**Note: This is an exercise only to show how Python/Django and Django-Rest-Framework can work together by using an existing API and creating a new one.**

**This new API has two endpoints:**

1. [**http://localhost:8000/v4/toppages/hostname**](http://localhost:8000/v4/toppages/hostname)
2. [**http://localhost:8000/v4/toppages/create**](http://localhost:8000/v4/toppages/create)

**The first one will show all the pages related to one host or domain and some statistics like visitors, page, path, etc. Also it can provide information about and increase of the concurrents or visitors.**

**The second one will help us create new domains or host through the API. Also can show us all the existents domains so far.**

**All of the responses from this API are in JSON format.**

[**http://localhost:8000/v4/toppages/hostname**](http://localhost:8000/v4/toppages/hostname)

**GET**

* **host:** receives a valid or existent hostname and can give you all the pages related to that host like the example below:

[**http://localhost:8000/v4/toppages/hostname?host=gizmodo.com**](http://localhost:8000/v4/toppages/hostname?host=gizmodo.com)

**…..**

**{  
 "i": "Buzz Aldrin Proves the Federal Government Has A Form for Everything",  
 "path": "/buzz-aldrin-proves-the-federal-government-has-a-form-fo-1721587666",  
 "visitors": 2057,  
 "domain\_id": 11  
 },**

**…..**

**The data is interpreted as follows:**

**i:** is the title of the page.

**path:** is the specific resource without the domain.

**visitors:** visitors so far in that page.

**domain\_id:** is the DB relation with the host or domain.

* **change:** will give you, every 5 seconds, the increasing concurrents or visitors to the page. For example:

[**http://localhost:8000/v4/toppages/hostname?host=gizmodo.com&change=True**](http://localhost:8000/v4/toppages/hostname?host=gizmodo.com&change=True)

**…..**

**{  
 "i": "Windows 10, Day Five",  
 "path": "/windows-10-day-five-1720608388",  
 "change": 452  
 },**

**…..**

**The data is interpreted as follows:**

**i:** is the title of the page.

**path:** is the specific resource without the domain.

**change:** represents the difference of visitors between now and 5 seconds(configurable)

[**http://localhost:8000/v4/toppages/hostname/create**](http://localhost:8000/v4/toppages/hostname/create)

**GET**

When you make a get request to this API you get back all the domains or hosts created so far:

**…..**

**{  
 "domain\_name": "**[**http://www.gizmodo.com**](http://www.gizmodo.com/)**"  
 }**

**…..**

**POST**

When you make a post request to this API you get back a 201 response that the host was created.

**Technical Specifications and Requirements**

**Requirements**

The requirements to make this API works are as follows:

**Python 2.7.6**

**Requests (2.7.0)**

**Django (1.8.3)**

**djangorestframework (3.1.3)**

**SQLite3**

**Simulators and tools**

In order to test this API faster I’ve created to additional tools:

**api\_requests.py**

This tool can help you create a host for the new API and calls the original API and retrieves all its pages. I have put a configurable limit. The default HOSTNAME is gizmodo.com. I took it from the AWS site that you put in the challenge.

**WARNING**

Since is a tool you cannot run it twice with the same host. It can be fix but there was no time for me to do it. Another thing is it must be run in the same directory that it is sitting right now because it uses Django’s settings in order to get to the DB. If you move it you’ll have to implement a bit more logic usings os or sys or both python modules.

**concurrent\_sym.py**

This is a simulator that will represent random users going to or visiting random websites that we have and will, every 5 seconds, change some fields in the DB in order to make the API work with the concurrent or visitors increase part (when you use change=True).

It’s all configurable inside this simulator. But I think is better to play with the time and the random numbers for visitors only.