



Team 8 Project Report

Team Members

Αλίκη Χρήστου

Δημήτρης Σώτος

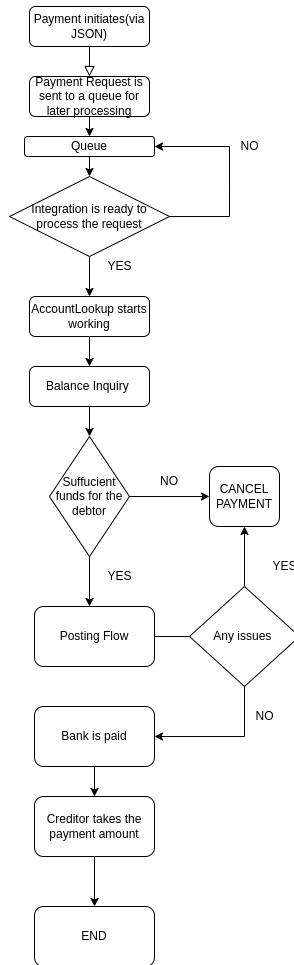
Κωνσταντίνος Τζιρβελάκης

Σάββας Σκουλίδης

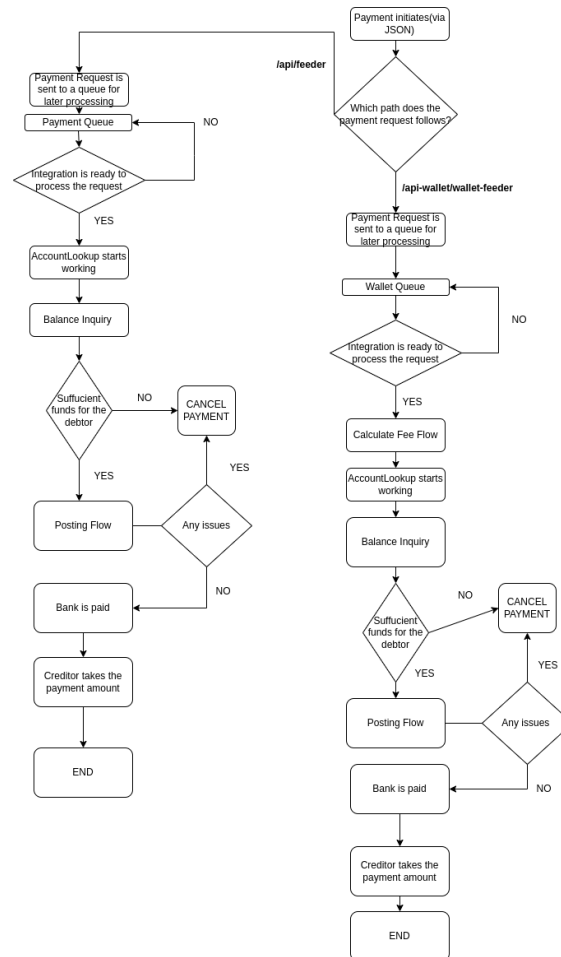
Σοφία Τερζοπούλου

Flow Diagrams

1. Diagram of business flow of the Payment System




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



RabbitMQ

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


Refreshed 2021-12-20 18:59:26
Refresh every 5 seconds

RabbitMQ 3.9.11
Erlang 24.2

Virtual host All
Cluster rabbit@0d83d125c728
User guest Log out

Overview
Connections
Channels
Exchanges

Queues
Admin

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

Add queue

[HTTP API](#)
[Server Docs](#)
[Tutorials](#)
[Community Support](#)
[Community Slack](#)
[Commercial Support](#)
[Plugins](#)
[GitHub](#)

[Changelog](#)

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/ Send

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

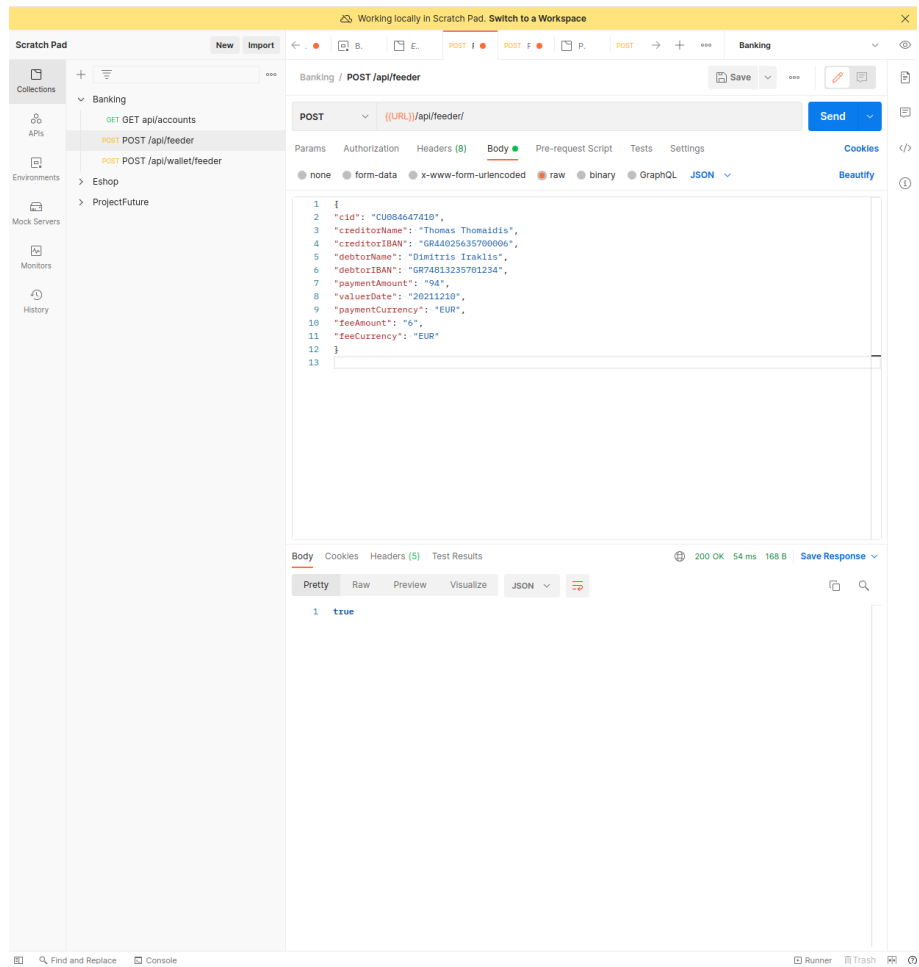
Body Cookies Headers (5) Test Results 200 OK 77 ms 685 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": "73477051-17f7-4898-9b4b-db1c12b5f193",
4     "name": "Thomas Thomaïdis",
5     "iban": "GR44025635700006",
6     "type": "NORMAL",
7     "balance": 100.00
8   },
9   {
10    "id": "65d84bb8-e717-459a-b369-473e1f593b83",
11    "name": "Dimitris Iraklis",
12    "iban": "GR74813235701234",
13    "type": "NORMAL",
14    "balance": 12000.00
15  },
16  {
17    "id": "6e96d8d3-b262-4868-96f8-e42f1ef351ed",
18    "name": "Ioannis Danis",
19    "iban": "GR42122635750096",
20    "type": "NORMAL",
21    "balance": 25000.00
22  },
23  {
24    "id": "cab314cb-29b0-402a-ac5a-07ee6a0ac009",
25    "name": "National Bank",
26    "iban": "GR00000000000001",
27    "type": "BANKER",
28    "balance": 0.00
29  }
30 }
```

Runner Trash

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



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

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

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

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 11 ms 685 B Save Response

Pretty Raw Preview Visualize JSON

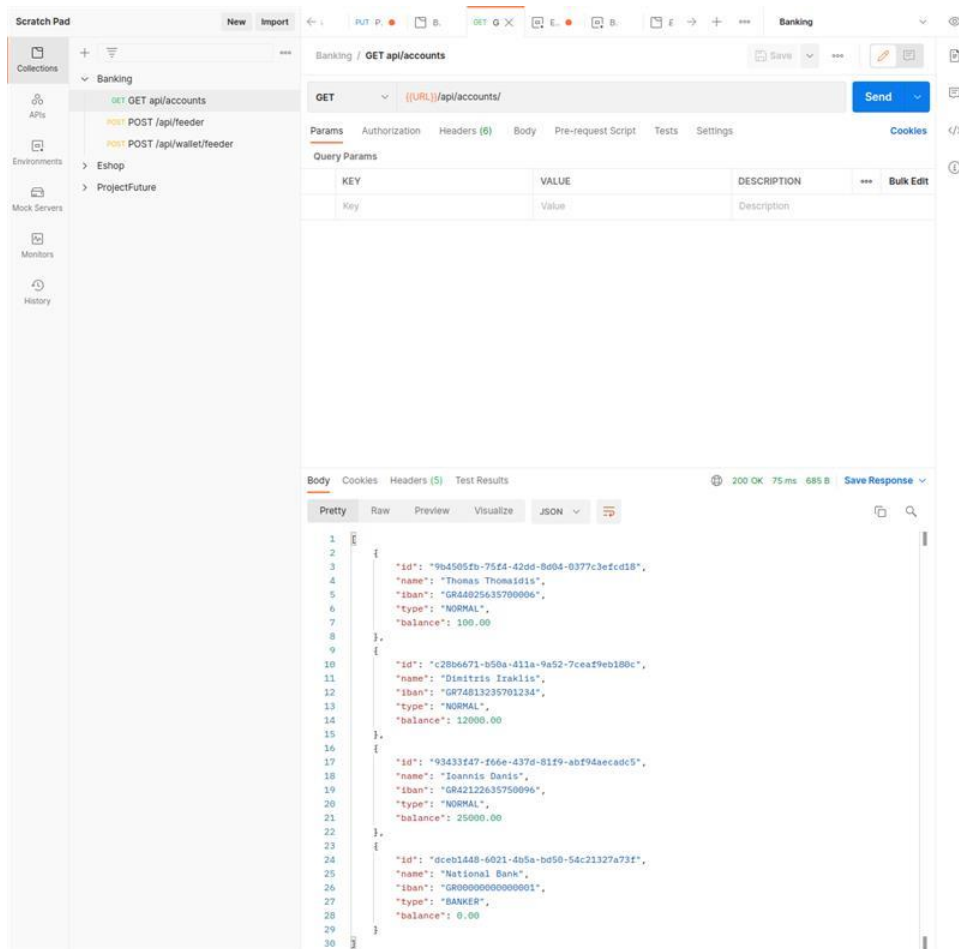
```
1 {
2   {
3     "id": "73477051-17f7-4898-9b4b-dblc12b5f193",
4     "name": "Thomas Thomaiddis",
5     "iban": "GR44025635700006",
6     "type": "NORMAL",
7     "balance": 194.00
8   },
9   {
10    "id": "65d84bb8-e717-459a-b369-473e1f593b83",
11    "name": "Dimitris Iraklis",
12    "iban": "GR74813235701234",
13    "type": "NORMAL",
14    "balance": 11900.00
15  },
16  {
17    "id": "6e96d8d3-b262-4868-96f8-e42f1ef351ed",
18    "name": "Ioannis Danis",
19    "iban": "GR42122635750096",
20    "type": "NORMAL",
21    "balance": 25000.00
22  },
23  {
24    "id": "cab314cb-29b0-402a-ac5a-07ee6a0ac009",
25    "name": "National Bank",
26    "iban": "GR00000000000001",
27    "type": "BANKER",
28    "balance": 6.00
29  }
30 }
```

Runner Trash

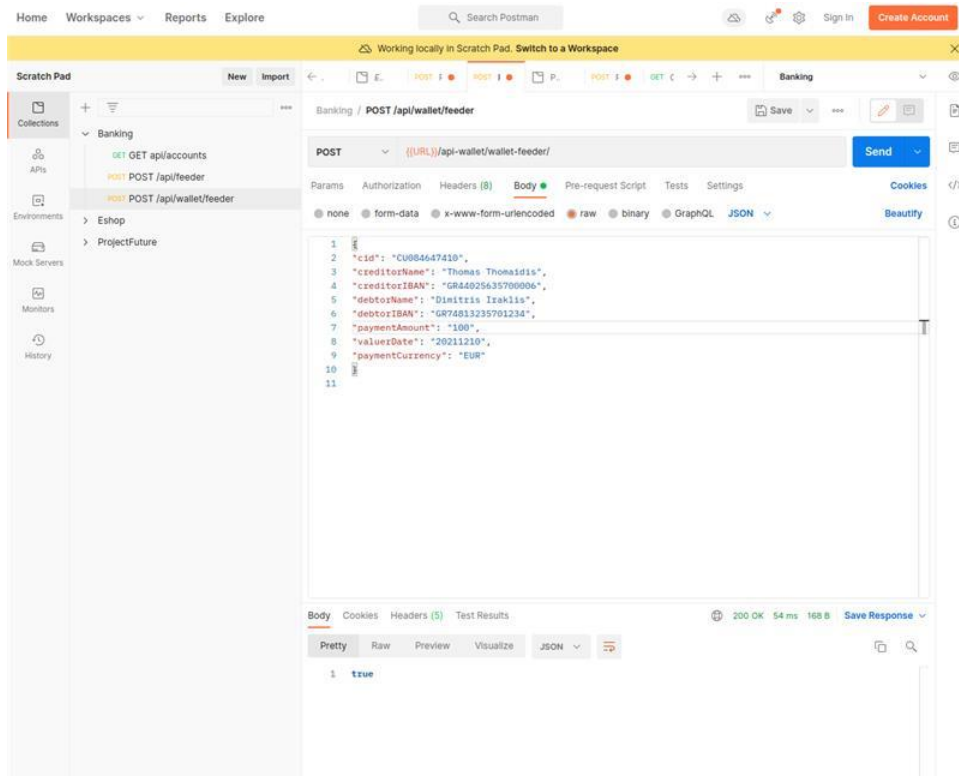
Case Study2: Successful Wallet Payment

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

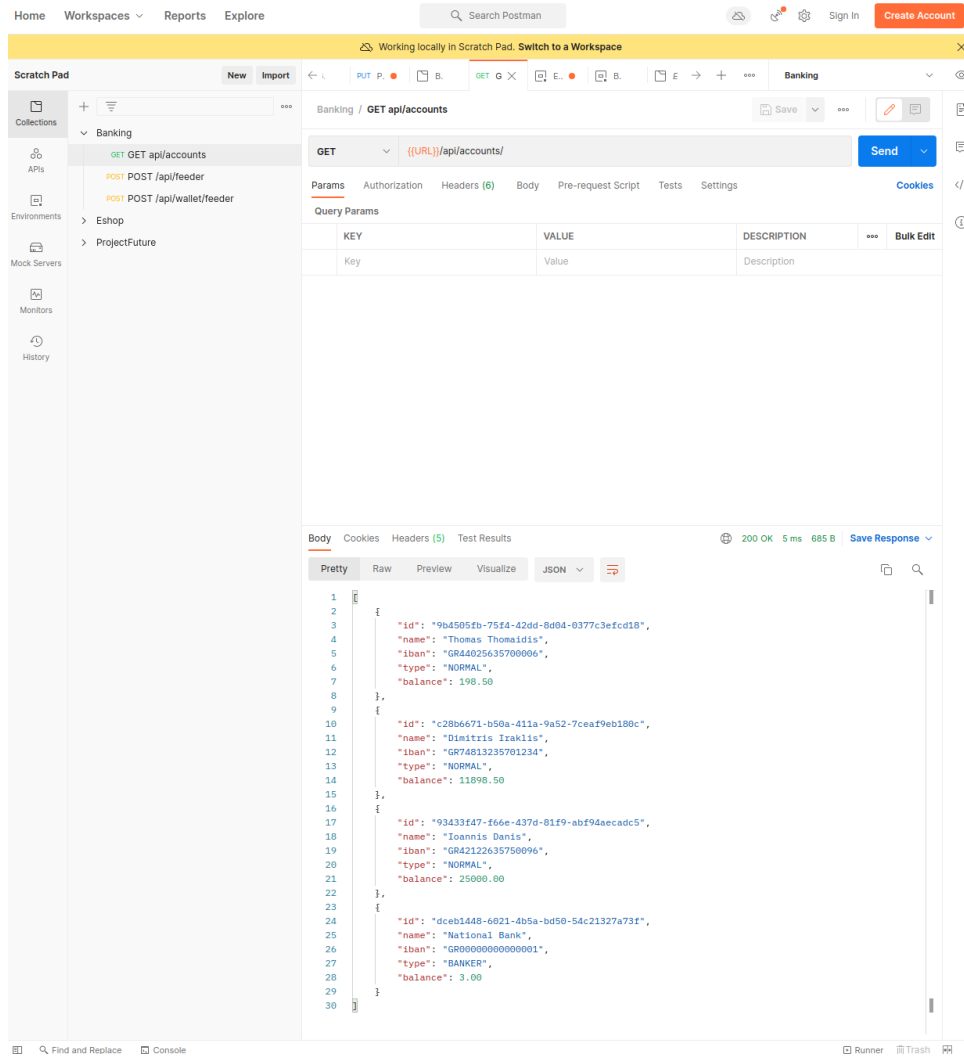
1. Check the accounts before the transaction (Api/accounts/)



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



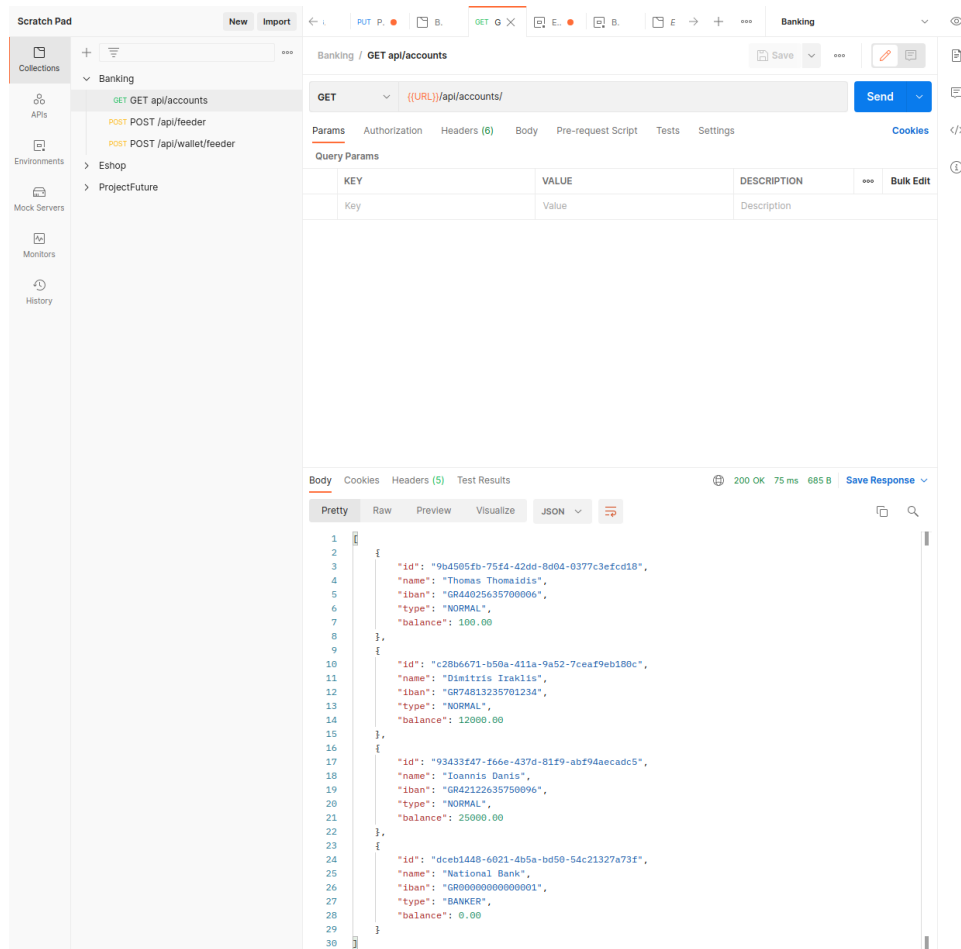
3. The amounts of the bank accounts changed, and the bank received the fee amount as described.



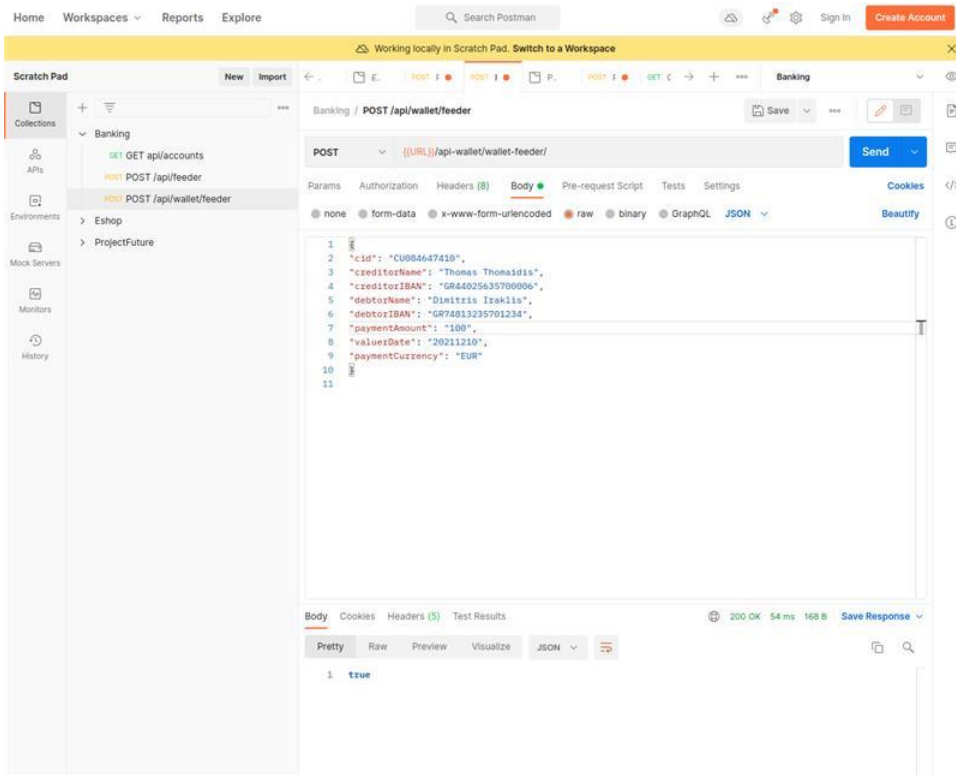
Case Study3: Unsuccessful Wallet Payment

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

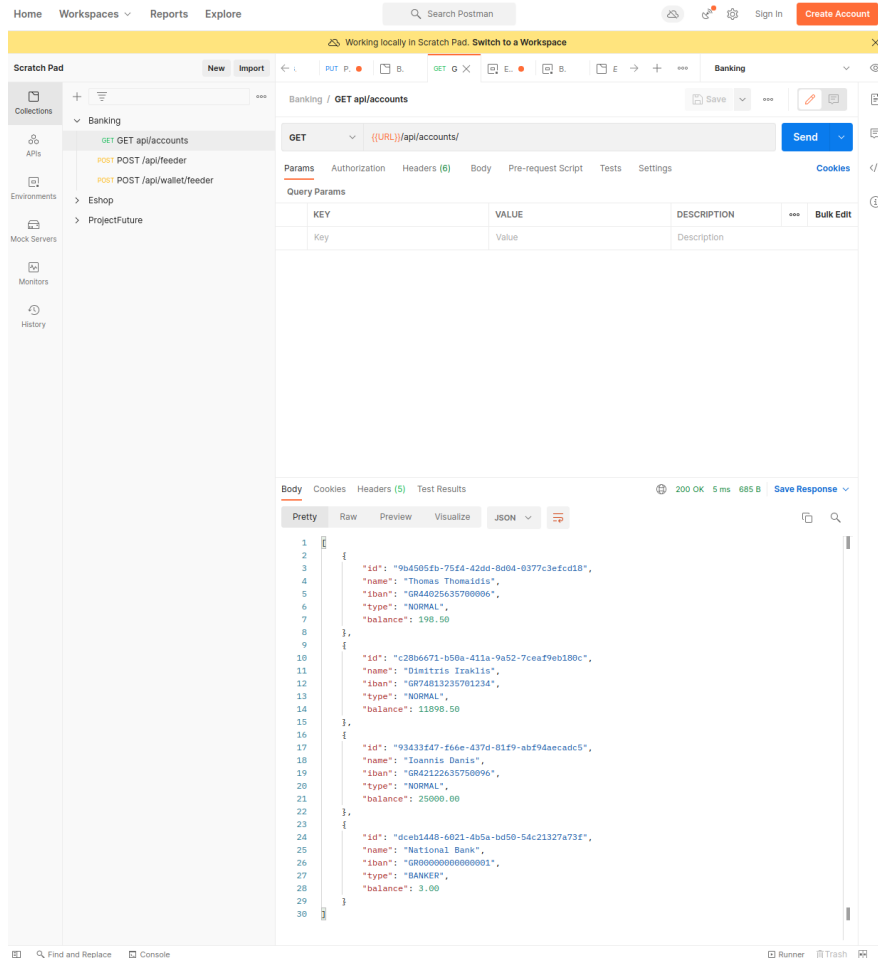
1. Check the accounts before the transaction (Api/accounts/)



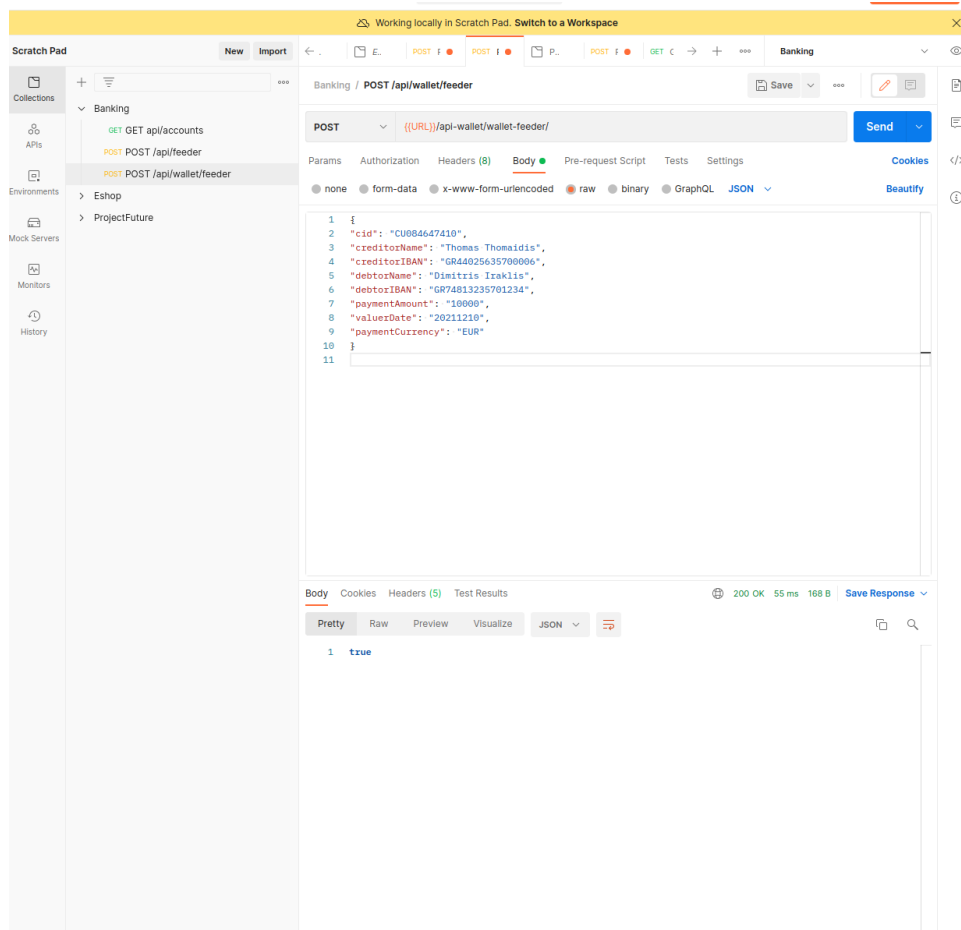
2. Send through Postman a post of a simple request for wallet bank transfer via JSON File described below.



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



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



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.

