dividing the monthly rate by the days in a month. And then use that daily interest rate to calculate the interest of each day.

So, if we want to calculate the interest for the first of January. We would first obtain the daily rate of January, which is $0.00797/31 \approx 0.00026$. Then we can calculate the interest of the day, which is $1000 * 0.00026 \approx 0.26$ euro's.

In this way you can calculate the interest for each day of the month, add all these interests together and calculate and charge the monthly interest.