**Структура API запроса**

Some parts of an API request are optional depending on what kind of request it is and what you are trying to do with it, but there are two pieces that are required for every API request. Every API request needs an **endpoint** and an **action**.

***API endpoints* -** Postman is asking you to put in a URL because an API endpoint is just a URL.

Ex: <https://api.github.com/users/djwester/repos>

You will often see an API endpoint specified without the base part of this API. EX: /users/:username/repos

If you see API endpoints listed that start with a / instead of with **http** or **www**, just remember that you need to go and find the base API URL for the endpoint in order to call it.

***API actions (also called Verbs)* -** The four actions (**GET**, **POST**, **PUT**, and **DELETE**) are sometimes summarized with  
the acronym **CRUD**. This stands for **C**reate, **R**ead, **U**pdate, and **D**elete. In an API, the **POST** action is used to create new objects, the **GET** action is used to read information about objects, the **PUT** action is used to modify existing objects, the **DELETE** action is used to delete objects. The rest of the actions (**PATCH**, **OPTIONS**. **HEAD**, …) are not often used and you will probably not encounter them much in the applications that you test and create.

API actions and endpoints are required for web APIs, but there are several other important pieces to API requests that we will consider.

***API parameters*** - API parameters are used to create structure and order in an API.

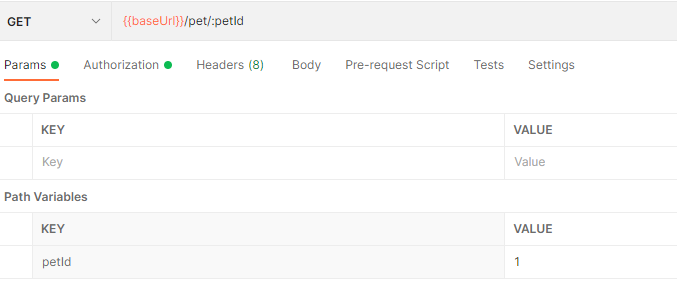
*Request parameters* - Part of the endpoint is a **parameter** (ex: name in github API). For example, the GitHub documentation uses a colon in front of the request parameter to indicate that it is a request parameter and not just another part of the endpoint. You will see endpoints specified like this in the GitHub documentation: /users/:username/repos.

In other APIs you will see request parameters enclosed in curly braces instead. In that case, the endpoint would look like /users/{{username}}/repos

We can either specify directly

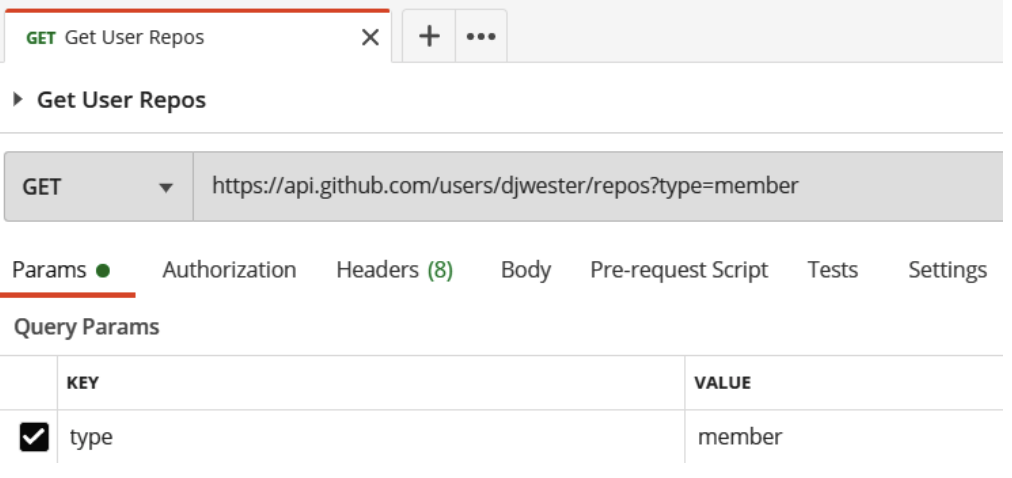


Or can use Path Variable



Often used to select specific user, or other entity.

*Query parameters* - A query parameter often acts like a kind of filter or additional action that you can apply to an endpoint. They are represented by a question mark in the API endpoint and are specified with a key that is the item you are querying for, and a value, which is what you want the query to return.



Make sure you have the **Params** tab selected and then put the name of the query parameter into the **Key** field and the value into the **Value** field. In this case, we will use the **type** parameter, so enter that word into the **Key** field.

Параметры можно редактировать через Bulk Edit. Полезно что бы кому то их отправить

***API headers*** - Every API request needs to include some **headers**. Headers include some of the background information that is often not that important to human users, but they help the server have some information about the client that is sending the request. Often we can just let the tool that we are using send the default headers that it needs to send without worrying about it.

***API body*** - If you want to create or modify resources with an API, you will need to give the server some information about what kind of properties you want the resource to have. This kind of information is usually specified in the **body** of a request. Most of the time, **GET** requests do not require you to specify a body. Other types of requests, such as **POST** and **PUT**, which do require you to specify a body, often require some form of authorization since they allow you to modify data

В основном используется row > JSON. По кнопке beatify можно выровнять JSON файл, что бы в нем были отступы

Можно передавать файлы. form-data > key(file) > values(selectFile)

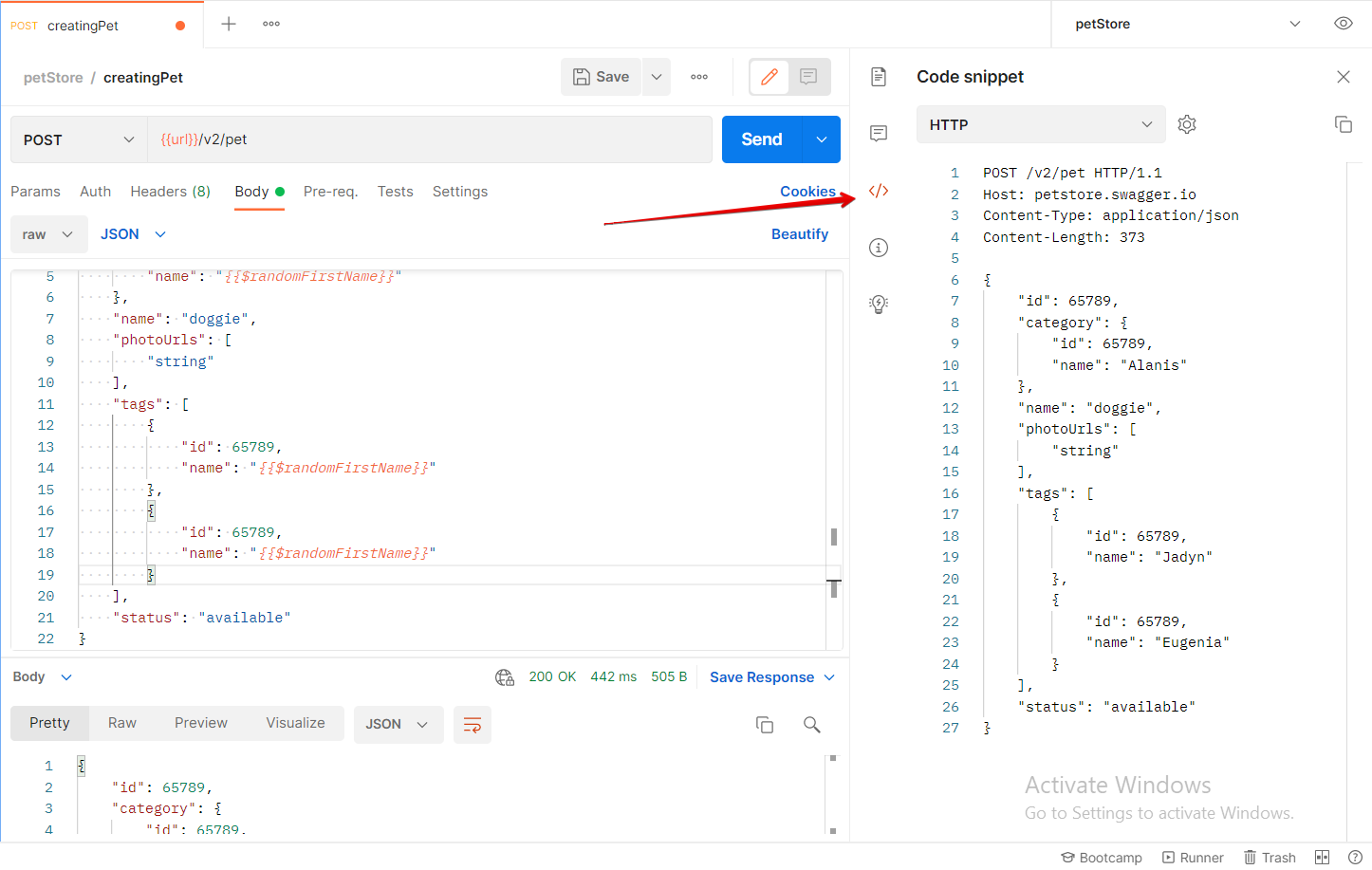
***API response*** - we send data to the server in the request, but then the server processes that request and sends back a response. There are a few different aspects to the response. The most obvious one is the **BODY** of the response. This is usually where most of the information that you are looking for will be included. An API response can also include a few other things such as cookies and headers. These kinds of things can be very important hints as to what is going on when testing or creating APIs

**JSON**

Если сервис поддерживает возможность передачи нескольких сущностей, то можно их передать, добавив в массив



**Code > HTTP** – можно смотреть в raw формате что мы отправляем на сервер



Можно также увидеть запрос в **cURL** формате

