

# Лабораторная работа 3

## Знакомство с VK API

### Задание

Создать HTML страницу с вложенным javascript, который бы выдавал список ID ваших друзей вконтакте. 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-3.3.1.min.js>.

Далее мы будем пользоваться функциейgetJSON вида

```
$.getJSON({
    url: "адрес_запроса",
    jsonp: "callback",
    dataType: "jsonp"
}).done(function( data ) {

    здесь описание того, что делать с ответом

});
```

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных [vk.com](https://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=5.52`

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 доступен [на этой странице](#).
- `?user_id=210700286&v=5.52` — 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 5.52 (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 ВКонтакте.

Откройте страницу «Управление» в левом меню, затем нажмите «Создать приложение» — вы попадете на страницу <https://vk.com/editapp?act=create>

Нужно выбрать один из трех типов приложений:

- **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.

- **Веб-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.
- **IFrame/Flash приложение** — это те самые приложения, которые Вы можете видеть в нашем каталоге <https://vk.com/apps>. Они загружаются непосредственно на сервер ВКонтакте (Flash) или встраиваются во фрейме с внешнего сайта.

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.

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