

Extension 7: Logging

For this extension we want you to implement a logging system into your bank server. The purpose of the logging system is twofold, the first reason being that you want to be able to know how you reached your current state and the second being to detect suspicious behaviour.

For the first purpose, you should log every action your system performs that changes it's state. In the end this log should contain sufficient information that you could replay all events that occurred on the initial state of the server and reach the current state once again, although we do not ask you to develop such a replay system.

For the second purpose, we want you to log every request the server receives that produces an error. For example failed login or pin attempts.

You are mostly free how to implement this logging mechanics. Such as the formatting of the messages or how you store them. We do however have a few requests for this implementation.

First off, we want you to attach a time-stamp to each log event. This time-stamp should adhere to RFC 3339 and be unique for every log event.

In addition, we want to be able to request all logs you made between two dates. This request should return an array of events that are logged between those two dates. The content of those event logs are however up to you :)

getEventLogs method

Method getEventLogs

Description Method that requests all events logged between two dates.

Parameters This method accepts the following parameters:

beginDate Begin date of period for which log are requested

endDate End date of period for which log are requested

Returns An array of dictionaries containing the following members:

timeStamp TimeStamp of the event

eventLog Info about the event

Example

```
--> {"method":"getEventLogs",
      "params":{"
        "beginDate": "2017-06-25 ",
        "endDate": "2017-06-28",
      }}
<-- {"result": [{
      "timeStamp": "2017-08-30T13:22:53.108Z",
      "eventLog": "Event X happend because Y",
    }, {...}, {...}]}
```