Extension 6: Savings account

We want to introduce a savings account to the banking system. Existing customer should be able to open a savings account that is connected to one of their regular accounts. The way that ING works however is that this does not create a completely separate bank account for the savings account. The savings account will receive the same IBAN as the regular (checking) account, giving you basically two banking accounts using the same ID.

Some more details for the savings account. A savings account must be connected to a regular checking account and cannot exist alone. Furthermore a savings account can only receive money from the checking account and can only transfer money bank to this checking account. A savings account should always have the same owner and set of people with access as the main account.

Your assignment is to give a customer the option to open and close a savings account for each bank account he/she is the owner of. Some more details: Once you open it, it's balance should be 0. It is not possible to go below zero with this account. Once you close it, all money should be returned to the regular checking account. Closing the regular account also closes the savings account.

In addition to this, the bank should give interest on the money on the savings account. This should happen on the first of January each year. Since the balance of the savings account can change, you should calculate the interest on a daily basis, using the lowest amount of each day as the value to calculate interest over for that day. Finally, you should use the interest rates as specified by ING in table 0.1.

This extension also requires some changes to the protocol. The first change is adding two methods to open and close a saving account. In addition we need a method to transfer money between the checking and savings account. We propose to use the transferMoney method for this, were we add the option to add an 'S' to the end of an IBAN to indicate that the source or target IBAN is the savings account.

Table 0.1: Interest rates of ING Savings Account

Balance	**Interest**
Until € 25.000	0,15%
From € 25.000 until € 75.000	0,15%
From € 75.000 until € 1.000.000	0,20%

Finally we need to update two info methods. First off it needs to be possible to see the balance of the savings account. In addition it needs to be possible to see the transactions of this account. We propose that you implement these as follows: The getBalance method should return a second value, namely "savingAccountBalance": 0,00". The default getTransactionOverview method already shows the all the transfers too and from the savings account, only the interest is not shown by this method. For simplicity we propose that the getTransactionOverview method returns the transactions of both accounts. So the interest on savings account transaction should be returned by the getTransactionOverview method in addition to the transactions of the checking account. In this case all transactions should differentiate between the regular and checking accounts using the S at the end of the IBAN.

openSavingsAccount method

Method openSavingsAccount

Description Method that attempts to open a savings account for 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 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":"openSavingsAccount",
        "params":{
            "authToken": "some.long.auth.token!%&...",
            "iBAN": "NL69INGB0123456789",
        }}
<-- {"result": {}}
```

closeSavingsAccount method

Method closeSavingsAccount

Description Method that attempts to close a savings account for 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 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":"closeSavingsAccount",
        "params":{
            "authToken": "some.long.auth.token!%&...",
            "iBAN": "NL69INGB0123456789",
        }}
<-- {"result": {}}
```