

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

**Дисциплина:** Фронт-энд разработка

Отчет  
Лабораторная работа 2

Выполнил:

Ковалев Валерий К33401

Проверил:

Добряков Д. И.

Санкт-Петербург  
2022 г.

## Задача

Привязать макет к внешнему API.

## Ход работы

Для начала я зарегистрировался на Unsplash и получил доступ к API. Чтобы делать запросы через ари, нужно в тело запроса добавляться поле “client\_id” с выданным ключом на сайте.

Реализовал класс ApiService:

```
import {fetchHandler} from "./utils.js";

export class ApiService{
    constructor() {
        this.api_key = "RcTUFyt0S2YUHSqsgEg3oL62h0FbB3Vq6I_dHi8qFls"
        this.base_url = "https://api.unsplash.com"
    }

    url(path, params:{} = {}){
        let query = "?";
        params["client_id"] = this.api_key
        Object.entries(params).forEach(e=>{
            query += `${e[0]}=${e[1]}&`
        })
        query = query.slice(0,-1)
        return `${this.base_url}${path}/${query}`
    }

    async fetchRandomPhotos(){
        return await fetchHandler(this.url( path: "/photos/random", params: {count:30}))
    }

    async fetchPhotosByQuery(query){
        const response = await fetchHandler(this.url( path: "/search/photos", params: {query:query, per_page:30}))
        return response.results
    }
}
```

Для отрисовки сетки фотографий реализовал функцию renderAlbum, которая принимает на вход массив фотографий и кнопку (на главной странице используется кнопка “Like”, на странице пользователя “Remove”).

```
export const renderAlbum = (photos, actionBar) => {
  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(actionBar(e))
    inner_card_body.appendChild(author_link_btn)

    card.appendChild(photo_img)
    card.appendChild(card_body)
    col.appendChild(card)
    row.appendChild(col)
  ])

  return album
}
```

Лайки обрабатываются и записываются в localStorage, так что на странице пользователя отображаются понравившееся фото:

```
export const likes = () => {
  let data = localStorage.getItem('likes')
  if(data){
    return JSON.parse(data)
  }else{
    return []
  }
}

export const handleLike = (photo) => {
  let data = likes()
  if(!data.find(e => e.id === photo.id)){
    data.push(photo)
    localStorage.setItem('likes', JSON.stringify(data))
  }
}

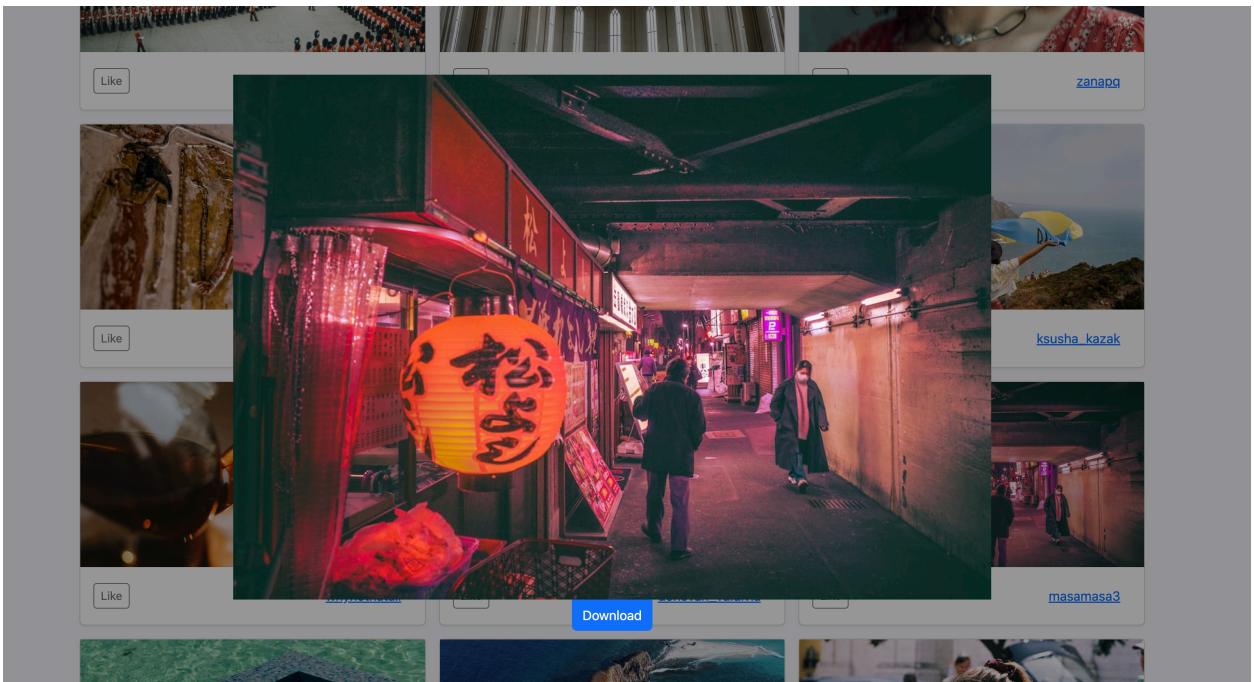
export const removeLike = (photo) => {
  localStorage.setItem('likes', JSON.stringify(likes().filter(e => e.id !== photo.id)))
  clearAlbum()
  renderAlbum(likes(), removeButton)
}
```

По API можно запросить фото по тематике. Реализация функций для поиска фото:

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

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

Реализовал модальное окно вместо отдельной страницы при клике на фото:



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

    const modal = document.querySelector( { selectors: ".myModal" })
    modal.style.display = "flex";

    const content = document.createElement( { tagName: "div" })
    content.className="myContent"

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

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

    window.onclick = function(event :MouseEvent ) {
        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.