

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнил:

Селезнев Илья

Группа К33412

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

В рамках данной лабораторной работы Вам предложено выбрать один из нескольких вариантов. Выбранный вариант останется единым на весь курс и будет использоваться в последующих лабораторных работах.

По выбранному варианту необходимо будет реализовать RESTful API средствами express + typescript (используя ранее написанный boilerplate).

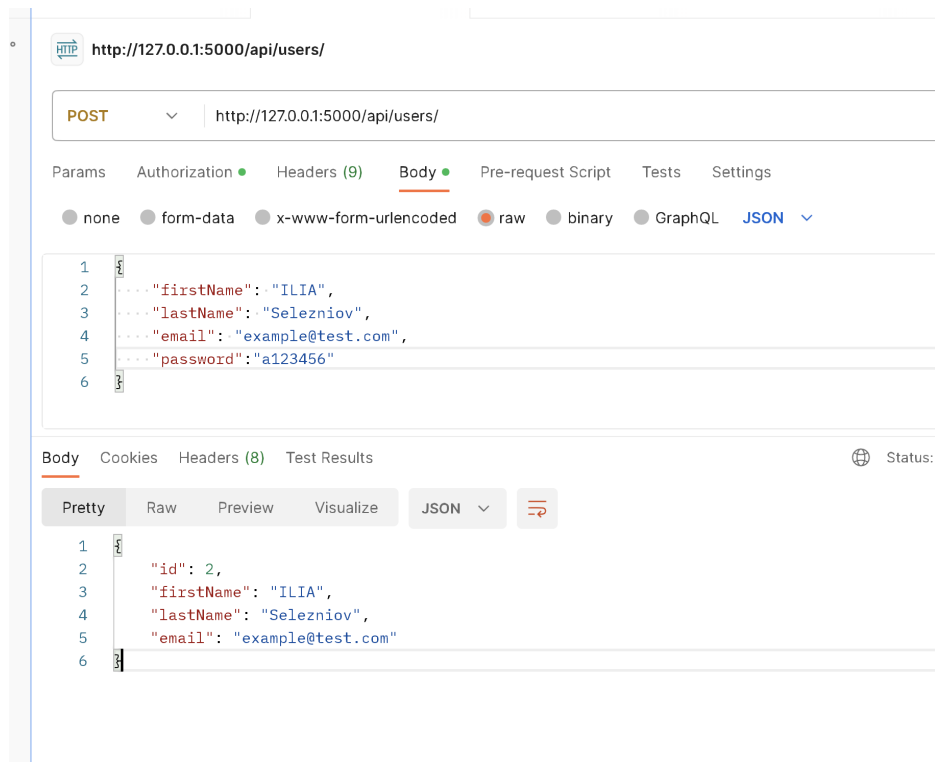
Выбранный вариант: 5. Сайт криптобиржи

- Вход
- Регистрация
- Портфель пользователя с указанием различных криптовалют и их количеством
- Графики роста криптовалют
- Поиск по криптовалютам с возможностью фильтрации по дате добавления на биржу

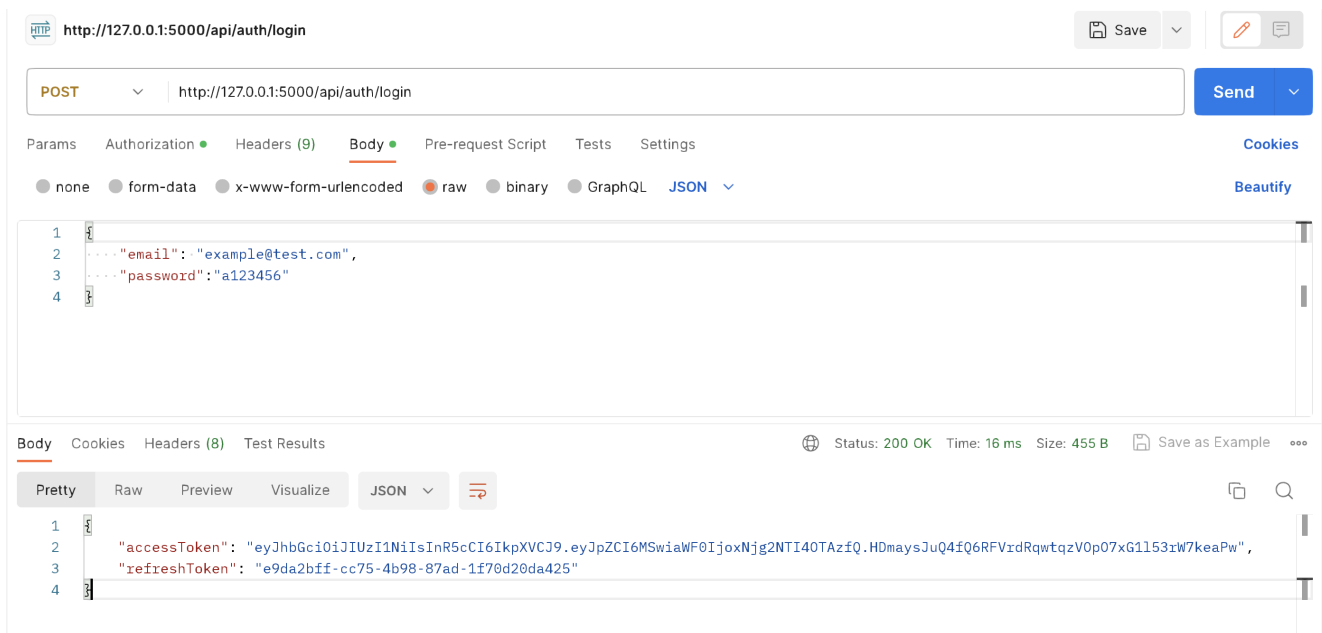
Ход работы

Пример работы API:

Создание пользователя:



Вход:



Получение профиля:

HTTP <http://127.0.0.1:5000/api/users/me> Save

GET <http://127.0.0.1:5000/api/users/me>

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Type Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Token `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`

Body Cookies Headers (8) Test Results Status: 200 OK Time: 11 ms Size: 344 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "firstName": "ILIA",
4   "lastName": "Seleznirov",
5   "email": "example@test.com"
6 }
```

Создание портфеля:

HTTP <http://127.0.0.1:5000/api/wallets> Save Send

POST <http://127.0.0.1:5000/api/wallets>

Params Authorization Headers (9) Body Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

[Cookies](#) [Beautify](#)

```
1 {
2   "name": "myWallet"
3 }
```

Body Cookies Headers (8) Test Results Status: 201 Created Time: 13 ms Size: 400 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "balance": 0,
3   "id": 1,
4   "name": "myWallet",
5   "userId": 1,
6   "updatedAt": "2023-06-12T00:18:19.774Z",
7   "createdAt": "2023-06-12T00:18:19.774Z"
8 }
```

Добавление денег в портфель:

HTTP http://127.0.0.1:5000/api/wallets/1/add_money Save

POST http://127.0.0.1:5000/api/wallets/1/add_money Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "amount": "1000"
3 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 23 ms Size: 336 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "myWallet",
4   "balance": 1000,
5   "userId": 1,
6   "coinWallets": []
7 }
```

Покупка криптовалюты:

HTTP http://127.0.0.1:5000/api/wallets/1/buy_coin Save

POST http://127.0.0.1:5000/api/wallets/1/buy_coin Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "ticker": "btc",
3   ... "amount": 5
4 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 272 ms Size: 385 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "myWallet",
4   "balance": 999999916834,
5   "userId": 1,
6   "coinWallets": [
7     {
8       "amount": 5,
9       "walletId": 1,
10      "coinId": "btc"
11     }
12   ]
13 }
```

Продажа криптовалюты:

HTTP http://127.0.0.1:5000/api/wallets/1/sell_coin Save

POST http://127.0.0.1:5000/api/wallets/1/sell_coin Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "ticker": "btc",
3   ... "amount": 3
4 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 261 ms Size: 385 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "myWallet",
4   "balance": 99999967333,
5   "userId": 1,
6   "coinWallets": [
7     {
8       "amount": 2,
9       "walletId": 1,
10      "coinId": "btc"
11     }
12   ]
13 }
```

Продажа всего объема конкретной криптовалюты:

HTTP http://127.0.0.1:5000/api/wallets/1/sell_all_coin Save

POST http://127.0.0.1:5000/api/wallets/1/sell_all_coin Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ... "ticker": "btc"
3 }
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 238 ms Size: 345 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "name": "myWallet",
4   "balance": 1000000000999,
5   "userId": 1,
6   "coinWallets": []
7 }
```

Вывод денег с биржи:

HTTP http://127.0.0.1:5000/api/wallets/1/withdraw_money Save

POST http://127.0.0.1:5000/api/wallets/1/withdraw_money Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {}
2 {
3   "amount": "1000000"
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 20 ms Size: 344 B Save as Example

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "id": 1,
3   "name": "myWallet",
4   "balance": 999999000999,
5   "userId": 1,
6   "coinWallets": []
7 }
```

Список доступных криптовалют:

HTTP <http://127.0.0.1:5000/api/coins> Save </>

GET <http://127.0.0.1:5000/api/coins> Send

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

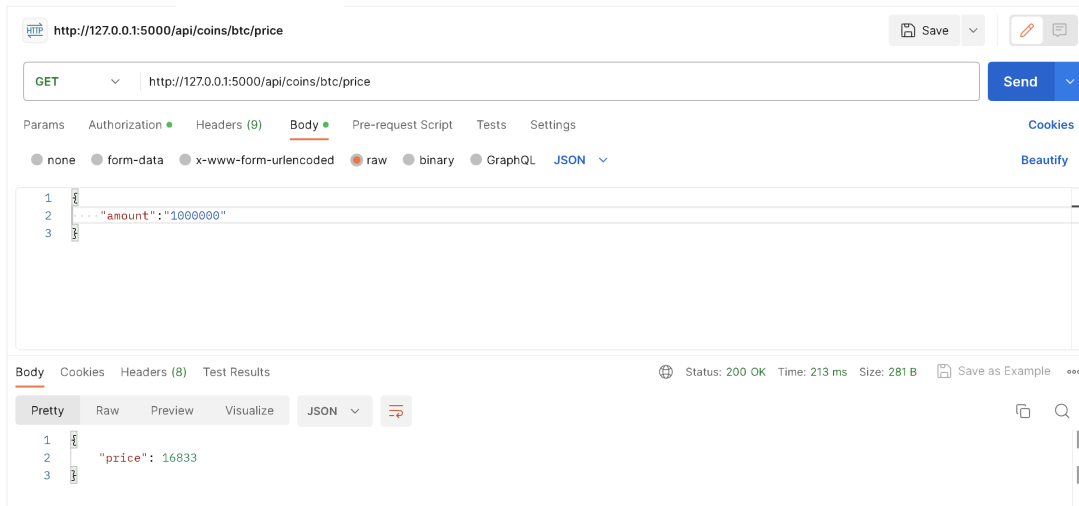
```
1 {}
2 {
3   "amount": "1000000"
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 7 ms Size: 319 B Save as Example

Pretty Raw Preview Visualize **JSON**

```
1 [
2   {
3     "ticker": "btc",
4     "name": "bitcoin",
5     "createdAt": null
6   }
7 ]
```

Цена криптовалюты в данный момент (используется API api.cryptowat.ch и данные биржи FTX):



Вывод

Я реализовал REST API криптобиржи с использованием фреймворка Express, ORM sequelize, Axios и другими библиотеками