Weather API

In the weather system, tmpst uses an API to fetch data depending on a user's query. The API used is provided by APIXU. The APIXU API provides multiple endpoints for different levels of a request such as current day, forecast and historical data as well as a forecast of astronomical data which is returned either in JSON or XML format. The APIXU data feeds require a key to be accessed and can be requested up to 10,000 times.

The available APIXU feed categories are shown below:

Available Data

- Current Weather
- Forecast up to 10 days
- Forecast Day (Which includes hourly updates and Astronomical data)
- Historical weather (For dates on or after the 1st of January 2015)

Data Fields

For each of the available data types there are similar fields that can be queried for information, some of the fields we have used on our website are;

- Temperature in Celsius
- Feels like temperature in Celsius
- Weather Condition
- Wind Speed
- Wind Direction
- Humidity
- Average Temp
- Max Temp

In this tutorial:

Pre-requisites:

- Understanding of JSON or XML format
- An APIXU API endpoint address (http://api.apixu.com/v1/current.json?key=&q=)
- An APIXU API key, which can be obtained from(https://www.apixu.com/signup.aspx)
- A suitable method for making a request (Postman: https://www.getpostman.com/downloads/ or in new tab)

Aims:

- Learn how to request data from one of the APIXU feeds in Postman

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

Postman Request Tutorial

In this tutorial, we will create a request to an APIXU 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

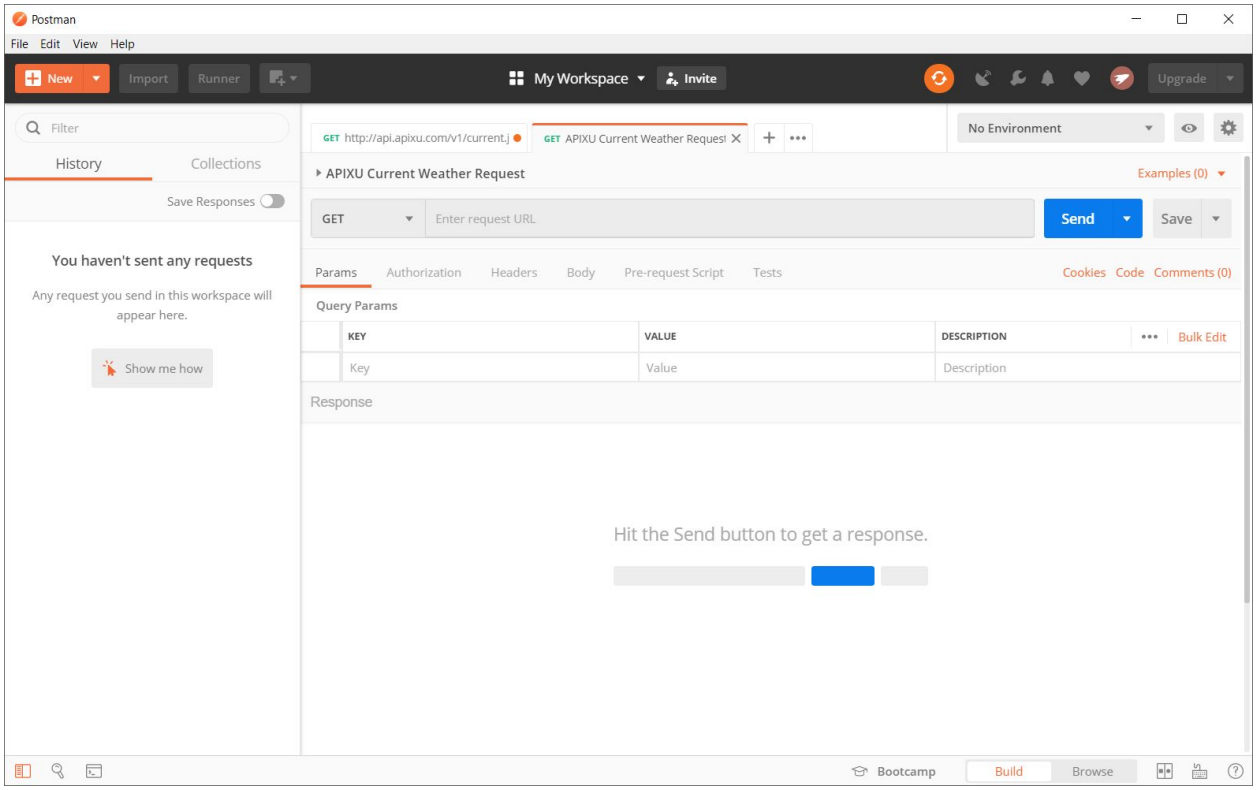
Step 1: Choose endpoint to request data from

As mentioned earlier, APIXU offers different weather feed endpoints to choose from. In this tutorial, not all sources will be shown because the request format is similar for all. It is important to keep in mind that not these sources should have data at the time of request that is either up to date or recent. For example, making a request for current weather should always return information as APIXU provides real-time data on the weather as it occurs. With that in mind, the endpoint that will be used for this tutorial is all current weather. This endpoint will almost always have data so it is good as an example. The endpoint is shown below, APIXU requires you search a city name, postcode, lat/long so in this example, we will just use Glasgow:

http://api.apixu.com/v1/current.json?key=[YOUR KEY HERE]&q=Glasgow

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 “APIXU Current Weather Request”. If you have set up 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:


```
{
    "location": {
        "name": "Glasgow",
        "region": "Glasgow City",
        "country": "United Kingdom",
        "lat": 55.86,
        "lon": -4.25,
        "tz_id": "Europe/London",
        "localtime_epoch": 1552844611,
        "localtime": "2019-03-17 17:43"
    },
    "current": {
        "last_updated_epoch": 1552843813,
        "last_updated": "2019-03-17 17:30",
        "temp_c": 8,
        "temp_f": 46.4,
        "is_day": 1,
        "condition": {
            "text": "Partly cloudy",
            "icon": "//cdn.apixu.com/weather/64x64/day/116.png",
            "code": 1003
        },
        "wind_mph": 5.6,
        "wind_kph": 9,
        "wind_degree": 310,
        "wind_dir": "NW",
        "pressure_mb": 1007,
        "pressure_in": 30.2,
        "precip_mm": 0.6,
        "precip_in": 0.02,
        "humidity": 66,
        "cloud": 50,
        "feelslike_c": 6.4,
        "feelslike_f": 43.6,
        "vis_km": 10,
        "vis_miles": 6.
```

The response is shown below in plain text:


```
{
   "location": {
      "name": "Glasgow",
      "region": "Glasgow City",
      "country": "United Kingdom",
      "lat": 55.86,
      "lon": -4.25,
      "tz_id": "Europe/London",
      "localtime_epoch": 1552844611,
      "localtime": "2019-03-17 17:43"
   },
   "current": {
      "last_updated_epoch": 1552843813,
      "last_updated": "2019-03-17 17:30",
      "temp_c": 8,
      "temp_f": 46.4,
      "is_day": 1,
      "condition": {
         "text": "Partly cloudy",
         "icon": "//cdn.apixu.com/weather/64x64/day/116.png",
         "code": 1003
      },
      "wind_mph": 5.6,
      "wind_kph": 9,
      "wind_degree": 310,
      "wind_dir": "NW",
      "pressure_mb": 1007,
      "pressure_in": 30.2,
      "precip_mm": 0.6,
      "precip_in": 0.02,
      "humidity": 66,
      "cloud": 50,
      "feelslike_c": 6.4,
      "feelslike_f": 43.6,
      "vis_km": 10,
      "vis_miles": 6,
      "uv": 3,
      "gust_mph": 18.1,
      "gust_kph": 29.2
   }
}
```

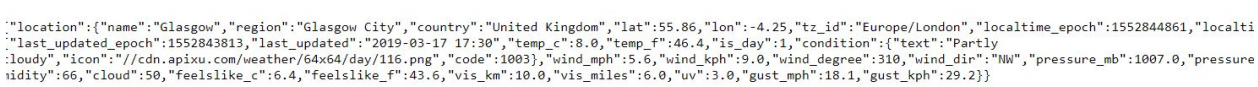
Browser Request Tutorial

In this tutorial, we will create a request to a APIXU 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.apixu.com/v1/current.json?key=[YOUR KEY HERE]&q=Glasgow

Step 1: Add URL to tab and run

This tutorial is easier than either the Postman method or the AJAX method 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:

{"location":{"name":"Glasgow","region":"Glasgow City","country":"United Kingdom","lat":55.86,"lon":-4.25,"tz_id":"Europe/London","localtime_epoch":1552844861,"localtime":"2019-03-17 17:47"},"current":{"last_updated_epoch":1552843813,"last_updated":"2019-03-17 17:30","temp_c":8.0,"temp_f":46.4,"is_day":1,"condition":{"text":"Partly cloudy","icon":"//cdn.apixu.com/weather/64x64/day/116.png","code":1003},"wind_mph":5.6,"wind_kph":9.0,"wind_degree":310,"wind_dir":"NW","pressure_mb":1007.0,"pressure_in":30.2,"precip_mm":0.6,"precip_in":0.02,"humidity":66,"cloud":50,"feelslike_c":6.4,"feelslike_f":43.6,"vis_km":10.0,"vis_miles":6.0,"uv":3.0,"gust_mph":18.1,"gust_kph":29.2}}

AJAX Tutorial

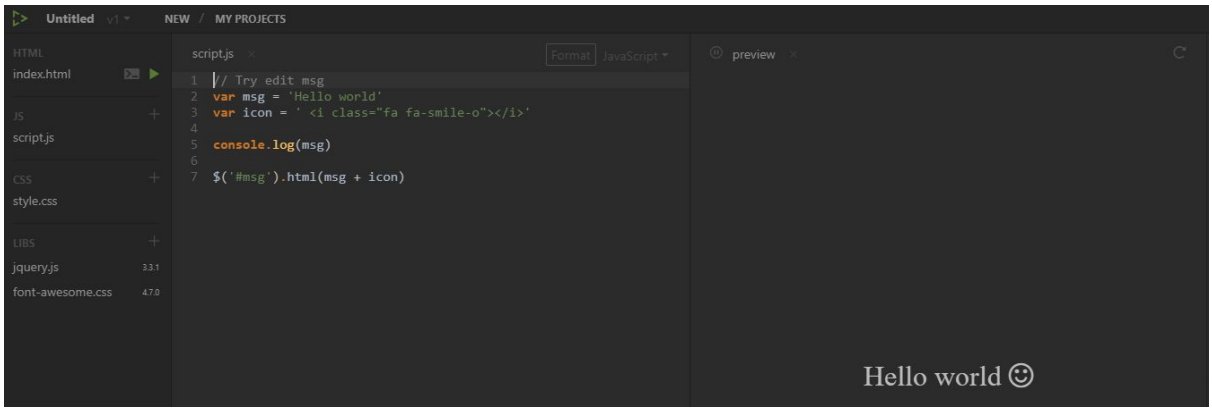
The above methods show you how to access a APIXU feed and view the data, however, neither method involves any coding and neither would work well in an application's context. So, in this tutorial, we will connect to the current weather feed using AJAX which can then be put into the javascript of any application

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 programmatically is:

```
$.ajax({
      type: "GET",
      url:
"https://api.apixu.com/v1/current.json?key=b2368c6f30c545c8b32105325190802&q=Glasgow",
      contentType: "application/json",
      success: function (result) {
         console.log('AJAX Response: ', result);
      },
```

```
    error: function (errorResult) {
        console.log('ERROR: ', errorResult.statusText);
    }
});
```

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.

Conclusion
These are the methods which can be used to request data from the APIXU feeds. The most useful of the methods is AJAX, however, the others can be used for testing and ensuring the quality of the data as well as understanding what information can be used to visualise.

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 lots of historical data sometimes, depending on your query. 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 APIXU feeds in Postman
- Request data from one of the APIXU feeds in a new tab
- Request data from one of the APIXU feeds 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:
- APIXU - Getting Started - https://www.apixu.com/doc/getting-started.aspx
- Functional playcode AJAX request - https://playcode.io/269219?tabs=console&script.js&output

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