SDK

LoftSchool
от мыслителя к создателю

Содержание

1

SDK

SDK (Software Development Kit) - это набор инструментов и библиотек (API) для разработки приложение под конкретную платформу.

В данной методичке мы рассмотрим SDK соц.сети vk.com

2

Использование SDK

Чтобы воспользоваться SDK, требуется:

- создать специальное приложение на сервере платформы (может и не требоваться)
- подключить JS-скрипт с SDK
- провести инициализацию приложения
- получить Access Token (может и не требоваться)

Создание приложения на сервере

В большинстве случаев, разработчик использует SDK для получения каких-либо данных с сервера. Например, SDK для социальной сети vk.com или facebook.com используется для связи с сервером и получением с него информации (фото, сообщения, новости, друзья и т.д.).

Соответственно, суть SDK - посылать запросы на сервер и получать ответ.

Нюанс заключается в том, что сервер не намерен раздавать эту информацию всем подряд. Информация между SDK и сервером передается через так называемое приложение.

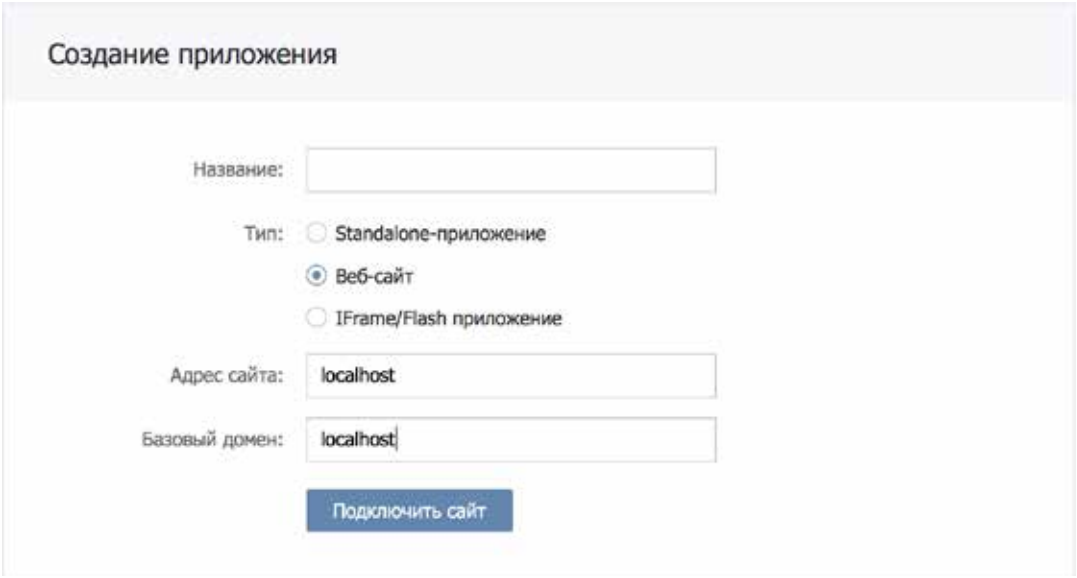
То есть, для того, чтобы работать с сервером, вы должны зарегистрировать приложение. После этого, вы получите специальный ID. Этот ID представляет собой регистрационный номер вашего приложения.

Чаще всего, страницу регистрации приложения можно найти в разделе “Разработчикам” соответствующей соц.сети:

- заходим - **http://vk.com/apps?act=manage**

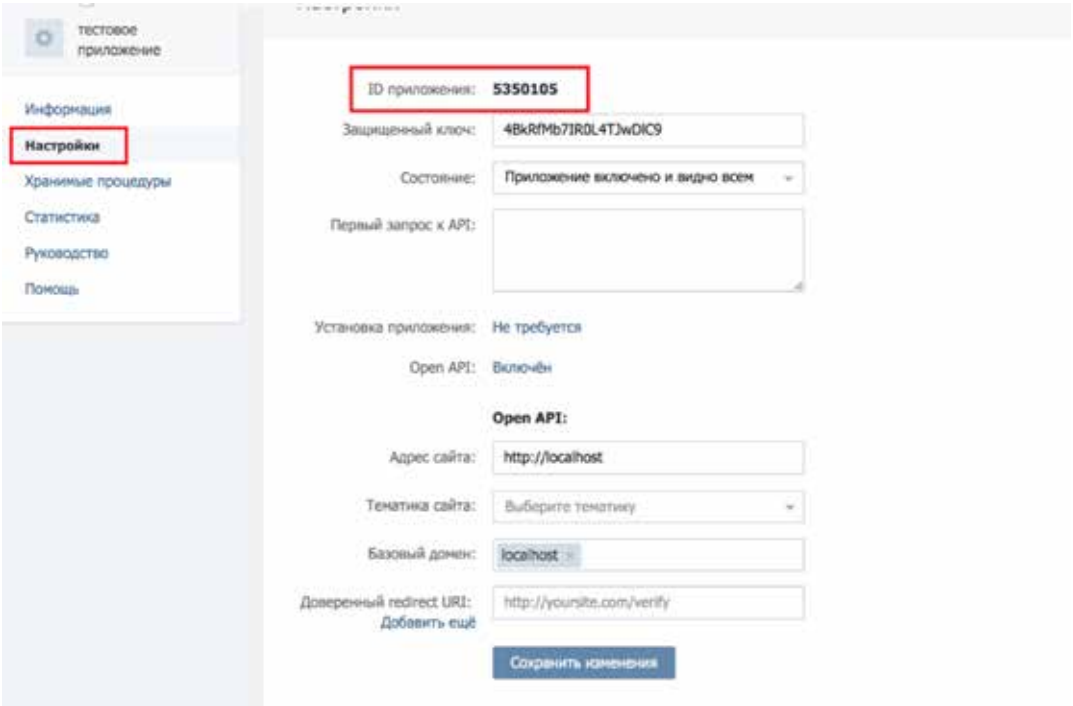
- нажимаем “создать приложение”

- заполняем инфо о вашем приложении



- подтвердите создание приложение через смс (вам придет код подтверждения)

- перейдите во вкладку “настройки” и увидите там id вашего нового приложения. Он-то нам и нужен



Подключение и инициализация SDK

После того, как мы получили ID приложения, необходимо подключить скрипт с SDK

Для соц.сети vk.com, этот скрипт подключается так:

```
<script src="http://vk.com/js/api/openapi.js" type="text/
javascript"></script>
```

После этого, мы можем инициализировать приложение для работы с SDK:

```
VK.init({
  apiId: 5350105
});

VK.Auth.login(function(response) {
  if (response.session) {
     console.log('всё ок!');
  } else {
     alert('Не удалось авторизоваться');
  }
}, 8);
```

VK.init - выполняет инициализацию.

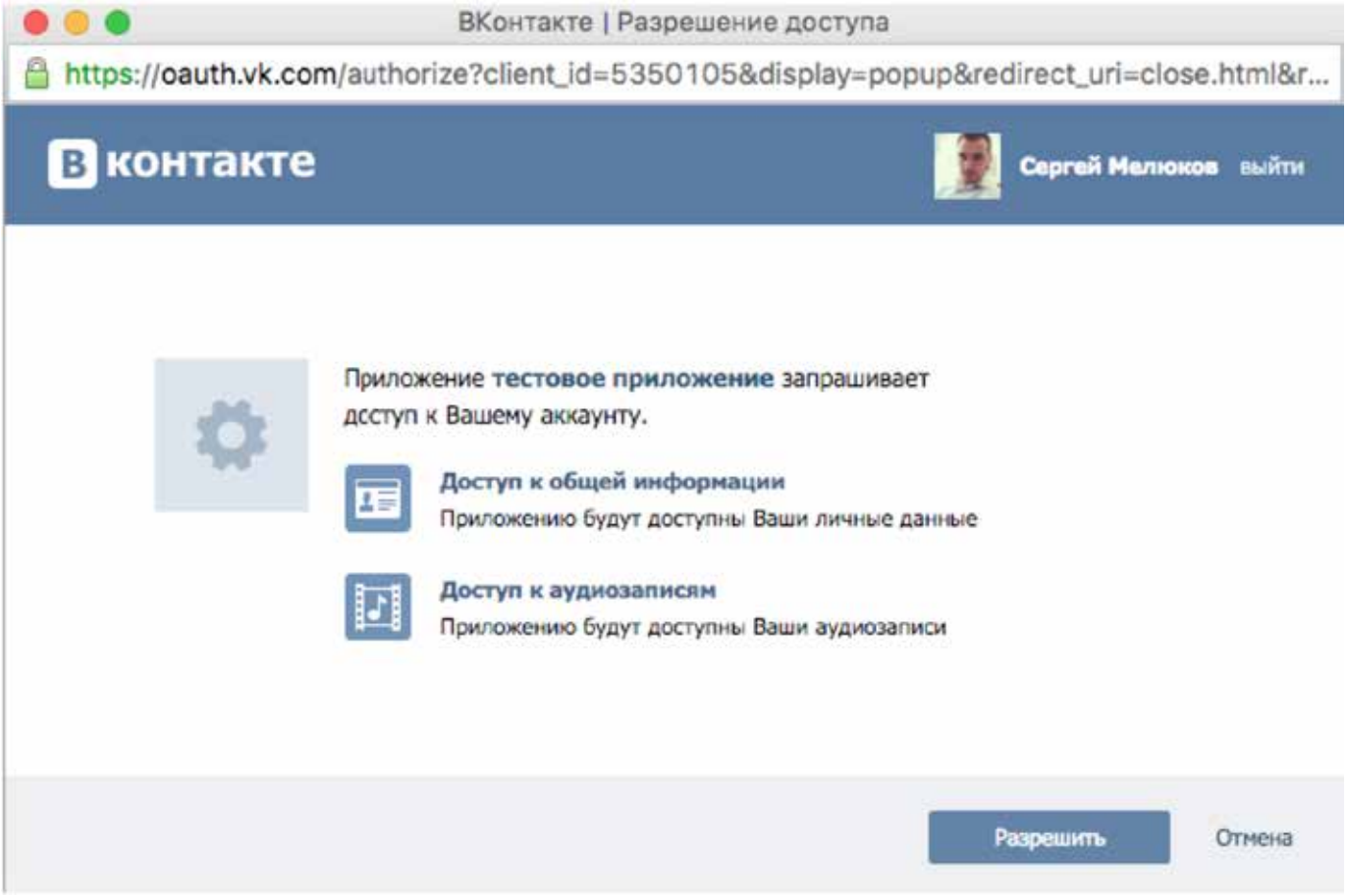
В качестве параметра принимает аргумент с настройками приложения. Обязательным свойством в этом объекте, является свойства с именем apiId. А в качестве значения необходимо подставить тот ID, который мы получили при регистрации приложения.

Получение Access Token (авторизация)

Далее, необходимо авторизовать пользователя, который зашел на вашу страницу.

То есть необходимо, чтобы пользователь подтвердил, что он дает свое согласие на то, чтобы ваше приложение имело доступ к его информации в соц.сети.

Для этого, необходимо вызвать метод **VK.Auth.login**, который откроет специальное окно, в котором, зашедший на страницу пользователь, мог бы подтвердить свою готовность предоставить информацию о своей странице нашему приложению.



VK.Auth.login имеет два параметра:

- функция, которая будет выполнена, когда пользователь согласится или не согласится предоставить информацию
- список той информации, которую требует наше приложение для корректной работы (например: доступ к списку друзей, доступ к новостям, к аудио/видео записям)

Список требуемой информации указывается в виде списка чисел, разделенных ||

Полный список таких числовых идентификаторов можно найти на странице **http://vk.com/dev/permissions**

В данном случае, мы запросили доступ к списку аудиозаписей пользователя.

Внутри функции, мы должны проверить - разрешил ли пользователь нашему приложению доступ к своей странице. Для этого достаточно просто проверить свойство session объекта с ответом. Теперь мы можем полноценно использовать VK JS SDK для доступа к информации о пользователе.

Работа с SDK

Дальнейшая работа с SDK производится при помощи вызова методом **VK.api**, который имеет 3 параматра:

- имя метода

- объект с параметрами метода

- функция, которая будет вызвана, когда VK-сервер вернет результат нашего запрос

Полный список методов и их описание, можно найти тут **https://vk.com/dev/methods**

Давайте отправим запрос для получения информации о пользователе. При этом, в параметрах укажем, что хотим получить имя и фамилию пользователя в родительном падеже:

```
VK.api('users.get', {'name_case': 'gen'},
function(response) {
  if (response.error) {
     alert(response.error.error_msg);
  } else {
     document.body.textContent = 'Музыка на странице'
+ response.response[0].first_name + ' ' + response.
response[0].last_name;
  }
});
```

Здесь всё очень просто - отправляем запрос, а в callback-функции проверяем ответ. Если нет ошибок, то выводим имя пользователя на html-страницу.

3

Пример

Заметьте, что все эти операции - асинхронны.

То есть браузер не будет ждать, пока сервер VK отдаст ответ на наш запрос.

Поэтому, для увеличения читаемости кода и удобства при работе с кодом, мы будет использовать промисы, для того, чтобы организовать асинхронный код:

```
new Promise(function(resolve) {
  if (document.readyState === 'complete') {
     resolve();
  } else {
     window.onload = resolve;
  }
}).then(function() {
  return new Promise(function(resolve, reject) {
     VK.init({
        apiId: 5267932
     });

     VK.Auth.login(function(response) {
        if (response.session) {
           resolve(response);
        } else {
           reject(new Error('Не удалось
авторизоваться'));
        }
     }, 8);
  });
}).then(function() {
  return new Promise(function(resolve, reject) {
     VK.api('users.get', {'name_case': 'gen'},
```

```
function(response) {
        if (response.error) {
           reject(new Error(response.error.error_msg));
        } else {
           headerInfo.textContent = 'Музыка на странице'
+ response.response[0].first_name + ' ' + response.
response[0].last_name;

           resolve();
        }
     });
  })
}).catch(function(e) {
  alert('Ошибка: ' + e.message);
});
```

Здесь всё очень просто:

- создаем промис

- промис ждет, пока наступит событие window.load

- следом, создаем еще промис, который ждет результата инициализации приложения и авторизации пользователя

- следом, создаем еще промис, который ждет ответа от сервера на наш запрос об информации о пользователе.

Таким образом, все асинхронные операции выполнятся последовательно.

Работа с SDK соц.сети fb.com схожа с описанным здесь алгоритмом.