

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа №2

Выполнил:

Конев Антон

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

```
function checkAuth() {
  if (localStorage.user) {
    window.location.href = 'http://localhost:63342/ITM0-ICT-Frontend-2022/labs/K33402/Konev_Anton/Lab_2/index.html'
  }
}

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

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

  const inputs = Array.from(event.target.querySelectorAll('.input'))

  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 oldUsers = JSON.parse(localStorage.getItem('User')) || []

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

      oldUsers.push(newUser)
      localStorage.setItem('user', JSON.stringify(oldUsers))
      window.location.href = 'http://localhost:63342/ITM0-ICT-Frontend-2022/labs/K33402/Konev_Anton/Lab_2/index.html'
      break
    }
  }
}

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

## Регистрация

```
function checkAuth() {
  if (localStorage.user) {
    window.location.href = 'http://localhost:63342/ITMO-ICT-Frontend-2022/labs/K33402/Konev_Anton/Lab_2/index.html'
  }
}

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 oldUsers = JSON.parse(localStorage.getItem('user')) || []

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

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

  window.location.href = 'http://localhost:63342/ITMO-ICT-Frontend-2022/labs/K33402/Konev_Anton/Lab_2/index.html'
}

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

## Портфель пользователя:

### Рендер валюты

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

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

    const response = await fetch(query)

    data = await response.json()
    amount = data.length

    btnPrev.style.visibility = currentPage === 1 ? "hidden" : "visible"
    btnNext.style.visibility = currentPage === Math.ceil(amount / coinsPerPage) ? "hidden" : "visible"

    for (let i = (currentPage - 1) * coinsPerPage; i < (currentPage * coinsPerPage); i++) {
        if (i > amount - 1) {
            break
        }

        if (amount < coinsPerPage) {
            btnPrev.style.visibility = "hidden"
            btnNext.style.visibility = "hidden"
        }

        document.querySelector('.coin-wrapper').innerHTML += getCoinHtml(data[i])
    }
    let page = document.querySelector('.page-number')
    page.innerHTML = currentPage;
}
```

### 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)
}
```

## Пагинация

```
let data;
let searchString = "";
let sortName = "";
let url = `http://localhost:3000/currency?${searchString}&${sortName}`;
let currentPage = 1;
let coinsPerPage = 10;
let amount;

const btnPrev = document.querySelector('.btn-prev')
const btnNext = document.querySelector('.btn-next')

function prev_page() {
  if (currentPage > 1) {
    currentPage--;
    loadCoins(url)
  }
}

function next_page() {
  if (currentPage < Math.ceil(amount / coinsPerPage)) {
    currentPage++;
    loadCoins(url)
  }
}

}
```

## 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++;
  }
}
```

```

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

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

```

## Создание элемента монеты

```

function getCoinHtml({image, name, current_price, symbol, market_cap_change_percentage_24h, atl_date}) {
  return `
    <tr class="">
      <td>
        <div class="d-flex align-items-center logo-item">
          
          <div class="ms-2">
            <p class="fw-bold mb-0">${name}</p>
            <p class="text-muted mb-0">${symbol.toUpperCase()}</p>
          </div>
        </div>
      </td>
      <td>
        <p class="fw-normal mb-0">RUB ${current_price}</p>
      </td>
      <td class="">
        ${changeHandler(market_cap_change_percentage_24h.toFixed(2))}
      </td>
      <td>
        <span class="start-date">${dateHandler(atl_date)}</span>
      </td>
      <td>
        <button type="button" class="btn btn-primary btn-link rounded-pill text-white text-decoration-none" value="${symbol}" onclick="buyCoin(event)">
          Купить
        </button>
      </td>
    </tr>
  `
}

```

## Графики

```
async function renderChart(event) {
  document.querySelector('.chart-wrapper').innerHTML = ''
  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: "rgba(0, 0, 0, 0)",
          borderColor: "rgb(14,96,249)",
          tension: 0,
        },
      ],
    },
  });
}
```



## Портфель

### Рендер монет в портфеле

```
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[0].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;

  btnPrev.style.visibility = currentPage === 1 ? "hidden" : "visible"
  btnNext.style.visibility = currentPage === Math.ceil(currCoins / coinsPerPage) ? "hidden" : "visible"

  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 = (currentPage - 1) * coinsPerPage; k < (currentPage * coinsPerPage); k++) {
      if (k > currCoins - 1) {
        break
      }

      if (currCoins < coinsPerPage) {
        btnPrev.style.visibility = "hidden"
        btnNext.style.visibility = "hidden"
      }

      document.querySelector('.coin-wrapper').innerHTML += getCoinHtml(renderQuery[k], amount[k])
    }
  }

  let page = document.querySelector('.page-number')
  page.innerHTML = currentPage;
  let money = document.querySelector('.balance')
  money.innerHTML = balance(coins)
}
```

## Продажа валюты

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

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

  const user = await fetch(`http://localhost:3000/users?id=${currentUser[0].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
  let f
  while (true) {
    amount = prompt("Сколько хотите продать?")
    if (amount === null) {
      break;
    } 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[0].id}`, {
      method: 'PUT',
      body: JSON.stringify(userJson[0]),
      headers: {
        'Content-Type': 'application/json',
      }
    })

    await loadCoins()
    location.reload()
  }
}

```

## Расчет баланса

```

function balance(coins) {
  let sum = 0;
  for (let i = 0; i < coins.length; i++) {
    sum += coins[i].price * coins[i].amount
  }
  return `<b>Баланс: </b>${sum.toFixed(2)}`
}

```

## Выход

```

function logout() {
  if (localStorage.user){
    localStorage.clear()
    window.location.href = "http://localhost:63342/ITMO-ICT-Frontend-2022/labs/K33402/Konev_Anton/Lab_2/sign_in.html"
  }
}

```

**Вывод:** json-server очень удобный Node Module, который позволяет реализовать взаимодействие приложения с внешним API при отсутствии реального backend'a.