Population API

The population section of the tmpst website is one of the more simpler of APIs to receive data from as there are no keys required to access the data from the website. The Population API is provided by the World Population API provides information on a variation of data from the world such as:

- World Population Rank
- Life Expectancy
- Population
- Mortality Distribution

These are divided by endpoints which are where the information will be accessed from. Each endpoint can return data in JSON, JSONP and HTML. For each endpoint you have a lot of customizability for querying the API such as:

- Sex
- Country
- Date
- Date of Birth
- Years Ago
- Has Diabetes
- Today VS Tomorrow

In this tutorial:

Pre-requisites:

- Understand the API (api.population.io) by using the sandbox displayed on the website to make calls to their API
- The endpoint that will be used will be the:
    - https://api.population.io/1.0/countries for a list of countries
    - https://api.population.io/1.0/population/{year}/{country}/ for population
- A suitable method for making a request (Postman: https://www.getpostman.com/downloads/ or in a new tab)

Aims:

- Learn how to request data from one of the World Population feeds in Postman
- Learn how to request data from one of the World Population feeds in a new tab
- Learn how to request data from one of the World Population feeds using AJAX call

Postman Tutorial

In this tutorial, we will create a request to a World Population API feed using the postman application on Windows. Before proceeding with this tutorial, please ensure you have access to the Postman application or are using the online version.

Steps:

1. Choose endpoint to request data from
2. Construct request in Postman

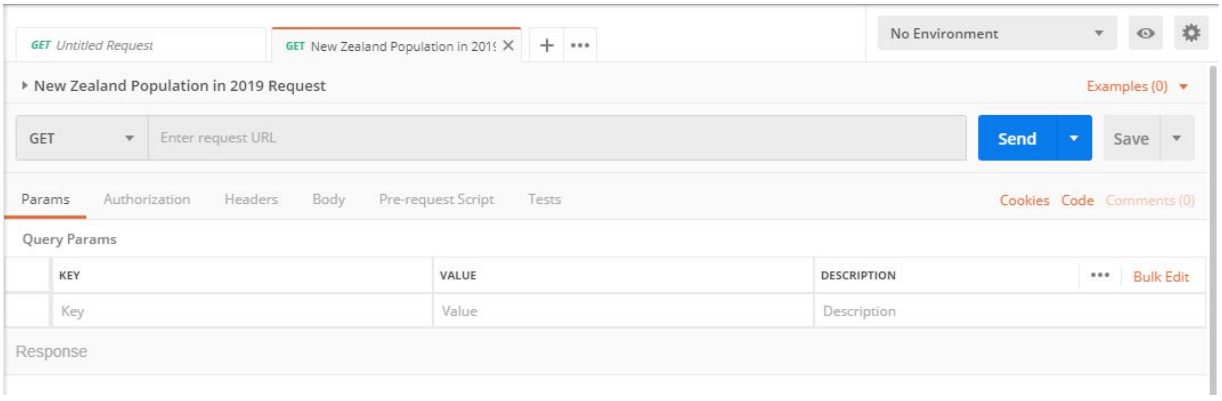
Step 1: Choose endpoint to request data from

As mentioned earlier, World Population offers many population feed endpoints to choose from. In this tutorial, not all sources will be shown because the request format is identical for all. It is important to keep in mind that not all sources will have data at the time of the request. For example, making a request for the population of a country 2000 years in the future won't yield any results from the request. With that in mind, the endpoint that will be used for this tutorial is New Zealand in the year 2019. This endpoint will always have data so it is good as an example. The endpoint is shown below:

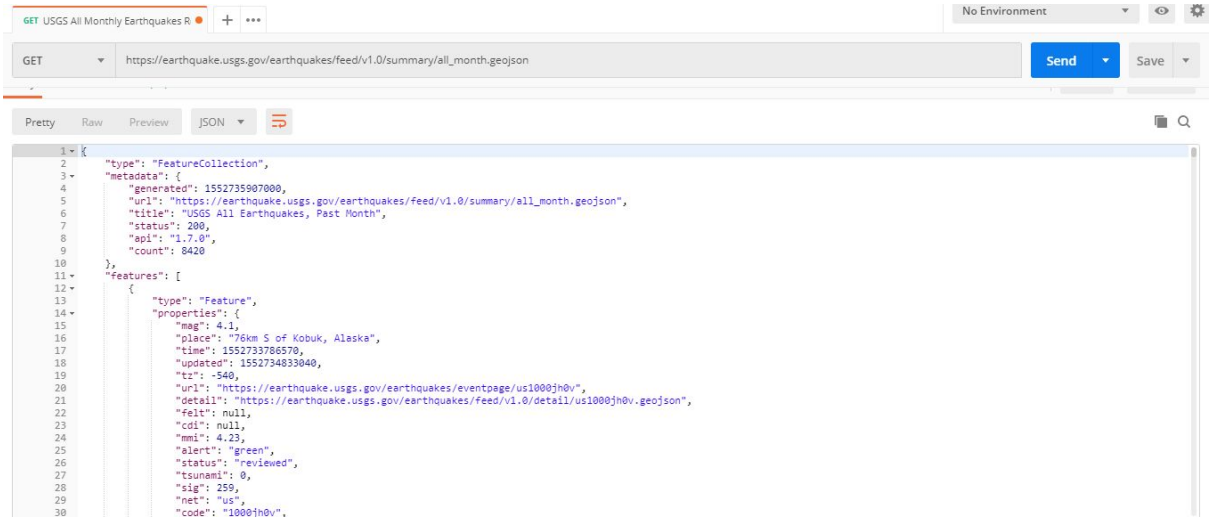
http://api.population.io/1.0/population/2019/New%20Zealand/

Step 2: Construct request in Postman

This tutorial assumes that you have a basic understanding of how to use Postman. If not, we provide a tutorial on how to utilise Postman properly that you may wish to read before you continue with this tutorial. If you know how to use Postman, create a new GET request and give it an appropriate name. The name I will use in this tutorial is "New Zealand Population in 2019 Request". If you have setup the request properly you should see the following:



Nothing will happen if you attempt to send the request as the url box is empty. Now all that is required is to add the url mentioned in step 1 to the url box and hit send. You should see the following:



Note: It is recommended that you add the format of the returned file. If you are going to use an AJAX call it is recommended to use JSONP format. Below are the formats the call is accepted in:

http://api.population.io/1.0/population/2019/New%20Zealand/?format=jsonp
http://api.population.io/1.0/population/2019/New%20Zealand/?format=json

Part of the JSON response is shown below in plain text:


```
[
 {
  "females": 28400,
  "country": "New Zealand",
  "age": 0,
  "males": 30000,
  "year": 2019,
  "total": 58400
 },
 {
  "females": 29000,
  "country": "New Zealand",
  "age": 1,
  "males": 30500,
  "year": 2019,
  "total": 59500
 },
```

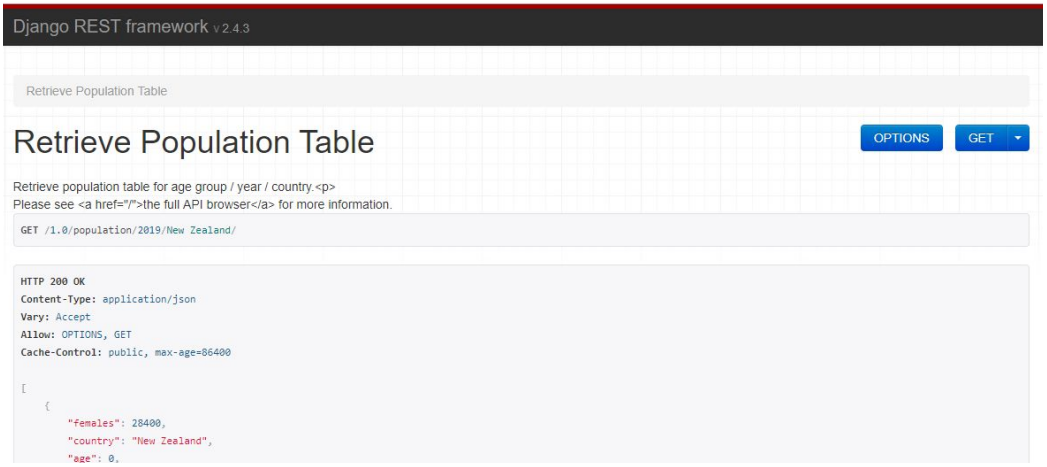

Not all the request will be show here as the response is too long for the document.

Browser Request Tutorial

In this tutorial, we will create a request to a World Population API population feed using a new tab on a Google Chrome browser. For this tutorial, we will use the same endpoint as is used in the Postman example:

http://api.population.io/1.0/population/2019/New%20Zealand/?format=jsonp

Step 1: Add url to tab and run
This tutorial is easier than either the Postman method or the AJAX method (still to come) as it only involves using a tab in a browser as if you are navigating to a web page. To get a result, add the url to the address bar and hit enter and you should see the following:



The result is shown in plain text below:

HTTP 200 OK
Content-Type: application/json
Vary: Accept
Allow: OPTIONS, GET
Cache-Control: public, max-age=86400


```
[
  {
    "females": 28400,
    "country": "New Zealand",
    "age": 0,
    "males": 30000,
    "year": 2019,
    "total": 58400
  },
  {
    "females": 29000,
```

```
    "country": "New Zealand",
    "age": 1,
    "males": 30500,
    "year": 2019,
    "total": 59500
},
```


Once again, the full response is not shown as it is too large.

AJAX Tutorial

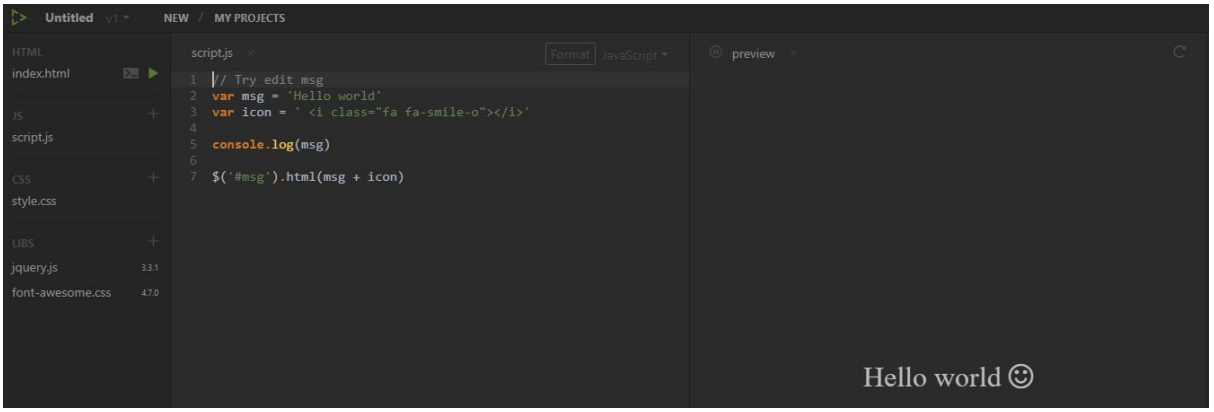
The above methods show you how to access the World Population feed and view the data, however, neither method involves any coding and neither would work well in an application's context. So, in this tutorial we will connect to the monthly feed using AJAX which can then be put into the javascript of any application

Steps:

1. Create a new playcode project
2. Add the provided AJAX code and run

Step 1: Create a new playcode project

For this tutorial, playcode will be used to demonstrate the AJAX request. Although we are using playcode, almost any online JavaScript IDE would work (JsFiddle). If you search for playcode and click on the link you should see the following default project setup:



Playcode is ideal for this tutorial as we only need to change the js file and view the console as jquery is referenced as standard.

Step 2: Add the provided AJAX code and run

The AJAX required to fetch the data programatically is:

```
$.ajax({
      type: "GET",
      url: "https://api.population.io/1.0/population/2019/New%20Zealand/?format=jsonp",
      contentType: "application/jsonp",
      success: function (result) {
         console.log('AJAX Response: ', result);
      },
      error: function (errorResult) {
         console.log('ERROR: ', errorResult);
      }
   });
```

If you replace the contents of script.js with the above AJAX and run the code, you should see the following:

As you can see this result looks similar to the other results we have seen earlier in the project, however, this code can be added to an application and the data can be fetched and altered dynamically.

Note: If playcode hangs or feels unresponsive when running this code there is no need to worry, their is a lot of data to process from this feed

Conclusion
That is the methods which can be used to request data from the World Population feeds. The most useful of the methods is AJAX, however, the others can be used for testing and ensuring the quality of the data.

Common Issues:
Normally, implementing the methods discussed in this tutorial are relatively easy, however, some of the most common issues are detailed below:

- Lack of responsiveness - Depending on the feed chosen, the method used to fetch the data can seem slow because it has to process thousands of ages in the population sometimes. Normally, the request should complete quite quickly without too much impact on the user.

Outcomes
You should now be able to:

- Request data from one of the World Population API population by year/country feed in Postman
- Request data from one of the World Population API population by year/country feed in a new tab
- Request data from one of the World Population API population by year/country feed using AJAX call

References
Although we hope this tutorial has been all the help you need, here are some useful links that may be of use:

Useful Resources:

- World Population API - http://api.population.io/
- Functional playcode AJAX request - https://playcode.io/268741?tabs=console&script.js&output

Tools:

- Postman - https://www.getpostman.com/downloads/
- Playcode - https://playcode.io/