Team 8 Project Report

Team Members

Αλίκη Χρήστου

Δημήτρης Σώτος

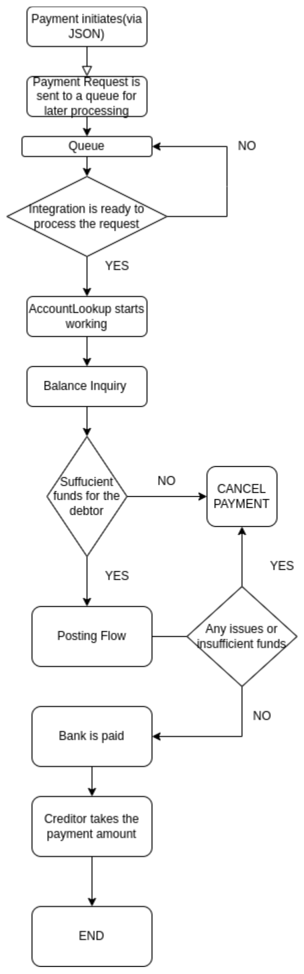
Κωνσταντίνος Τζιρβελάκης

Σάββας Σκουλίδης

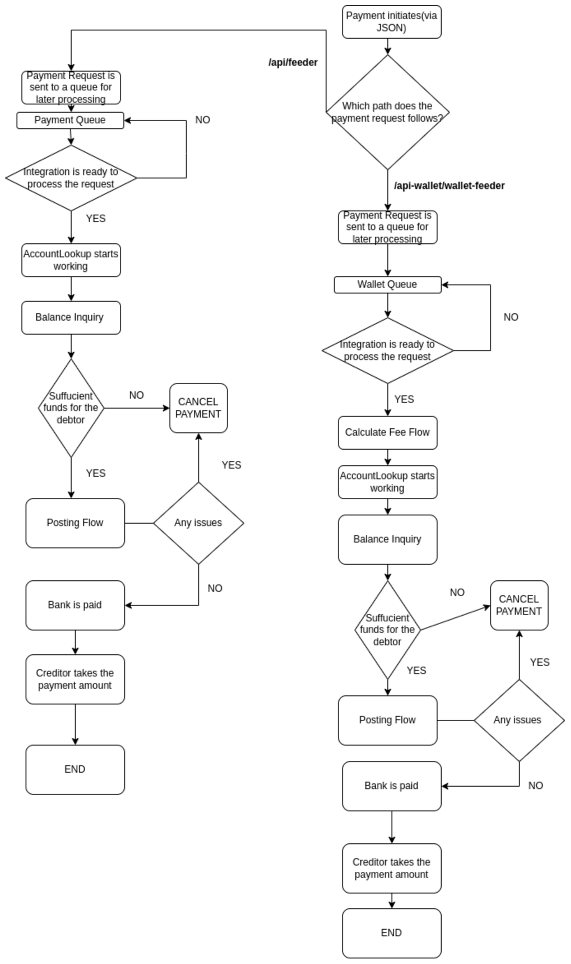
Σοφία Τερζοπούλου

Flow Diagrams

1. Diagram of business flow of the Payment System

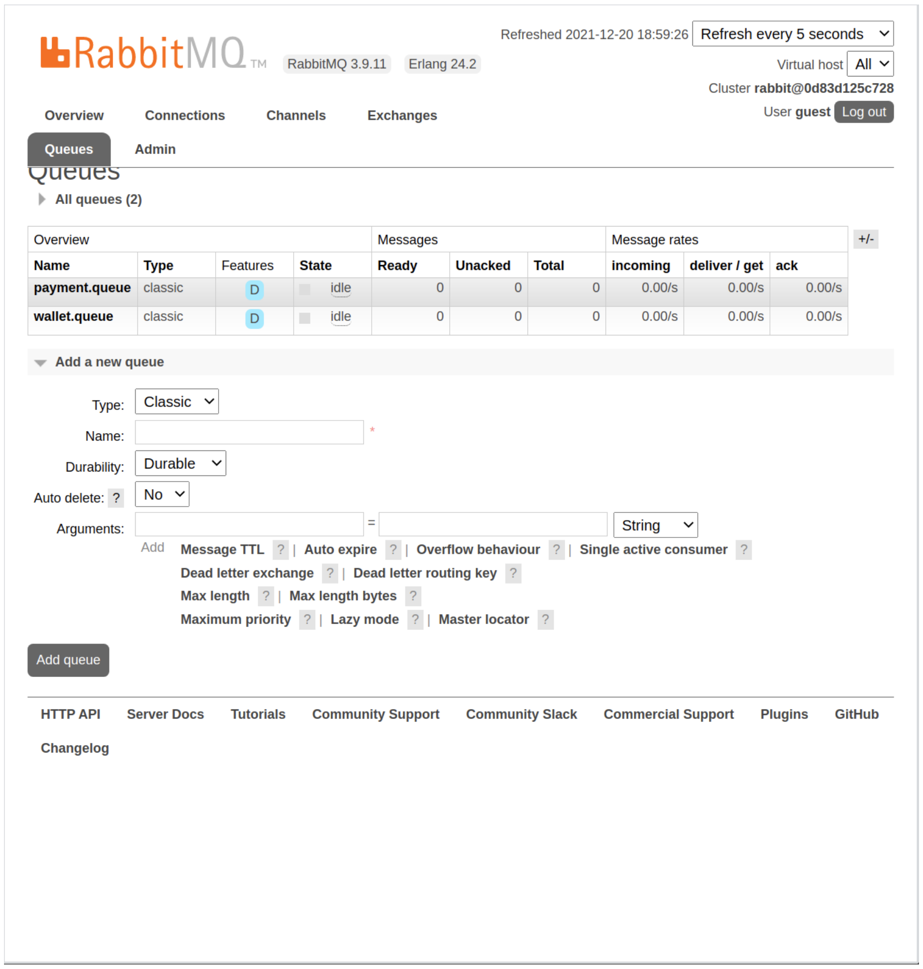

Payment initiates(via JSON)
Payment Request is sent to a queue for later processing
Queue
NO
Integration is ready to process the request
YES
AccountLookup starts working
Balance Inquiry
Suffucient funds for the debtor
NO
CANCEL PAYMENT
YES
YES
Posting Flow
Any issues or insufficient funds
NO
Bank is paid
Creditor takes the payment amount
END

2. Diagram of business flow of the improved Payment System (Normal and Wallet)


Payment initiates(via JSON)
/api/feeder
Which path does the payment request follows?
/api-wallet/wallet-feeder
Payment Request is sent to a queue for later processing
Payment Queue
NO
Integration is ready to process the request
YES
AccountLookup starts working
Balance Inquiry
Suffucient funds for the debtor
NO
CANCEL PAYMENT
YES
Posting Flow
Any issues
YES
NO
Bank is paid
Creditor takes the payment amount
END
Payment Request is sent to a queue for later processing
Wallet Queue
NO
Integration is ready to process the request
YES
Calculate Fee Flow
AccountLookup starts working
Balance Inquiry
Suffucient funds for the debtor
NO
CANCEL PAYMENT
YES
Posting Flow
Any issues
YES
NO
Bank is paid
Creditor takes the payment amount
END

RabbitMQ

The following image represents the two queues of Normal Payments and Wallet Payments running in RabbitMQ.



Queues

All queues (2)

Overview				Messages			Message rates		
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
payment.queue	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
wallet.queue	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s

Add a new queue

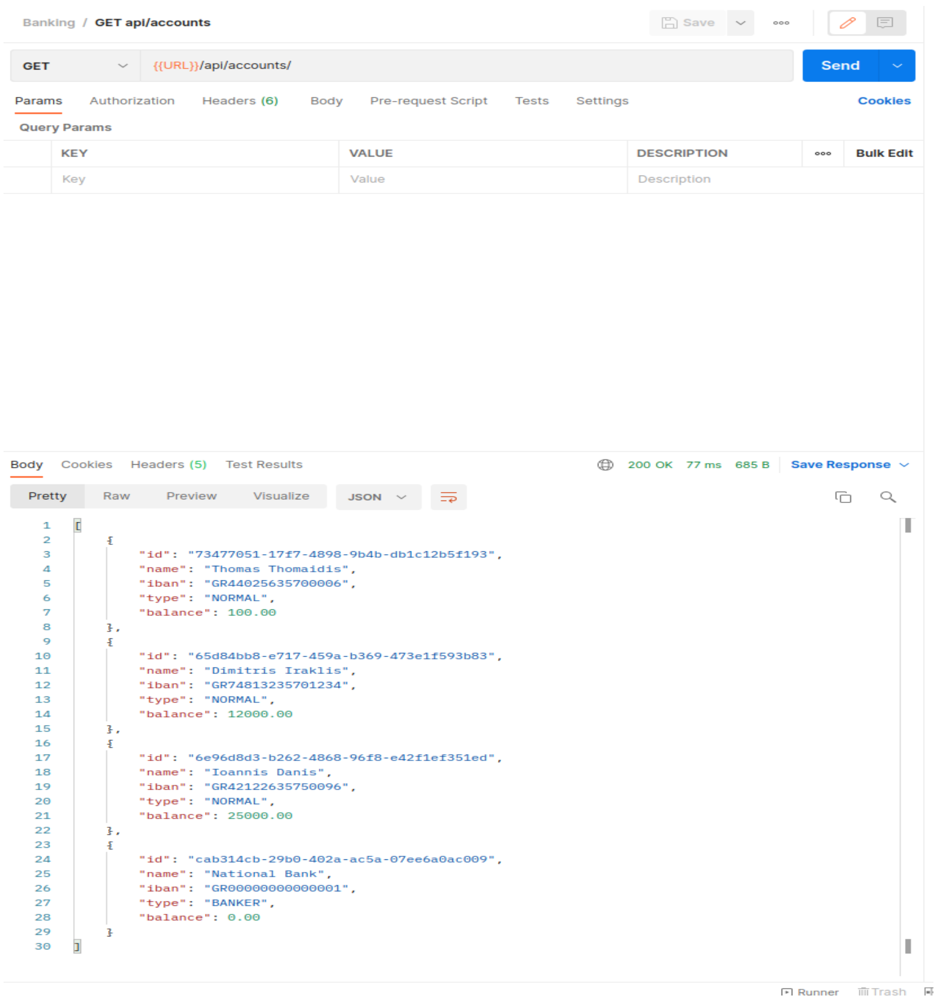
Type:	Classic
Name:	
Durability:	Durable
Auto delete:	No
Arguments:	= String

Add Message TTL | Auto expire | Overflow behaviour | Single active consumer
Dead letter exchange | Dead letter routing key
Max length | Max length bytes
Maximum priority | Lazy mode | Master locator

Case Study1: Successful Normal Payment

In the first case study we present a successful normal payment in the transaction system.

1. Check the accounts before the transaction (Api/accounts/)



Banking / GET api/accounts

```
GET {{URL}}/api/accounts/
```

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests | Settings | Cookies

Query Params

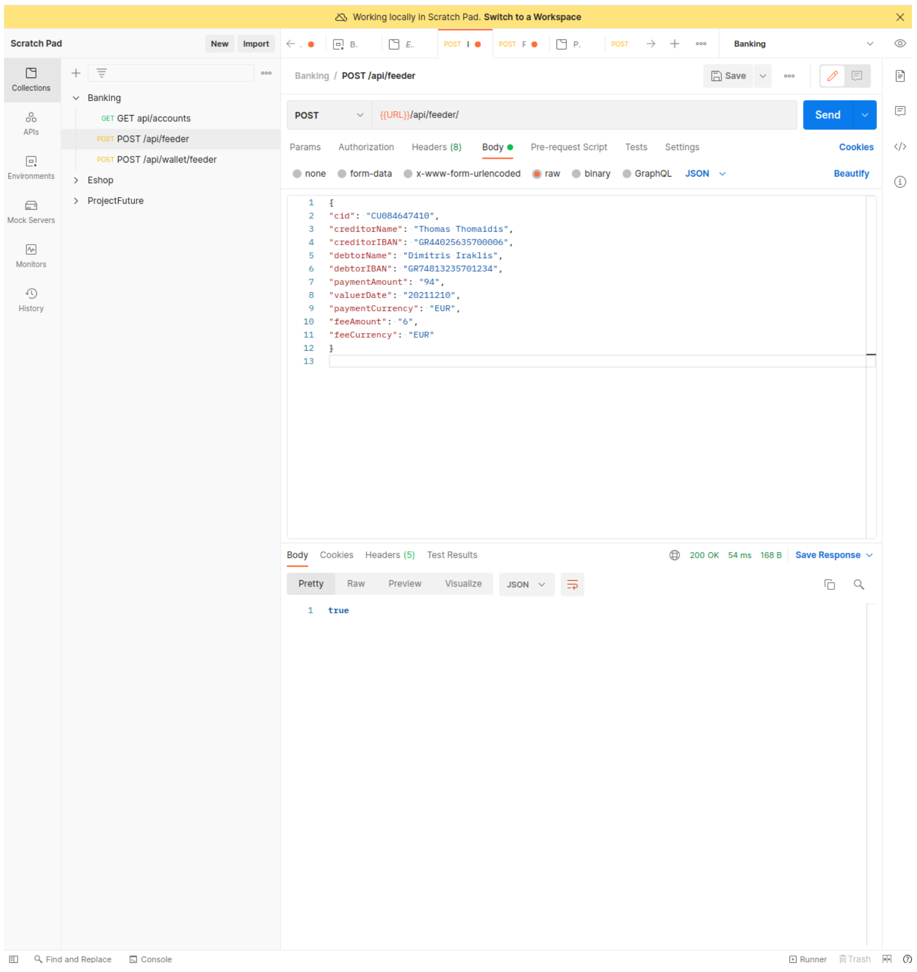
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body | Cookies | Headers (5) | Test Results — 200 OK 77 ms 685 B — Save Response

Pretty | Raw | Preview | Visualize | JSON

```
[
    {
        "id": "73477051-17f7-4898-9b4b-db1c12b5f193",
        "name": "Thomas Thomaidis",
        "iban": "GR44025635700006",
        "type": "NORMAL",
        "balance": 100.00
    },
    {
        "id": "65d84bb8-e717-459a-b369-473e1f593b83",
        "name": "Dimitris Iraklis",
        "iban": "GR74813235701234",
        "type": "NORMAL",
        "balance": 12000.00
    },
    {
        "id": "6e96d8d3-b262-4868-96f8-e42f1ef351ed",
        "name": "Ioannis Danis",
        "iban": "GR42122635750096",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "cab314cb-29b0-402a-ac5a-07ee6a0ac009",
        "name": "National Bank",
        "iban": "GR00000000000001",
        "type": "BANKER",
        "balance": 0.00
    }
]
```

2. Send through Postman a post of a simple request for bank transfer via JSON File described below.



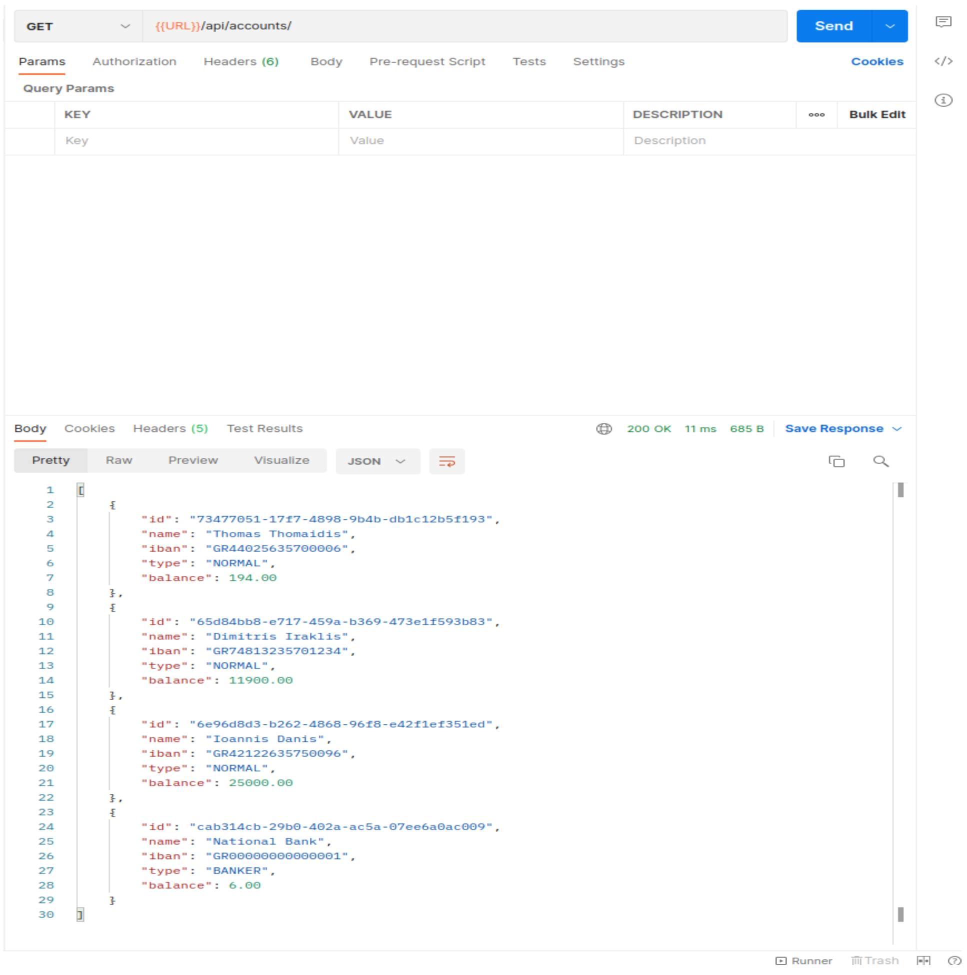
POST {{URL}}/api/feeder/

```
{
"cid": "CU084647410",
"creditorName": "Thomas Thomaidis",
"creditorIBAN": "GR44025635700006",
"debtorName": "Dimitris Iraklis",
"debtorIBAN": "GR74813235701234",
"paymentAmount": "94",
"valuerDate": "20211210",
"paymentCurrency": "EUR",
"feeAmount": "6",
"feeCurrency": "EUR"
}
```

Body — 200 OK 54 ms 168 B

```
true
```

3. The amounts of the bank accounts changed, and the bank received the fee amount as described.


GET {{URL}}/api/accounts/

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

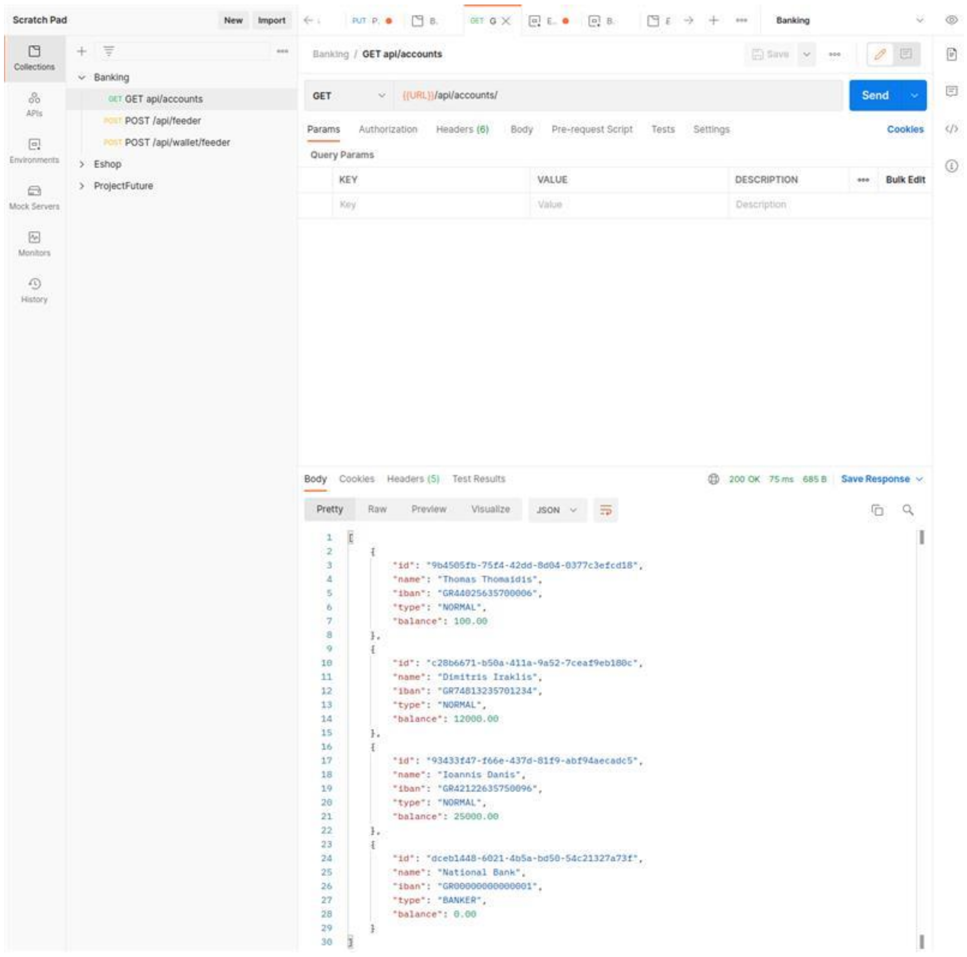
200 OK 11 ms 685 B

```
[
    {
        "id": "73477051-17f7-4898-9b4b-db1c12b5f193",
        "name": "Thomas Thomaidis",
        "iban": "GR44025635700006",
        "type": "NORMAL",
        "balance": 194.00
    },
    {
        "id": "65d84bb8-e717-459a-b369-473e1f593b83",
        "name": "Dimitris Iraklis",
        "iban": "GR74813235701234",
        "type": "NORMAL",
        "balance": 11900.00
    },
    {
        "id": "6e96d8d3-b262-4868-96f8-e42f1ef351ed",
        "name": "Ioannis Danis",
        "iban": "GR42122635750096",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "cab314cb-29b0-402a-ac5a-07ee6a0ac009",
        "name": "National Bank",
        "iban": "GR00000000000001",
        "type": "BANKER",
        "balance": 6.00
    }
]
```

Case Study2: Successful Wallet Payment

In the second case study we present a successful wallet payment in the transaction system.

1. Check the accounts before the transaction (Api/accounts/)

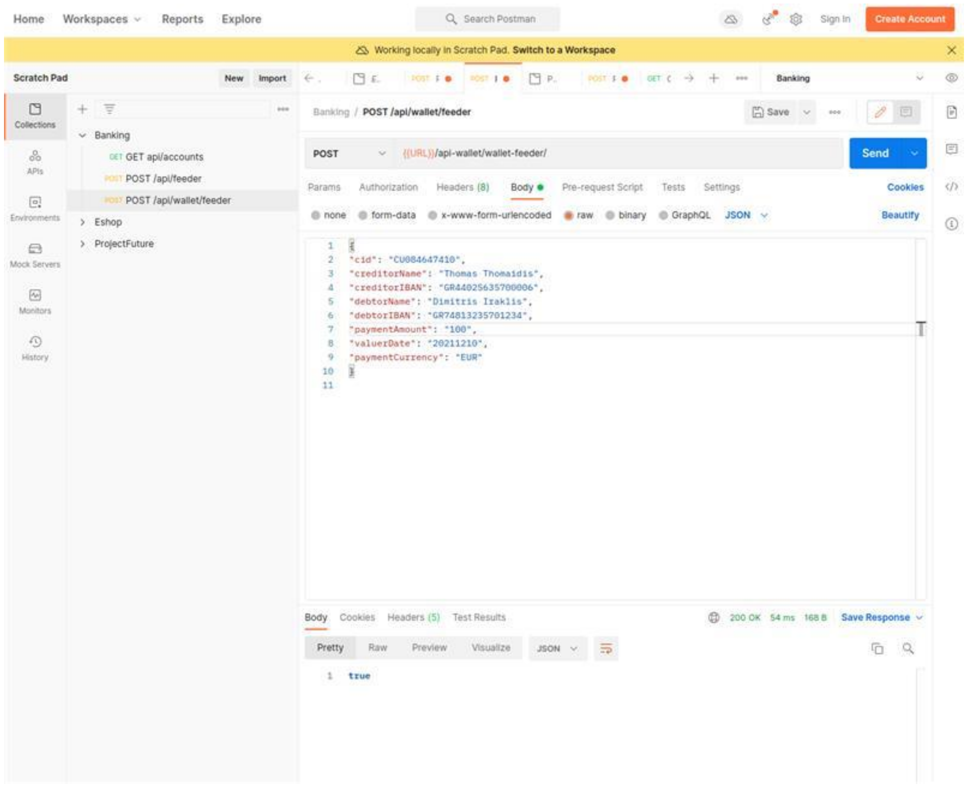


GET {{URL}}/api/accounts/

200 OK 75 ms 685 B

```
[
    {
        "id": "9b4505fb-75f4-42dd-8d04-0377c3efcd18",
        "name": "Thomas Thomaidis",
        "iban": "GR44025635700006",
        "type": "NORMAL",
        "balance": 100.00
    },
    {
        "id": "c28b6671-b50a-411a-9a52-7ceaf9eb180c",
        "name": "Dimitris Iraklis",
        "iban": "GR74813235701234",
        "type": "NORMAL",
        "balance": 12000.00
    },
    {
        "id": "93433f47-f66e-437d-81f9-abf94aecadc5",
        "name": "Ioannis Danis",
        "iban": "GR42122635750096",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "dceb1448-6021-4b5a-bd50-54c21327a73f",
        "name": "National Bank",
        "iban": "GR00000000000001",
        "type": "BANKER",
        "balance": 0.00
    }
]
```

2. Send through Postman a post of a simple request for wallet bank transfer via JSON File described below.



Banking / POST /api/wallet/feeder

POST {{URL}}/api-wallet/wallet-feeder/

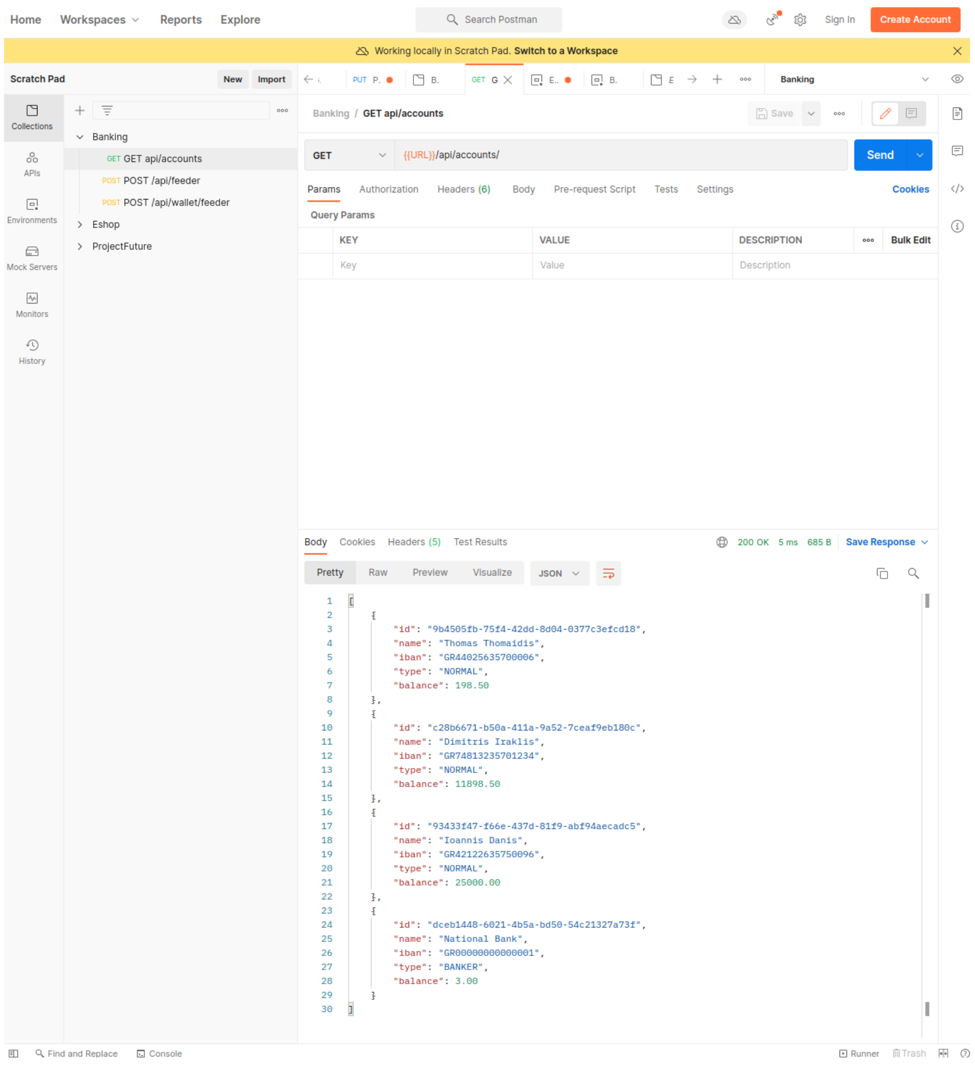
Body raw JSON

```
{
"cid": "CU084647410",
"creditorName": "Thomas Thomaidis",
"creditorIBAN": "GR44025635700006",
"debtorName": "Dimitris Iraklis",
"debtorIBAN": "GR74813235701234",
"paymentAmount": "100",
"valuerDate": "20211210",
"paymentCurrency": "EUR"
}
```

200 OK 54 ms 168 B

```
true
```

3. The amounts of the bank accounts changed, and the bank received the fee amount as described.


GET {{URL}}/api/accounts/

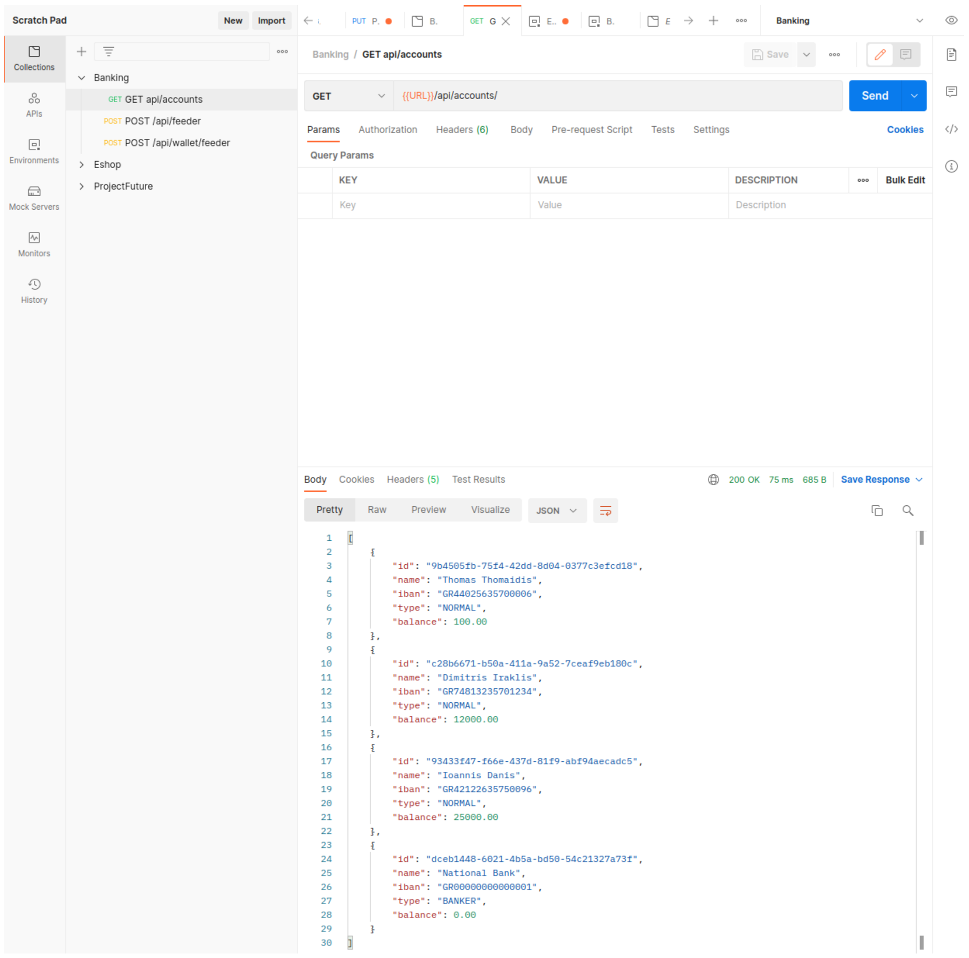
200 OK 5 ms 685 B

```
[
    {
        "id": "9b4505fb-75f4-42dd-8d04-0377c3efcd18",
        "name": "Thomas Thomaidis",
        "iban": "GR4402563570006",
        "type": "NORMAL",
        "balance": 198.50
    },
    {
        "id": "c28b6671-b50a-411a-9a52-7ceaf9eb180c",
        "name": "Dimitris Iraklis",
        "iban": "GR7481323570I234",
        "type": "NORMAL",
        "balance": 11898.50
    },
    {
        "id": "93433f47-f66e-437d-81f9-abf94aecadc5",
        "name": "Ioannis Danis",
        "iban": "GR4212263575009G",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "dceb1448-6021-4b5a-bd50-54c21327a73f",
        "name": "National Bank",
        "iban": "GR000000000000001",
        "type": "BANKER",
        "balance": 3.00
    }
]
```

Case Study3: Unsuccessful Wallet Payment

In the first case study we present an unsuccessful normal payment in the transaction system.

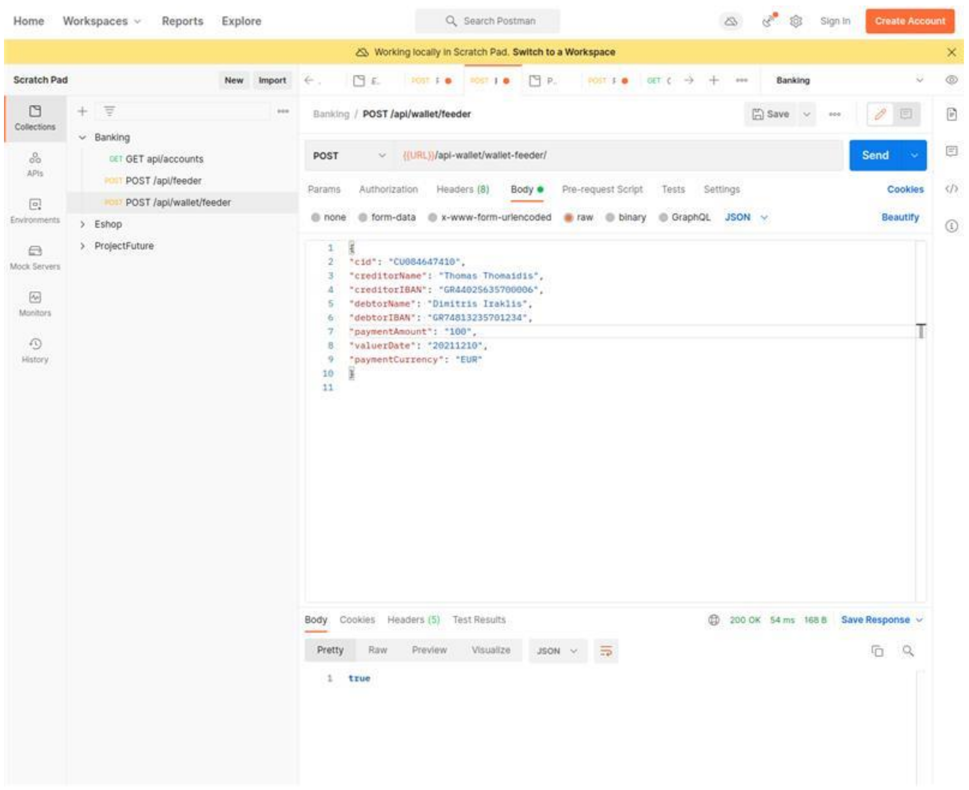
1. Check the accounts before the transaction (Api/accounts/)



GET {{URL}}/api/accounts/ Send

```
[
    {
        "id": "9b4505fb-75f4-42dd-8d04-0377c3efcd18",
        "name": "Thomas Thomaidis",
        "iban": "GR44025635700006",
        "type": "NORMAL",
        "balance": 100.00
    },
    {
        "id": "c28b6671-b50a-411a-9a52-7ceaf9eb180c",
        "name": "Dimitris Iraklis",
        "iban": "GR74813235701234",
        "type": "NORMAL",
        "balance": 12000.00
    },
    {
        "id": "93433f47-f66e-437d-81f9-abf94aecadc5",
        "name": "Ioannis Danis",
        "iban": "GR42122635750096",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "dceb1448-6021-4b5a-bd50-54c21327a73f",
        "name": "National Bank",
        "iban": "GR00000000000001",
        "type": "BANKER",
        "balance": 0.00
    }
]
```

2. Send through Postman a post of a simple request for wallet bank transfer via JSON File described below.

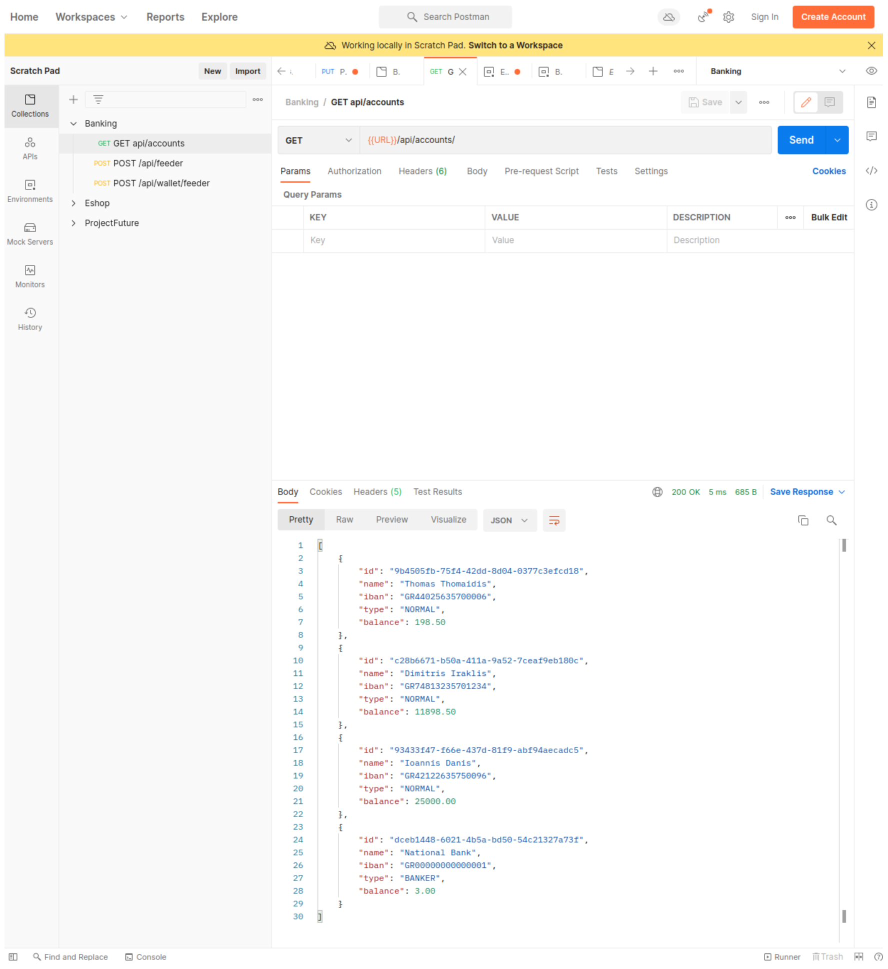


POST {{URL}}/api-wallet/wallet-feeder/ Send

```
{
"cid": "CU084647410",
"creditorName": "Thomas Thomaidis",
"creditorIBAN": "GR44025635700006",
"debtorName": "Dimitris Iraklis",
"debtorIBAN": "GR74813235701234",
"paymentAmount": "100",
"valuerDate": "20211210",
"paymentCurrency": "EUR"
}
```

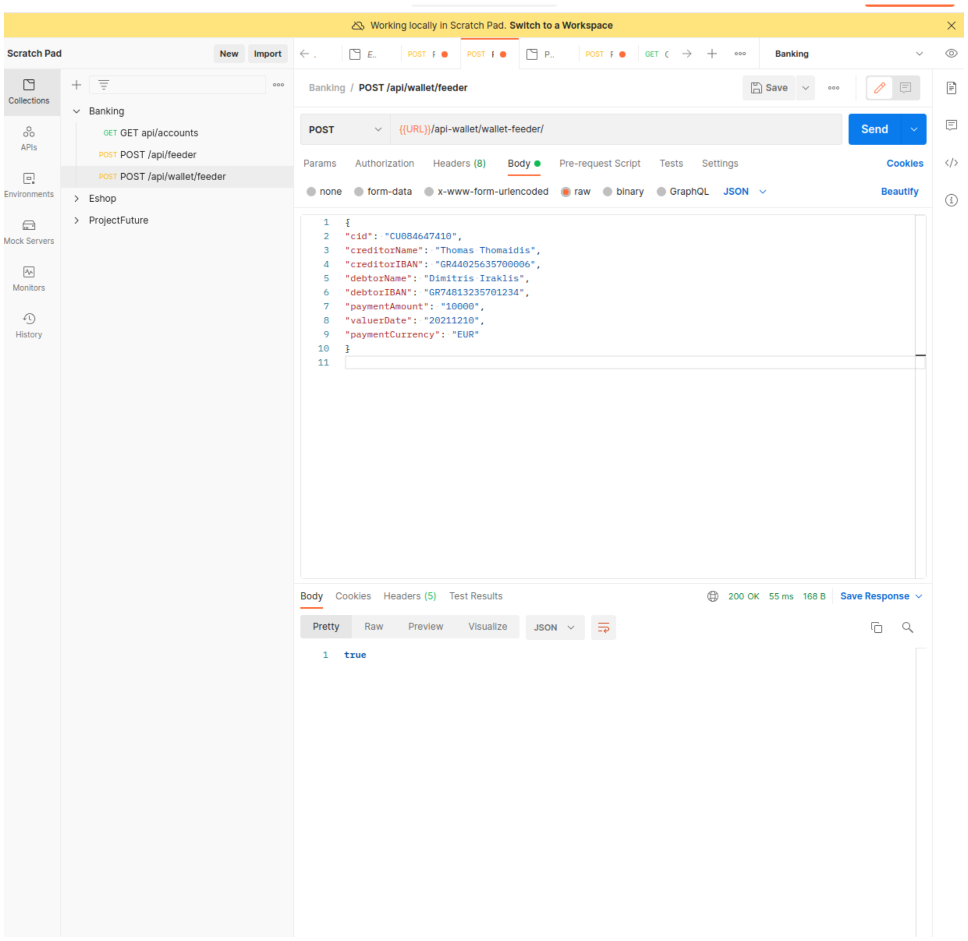
```
true
```

3. In the first case the payment is successful. To confirm this, we check the balances of the accounts.



```
[
    {
        "id": "9b4505fb-75f4-42dd-8d04-0377c3efcd18",
        "name": "Thomas Thomaidis",
        "iban": "GR4402563570006",
        "type": "NORMAL",
        "balance": 198.50
    },
    {
        "id": "c28b6671-b50a-411a-9a52-7ceaf9eb180c",
        "name": "Dimitris Iraklis",
        "iban": "GR74813235701234",
        "type": "NORMAL",
        "balance": 11898.50
    },
    {
        "id": "93433f47-f66e-437d-81f9-abf94aecadc5",
        "name": "Ioannis Danis",
        "iban": "GR42122635750096",
        "type": "NORMAL",
        "balance": 25000.00
    },
    {
        "id": "dceb1448-6021-4b5a-bd50-54c21327a73f",
        "name": "National Bank",
        "iban": "GR00000000000001",
        "type": "BANKER",
        "balance": 3.00
    }
]
```

4. Send through Postman a post of the second request for wallet bank transfer via JSON File described below.

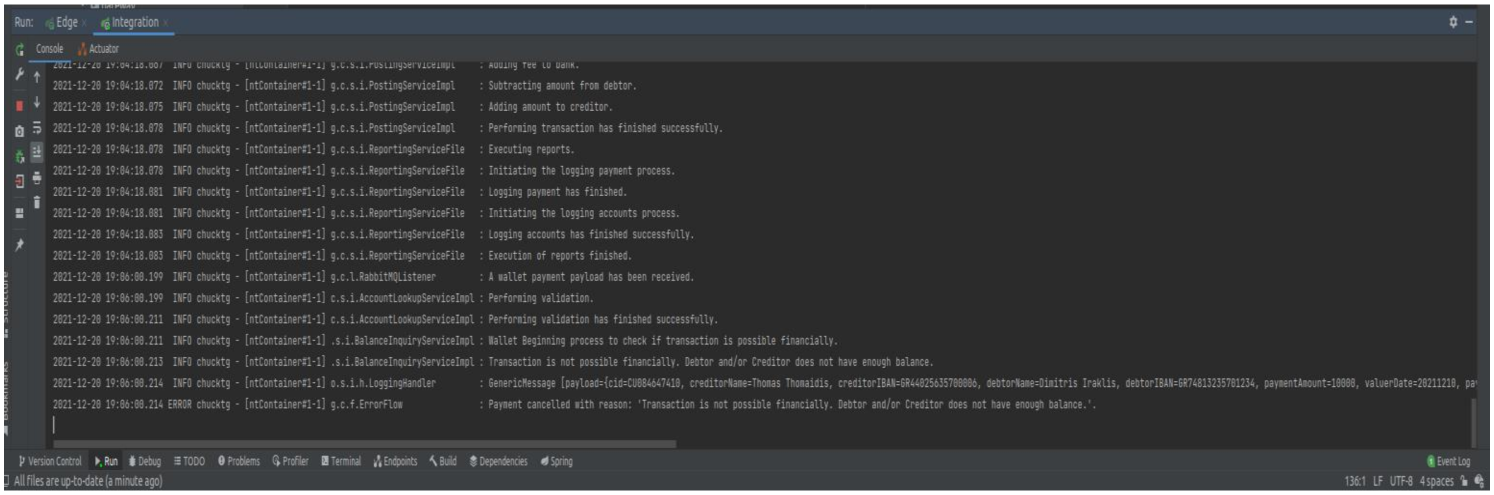


```
POST  {{URL}}/api-wallet/wallet-feeder/
```

```
{
"cid": "CU084647410",
"creditorName": "Thomas Thomaidis",
"creditorIBAN": "GR44025635700006",
"debtorName": "Dimitris Iraklis",
"debtorIBAN": "GR74813235701234",
"paymentAmount": "10000",
"valuerDate": "20211210",
"paymentCurrency": "EUR"
}
```

```
true
```

5. In this case the payment is unsuccessful because the creditor balance does not cover the fees. During the execution of the program the following message appears.



```
2021-12-20 19:04:18.067  INFO chucktg - [ntContainer#1-1] g.c.s.i.PostingServiceImpl    : Adding fee to bank.
2021-12-20 19:04:18.072  INFO chucktg - [ntContainer#1-1] g.c.s.i.PostingServiceImpl    : Subtracting amount from debtor.
2021-12-20 19:04:18.075  INFO chucktg - [ntContainer#1-1] g.c.s.i.PostingServiceImpl    : Adding amount to creditor.
2021-12-20 19:04:18.078  INFO chucktg - [ntContainer#1-1] g.c.s.i.PostingServiceImpl    : Performing transaction has finished successfully.
2021-12-20 19:04:18.078  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Executing reports.
2021-12-20 19:04:18.078  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Initiating the logging payment process.
2021-12-20 19:04:18.081  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Logging payment has finished.
2021-12-20 19:04:18.081  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Initiating the logging accounts process.
2021-12-20 19:04:18.083  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Logging accounts has finished successfully.
2021-12-20 19:04:18.083  INFO chucktg - [ntContainer#1-1] g.c.s.i.ReportingServiceFile  : Execution of reports finished.
2021-12-20 19:06:00.199  INFO chucktg - [ntContainer#1-1] g.c.l.RabbitMQListener        : A wallet payment payload has been received.
2021-12-20 19:06:00.199  INFO chucktg - [ntContainer#1-1] c.s.i.AccountLookupServiceImpl : Performing validation.
2021-12-20 19:06:00.211  INFO chucktg - [ntContainer#1-1] c.s.i.AccountLookupServiceImpl : Performing validation has finished successfully.
2021-12-20 19:06:00.211  INFO chucktg - [ntContainer#1-1] .s.i.BalanceInquiryServiceImpl : Wallet Beginning process to check if transaction is possible financially.
2021-12-20 19:06:00.213  INFO chucktg - [ntContainer#1-1] .s.i.BalanceInquiryServiceImpl : Transaction is not possible financially. Debtor and/or Creditor does not have enough balance.
2021-12-20 19:06:00.214  INFO chucktg - [ntContainer#1-1] o.s.i.h.LoggingHandler        : GenericMessage [payload={cid=CU084647410, creditorName=Thomas Thomaidis, creditorIBAN=GR44025635700006, debtorName=Dimitris Iraklis, debtorIBAN=GR74813235701234, paymentAmount=10000, valuerDate=20211210, pa
2021-12-20 19:06:00.214 ERROR chucktg - [ntContainer#1-1] g.c.f.ErrorFlow               : Payment cancelled with reason: 'Transaction is not possible financially. Debtor and/or Creditor does not have enough balance.'.
```