**Загрузка сообщения с картинкой на стену ВК (API)**

**(инструкция)**

Загрузка фотографий на стену осуществляется методом **wall.get**, который можно осуществить только с помощью ключа доступа (**acces\_token**) или сервисным ключом доступа.

Чтобы обратиться к методу API ВКонтакте, нужно выполнить либо POST-, либо GET-запрос (запрос с каким HTTP-глаголом вы отправите, не имеет значения) вида:

POST https://api.vk.com/method/<METHOD>?<PARAMS> HTTP/1.1

**Забегая наперед: создайте переменную в коллекции: owner\_id**



Его можно посмотреть, кликнув на главную страницу своего профиля ВК



**Внимание: не забывайте указывать параметры: «acces\_token» и «v» (используемая версия API)**

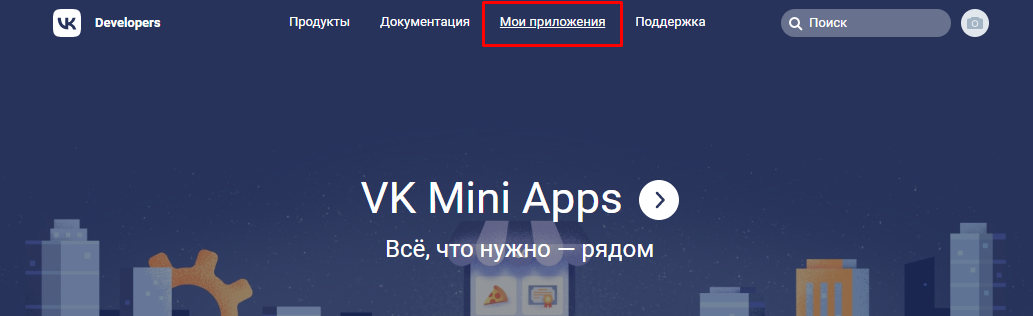
**Этапы загрузки фотографии и сообщения на стену:**

1. Получение ключа доступа (acces\_token)
2. Получение списка сообщений на стене
3. Получение адреса для загрузки картинки
4. Передача файла (картинки) на полученный адрес
5. Сохранение картинки на полученном адресе
6. Публикация картинки и сообщения на стену
7. Просмотр конкретного созданного сообщения на стене
8. Получение списка сообщений на стене (удостоверяемся, что наше сообщение есть в списке)

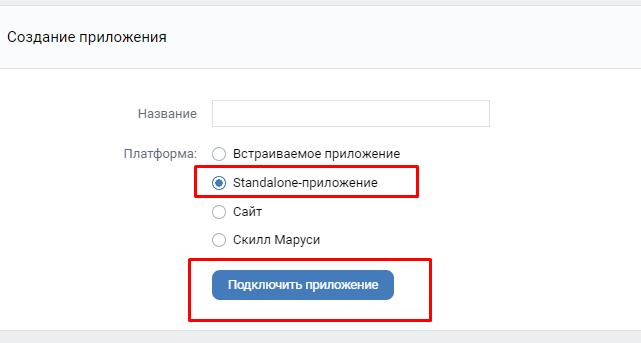
**1. Получение ключа доступа (acces\_token)**

Прежде всего необходимо создать Standalone-приложение. Сделать это можно, следуя следующим пунктам:

1. Перейдем в раздел [разработчикам](http://vk.com/dev) и нажмем кнопку "Создать приложение".



1. Выбираем тип приложения "Standalone-приложение" и вводим любое имя приложения. Например: Auto msg.



1. После нажатия по кнопке «Подключить приложение» у вас затребуют код, который придет вам в виде СМС в течении минуты на мобильный номер телефона, указанный в профиле пользователя, под которым вы авторизованы.
2. Отлично! У Вас уже есть standalone-приложение.

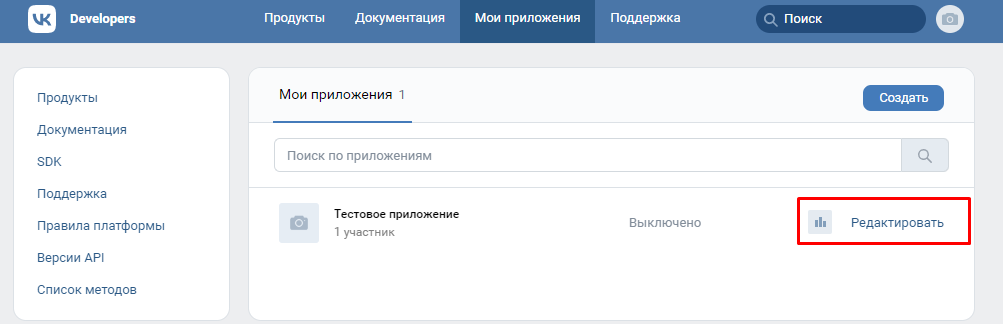
PS в дальнейшем информацию вашего приложения вы можете редактировать на странице [Приложения](http://vk.com/apps).

PS. Кстати, вы не сможете удалить созданные вами приложения. После осознания этой информации у нас задергался левый глаз и зачесалась правая пятка! Мы не привыкли работать с сервисами, где нет возможности вернуться на исходную!

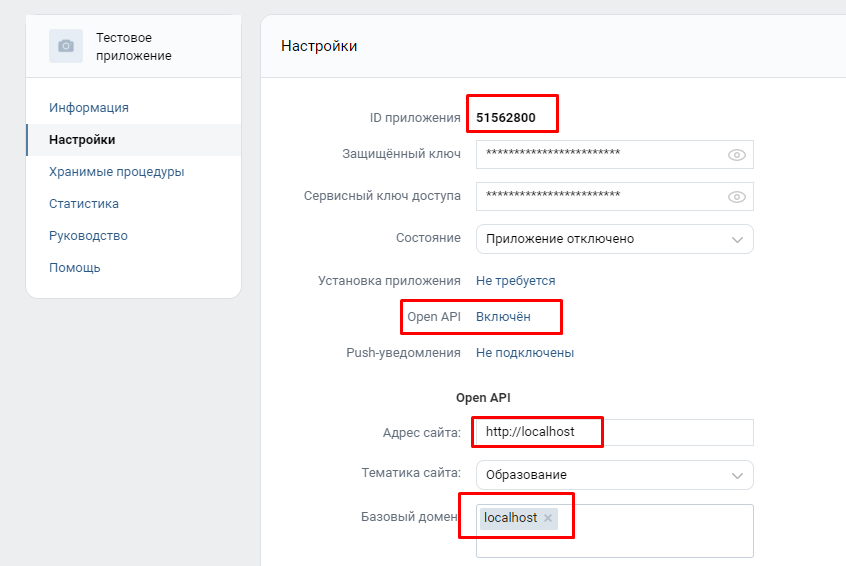
Далее нам потребуется провести первоначальную конфигурацию созданного приложения, без которой мы не сможем получить доступ к API вконтакте с вашего сайта.

Итак, опять по шагам:

1. Переход на страницу [Приложения](http://vk.com/apps) и жмем кнопку "Настройки". Далее ищем ваше созданное приложение и жмем кнопку "Редактировать".



1. При желании вы можете пройтись по всем вкладкам и заполнить все необходимые для вас поля.
2. Переходим на вкладку "Настройки" и заполняем обязательные поля: адрес сайта, Базовый домен (домены которым разрешен доступ к API). Значение "ID приложения" вам необходимо запомнить, оно нам пригодится в дальнейшем.

**Также следует Включить Oopen API**

1. Не забудьте сохранить изменения.

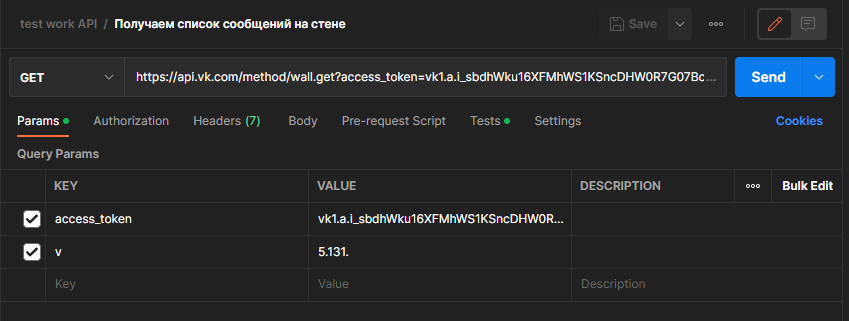
Далее, что нам необходимо - это получить доступ вашего сайта к методам API. Для этой процедуры ВК использует [Авторизация на базе OAuth 2.0](http://vk.com/dev/auth_sites). Принцип авторизации очень простой:

1. Формируем ссылку http://api.vkontakte.ru/oauth/authorize?client\_id=&scope=&redirect\_uri=http://api.vk.com/blank.html&response\_type= где:  
   client\_id - Ваш id приложения;  
   scope - запрашиваемые права доступа приложения, которые можно перечислять через запятую. В данном случае приложение запрашивает доступ к API со стороннего сервера в любое время (offline) и wall – доступ к методам работы со стеной;  
   **redirect\_uri** - ссылка для возврата ответа сервера. В нашем случае обязательный параметр равный http://api.vk.com/blank.html, т.к. только данный способ авторизации имеет возможность получить доступ к методам работы со стеной.  
   **response\_type** - собственно тот параметр который мы хотим получить - access\_token.
2. Переходим по получившейся ссылке(в этот момент вы должны быть авторизованы) и разрешаем доступ приложению.
3. Далее Вас перебрасывает на ссылку http://api.vk.com/blank.html с рядом параметров. Нужный нам параметр access\_token. Копируем его значение куда-нибудь. Оно нам необходимо для доступа к API.

**2. Получение списка сообщений на стене**

Для получения списка сообщений на стене используем следующий GET запрос:

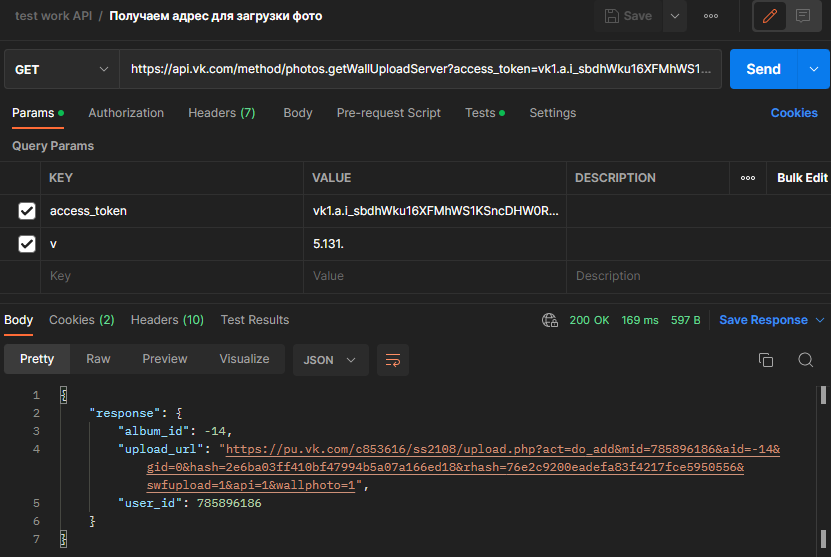
https://api.vk.com/method/wall.get?access\_token=(**значение полученного токена без скобок**) &v=5.131.



* 1. **Получение адреса для загрузки картинки**

Для получение адреса для загрузки картинки отправляем GET запрос с методом [photos.getWallUploadServer](https://vk.com/dev/photos.getWallUploadServer).

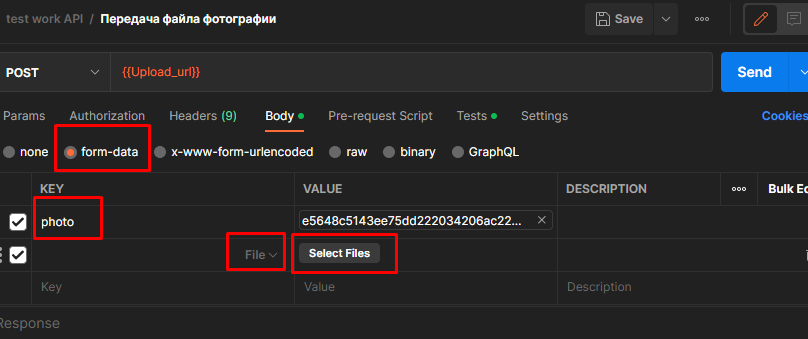
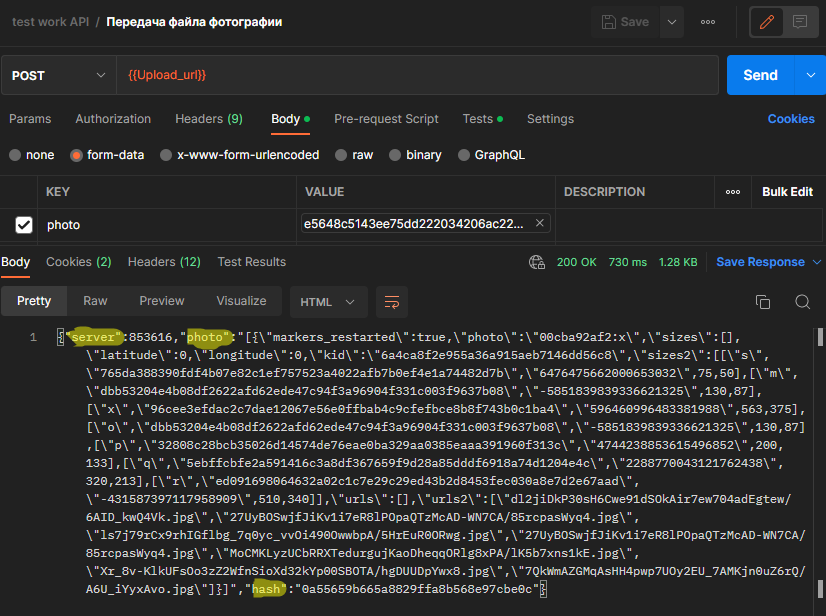
Пример запроса: https://api.vk.com/method/**photos.getWallUploadServer**?access\_token=(**значение полученного токена, без скобок**)&v=5.131.

Из ответа сервера нам нужно будет запомнить параметр «upload\_url», для следующей операции.  


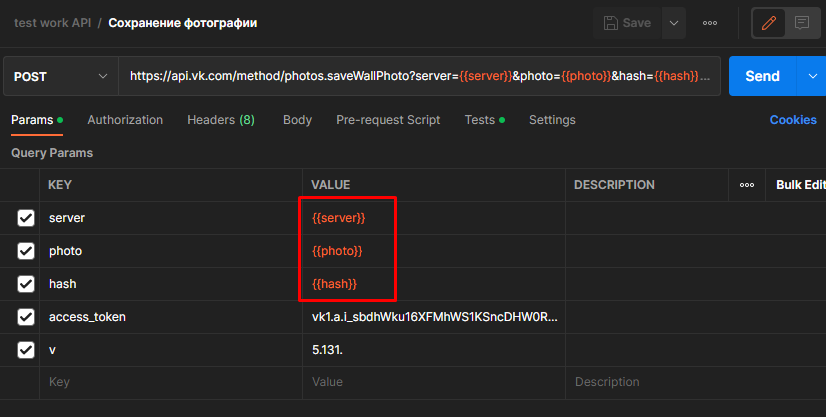
* 1. **Передача файла (картинки) на полученный адрес**

Передача картинки осуществляется методом POST на адрес, который мы получили ранее («**upload\_url**»). Параметры в данном запросе не заполняем. В теле запроса указываем параметр передачи данных **form-data**. Указываем поле **photo** (тип данных file). Далее прикрепляем картинку, нажав на кнопку «**select file**».

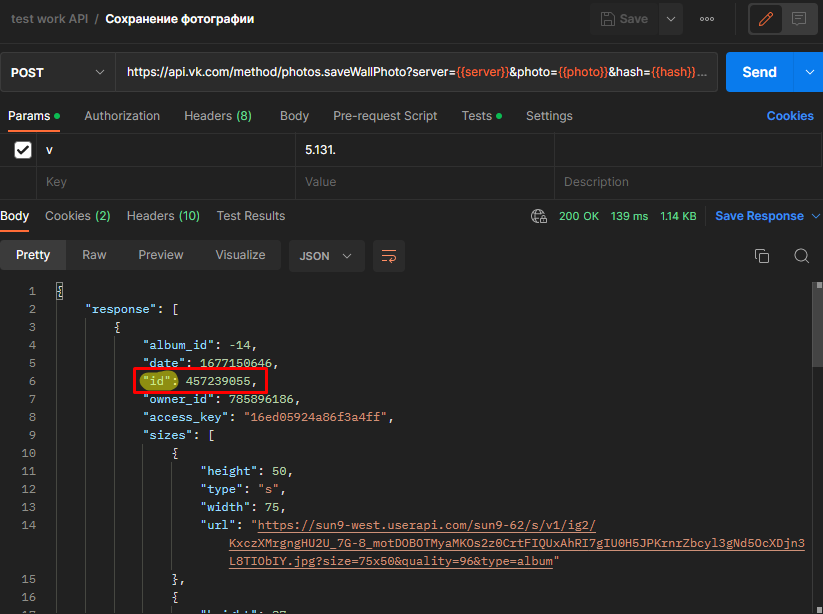
Допустимые форматы картинки: JPG, PNG, GIF.  
Ограничения: не более **6** фотографий за один раз в методе [photos.saveWallPhoto](https://vk.com/dev/photos.saveWallPhoto), сумма высоты и ширины не более 14000px, файл объемом не более 50 МБ, соотношение сторон не менее 1:20.

  
  
После успешной загрузки сервер возвращает в ответе JSON-объект с полями **server**, **photo**, **hash**:  
  
  
  
Поле **server** содержит числовое значение, в **hash** возвращается строка, а в поле **photo** находится JSON-объект, структуру которого важно сохранить в исходном виде.

1. **Сохранение картинки на полученном адресе**

Чтобы сохранить фотографию, нужно вызвать метод [**photos.saveWallPhoto**](https://vk.com/dev/photos.saveWallPhoto) с параметрами **server**, **photo**, **hash**, полученными на предыдущем этапе и осуществить POST запрос.  
Пример запроса: https://api.vk.com/method/photos.saveWallPhoto?server={{server}}&photo={{photo}}&hash={{hash}}&access\_token=(**значение полученного токена, без скобок**)&v=5.131.  


В полученном ответе от сервера мы запоминаем **ID добавленной картинки**

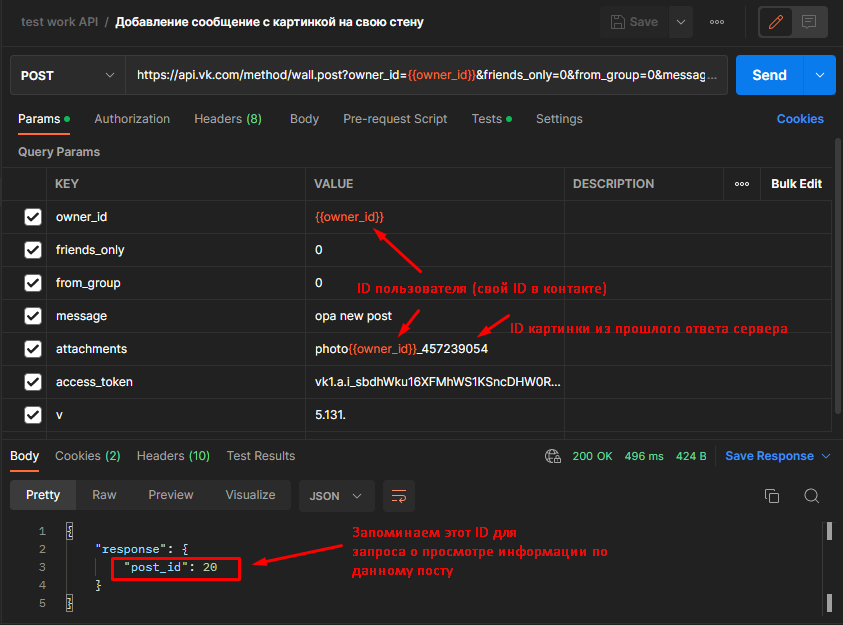


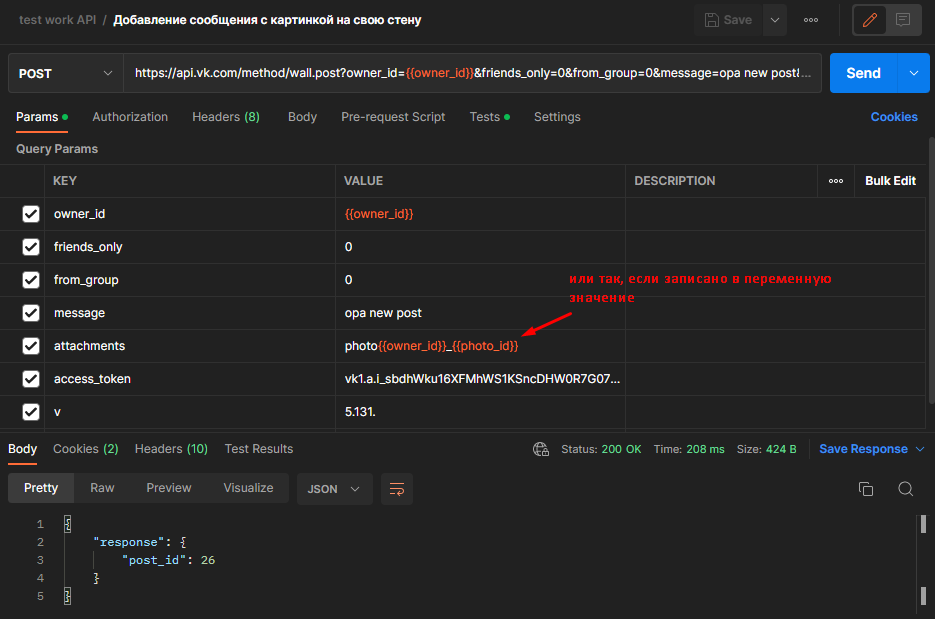
1. **Публикация картинки и сообщения на стену**

После успешной загрузки фотографии Вы можете разместить её на стене, опубликовав запись с помощью метода [wall.post](https://vk.com/dev/wall.post) и указав идентификатор фотографии в формате "photo" + {owner\_id} + "\_" + {photo\_id} (например, "photo12345\_654321") в параметре **attachments**. В {owner\_id} необходимо указывать то же значение, которое пришло Вам в ответе от метода [photos.saveWallPhoto](https://vk.com/dev/photos.saveWallPhoto).

В параметре **message** прописываем текст нашего сообщения.

Пример запроса: https://api.vk.com/method/wall.post?owner\_id={{owner\_id}}&friends\_only=0&from\_group=0&message=opa new post&attachments=photo{{owner\_id}}\_457239054&access\_token=(**значение полученного токена, без скобок**)&v=5.131.





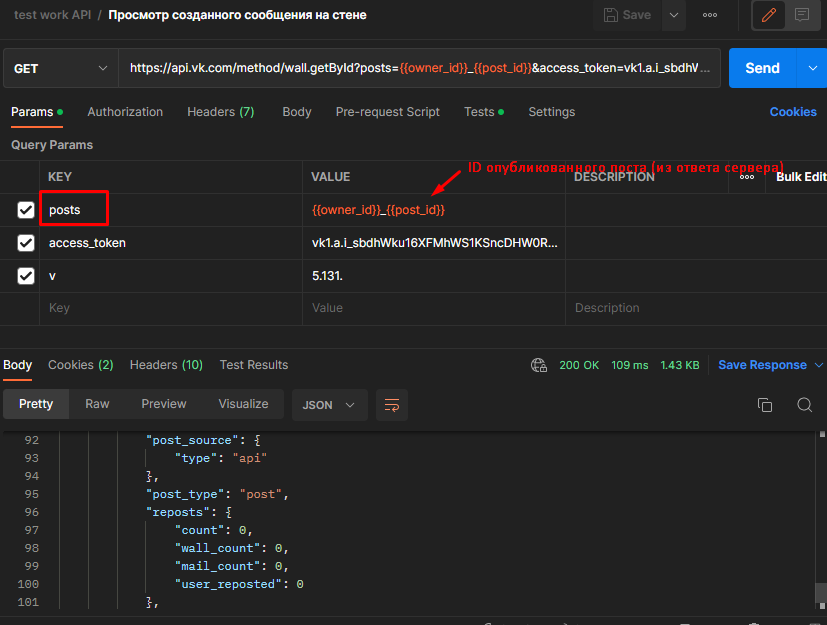
В ответе от сервера будет «**post\_id**» - это идентификатор нашего поста. Запомним его для того чтобы посмотреть информацию только об этом посте.

1. **Просмотр конкретного созданного сообщения на стене**

Для просмотра информации по созданной записи на стене используем метод **getById** и следующий GET запрос:

https://api.vk.com/method/wall.getById?posts={{owner\_id}}\_{{post\_id}}&access\_token==(**значение полученного токена, без скобок**)&v=5.131.

В параметрах необходимо указать: posts со значениями owner\_id\_post\_id (между этими значениями используем нижнее подчеркивание).



1. **Получение списка сообщений на стене**

**(удостоверяемся, что наше сообщение есть в списке)**

Для получения списка сообщений на стене используем следующий GET запрос:

https://api.vk.com/method/wall.get?access\_token=(**значение полученного токена без скобок**) &v=5.131.

В ответе сервера мы получаем JSON в котором визуально можем удостовериться, что наша запись существует на стене. Также должно увеличиться на +1 общее количество записей на странице («**count**»).

