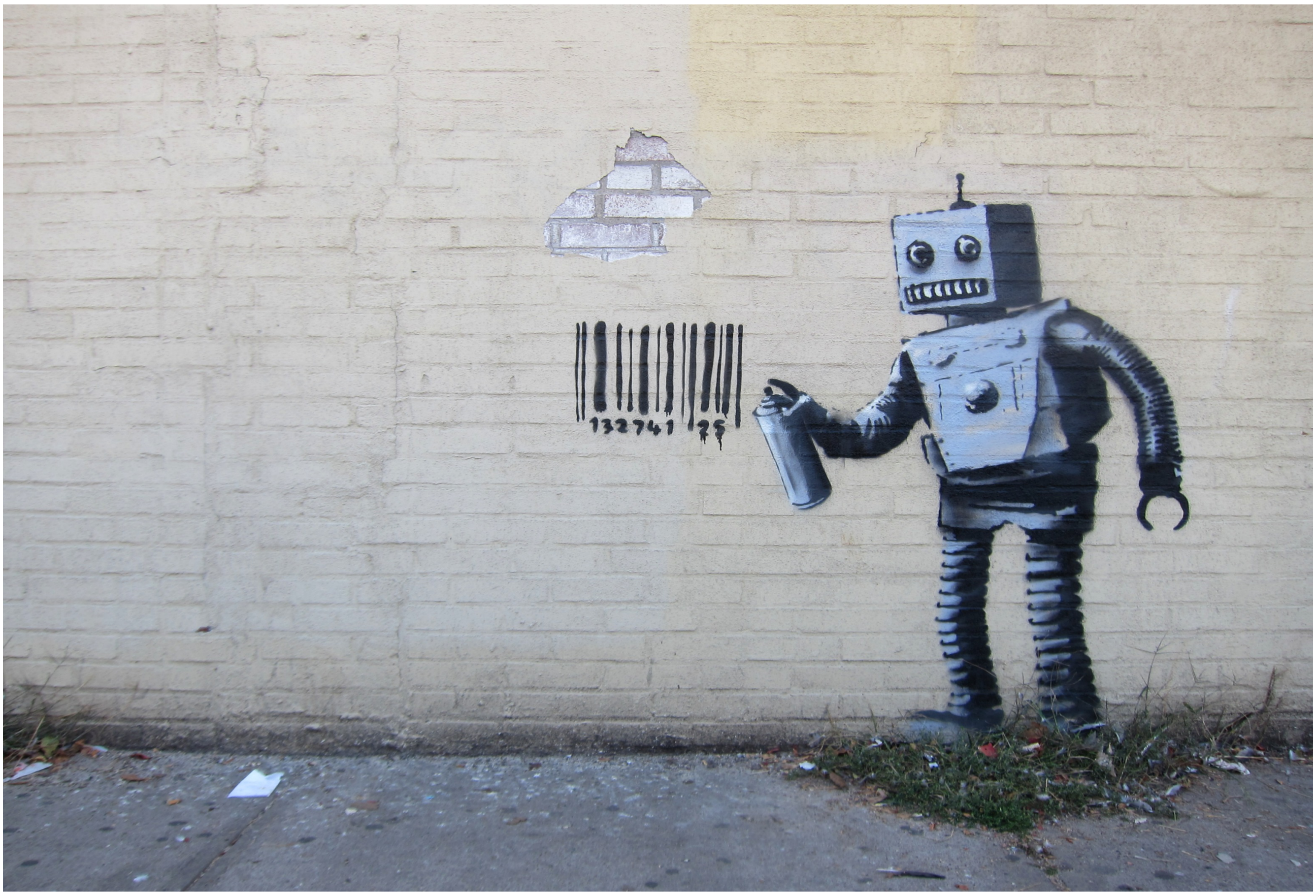
BEYOND THE GRAPHICAL USER INTERFACE:

INTRODUCTION TO MEDIA WIKI API USE AND BOT CREATION PROCESS



©Scott Lynch

User : **Mah3110** – WikiIndaba Tunis 2018

This tutorial is to have you go beyond the graphical User Interface (GUI) and to discover how to use the Media Wiki API and the bot creation process. We will first discover the API use and after that discover the bot creation process.

MEDIA WIKI API USE

What is an API?

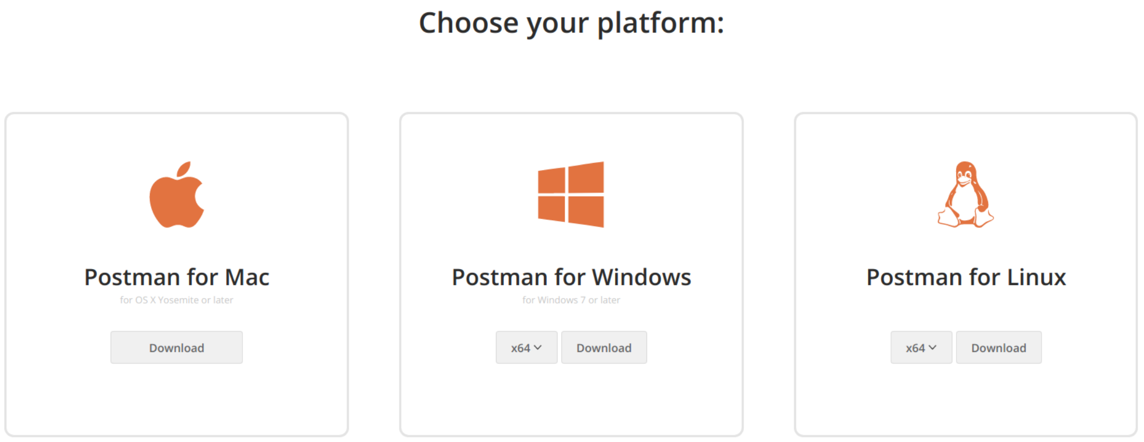
An Application Programming Interface is an interface used to fetch or to post data from or to a distant server. This process is based on the couple **request – answers** passing parameters to a URL. An API is based on the HTTP protocol. Requests are made using some methods.
To fetch data from a server, the GET method is the one to be used and to post a data the method to use is POST. There are others methods (delete, update) that we are not going to cover in this tutorial. The response returned by the server can be in different format: JSON, XML, HTML etc…
The format you choose depends on what you will do with the data. In the case of this tutorial we will be using the JSON format.

Fetching data from media wiki server

Let's see how you can get data from wikimedia server .
First, we need to install Postman that is a free tool to make API use easier.

Installing Postman app

Postman is available for different operating system(windows, Linux and Unix).
Go to postman download page (postman download link) and grab the version matching your platform characteristics.



If you don’t want to install the Postman native app, there is a chrome extension you can add to your browser. Grab the extension here. (postman chrome extension)

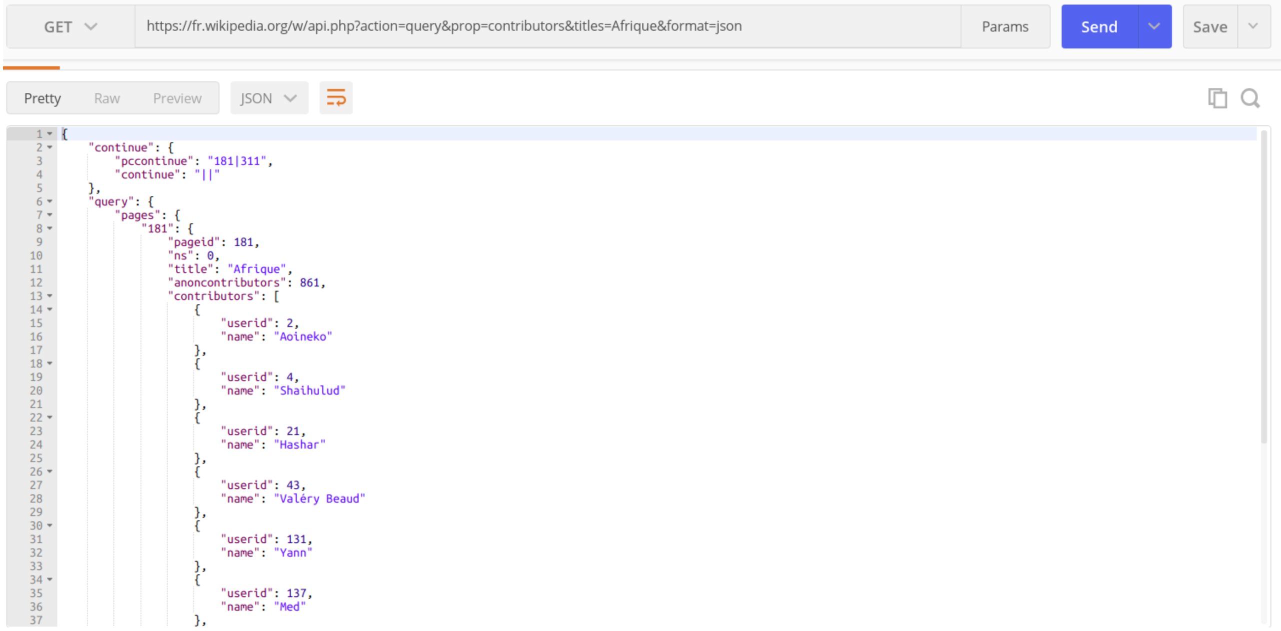
Making requests

Now that we have postman installed, let’s start making our first **GET** request to Media Wiki server. For that, we need first to know how Media Wiki API works. There is a complete documentation on that with methods and parameters to pass to urls. The documentation is available here (wikimedia api doc).
As per the documentation to get data from there server, request should be sent to the endpoint(where the resource is located) https://en.wikipedia.org/w/api.php with valid parameters. For French site, change **en** to **fr** in the endpoint. Endpoint and parameters are separated with the **(?)** character. Parameters are separated with the **(&)** character. If many values, there should be separated with **(|)** character.
Here are some parameters to pass in the url:
action = query: to fetch data from and about mediaWiki.
action = edit: to create and edit pages
action = thanks: to send a thank you message to an editor.
For each action, there are others specifics parameters to be associated. Check full list of action and there usage in the documentation.

Let’s see some cases:
<u>Case 1</u>: **Get username, id of contributors to the page Afrique (in french site)**
here is the request url:

https://fr.wikipedia.org/w/api.php?action=query&prop=contributors&titles=Afrique&format=json

Then open postman click on get(as you’re making a get request). Enter the request name(any name) and the request description(what your api request is for) and then add it to a collection or create a new one and save. On the next page, enter the request url and click on send to submit the request.
In the body field, you should get the response:



https://fr.wikipedia.org/w/api.php?action=query&prop=contributors&titles=Afrique&format=json

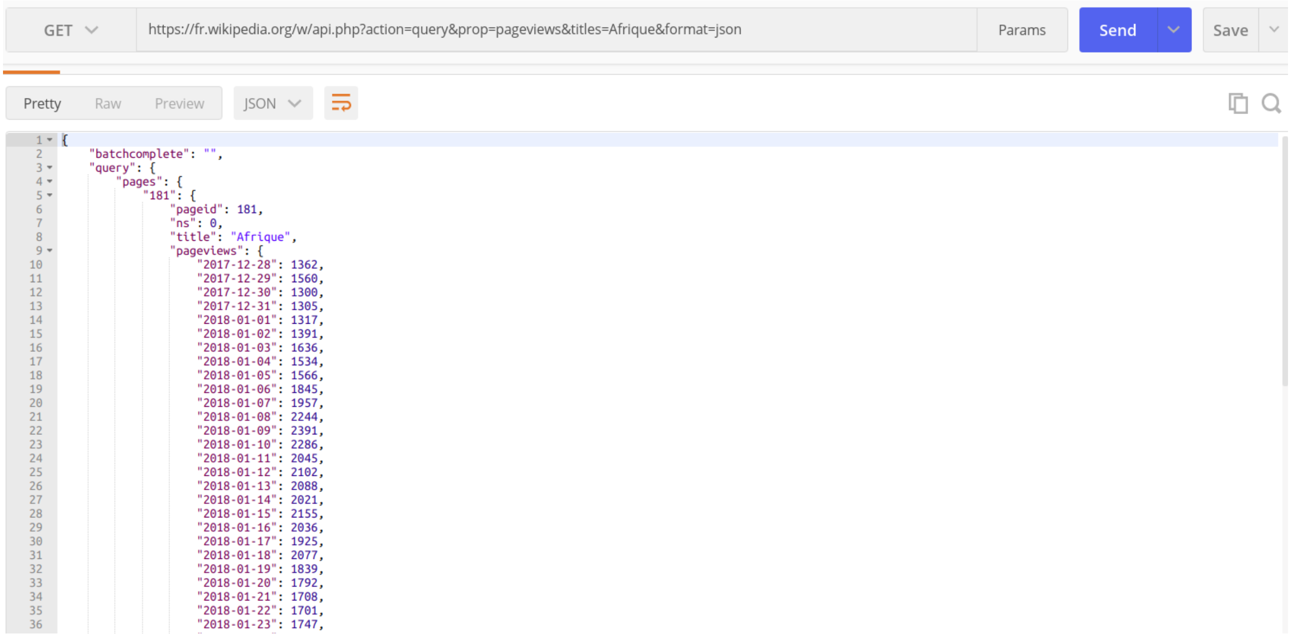
```json
{
    "continue": {
        "pccontinue": "181|311",
        "continue": "||"
    },
    "query": {
        "pages": {
            "181": {
                "pageid": 181,
                "ns": 0,
                "title": "Afrique",
                "anoncontributors": 861,
                "contributors": [
                    {
                        "userid": 2,
                        "name": "Aoineko"
                    },
                    {
                        "userid": 4,
                        "name": "Shaihulud"
                    },
                    {
                        "userid": 21,
                        "name": "Hashar"
                    },
                    {
                        "userid": 43,
                        "name": "Valéry Beaud"
                    },
                    {
                        "userid": 131,
                        "name": "Yann"
                    },
                    {
                        "userid": 137,
                        "name": "Med"
                    },
```

<u>Case 2</u>: Get views of the page Afrique (in french site)

Here is the request url:

https://fr.wikipedia.org/w/api.php?action=query&prop=pageviews&titles=Afrique&format=json

And response:



https://fr.wikipedia.org/w/api.php?action=query&prop=pageviews&titles=Afrique&format=json

```json
{
    "batchcomplete": "",
    "query": {
        "pages": {
            "181": {
                "pageid": 181,
                "ns": 0,
                "title": "Afrique",
                "pageviews": {
                    "2017-12-28": 1362,
                    "2017-12-29": 1560,
                    "2017-12-30": 1300,
                    "2017-12-31": 1305,
                    "2018-01-01": 1317,
                    "2018-01-02": 1391,
                    "2018-01-03": 1636,
                    "2018-01-04": 1534,
                    "2018-01-05": 1566,
                    "2018-01-06": 1845,
                    "2018-01-07": 1957,
                    "2018-01-08": 2244,
                    "2018-01-09": 2391,
                    "2018-01-10": 2286,
                    "2018-01-11": 2045,
                    "2018-01-12": 2102,
                    "2018-01-13": 2088,
                    "2018-01-14": 2021,
                    "2018-01-15": 2155,
                    "2018-01-16": 2036,
                    "2018-01-17": 1925,
                    "2018-01-18": 2077,
                    "2018-01-19": 1839,
                    "2018-01-20": 1792,
                    "2018-01-21": 1708,
                    "2018-01-22": 1701,
                    "2018-01-23": 1747,
```

BOTS CREATION

What is a bot?

A bot is a script developed to automate some actions.
On Wikimedia there are many bots that administrators used to automate there tasks like mass pages deletion, account creations, block a user or to send emails.
In this tutorial, we will be using the Media Wiki API as we seen it before but now using **POST** method as we are going to make changes on the server.

Methods

As per the document there are many actions to be used to have a bot do an action depending on user’s rights. Here are some methods:
Action = edit: to create a new page or to edit an existing one.
action = delete: to delete a page etc. Check documentation for full list of actions used and specific parameters associated to each action.
Let see in action how it works :

Creating your first bot

Case 1**: Create a bot that will edit the user sandbox page. The bot will create the page if this doesn’t already exist**

To do this, we need first to request a page edit /create page token from media Wiki server. This token is for security issues. You can get your token by making a GET request to the Media Wiki server. Here is the url:

https://fr.wikipedia.org/w/api.php?action=query&meta=tokens&format=json

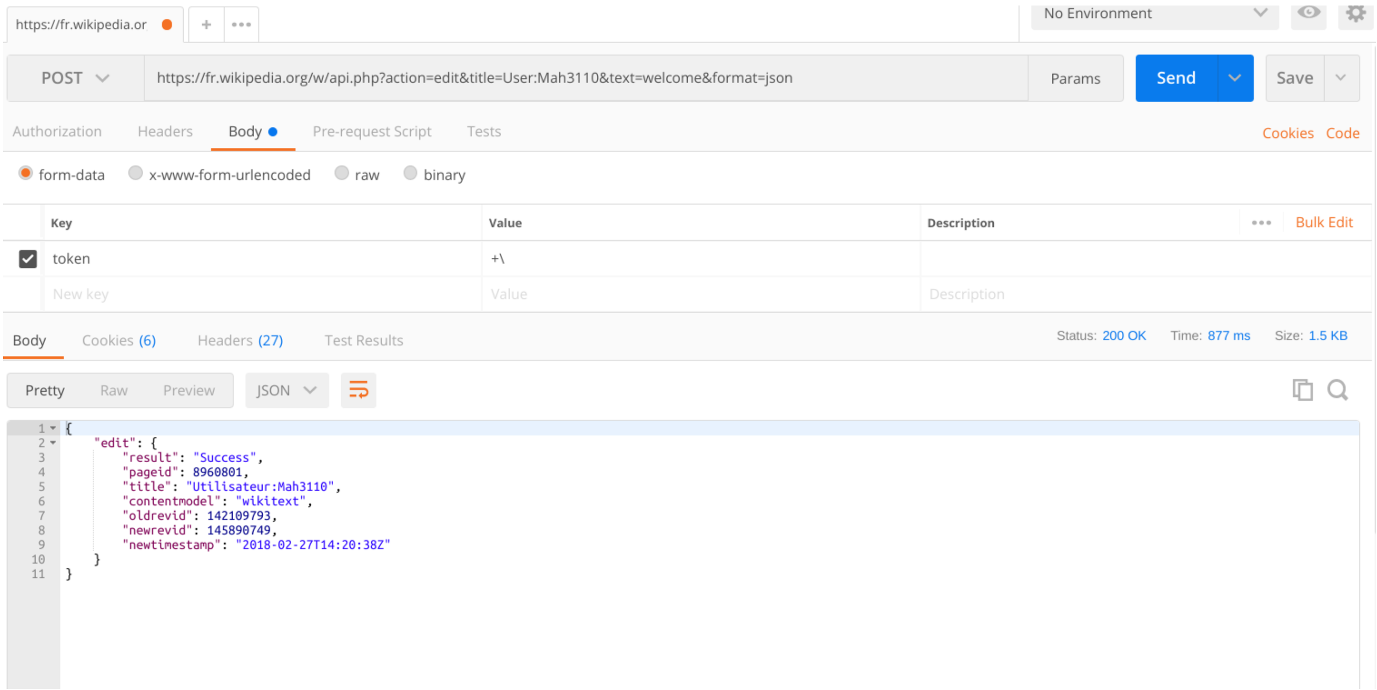
The token you get depends on your rights on Wikimedia . By default you get the edit & create right.

Here is the url to use to edit my sandbox page:

https://fr.wikipedia.org/w/api.php?action=edit&title=User:Mah3110/Brouillon&text=Hello world&format=json

Access postman, then change the GET method to POST and enter the request url.
Add the token in the request body.

When using postman, don’t add the token you get to the url. Set it in the request body as you can see in the screen shot. If your request is valid, you should get a success response from the server.



POST https://fr.wikipedia.org/w/api.php?action=edit&title=User:Mah3110&text=welcome&format=json

Body: ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key	Value	Description
☑	token	+\	
	New key	Value	Description

Body — Status: 200 OK Time: 877 ms Size: 1.5 KB

JSON

```
{
    "edit": {
        "result": "Success",
        "pageid": 8960801,
        "title": "Utilisateur:Mah3110",
        "contentmodel": "wikitext",
        "oldrevid": 142109793,
        "newrevid": 145890749,
        "newtimestamp": "2018-02-27T14:20:38Z"
    }
}
```

The edit action overwrites the page content. To avoid that, you can pass in the url, the appendtext or the prependtext parameters to add text at the end of a section or to add text before a section.

<u>Case 2</u>: append a new text block to the user sansbox content.

Here is the request url:

https://fr.wikipedia.org/w/api.php?action=edit&title=User:Mah3110/Brouillon&appendtext= Hello world after&format=json

Final notes

it is recommended to have your bot validated by Media Wiki dev team. That way others contributors can use. Have a look at the already existing bots before starting developing a new one.
There are many libraries with pre-built bots functions that you can reuse. Th most common one is **Pywikibot**. You can check all existing bots here: bots list
Now guys, you know what is behind the GUI, API use and how to automate task.

Happy coding!