

# FastAPI (0.1.0)

Download OpenAPI specification: [Download](#)

## auth

### Pre-registration form

Form to create temp user before verification of email. Used to check whenever account data is unique.  
Check UserTemp schema. Pass UserTemp body schema

REQUEST BODY SCHEMA: `application/json`

<code>email</code> <code>required</code>	string <email> (Email)
---------------------------------------------	------------------------

<code>phone_number</code> <code>required</code>	string (Phone Number)
----------------------------------------------------	-----------------------

<code>social_security</code> <code>required</code>	string (Social Security)
-------------------------------------------------------	--------------------------

### Responses

> **200** Returns positive status of pre-registration

— **400** Internal error accused by inprocessable data which crashed database or SMTP server

— **409** Conflict. The main unique data is already exists for another user

> **422** Validation Error

POST /auth/register

Request samples

Payload

Content type  
application/json

Copy

```
{
  "email": "user@example.com",
  "phone_number": "string",
  "social_security": "string"
}
```

Response samples

200

422

Content type  
application/json

Copy

```
null
```

Full account registration

Account being created with full provided information about user. Additionally to user information, verification code from email is passed. Check UserCreate schema. Pass UserCreate body schema

REQUEST BODY SCHEMA: application/json

first_name required	string (First Name)
last_name required	string (Last Name)
email required	string <email> (Email)

phone_number required	string (Phone Number)
date_of_birth required	string <date> (Date Of Birth)
social_security required	string (Social Security)
address required	string (Address)
city required	string (City)
state required	string (State)
post_code required	string (Post Code)
password required	string (Password)
verification_code required	string (Verification Code)

Responses

> **200** Returns positive status of registration and cookie for first account access

— **406** Bad verification code or this account is not needed for account

> **422** Validation Error

POST    /auth/verify-email

Request samples

Payload

Content type  
application/json

Copy

```
{
  "first_name": "string",
  "last_name": "string",
  "email": "user@example.com",
  "phone_number": "string",
  "date_of_birth": "2019-08-24",
  "social_security": "string",
  "address": "string",
  "city": "string",
  "state": "string",
  "post_code": "string",
  "password": "string",
  "verification_code": "string"
}
```

Response samples

200

422

Content type  
application/json

Copy

null

First step for login

First step for login to send verification code. Pass email and password fields of UserLogin body schema

REQUEST BODY SCHEMA: application/json

email required	string (Email)
password >	Password (string) or Password (null) (Password)
twofa_code >	Twofa Code (string) or Twofa Code (null) (Twofa Code)

Responses

- > 200 Returns positive status of first step
- 401 Account is not exists or provided data is incorrect
- > 422 Validation Error

POST /auth/2fa/request

Request samples

Payload

Content type  
application/json

Copy

```
{
  "email": "string",
  "password": "string",
  "twofa_code": "string"
}
```

Response samples

200

422

Content type  
application/json

Copy

```
null
```

Last step for login

Last step for login to end login process. Pass email and twofa\_code fields of UserLogin body schema

REQUEST BODY SCHEMA: application/json

email required	string (Email)
password >	Password (string) or Password (null) (Password)
twofa_code >	Twofa Code (string) or Twofa Code (null) (Twofa Code)

Responses

> 200 Returns positive status of login and cookie for account access

— 401 Account is not exists or provided data is incorrect

> 422 Validation Error

POST /auth/2fa/confirm

Request samples

Payload

Content type  
application/json

Copy

```
{
  "email": "string",
  "password": "string",
  "twofa_code": "string"
}
```

Response samples

200

422

Content type  
application/json

Copy

null

# Base user data

User must be logged into account to perform this option.

## COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

## Responses

- > **200** Returns base user data of user. Name and balance
- **401** Account is not exists, user not logged into account or provided data is incorrect
- > **422** Validation Error

GET /auth/me

## Response samples

200

422

Content type  
application/json

null

Copy

# Reset password request

Requesting reset password for account. To confirm it's actual owner pass UserTemp body schema

REQUEST BODY SCHEMA: application/json

email required	string <email> (Email)
phone_number required	string (Phone Number)
social_security required	string (Social Security)

Responses

- > 200 Returns positive status of sending reset link
- 401 Account is not exists or provided data is incorrect
- > 422 Validation Error

POST /auth/reset

Request samples

Payload

Content type  
application/json

Copy

```
{
  "email": "user@example.com",
  "phone_number": "string",
  "social_security": "string"
}
```

Response samples

200 422



Content type

application/json

Copy

null

# Reset password confirm

Confirmation of password resetting. Pass UserPasswordReset body schema

REQUEST BODY SCHEMA: application/json

token required	string (Token)
new_password required	string (New Password)

## Responses

- > 200 Returns positive status of process
- 401 Account is not requests password reset or token is incorrect
- > 422 Validation Error

PATCH /auth/reset

## Request samples

Payload

Content type

application/json

Copy

```
{
  "token": "string",
  "new_password": "string"
}
```

Response samples

200

422

Content type  
application/json

Copy

null

# Log out

User must be logged into account to perform this option.

## Responses

> 200 Deleting the cookie

POST /auth/logout

Response samples

200

Content type  
application/json

Copy

null

---

## miscallenious

### Cleanup unverified users routine

Endpoint created to cleanup users in 24h timespan. Used in routine

#### Responses

> **200** Each time is called returns amount of users deleted

DELETE /cleanup-unverified

#### Response samples

200

**Content type**

application/json

Copy

null

---

# card

## Get single card by 4 last numbers

User must be logged into account to perform this option. Provide into route of endpoint last 4 digits of card to get the card. Example: url/card/0443

### PATH PARAMETERS

---

<code>four_digits</code> <code>required</code>	string (Four Digits)
---------------------------------------------------	----------------------

---

### COOKIE PARAMETERS

---

<code>authorization</code>	string (Authorization)
----------------------------	------------------------

---

## Responses

> **200** Returns all card information

— **400** Internal error accused by inprocessable data which crashed database

— **401** Credential exception. User is not logged or cookie is corrupted

> **422** Validation Error

— **500** Function is not properly implemented, yet

GET /card/{four\_digits}

### Response samples

200

422

Content type  
application/json

Copy

null

# Card creation

User must be logged into account to perform this option. Creating card using information about user in database

## COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

## Responses

- > 200 Returns card data
- 400 Internal error accused by inprocessable data which crashed database
- 401 Credential exception. User is not logged or cookie is corrupted
- 406 User for who card will be created has no wallet
- > 422 Validation Error

POST /card/create

## Response samples

200

422

Content type  
application/json

null

## Deleting card

User must be logged into account to perform this option. Deleting card of user. Pass CardDelete body schema

### COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

### REQUEST BODY SCHEMA: application/json

card_number required	string (Card Number)
-------------------------	----------------------

## Responses

> 200

— 400 Internal error accused by inprocessable data which crashed database

— 401 Credential exception. User is not logged or cookie is corrupted

— 403 Card balance is positive. Cannot delete card with funds on it

— 406 User for who card will be deleted has no wallet

> 422 Validation Error

DELETE /card/delete

## Request samples

Payload

Content type

application/json

Copy

```
{
  "card_number": "string"
}
```

Response samples

200

422

Content type

application/json

Copy

```
null
```

# Transferring money from card to card

User must be logged into account to perform this option. Transferring money from card to card. Transfer operation will be added to history. Pass TransferRequest body schema

COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

REQUEST BODY SCHEMA: application/json

from_card_number required	string (From Card Number)
------------------------------	---------------------------

to_card_number required	string (To Card Number)
----------------------------	-------------------------

amount required	number (Amount) > 0 Amount must be positive
--------------------	------------------------------------------------

## Responses

> **200** Returns how much money was transfered

— **400** Internal error accused by inprocessable data which crashed database

— **401** Credential exception. User is not logged or cookie is corrupted

— **403** Sender has no funds to perform operation

— **406** Reciever/Sender for who transferring operation will be called has no wallet

> **422** Validation Error

POST /card/transfer

## Request samples

### Payload

#### Content type

application/json

Copy

```
{
  "from_card_number": "string",
  "to_card_number": "string",
  "amount": 0
}
```

## Response samples

**200**

**422**

#### Content type

application/json

Copy

null



# Get list of user's cards

User must be logged into account to perform this option. Getter for list of user cards. List contains of card data

## COOKIE PARAMETERS

---

authorization

string (Authorization)

---

## Responses

- > 200 Returns list of cards data
- 400 Internal error accused by inprocessable data which crashed database
- 401 Credential exception. User is not logged or cookie is corrupted
- 406 User for who this operation will be called has no wallet
- > 422 Validation Error

GET /card

## Response samples

200

422

Content type  
application/json

null

Copy

# Get transfer history

User must be logged into account to perform this option. Getter for list of transfers performed by user with n-card. Pass CardHistoryRequest body schema

COOKIE PARAMETERS

authorization      string (Authorization)

REQUEST BODY SCHEMA:    application/json

card\_number  
required      string (Card Number)

## Responses

- > 200 Return list of transfers
- 400 Internal error accused by inprocessable data which crashed database
- 401 Credential exception. User is not logged or cookie is corrupted
- 406 User for who this operation will be called has no wallet
- > 422 Validation Error

POST    /card/history

## Request samples

Payload

Content type  
application/json

Copy

```
{  
  "card_number": "string"
```

```
}
```

Response samples

200

422

Content type  
application/json

Copy

null

savings

Creation of saving account

User must be logged into account to perform this option. Setting goal for users desire. Pass Saving\_Account\_creation body scheme

COOKIE PARAMETERS

authorization      string (Authorization)

REQUEST BODY SCHEMA:    application/json

name  
required              string (Name)

goal  
required              number (Goal)    > 0  
Goal must be positive

Responses

> 200 Returns saving data on success

- **400** Internal error accused by inprocessable data which crashed database
- **401** Account is not exists, user not logged into account or provided data is incorrect
- **406** User for who this operation will be called has no wallet
- > **422** Validation Error

POST /savings/create

## Request samples

### Payload

**Content type**  
application/json

```
{  
  "name": "string",  
  "goal": 0  
}
```

Copy

## Response samples

**200**

**422**

**Content type**  
application/json

null

Copy

---

## Getter for savings

User must be logged into account to perform this option.

COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

Responses

- > 200 Returns list of Saving\_Account\_out. Check schema
- 400 Internal error accused by inprocessable data which crashed database
- 401 Account is not exists, user not logged into account or provided data is incorrect
- 406 User for who this operation will be called has no wallet or saving account
- > 422 Validation Error

GET /savings

Response samples

200

422

Content type  
application/json

Copy Expand all Collapse all

```
[
  - {
    "id": 0,
    "name": "string",
    "balance": 0,
    "goal": 0,
    "remain": 0
  }
]
```

# Topping up savings

User must be logged into account to perform this option. Used to topping up savings. Topping up is mirrored into transfer history. Pass Saving\_Account\_TopUp body scheme

## COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

## REQUEST BODY SCHEMA: application/json

amount required	number (Amount)
--------------------	-----------------

saving_account_id required	integer (Saving Account Id)
-------------------------------	-----------------------------

card_id required	integer (Card Id)
---------------------	-------------------

## Responses

- > 200
- 400 Internal error accused by inprocessable data which crashed database
- 401 Account is not exists, user not logged into account or provided data is incorrect
- 403 Forbidden wallet action. Savings are not existed or card which tried to pay with has no funds
- 406 User for who this operation will be called has no wallet or saving account
- > 422 Validation Error

POST /savings/topUp

## Request samples

Payload

Content type

application/json

Copy

```
{
  "amount": 0,
  "saving_account_id": 0,
  "card_id": 0
}
```

Response samples

200

422

Content type

application/json

Copy

```
null
```

# Decreasing savings

User must be logged into account to perform this option. Taking funds from savings to card. Pass Saving\_Account\_TopUp body scheme

COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

REQUEST BODY SCHEMA: application/json

amount required	number (Amount)
saving_account_id required	integer (Saving Account Id)
card_id required	integer (Card Id)

## Responses

> 200

— 400 Internal error accused by inprocessable data which crashed database

— 401 Account is not exists, user not logged into account or provided data is incorrect

— 403 Forbidden wallet action. Card is not exists or savings which tried to decrease of has no funds

— 406 User for who this operation will be called has no wallet or saving account

> 422 Validation Error

POST /savings/decrease

## Request samples

Payload

Content type

application/json

Copy

```
{
  "amount": 0,
  "saving_account_id": 0,
  "card_id": 0
}
```

## Response samples

200

422

Content type

application/json

Copy

null



# Deletes savings account

User must be logged into account to perform this option. Deletes savings account. Pass Saving\_Account\_Delete body schema

## COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

## REQUEST BODY SCHEMA: application/json

saving_account_id required	integer (Saving Account Id)
-------------------------------	-----------------------------

## Responses

- > 200 Returns id and status of deleted saving account
- 400 Internal error accused by inprocessable data which crashed database
- 401 Account is not exists, user not logged into account or provided data is incorrect
- 403 Forbidden wallet action. Savings has positive amount of funds or savings is not exists
- 406 User for who this operation will be called has no wallet
- > 422 Validation Error

DELETE /savings/delete

## Request samples

Payload

Content type  
application/json

Copy

```
{
  "saving_account_id": 0
}
```

Response samples

200

422

Content type  
application/json

Copy

null

bills

Bill creation

User must be logged into account to perform this option. Pass the BillCreate body schema to create bill in database

COOKIE PARAMETERS

authorization      string (Authorization)

REQUEST BODY SCHEMA:    application/json

name  
required              string (Name)

amount  
required              number (Amount)

due\_date  
required              string <date-time> (Due Date)

## Responses

> **200** Displaying the created bill

— **400** Internal error accused by inprocessable data which crashed database

— **401** Credential exception. User is not logged or cookie is corrupted

— **406** User for who bill will be created has no wallet

> **422** Validation Error

POST    /bills/create

### Request samples

Payload

Content type

application/json

Copy

```
{
  "name": "string",
  "amount": 0,
  "due_date": "2019-08-24T14:15:22Z"
}
```

### Response samples

**200**

**422**

Content type

application/json

Copy

null

# Bill list getter

User must be logged into account to perform this option. Returns the list of bills which user created earlier. Displays current values of each user's bills

## COOKIE PARAMETERS

authorization	string (Authorization)
---------------	------------------------

## Responses

- > **200** List of Bill. Check BillOut schema
- **400** Internal error accused by inprocessable data which crashed database
- **401** Credential exception. User is not logged or cookie is corrupted
- **406** User for who bills will be shown has no wallet or actual bills
- > **422** Validation Error

GET /bills

## Response samples

200

422

Content type  
application/json

Copy Expand all Collapse all

```
[
  - {
    "id": 0,
    "name": "string",
    "amount": 0,
    "due_date": "2019-08-24T14:15:22Z",
    "paid": true
  }
]
```

---

## Paying of the bill

User must be logged into account to perform this option. Bill paying is replicated into transfer history.  
Pass the BillPay body schema

### COOKIE PARAMETERS

---

authorization	string (Authorization)
---------------	------------------------

---

### REQUEST BODY SCHEMA: application/json

---

bill_id required	integer (Bill Id)
---------------------	-------------------

---

card_number required	string (Card Number)
-------------------------	----------------------

---

## Responses

> **200** Returns feedback information about card balance

— **400** Internal error accused by inprocessable data which crashed database

— **401** Credential exception. User is not logged or cookie is corrupted

— **403** Forbidden wallet action. Bill is paid off or card which tried to pay with has no funds

— **406** User for who bill will be paid has no wallet, card to pay or actual bills

> 422 Validation Error

POST /bills/pay

Request samples

Payload

Content type  
application/json

```
{
  "bill_id": 0,
  "card_number": "string"
}
```

Copy

Response samples

200

422

Content type  
application/json

null

Copy

Read Root

Responses

> 200 Successful Response

GET /

Response samples

200

Content type  
application/json

Copy

null