Extension 4: Time simulation

For some components of the banking system, time plays a important role. Think for example about the expiration date of a PIN card. So far the time aspect has not really received attention in the protocol however, as it was not really possible to test time related aspects as the expiration date of a PIN card.

The task of this assignment is to give the bank system its own date, separate from the actual date (even though these should initially be the same). The date of the system should be just as persistent as your other data. The bank system should only use this date for all actions involving a date, such as processing a transaction or checking if a PIN card is expired.

In addition to this we ask you to implement a method to process the passing of a days. This method should update the state of the system accordingly, such as updating the date of the system with the number of days passed. The banking system can be requested to process the passing of a specified amount of days through the protocol.

We also ask that you implement a method to reset the state of the system to the initial state, including the date of the system. We for-see this feature being used in test cases involving the time simulation. If you want to reset the date of the system, you also need to reset the other data, otherwise you could have transactions in the future.

Finally we ask that you implement a protocol method to return the date of the system to any client that requests it.

This change introduces a number of changes onto the protocol. First is a new method to initiate the passing of time, see the next page. In addition all methods that create a new PIN card should now also return an expiration date (add parameter like this: "experidationDate": "2021-02-19") and attempts to use an expired PIN card should result in an InvalidPIN-Error. Furthermore a method that requests the system to reset its state and a method that requests the date of the server should be added.

An example of the behaviour that we expect from the system. When you start the system for the first time, it starts with an empty data-set and at the current date. We create a new user account and receive back the required information including the expiration date (e.g."experidationDate": "2021-02-19"). We then ask the system to simulate the passing of 2000days (around 5 years). If the user transfers money at that time, the new date should be stored as the transfer date. Payments with the card of this bank account should be blocked. If we request the date of tje system it should return a date 2000 days after the date when the system was initialized. If we than request that the system resets itself it should return the state of the system to the initial state, i.e. remove all of it's data.

simulateTime method

Method simulateTime

Description Method that asks the server to process the passing of a specified number of days.

Parameters This method accepts the following parameters:

nrOfDays The number of days that should be simulated.

Returns An empty dictionary if successful

Errors InvalidParamValueError: One or more parameter has an invalid value. See message.

Example

```
--> {"method":"simulateTime",
        "params":{
            "nrOfDays": 1337
        }}
<-- {"result": {}}
```

reset method

Method reset

Description Method that asks the server to restore the state to it’s initial value

Parameters The parameter of this method is an empty dictionary.

Returns An empty dictionary if successful

Example

```
--> {"method":"reset",
        "params":{}}
<-- {"result": {}}
```

getDate method

Method getDate

Description Method that asks the server to get it’s simulated date

Parameters The parameter of this method is an empty dictionary.

Returns A dictionary containing the following members:

date The date of the system

```
--> {"method":"getDate",
        "params":{}}
<-- {"result": {"date": "2021-06-14"}}
```