САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронд-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Хайрнасов Андрей

К33402

Проверил:
Добряков Д. И.

Санкт-Петербург

2022 г.

Задача: нужно привязать то, что Вы делали в ЛР1 к внешнему API средствами fetch/axios/xhr.

Ход работы:

Вход

```
let url = 'http://localhost:3000/users'

async function login(event) {
    event.preventDefault()

    const inputs = Array.from(event.target.querySelectorAll('.form-control'))

    const credentials = {}

    for (const input of inputs) {
        credentials[input.name] = input.value
    }

    const response = await fetch(url)

    const responseJson = await response.json()
    const amount = responseJson.length

    for (let i = 0; i < amount; i++) {
        let {id, email, password} = responseJson[i]
        if (email === credentials.email && password === credentials.password) {

            let newUser = {
                'id': id,
                'email': email,
                'password': JSON.stringify(password),
            }

            localStorage.setItem('user', JSON.stringify(newUser))
            window.location.href = 'http://localhost:63342/frontLabs!/personal.html'
            break
        }
    }
}

document.addEventListener('DOMContentLoaded', () => checkAuth())
```

Проверка авторизации

```
function redirect(event) {
    event.preventDefault()
    window.location.href = 'http://localhost:63342/frontLabs!/register.html'
}


function checkAuth() {
    if (localStorage.user) {
        window.location.href = 'http://localhost:63342/frontLabs!/personal.html'
    }
}
```

Регистрация

```
async function register(event) {
    event.preventDefault()

    const inputs = Array.from(event.target.querySelectorAll('.form-control'))

    const credentials = {}

    for (const input of inputs) {

        credentials[input.name] = input.value
    }

    credentials.coins = []

    const response = await fetch('http://localhost:3000/users', {
            method: 'POST',
            body: JSON.stringify(credentials),
            headers: {
                'Content-Type': 'application/json',
            }
        }
    )

    const responseJson = await response.json()

    const {id, email, password} = responseJson

    let newUser = {
        'id': id,
        'email': email,
        'password': JSON.stringify(password),
    }

    localStorage.setItem('user', JSON.stringify(newUser))

    window.location.href = 'http://localhost:63342/frontLabs!/personal.html'
}
```

Создание элемента монеты

```
function getCoinHtml({image, name, current_price, symbol}, amount) {
    return `
        <div style="...">
            <div class="dlogo-item p-0" style="...">
                <img src="${image}" alt="" style="..." class="rounded-circle"/>
                <div class="">
                    <p class="fw-bold m-0">${name}</p>
                    <p class="text-muted m-0">${symbol.toUpperCase()}</p>
                </div>
            </div>
            <div class="d-flex align-items-center" style="...">
                <p class="fw-normal mb-0">RUB ${current_price}</p>
            </div>
            <div class="amount d-flex align-items-center" style="...">
                ${amount}
            </div>
            <div class="d-flex align-items-center" style="...">
                <span class="total">
                RUB ${(amount * current_price).toFixed(2)}
                </span>
            </div>
            <div class="btn-wrap" style="...">
                <div class="d-flex justify-content-center w-50" >
                    <button type="button" class="btn btn-primary btn-link rounded-pill text-white text-decoration-none" style="..." value="${symbol}" onclick="buyCoin(event)">
                        Купить
                    </button>
                </div>
                <div class="d-flex justify-content-center w-50" >
                    <button type="button" class="btn btn-danger btn-link rounded-pill text-white text-decoration-none" style="..." value="${symbol}" onclick="sellCoin(event)">
                        Продать
                    </button>
                </div>
            </div>
        </div>
    `
}
```

Рендер монет

```
async function loadCoins(query) {
    document.querySelector('.coin-wrapper').innerHTML = "";

    let currentUser = JSON.parse(localStorage.getItem('user'));

    const user = await fetch(`http://localhost:3000/users?id=${currentUser.id}`);
    const userJson = await user.json();

    const {coins} = userJson[0];

    if (query === undefined) {
        query = url;
    }

    const response = await fetch(query);

    data = await response.json();
    currCoins = coins.length;

    let renderQuery = [];

    if (coins.length !== null) {
        for (let i = 0; i < data.length; i++) {
            for (let j = 0; j < currCoins; j++) {
                if (coins[j].id === data[i].id) {
                    amount.push(coins[j].amount);
                    renderQuery.push(data[i])
                }
            }
        }
    }

    if (coins.length !== null) {
        for (let k = 0; k < coins.length; k++) {
            document.querySelector('.coin-wrapper').innerHTML += getCoinHtml(renderQuery[k], amount[k])
        }

        let money = document.querySelector('.balance')
        money.innerHTML = balance(coins)
    }
}
```

Покупка монет

```
async function buyCoin(event) {
    checkAuth()

    const symbol = event.target.value;
    let url = `http://localhost:3000/currency?symbol=${symbol}`;
    const response = await fetch(url);
    data = await response.json();
    const {id, current_price} = data[0];
    let coin = {
        'id': id,
        'price': current_price,
        'amount': 0,
    }

    let amount;
    while (true) {
        amount = prompt("Сколько хотите купить?")
        if (amount === null) {
            break
        } else if (isNaN(amount)) {
            alert("Введите число")
        } else if (amount <= 0) {
            alert("Количество должно быть положительным")
        } else {
            break
        }
    }
    if (amount !== null) {
        amount = parseFloat(amount)
        coin.amount = amount;
    }

    let currentUser = JSON.parse(localStorage.getItem('user'));
    const user = await fetch(`http://localhost:3000/users?id=${currentUser.id}`);
    const userJson = await user.json();
    let dup = -1;
    let i = 0;
    while (i < userJson[0].coins.length) {
        if (userJson[0].coins[i].id === coin.id) {
            dup = i;
        }
        i++;
    }

    if (dup === -1) {
        userJson[0].coins.push(coin)
    } else {
        userJson[0].coins[dup].amount += amount;
    }

    await fetch(`http://localhost:3000/users/${currentUser.id}`, {
        method: 'PUT',
        body: JSON.stringify(userJson[0]),
        headers: {
            'Content-Type': 'application/json',
        }
    })
```

Продажа монет

```
async function sellCoin(event) {
    checkAuth()

    let currentUser = JSON.parse(localStorage.getItem('user'));
    const user = await fetch(`http://localhost:3000/users?id=${currentUser.id}`);
    const userJson = await user.json();
    const symbol = event.target.value;
    let url = `http://localhost:3000/currency?symbol=${symbol}`;
    const response = await fetch(url);
    data = await response.json();
    const {id} = data[0];
    let index;
    let i = 0;
    while (i < userJson[0].coins.length) {
        if (userJson[0].coins[i].id === id) {
            index = i;
        }
        i++;
    }

    let currentAmount = userJson[0].coins[index].amount;
    let amount
    let ok
    while (true) {
        amount = prompt("Сколько хотите продать?")
        if (amount === null) {
            break;
        } else if (isNaN(amount)){
            alert("Введите число")
        } else if (amount <= 0) {
            alert("Количество должно быть положительным")
        } else if (amount > currentAmount) {
            alert("У вас нет столько")
        } else if (amount === currentAmount) {
            ok = confirm("Хотите продать все?")
            if (ok) {
                break;
            }
        } else {
            break;
        }
    }

    if (amount !== null) {
        amount = parseFloat(amount)
        currentAmount -= amount
    }

    if (currentAmount === 0) {
        userJson[0].coins.splice(index, 1)
    } else {
        userJson[0].coins[index].amount = currentAmount;
    }

    await fetch(`http://localhost:3000/users/${currentUser.id}`, {
        method: 'PUT',
        body: JSON.stringify(userJson[0])
```

Поиск с очисткой

```
function search() {
    const searchValue = document.querySelector('input').value

    const searchParams = new URLSearchParams()
    searchParams.set('q', searchValue)

    searchString = searchParams.toString()
    url = `http://localhost:3000/currency?${searchString}&${sortName}`

    loadCoins(url)
}

function clear_search() {
    const searchValue = ''
    document.querySelector('input').value = searchValue

    const searchParams = new URLSearchParams()
    searchParams.set('q', searchValue)

    searchString = searchParams.toString()
    url = `http://localhost:3000/currency?${searchString}&${sortName}`

    loadCoins(url)
}
```

Покупка валюты

```
async function buyCoin(event) {
    checkAuth()

    const symbol = event.target.value;
    let url = `http://localhost:3000/currency?symbol=${symbol}`;

    const response = await fetch(url);

    data = await response.json();

    const {id, current_price} = data[0];

    let coin = {
        'id': id,
        'price': current_price,
        'amount': 0,
    }

    let amount;
    while (true) {
        amount = prompt("Сколько хотите купить?")
        if (amount === null) {
            break;
        } else if (amount <= 0) {
            alert("Количество должно быть положительным")
            amount = 0
        } else {
            break
        }
    }

    if(amount !== null){
        amount = parseFloat(amount)
        coin.amount = amount;
    }

    let currentUser = JSON.parse(localStorage.getItem('user'));

    const user = await fetch(`http://localhost:3000/users?id=${currentUser[0].id}`);
    const userJson = await user.json();

    let dup = -1;
    let i = 0;
    while (i < userJson[0].coins.length) {
        if (userJson[0].coins[i].id === coin.id) {
            dup = i;
        }
        i++;
    }
```

Рендер графика

```
async function renderChart(event) {
    document.querySelector('.chart-wrapper').innerHTML = ''
    console.log(event)
    let response = await fetch(`http://localhost:3000/charts?id=${event.target.id}`)
    let chartData = await response.json()

    document.querySelector('.chart-wrapper').innerHTML +=
        `
                <canvas style="..." class="${chartData[0].id}"></canvas>
        `;

    let ctx = document.querySelector(`.${chartData[0].id}`)
    let timeSet = []
    let dataSet = []

    for (let i = 0; i < chartData[0].prices.length; i++) {
        timeSet.push(chartData[0].prices[i][0])
        dataSet.push(chartData[0].prices[i][1])
    }

    new Chart(ctx, {
        type: "line",
        options: {
            responsive: true,
        },
        data: {
            labels: timeSet,
            datasets: [
                {
                    label: chartData[0].id,
                    data: dataSet,
                    backgroundColor: "rgb(3,233,244)",
                    borderColor: "rgb(3,233,244)",
                    tension: 0,
                },
            ],
        },
    });
}
```

Выход

```
function logout() {
    if (localStorage.user) {
        localStorage.clear()
        window.location.href = "http://localhost:63342/frontLabs!/"
    }
}
```

Сортировка по имени

```
let cnt_name = 1

function sort_name() {
    switch (cnt_name % 3) {
        case 0:
            sortName = ``
            cnt_name++
            break
        case 1:
            sortName = `_sort=name&_order=desc`
            cnt_name++
            break
        case 2:
            sortName = `_sort=name&_order=asc`
            cnt_name++
            break
    }
    url = `http://localhost:3000/currency?${searchString}&${sortName}`

    loadCoins(url)
}
```

Поиск и очистка

```
function search() {
    const searchValue = document.querySelector('input').value

    const searchParams = new URLSearchParams()
    searchParams.set('q', searchValue)

    searchString = searchParams.toString()
    url = `http://localhost:3000/currency?${searchString}&${sortName}`

    loadCoins(url)
}

function clear_search() {
    const searchValue = ''
    document.querySelector('input').value = searchValue

    const searchParams = new URLSearchParams()
    searchParams.set('q', searchValue)

    searchString = searchParams.toString()
    url = `http://localhost:3000/currency?${searchString}&${sortName}`

    loadCoins(url)
}
```

Вывод: json-server довольно удобный Node Module, который позволяет эмитировать взаимодействие приложения с API при отсутствии backend'a.