

Requisitos

O projeto foi desenvolvido utilizando Python 3.8 com Flask 2.1 .

Servidor de Dados

Código disponível em: data_server.py .

O comando abaixo inicializa o servidor de dados. Por padrão, ele é executada na porta 5000.

```
python3.8 data_server.py
```

Todo o log das operações executadas pelo servidor de dados são salvas no arquivo data_server.log no formato a seguir:

TIMESTAMP, NUMERO OPERACAO, ID_NEGOC, OPERACAO, *PARAMETROS DA OPERAÇÃO

Rotas

endpoint	método	headers	query string	descrição
/status	GET			verifica status do servidor
/autentica	GET	token	id_negoc	autentica o token informado
/getSaldo	GET	token	id_negoc, conta	retorna o saldo da conta informada
/setSaldo	GET	token	id_negoc, conta, valor	atualiza o saldo com o valor informado
/getLock	GET	token	id_negoc, conta	bloqueia a conta para escrita de outros servidores de negócio
/setLock	GET	token	id_negoc, conta	libera a conta para escrita de outros servidores de negócio

Servidor de Negócio

Código disponível em: business_server.py

O código abaixo inicializa um servidor de negócios, com o servidor de dados executando na url http://localhost:5000 , na porta 5001 . A porta é opcional, o valor padrão é 5000 .

```
python3.8 business_server.py http://localhost:5000 5001
```

Todo o log das operações executadas pelo servidor de negócio são salvas no arquivo business_server.log no formato a seguir:

TIMESTAMP, NUMERO OPERACAO, ID_NEGOC, OPERACAO, *PARAMETROS DA OPERAÇÃO

Rotas

endpoint	método	headers	descrição
/status	GET		verifica status do servidor
/deposito/<conta>/<valor>	GET	token	faz um deposito de <valor> na conta <conta>
/saque/<conta>/<valor>	GET	token	faz um saque de <valor> na conta <conta>
/saldo/<conta>	GET	token	consulta o saldo da <conta>
/transferencia/<conta_orig>/<conta_dest>/<valor>	GET	token	faz uma transferência de <valor> da conta <conta_orig> para a conta <conta_dest>

Cliente

Código disponível em: `client.py`.

O código abaixo inicializa um cliente utilizando o servidor de negócios na url `http://localhost:5001` e se autenticando pela url `http://localhost:5000/autentica`.

```
from client import *

client = Client("http://localhost:5001", "http://localhost:5000/autentica")

# Depósito de 5000 para a conta 2
client.deposito(2, 5000)

# Transferência de 2000, da conta 2 para a conta 1
client.transferencia(2, 1, 2000)

# Saque de 1000 da conta 1
client.saque(1, 1000)

# Saldo da conta 2
client.saldo(2)
```
