

Лабораторная работа 3

Знакомство с VK API

Задание

Создать HTML страницу с вложенным javascript, который бы выдавал список ID ваших друзей в контакте. Для каждого друга необходимо вывести дополнительную информацию, согласно варианту.

Дополнительное задание:

Оформить возвращенный JSON в виде таблицы HTML.

Теория

При использовании кроссдоменных запросов (а скрипт с нашей страницы будет, очевидно, обращаться к другому домену) использовать метод XMLHttpRequest невозможно, поскольку вступает в действие ограничение «Same Origin» («тот же источник»). Эта политика ограничивает доступ окон и фреймов друг к другу, а также влияет на AJAX-запросы к серверу. Причина, по которой оно существует – безопасность. Если есть два окна, в одном из которых vasya-purkin.com, а в другом gmail.com, то мы бы не хотели, чтобы скрипт из первого мог читать нашу почту.

Для кроссдоменных запросов нам необходимо использовать JSONP. Для удобства использования мы будем пользоваться фреймворком jQuery. В заголовке нашей страницы необходимо подключить его, например, с адреса <https://code.jquery.com/jquery-3.6.0.min.js>

Как подключить: <https://releases.jquery.com/> (нужна minified версия) Далее мы будем пользоваться функцией getJSON:

```
<script
src="https://code.jquery.com/jquery-3.6.0.min.js"
integrity="sha256-/xUj+30JU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"></script>
<script>
  const method = 'friends.get'
  const userId = '123'
  const token = '231213123'
  const version = '5.131'
  const fields = '*'
  const data = $.getJSON(
    `https://api.vk.com/method/${method}?
user_id=${userId}&access_token=${token}&v=${version}&fields=${fields}`,
    (data) => {

      console.log('Out data is: ', data);
    }
  )
</script>
```

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных vk.com с помощью http-запросов к специальному серверу. Вам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит — достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса.

Например, для получения данных о пользователе с идентификатором 210700286 необходимо составить запрос такого вида: `https://api.vk.com/method/users.get?user_id=210700286&access_token=${service_key}&v=5.131&fields=*`

Рассмотрим отдельно все его составляющие:

- **https://** — протокол соединения.
- **api.vk.com/method** — адрес API-сервиса.
- **users.get** — название метода API ВКонтакте. Методы представляют собой условные команды, которые соответствуют той или иной операции с базой данных — получение информации, запись или удаление. Например, *users.get* — метод для получения информации о пользователе, *video.add* — метод для добавления видеозаписи в свой список, *likes.delete* — метод для удаления отметки «Мне нравится». Все методы разделены на секции. Например, для работы с сообществами Вам нужны методы секции *groups*, для работы с фотографиями — *photos*, и так далее. Полный список методов по секциям, а также консоль для отправки запросов доступна [этой странице](#)
- **?user_id=210700286&v=5.131** — параметры запроса. После названия метода нужно передать его входные данные (если они есть) — как обычные GET-параметры в http-запросе. В нашем примере мы сообщаем серверу, что хотим получить данные о пользователе с id=210700286 и формат этих данных должен соответствовать версии API 5.131 (о версиях мы еще поговорим позже). Также для работы вам необходимо указать сервисный ключ доступа. Входные параметры всегда перечислены на странице с описанием метода.

Ответ на наш запрос выглядит так:

```
{
  "response": [
    {
      "id": 210700286,
      "first_name": "Lindsey",
      "last_name": "Stirling"
    }
  ]
}
```

Структура ответа каждого метода также строго задана, и при работе с API вы заранее знаете, что в поле *id* придет число, а в поле *first_name* — строка. Такие правила оговариваются на страницах с описанием метода и соответствующих объектов, которые он возвращает в ответе приложения.

В примере был использован метод `users.get`, для вызова которого с апреля 2017 года необходимо зарегистрировать свое приложение, чтобы использовать возможности API ВКонтakte.

Откройте страницу "Управление" в левом меню, затем нажмите "Создать" — вы попадете на [страницу](#) Нужно выбрать один из трех типов приложений:

- **Встраиваемое приложение** — это те самые приложения, которые Вы можете видеть в нашем каталоге <https://vk.com/apps> . Они загружаются непосредственно на сервер ВКонтakte (Flash) или встраиваются во фрейме с внешнего сайта.
- **Standalone-приложение** — это API_ID для мобильного или десктопного клиента, внешнего сайта, где работа с API будет вестись из Javascript. Основная мысль в том, что запросы к API должны осуществляться с устройства пользователя. В интерфейсе приложения с таким типом доступны настройки SDK и подключение сертификатов для push-уведомлений.
- **Сайт** — регистрация API_ID для внешнего сайта и работы с API с сервера. Например, если Вы хотите написать скрипт на PHP с использованием API ВК, Вам нужен именно этот вариант.

Выберите первый тип, **Standalone**. После подтверждения действия вы попадете на страницу с информацией о приложении.

Откройте вкладку **Настройки** в меню слева. Вы увидите поле "ID приложения", в котором будет указано число, например, 5490057. Это число — идентификатор приложения, он же *API_ID*, *APP_ID*, *client_id*, оно потребуется Вам в дальнейшей работе.

Сервисный ключ доступа

Сервисный ключ нужен для запросов, которые не требуют авторизации пользователя или сообщества. Это такие методы, как `secure.sendNotification` для отправки уведомлений от приложения, или `secure.addAppEvent` для добавления информации о достижениях, а также, начиная с апреля 2017 года, открытые методы, например, `users.get`.

Получить сервисный ключ доступа можно в настройках вашего приложения. Ключ не привязан к IP-адресу при использовании с открытыми методами, срок его действия не ограничен. Если ключ был скомпрометирован, вы можете сгенерировать новый ключ, при этом старый будет аннулирован.

Сервисный ключ доступа идентифицирует Ваше приложение. Все запросы к API, совершённые с использованием вашего ключа доступа, будут считаться совершёнными от имени вашего приложения. Сервисный ключ доступа можно использовать только для запросов с серверной стороны приложения, его нельзя передавать и хранить на клиенте. А иначе злоумышленники будут совершать действия от имени вашего приложения так как весь код клиентского приложения можно легко просмотреть через панель разработчика.

Для запросов к методам `secure` сервисный ключ привязан к IP-адресу, с которого был сгенерирован.

Варианты

Для удобства во время написания кода можно тестировать запросы прямо на странице конкретного метода в документации к [VPAPI](#)

Номер варианта	Тема
1	для 2 пользователей: "информацию о пользователе", "количество подписчиков", "имя в предложном падеже"
2	для 1 пользователя: "пол", "город", "имя в родительном падеже"
3	для 6 пользователей: "статус онлайн", "имя в творительном падеже"
4	для 3 пользователей: "ссылку на оригинал фотографии в максимальном качестве", "доменное имя (которое служит сокращением в ссылке)"
5	для 4 пользователей: "день рождение", "контактную информацию", "имя в дательном падеже"
6	для 1 пользователя: "дату последнего захода в онлайн" + перевести дату в читабельный вид