

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа №2  
“Взаимодействие с внешним API”

Выполнила:

Фамилия Имя:  
Балдина Дарья

Группа: К33401

Проверил:  
Добряков Д. И.

Санкт-Петербург 2022 г.

## Задача

Подключить сайт из предыдущей лабораторной работы к внешнему API.  
Добавить фильтрацию на страницу с помощью JS.

## Ход работы

Вариант лабораторной — приложение для администратора магазина

Выбранный API: json-server-auth

### 1. Регистрация/авторизация с помощью json-server-auth

```
1  {
2    "users": [
3      {
4        "email": "ddbaldina@mail.ru",
5        "password": "$2a$10$5JvSkGy9NErQy693Prmq.Omm/ht5bYhoGwlnHZLKiefreq2.tIv.",
6        "lastname": "Балдина",
7        "name": "Дарья",
8        "id": 1
9      },
10     {
11       "email": "312407@niuitmo.ru",
12       "password": "$2a$10$2nYkTPYx2Fw2kj6QmK6ugOTEGS3V4AkSCDTW00MQEKe3nvM6Z5R92",
13       "lastname": "Матвеев",
14       "name": "Иван",
15       "id": 2
16     },
17     {
18       "email": "aleks@mail.ru",
19       "password": "$2a$10$3/MgnU2p2g5LGK/wERyE7uH5kHmMYNc7XBbNV3acBO/uDky7cOn6.",
20       "lastname": "Алексеев",
21       "name": "Олег",
22       "id": 3
23     }
24   ],
25 }
```

```

<script>
  async function createAdmin(event) {
    event.preventDefault()

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

    const infoAdmin = {}

    for (const input of inputs) {
      infoAdmin[input.name] = input.value
    }
    console.log('admin data', infoAdmin)

    const response = await fetch('http://localhost:3000/signup', {
      method: "POST",
      body: JSON.stringify(infoAdmin),
      headers: {
        'Content-Type': 'application/json',
      }
    })
  })
  window.location.href = "http://127.0.0.1:5500/labs/K33401/Baldina_Daria/LR_1/main_page.html"
}

</script>

```

```

function checkAuth(response) {
  if (response == 'Incorrect password' || response == 'Password is too short') {
    let div = document.createElement('div')
    div.className = "alert alert-danger"
    div.innerHTML = "<strong>Пароль неверный</strong>"
    document.body.append(div)
  } else if (localStorage.accessToken && response != 'Cannot find user') {
    window.location.href = "http://127.0.0.1:5500/labs/K33401/Baldina_Daria/LR_1/manage.html"
  } else {
    let div = document.createElement('div')
    div.className = "alert alert-danger"
    div.innerHTML = "<strong>Логин и пароль неверные</strong>"
    document.body.append(div)
  }
}

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

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

  const loginData = {}

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

```

```

    console.log('login data', loginData)

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

    const responseJson = await response.json()

    const { accessToken, user } = responseJson

    localStorage.accessToken = accessToken
    localStorage.user = JSON.stringify(user)

    console.log('response', responseJson)

    checkAuth(responseJson)
  }
</script>

```

## 2. Получение информации о сотрудниках их БД “workers”

```

function createTable (id, last_name, first_name, middle_name, department) {
  return `<tr>
    <td>${id}</td>
    <td>${last_name}</td>
    <td>${first_name}</td>
    <td>${middle_name}</td>
    <td>${department}</td>
  </tr>`
}

function getAuthToken() {
  return localStorage.accessToken
}

async function getDataFromDb(){
  const response = await fetch('http://localhost:3000/600/workers', {
    method: "GET",
    headers: {
      "Authorization": `Bearer ${getAuthToken()}`
    }
  })

  responseJson = await response.json()
  parcedData = JSON.parse(JSON.stringify(responseJson))

  for (i = 0; i < Object.keys(responseJson).length; i++){
    var id = responseJson[i].id
    var last_name = responseJson[i].last_name

```

```

responseJson = await response.json()
parcedData = JSON.parse(JSON.stringify(responseJson))

for (i = 0; i < Object.keys(responseJson).length; i++){
  var id = responseJson[i].id
  var last_name = responseJson[i].last_name
  var name = responseJson[i].name
  var middle_name = responseJson[i].middle_name
  var department = responseJson[i].departure_name
  document.querySelector("#new_row").innerHTML += createTable(id, last_name,name,middle_name,department)
}
}
document.addEventListener('DOMContentLoaded', () => { getDataFromDb() } )

```

### 3. Фильтрация по отделам сотрудников

```

<script>
function departmentFilter() {
  var input, filter, table, tr, td, i;
  input = document.getElementById("myInput");
  filter = input.value.toUpperCase();
  table = document.getElementById("myTable");
  tr = table.getElementsByTagName("tr");

  for (i = 0; i < tr.length; i++) {
    td = tr[i].getElementsByTagName("td")[4];
    if (td) {
      if (td.innerHTML.toUpperCase().indexOf(filter) > -1) {
        tr[i].style.display = "";
      } else {
        tr[i].style.display = "none";
      }
    }
  }
}

function logout() {

```

Intro		Информация о сотрудниках компании			
Домашняя страница		Табельный номер	Фамилия	Имя	Отчество
Учет товара		1,001	Иванов	Иван	Андреевич
Персонал		1,008	Иванов	Иван	Андреевич
Продажи		1,009	Столетов	Александр	Дмитриевич

## 4. Фильтрация по категориям товаров

```
function categoryFilter() {  
    // Declare variables  
    var input, filter, table, tr, td, i;  
    input = document.getElementById("myInput");  
    filter = input.value.toUpperCase();  
    table = document.getElementById("myTable");  
    tr = table.getElementsByTagName("tr");  
  
    // Loop through all table rows, and hide those who don't match the search query  
    for (i = 0; i < tr.length; i++) {  
        td = tr[i].getElementsByTagName("td")[1];  
        if (td) {  
            if (td.innerHTML.toUpperCase().indexOf(filter) > -1) {  
                tr[i].style.display = "";  
            } else {  
                tr[i].style.display = "none";  
            }  
        }  
    }  
}
```

Intro

Но

Выйти


Домашняя страница

Учет товара

Персонал

Продажи

Товар

Артикул	Категория	Название	Внешний вид	Стоимость	Остаток(шт.)
22489	Ноутбуки	McBook Pro		215900	215

## 5. Фильтр по датам для графика

```

<section class="col-lg-8 vh-100" style="background-color: #f7f7f7">
  <form onsubmit="filterDate(event)">
    <div class="mb-3">
      <input type="date" class="" id="start" name="start">
      <label for="form3Example2">Начальная дата</label>
    </div>
    <div class="mb-3">
      <input type="date" class="" id="end" name="end">
      <label class="form-label">Конечная дата</label>
    </div>
    <button type="submit" class="btn btn-primary col-lg-3">Искать</button>
  </form>
  <button id="clear" type="submit" class="btn btn-danger col-lg-3">Сброс </button>
  <canvas class="my-4" id="myChart" width="900" height="380"></canvas>
  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.7.1/dist/Chart.min.js"></script>
  <script>
    function needData(event){
      event.preventDefault()
      var firstDate = new Date (document.getElementById("start").value)
      var secondDate = new Date (document.getElementById("end").value)
      const filterDates = [firstDate, secondDate]
      return filterDates
    }

    function filterDate(event){
      var dates = needData(event)
      var arr = ['январь', 'февраль', 'март', 'апрель', 'май', 'июнь', 'июль', 'август', 'сентябрь', 'октябрь', 'ноябрь', 'декабрь']
      var month = [arr[dates[0].getMonth()], arr[dates[1].getMonth()]]
      var days = []
      var values = []

      if (month[0] == month[1]) {
        for (i = dates[0].getDate(); i <= dates[1].getDate(); i++){
          days.push(i)
          min = Math.ceil(10000);
          max = Math.floor(30000);
          values.push(Math.floor(Math.random() * (max - min) + min))
        }
      } else {
        let thirtyOne = ['январь', 'март', 'май', 'июль', 'август', 'октябрь', 'декабрь']
        let thirty = ['апрель', 'июнь', 'сентябрь', 'ноябрь']
        console.log(thirtyOne.indexOf(month[0]))
        console.log(thirty.indexOf(month[0]))
        console.log(arr.indexOf(month[0]))
        if (thirtyOne.indexOf(month[0]) != -1 ){
          for (var i = dates[0].getDate(); i <= 31; i++){
            days.push(i)
            min = Math.ceil(10000);
            max = Math.floor(30000);
            values.push(Math.floor(Math.random() * (max - min) + min))
          }
        }
      }
    }
  </script>

```

```

        for (var i = 1; i <= dates[1].getDate(); i++){
            days.push(i)
            min = Math.ceil(10000);
            max = Math.floor(30000);
            values.push(Math.floor(Math.random() * (max - min) + min))
        }
    } else if (thirty.indexOf(month[0]) != -1){
        for (var i = dates[0].getDate(); i < 31; i++){
            days.push(i)
            min = Math.ceil(10000);
            max = Math.floor(30000);
            values.push(Math.floor(Math.random() * (max - min) + min))
        }
        for (var i = 1; i <= dates[1].getDate(); i++){
            days.push(i)
            min = Math.ceil(10000);
            max = Math.floor(30000);
            values.push(Math.floor(Math.random() * (max - min) + min))
        }
    }
} else {
    for (var i = dates[0].getDate(); i <= 28; i++){
        days.push(i)
        min = Math.ceil(10000);
        max = Math.floor(30000);
        values.push(Math.floor(Math.random() * (max - min) + min))
    }

    for (var i = 1; i <= dates[1].getDate(); i++){
        days.push(i)
        min = Math.ceil(10000);
        max = Math.floor(30000);
        values.push(Math.floor(Math.random() * (max - min) + min))
    }
}

clear .insertAdjacentHTML('afterend', tittleGraph(month))
drawGraph(days, values)
}

function tittleGraph(month){
    if (month[0] == month[1]){
        return `<h2> Продажи за ${month[0]}</h2>`
    } else {
        return `<h2> Продажи за ${month[0]} - ${month[1]}</h2>`
    }
}

function drawGraph(days, val){
    var ctx = document.getElementById("myChart");
    var myChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: days,
            datasets: [{

```



```

data: {
  labels: days,
  datasets: [{
    data: val,
    lineTension: 0,
    backgroundColor: 'transparent',
    borderColor: '#007bff',
    borderWidth: 4,
    pointBackgroundColor: '#007bff'
  }]
},
options: {
  scales: {
    yAxes: [{
      ticks: {
        beginAtZero: false
      }
    }]
  },
  legend: {
    display: false,
  }
}
})
}

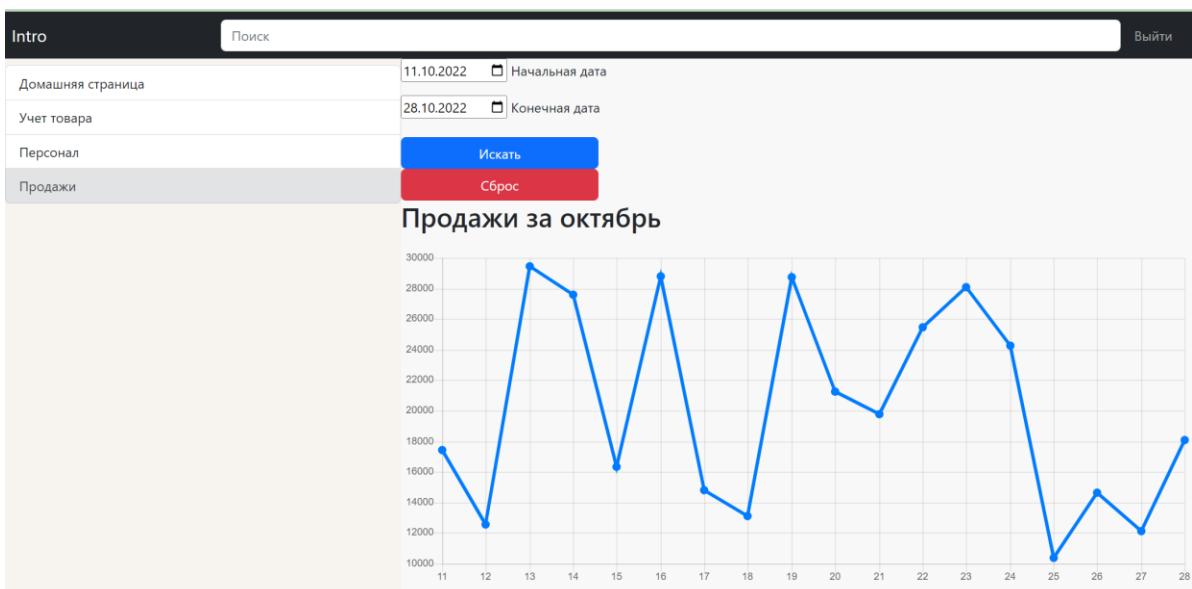
document.querySelector("#clear").onclick = function(){
  window.location.reload()
}

```

```

> function Graph(){...
document.addEventListener('DOMContentLoaded', () => { Graph() } )
function logout() {
  localStorage.clear()
  window.location.href = "http://127.0.0.1:5500/labs/K33401/Baldina_Daria/LR_1/main_page.html"
}
</script>

```



## 6. Выход из системы

```
function logout() {  
  localStorage.clear()  
  window.location.href = "http://127.0.0.1:5500/labs/K33401/Baldina_Daria/LR_1/main_page.html"  
}
```

## Вывод

В ходе работы были написаны одностраничные фильтры для работников, графиков продаж и категорий товаров с помощью ЯП javascript и реализована работа с внешним API посредством json-server-auth.