

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО

**Дисциплина:** Фронт-энд разработка

**Отчет** Лабораторная работа 2

Выполнил: Бункута Натан

Груаа: K33412

Проверил: Добряков Д. И.

Санкт-Петербург  
2022 г.

## Задача:

Привязать веб-страницу к внешнему API.

## Ход работы

Сначала я зарегистрировался на Unsplash и получил доступ к API. Чтобы делать запросы через api, нужно в тело запроса добавляться поле "client\_id" с выданным ключом на сайте.

```
scripts > JS ApiService.js > ApiService
1  import {fetchHandler} from "../utils.js";
2
3  export class ApiService{
4    constructor() {
5      this.api_key = "RcTUFyt0S2YUHSqsgEg3oL62h0FbB3Vq6I_qHi8qFls"
6      this.base_url = "https://api.unsplash.com"
7    }
8
9    url(path, params= {}){
10     let query = "?";
11     params["client_id"]=this.api_key
12     Object.entries(params).forEach(e=>{
13       query+=` ${e[0]}=${e[1]}&`
14     })
15     query = query.slice(0,-1)
16     return `${this.base_url}${path}/${query}`
17   }
18
19   async fetchRandomPhotos(){
20     return await fetchHandler(this.url("/photos/random", {count:30}))
21   }
22
23   async fetchPhotosByQuery(query){
24     const response = await fetchHandler(this.url("/search/photos", {query:query, per_page:30}))
25     return response.results
26   }
27 }
```

**renderAlbum** здесь принимает на вход массив фотографий и кнопку. Эту функцию позволяет отрисовать сетки фотографий. (на главной странице используется кнопка “download”, на странице пользователя “Remove”).

```
export const renderAlbum = (photos, actionButton) => {
  const album = document.getElementById('album')
  const album_div = document.createElement('div')
  album_div.className = "album bg-light"
  const container = document.createElement('div')
  container.className = "container"
  const row = document.createElement('div')
  row.className = "row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3"

  album.appendChild(album_div)
  album_div.appendChild(container)
  container.appendChild(row)

  photos.forEach(e=>{
    let author_link = e.user.links.html;
    let author_username = e.user.username;
    let preview_link = e.urls.regular;

    const col = document.createElement('div')
    col.className = "col"
    const card = document.createElement('div')
    card.className = "card shadow-sm"
    const photo_img = document.createElement('img')
    photo_img.src = preview_link
    photo_img.alt = "photo"
    photo_img.className = "card-photo"
    photo_img.addEventListener("click", ()=>{
      viewPhotoModal(e)
    })

    const card_body = document.createElement('div')
    card_body.className = "card-body"
    const inner_card_body = document.createElement('div')
    inner_card_body.className = "d-flex justify-content-between align-items-center"
    const author_link_btn = document.createElement('a')
    author_link_btn.href = author_link
    const author_button = document.createElement('button')
    author_button.type="button"
    author_button.className="btn btn-link"
    author_button.textContent=author_username
    author_link_btn.appendChild(author_button)

    card_body.appendChild(inner_card_body)
    inner_card_body.appendChild(actionButton(e))
    inner_card_body.appendChild(author_link_btn)
  })
}
```

Реализация функций для поиска фото по API:

```
export const searchBar = () => {
  const search = document.querySelector('[type="search"]')
  search.addEventListener('keypress', async (e) => {
    if (e.key === 'Enter') {
      await queryFilter(e.target.value.toLowerCase())
    }
  })
}

export const queryFilter = async (filter) => {
  photos = await api.fetchPhotosByQuery(filter)
  clearAlbum()
  renderAlbum(photos, downloadButton)
}

export const renderFilters = () => {
  const filters = ["Cities", "Health", "", "Fashion & Style", "People", "Animals", "Arts & Culture", "Sport", "Food and Drinks", "Houses", "Tech"]
  const filtersField = document.getElementsByClassName("categories_filter")
  filters.forEach(filter => {
    const btn = document.createElement('button')
    btn.className = "btn btn-dark"
    btn.textContent = filter
    btn.addEventListener("click", async () => {
      await queryFilter(filter.toLowerCase())
    })
    filtersField[0].appendChild(btn)
  })
}
```

**viewPhotomodal**, функция для реализации модального окна для предпросмотра при клике.

```
export const viewPhotoModal = (photo) => {

  const modal = document.querySelector(".myModal")
  modal.style.display = "flex";

  const content = document.createElement("div")
  content.className = "myContent"

  const img = document.createElement("img")
  img.src = photo.urls.regular;
  img.className = "content"

  const download = document.createElement("a")
  download.href = photo.links.download
  download.className = "rounded-0 rounded-bottom btn btn-primary"
  download.textContent = "Download"

  window.onclick = function(event) {
    if (event.target === content || event.target === modal) {
      modal.style.display = "none";
      while (modal.firstChild) {
        modal.firstChild.remove();
      }
    }
  }

  modal.appendChild(content)
  content.appendChild(img)
  content.appendChild(download)
}
```

**Вывод:**

В ходе выполнения лабораторной работы были реализованы запросы по API. Реализован функционал по поиску фото и добавлению фото в избранное. Сделано модальное окно для предпросмотра фото.