

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 2

Выполнила:

Афанасьева Ирина Максимовна

Группа:

К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задание

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

Ход работы

Выбранный вариант: 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

Создание начальных значений для валют:

```
PS C:\Users\Irina\Desktop\Семинары\Бекендеки\LW2> npx sequelize-cli db:seed:all

Sequelize CLI [Node: 18.15.0, CLI: 6.4.1, ORM: 6.20.1]

Loaded configuration file "src\configs\sequelize.js".
Using environment "development".
== 20220609203417-coins: migrating =====
== 20220609203417-coins: migrated (0.229s)

PS C:\Users\Irina\Desktop\Семинары\Бекендеки\LW2> 
```

Пример работы API:

Создание пользователя:

The screenshot shows a REST client interface with a POST request to `http://localhost:5000/api/users`. The 'Body' tab is selected, and the request body is a JSON object: `{ "firstName": "Irina", "lastName": "Afanasyeva", "email": "multifructina@gmail.ru", "password": "5snemELX" }`. Below the request, the 'Body' tab of the response is shown, displaying a JSON object: `{ "id": 2, "firstName": "Irina", "lastName": "Afanasyeva", "email": "multifructina@gmail.ru" }`.

```
POST http://localhost:5000/api/users
```

Params Headers Body

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

```
1 { "firstName": "Irina", "lastName": "Afanasyeva", "email": "multifructina@gmail.ru", "password": "5snemELX" }
```

Body Headers (8)

Pretty Raw Preview JSON

```
1 {
2   "id": 2,
3   "firstName": "Irina",
4   "lastName": "Afanasyeva",
5   "email": "multifructina@gmail.ru"
6 }
```

Вход:

The screenshot shows a REST client interface with a POST request to `http://localhost:5000/api/auth/login`. The 'Body' tab is selected, and the request body is a JSON object: `{ "email": "multifructina@gmail.ru", "password": "5snemELX" }`. Below the request, the 'Body' tab of the response is shown, displaying a JSON object with access and refresh tokens: `{ "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiYW90IjoxNjg2MzM3MTM2fQ.LAYzakleUt4Va98smVh5jn15faHjmc0q0sw8tP1BPrg", "refreshToken": "4799fba7-b9c3-48e6-9e31-606585a5369e" }`. The status bar at the bottom indicates a 200 OK response with a time of 171 ms and a size of 455 B.

```
POST http://localhost:5000/api/auth/login
```

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1 { "email": "multifructina@gmail.ru", "password": "5snemELX" }
```

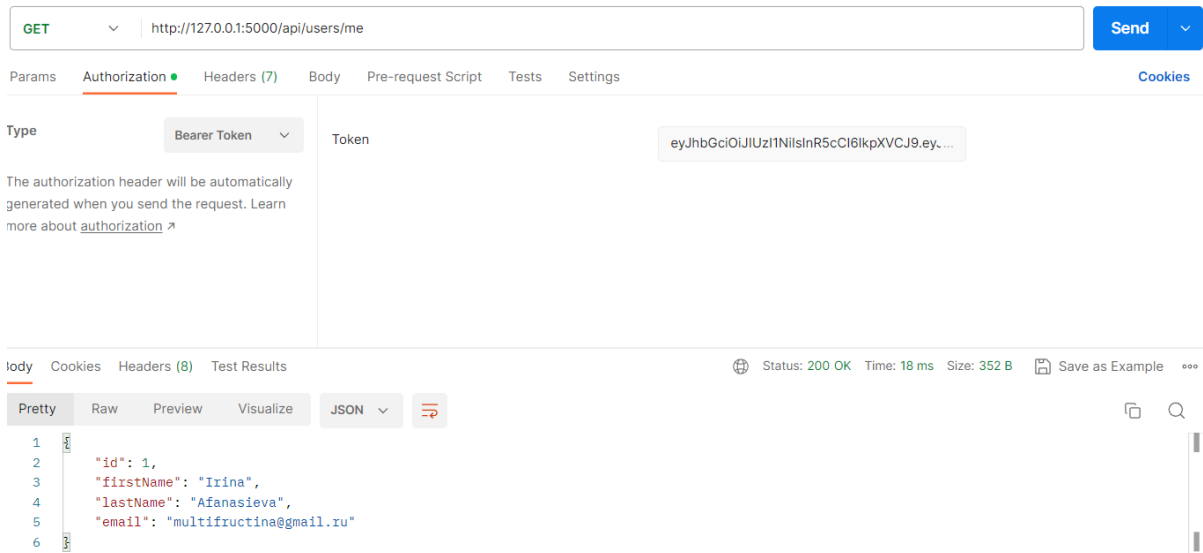
Body Cookies Headers (8) Test Results

Status: 200 OK Time: 171 ms Size: 455 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiYW90IjoxNjg2MzM3MTM2fQ.LAYzakleUt4Va98smVh5jn15faHjmc0q0sw8tP1BPrg",
3   "refreshToken": "4799fba7-b9c3-48e6-9e31-606585a5369e"
4 }
```

Получение профиля:



GET ▼ http://127.0.0.1:5000/api/users/me Send ▼

Params **Authorization** ● Headers (7) Body Pre-request Script Tests Settings Cookies

Type Bearer Token ▼ Token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

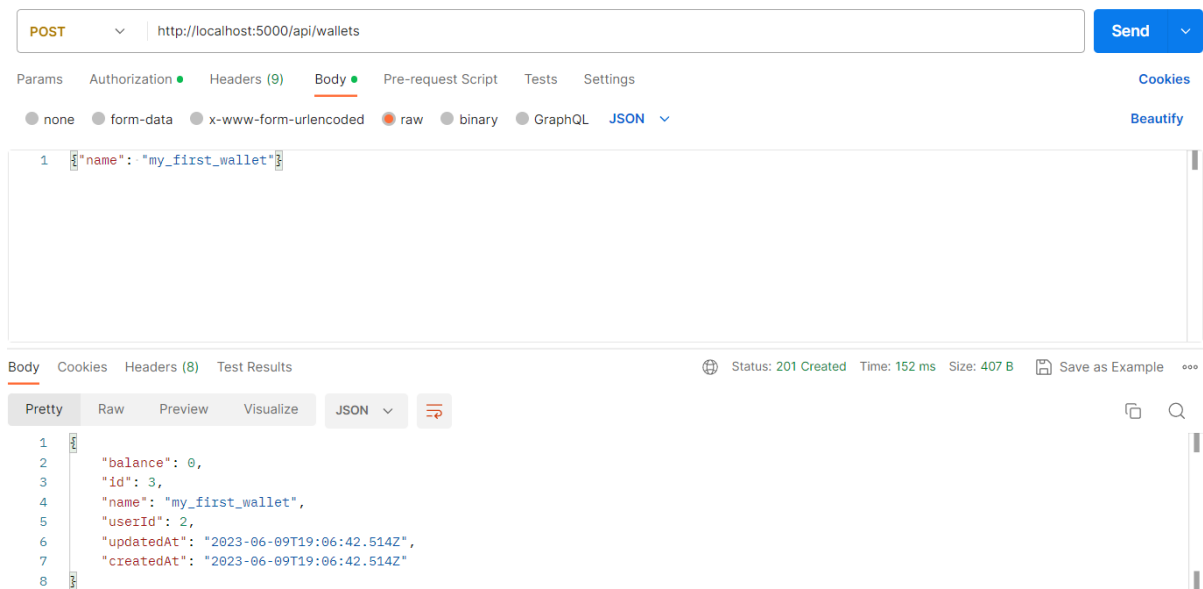
The authorization header will be automatically generated when you send the request. Learn more about [authorization](#) ↗

body Cookies Headers (8) Test Results Status: 200 OK Time: 18 ms Size: 352 B Save as Example

Pretty Raw Preview Visualize JSON ▼ ⋮

```
1 {
2   "id": 1,
3   "firstName": "Irina",
4   "lastName": "Afanasieva",
5   "email": "multifructina@gmail.ru"
6 }
```

Создание портфеля:



POST ▼ http://localhost:5000/api/wallets 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 { "name": "my_first_wallet" }
```

body Cookies Headers (8) Test Results Status: 201 Created Time: 152 ms Size: 407 B Save as Example

Pretty Raw Preview Visualize JSON ▼ ⋮

```
1 {
2   "balance": 0,
3   "id": 3,
4   "name": "my_first_wallet",
5   "userId": 2,
6   "updatedAt": "2023-06-09T19:06:42.514Z",
7   "createdAt": "2023-06-09T19:06:42.514Z"
8 }
```

Добавление денег в портфель:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:5000/api/wallets/3/add_money
- Body:**

```
{ "amount": 1000000 }
```
- Response:**

```
{ "id": 3, "name": "my_first_wallet", "balance": 1000000, "userId": 2, "coinWallets": [] }
```
- Status:** 200 OK
- Time:** 207 ms
- Size:** 346 B

Покупка криптовалюты:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:5000/api/wallets/3/buy_coin
- Body:**

```
{ "ticker": "btc", "amount": "5" }
```
- Response:**

```
{ "id": 3, "name": "my_first_wallet", "balance": 915835, "userId": 2, "coinWallets": [ { "amount": 5, "walletId": 3, "coinId": "btc" } ] }
```
- Status:** 200 OK
- Time:** 945 ms
- Size:** 386 B

Продажа криптовалюты:

POST http://localhost:5000/api/wallets/3/sell_coin

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

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

```
1 [{"ticker": "btc", "amount": 3}]
```

body Cookies Headers (8) Test Results Status: 200 OK Time: 552 ms Size: 386 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 [{"id": 3,
2   "name": "my_first_wallet",
3   "balance": 966334,
4   "userId": 2,
5   "coinWallets": [
6     {
7       "amount": 2,
8       "walletId": 3,
9       "coinId": "btc"
10    }
11  ]
12 }
13 ]
```

Вывод денег с биржи:

POST http://localhost:5000/api/wallets/3/withdraw_money

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

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

```
1 [{"amount": 960000}]
```

body Cookies Headers (8) Test Results Status: 200 OK Time: 207 ms Size: 384 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 [{"id": 3,
2   "name": "my_first_wallet",
3   "balance": 6334,
4   "userId": 2,
5   "coinWallets": [
6     {
7       "amount": 2,
8       "walletId": 3,
9       "coinId": "btc"
10    }
11  ]
12 }
13 ]
```

Список доступных криптовалют:

GET

http://localhost:5000/api/coins

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 11 msSize: 788 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1 {
2   {
3     "ticker": "btc",
4     "name": "Bitcoin",
5     "createdAt": "2012-01-01T00:00:00.000Z"
6   },
7   {
8     "ticker": "eth",
9     "name": "Ethereum",
10    "createdAt": "2018-03-05T00:00:00.000Z"
11  },
12  {
13    "ticker": "doge",
14    "name": "Dogecoin",
15    "createdAt": "2013-12-01T00:00:00.000Z"
16  }
```

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

GET

http://localhost:5000/api/coins/btc/price

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (8)Test Results

Status: 200 OKTime: 306 msSize: 281 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1 {
2   "price": 16833
3 }
```

Вывод

В ходе работы был реализован RESTful API для сайта криптобиржи с использованием фреймворка Express, ORM Sequelize, Axios и другими библиотеками.