Дипломная работа по курсу «JavaScript»

Что должно делать приложение?

Пользователь приложения должен иметь возможность:

- просмотреть ленту фотографий
- просмотреть фотографию детально
- поставить/снять лайк

Как должна работать лента фотографий?

По умолчанию в ленте загружаются 10 последних фотографий. Загрузить следующие 10 фотографий можно нажатием на кнопку «Загрузить еще» или прокруткой к концу страницы.

Для каждой фотографии в ленте отображается:

- сама уменьшенная фотография (превью)
- имя автора и ссылка на его Unsplash-профиль
- дата публикации
- кол-во лайков

При нажатии на фотографию происходит переход на страницу просмотра данной фотографии (в вашем приложении, не на сайте Unsplash). Переход должен происходить без перезагрузки страницы (пригодится `react-router`).

Как должна работать страница просмотра фотографии?

На странице просмотра фотографии отображается следующие элементы:

- полноразмерная фотография
- имя автора и ссылка на его Unsplash-профиль
- дата публикации

- кол-во лайков
- кнопка лайка
- кнопка возврата к ленте фотографий

Кнопка лайка должна работать соответствующим образом (ставить или снимать лайк). При этом важно, чтобы лайки не только отображались в интерфейсе приложения, но и реально ставились от лица пользователя для данной фотографии через Unsplash API.

Откуда взять фотографии?

Фотографии нужно загружать с сайта Unsplash при помощи их API.

Почему не взять фотографии из Instagram? Дело в том, что у Instagram есть масса ограничений по использованию их API для приложений, которые не прошли проверку и находятся в так называемом Sandbox Mode (режим песочницы). Отправлять дипломную работу на проверку в Instagram не имеет смысла и может занять дополнительное время, а работать в рамках ограничений довольно некомфортно. Поэтому лучше воспользоваться бесплатным и более доступным Unsplash API. Ограничения там тоже есть, но они более мягкие (не более 50 запросов в сутки в режиме в Development-режиме).

Как воспользоваться Unsplash API?

Чтобы использовать Unsplash API, нужно пройти следующие шаги:

1. Зарегистрироваться на Unsplash как обычный пользователь сайта
2. В разделе API/Developers зарегистрироваться как разработчик для использования Unsplash API
3. Создать в разделе API/Developers новое приложение (Application), указав необходимые данные:

`Application name` — название приложения (например, *Диплом Skillbox JS*)

`Description` — краткое описание приложения (например, *Приложение для просмотра фотографий*)

`Redirect URI` — путь к странице вашего приложения, на которую будет произведено перенаправление после авторизации приложения в Unsplash API. Об этом подробнее написано ниже.

`Permissions` — разрешения для вашего приложения. Нам нужны пункты `Public access` и `Write likes access`.

1. После сохранения скопировать полученные строки `accesskey` и `Application Secret`
2. Ознакомиться с документацией официальной JavaScript-библиотеки unsplash-js. По ссылке перечислены все возможные методы API. Нас интересуют в первую очередь методы `photos.listPhotos`, `photos.likePhoto`, `photos.unlikePhoto`.
3. При написании приложения с использованием Unsplash API желательно следовать их требованиямкасательно ссылок на профили авторов и наличия UTM-параметров в ссылках.

Что такое авторизация приложения (OAuth) и как ей пользоваться?

Unsplash, как и многие другие крупные проекты (Facebook, Twitter, Instagram, ...) использует механизм OAuth для авторизации сторонних приложений. Сторонние приложения — это сайты, мобильные приложения и вообще любой внешний код, который взаимодействует с их API.

Зачем нужна авторизация? Чтобы дать стороннему приложению разрешение выполнять действия на сайте от лица пользователя, который это разрешение даст. Действия могут быть разные: создание, редактирование и удаление контента, лайки, комментарии и т.д.

Принцип работы OAuth следующий:

1. Разработчик пишет приложение (назовем его TestApp), подключает его к Unsplash API и выкладывает на своем сервере по какому-либо адресу (к примеру, `www.example.com`)
2. Пользователь заходит по адресу `www.example.com`, после чего его перекидывает на страницу авторизации на сайте `unsplash.com`, где сказано следующее: «Приложение TestApp запрашивает разрешение на выполнение следующих действий от вашего имени: ...». Предполагается, что пользователь уже зарегистрирован и авторизован на Unsplash.
3. Если пользователь дает приложению TestApp соответствующие разрешешения, то его перекидывает обратно на `www.example.com` на специальную страницу, которую разработчик указал в настройках своего приложения в разделе API/Developers. Эта страница называется `Redirect URI` или `Callback URL`. Например, она может располагаться по адресу `www.example.com/auth`.

4. Можно увидеть, что в строке адреса появится GET-параметр `code`: `www.example.com/auth?code=abcdef123456...`. Это специальный уникальный код, который позволит приложению запросить и получить *токен доступа*. Токен — это нечто вроде временного пароля, длинная строка, которая добавляется к каждому запросу от приложения к API, и которая дает возможность выполнять операции от лица конкретного пользователя.
5. После получения токена приложение считается авторизованным и может выполнять все необходимые действия

Пример кода авторизации

Главная страница `www.example.com`:

```
// Подключаем библиотеку unsplash-
                  js

   // (при настроенной webpack-
              сборке)

 import Unsplash from 'unsplash-
                 js';
```

```
// Создаем экземпляр объекта для доступа к API
const unsplash = new Unsplash({
  // accesskey из настроек вашего приложения
  accesskey: " ......",
  // Application Secret из настроек вашего приложения
  secret: " ..",
  // Полный адрес страницы авторизации приложения (Redirect URI)
  // Важно: этот адрес обязательно должен быть указан в настройках приложения
на сайте Unsplash API/Developers
  callbackUrl: "http://www.example.com/auth"
});

// Генерируем адрес страницы аутентификации на unsplash.com
// и указываем требуемые разрешения (permissions)
const authenticationUrl = unsplash.auth.getAuthenticationUrl([
  "public",
  "write_likes"
]);

// Отправляем пользователя по этому адресу
location.assign(authenticationUrl);
```

Страница `www.example.com/auth`:

```
// Подключаем библиотеку unsplash-
                  js

   // (при настроенной webpack-
              сборке)

 import Unsplash from 'unsplash-
                 js';
```

```
// Создаем экземпляр объекта для доступа к API
const unsplash = new Unsplash({
```

```
  accesskey: "...",
  secret: "...",
  callbackUrl: "http://www.example.com/auth"
});

// Считываем GET-параметр code из URL
// www.example.com/auth?code=abcdef123456...
const code = location.search.split('code=')[1];

// Если код передан, отправляем запрос на получение токена
if (code) {
  unsplash.auth.userAuthentication(code)
    .then(res =>
    res.json())
    .then(json =>
    {
      // Сохраняем полученный
      токен
      unsplash.auth.setBearerToken(json.access_token);

      // Теперь можно сделать что-то от имени пользователя
      // Например, поставить лайк фотографии
      unsplash.photos.likePhoto("kBJEJqWNtNY");
    });
}
```

Как должно выглядеть приложение?

Дизайн и верстка приложения остаётся полностью на ваше усмотрение. В качестве примеров можете использовать сайты Instagram и Unsplash.

В каком виде нужно сдать дипломную работу?

Дипломную работу необходимо сдать в виде двух ссылок:

1. Результат — готовое работающее приложение, выложенное в интернете для просмотра и тестирования. Для публикации можно использовать любой имеющийся хостинг.
2. Исходный код — выложить на GitHub.

Каковы технические требования к приложению?

- Использование Unsplash API и библиотеки `unsplash-js`
- Использование `React` и `Redux`
- Сборка при помощи `webpack`

- Single page application (SPA) — переход от ленты к просмотру фотографии и обратно должен происходить без перезагрузки страницы. Для этого пригодится модуль `react-router`.