GetMyWeather API

You may be interesetd in using the GetMyWeather API if you're developing a website or application related to the outdoors, including anything from camping to biking to outdoor wedding planning. You can use the GetMyWeather API to give your user the weather forecast at a specific area, including the temperature, wind speed, and chance of rain.

What makes GetMyWeather so special? The API provides a highly-customizable forecast, allowing you to get the forecast at a very specific area, which is called a "micro-forecast." For example, you can use getWeather to show the weather forecast at a user or client's exact campground. You can use the getWeather API to provide your users with the forecast at a specific area that is important for them.

What is getWeather?

getWeather is the only method that the GetMyWeather API provides. You can use getWeather when you need to embed "micro-forecasts" into your website or page. A micro-forecast is a weather forecast that is specific to a small area, and the getWeather microforecast includes temperature, wind speed, and the chance of rain, as well as the level of confidence in the forecast.

Register the GetMyWeather API

You can register and receive a key at http://www.getmyweather.com/getWeather. You can get a no-cost key for development and testing, but generally the cost will depend on how much you use the API. You can find more cost information on the website.

What code do you need to use getWeather?

You can use getWeather with JavaScript and HTML, and the following sections have example snippets of code you can use to initialize and invoke getWeather and to insert it's return package.

Initialize getWeather

You must initialize the GetMyWeather API in JavaScript to establish a connection to the GetMyWeather server. You can find an example of how you might do this below.

```
 <script type="text/JavaScript">
var weatherForecast;
function init() {
weatherForecast =
new getWeather(document.getElementById('forecast')}
</script>
```

Invoke getWeather

Invoke, or call, getWeather to get a forecast with the desired parameters. You may call getWeather multiple times per session and can input different parameter values each time to get a different forecast.

```
 <script async defer
src="https://www.getmyweather.com/getWeather?
key=<YOUR_KEY>&
callback=init&
location=<LATITUDE>:<LONGITUDE>&
specificity=<SPECIFICITY>&
time=<TIME>">
</script>
```

Insert the return package

To input getWeather's return package into your website, you can include an empty element in which the browser will insert the return package. To do this, you can use the following line of code.

```
 <div id="forecast"></div>
```

getWeather returns an HTML structure that contains the classes `temperature`, `windspeed`, `chanceRain`, and `trust`. You can see an example return package below.

```
 <div class="forecast">
     <div class="temperature">78</div>
     <div class="windspeed">15</div>
     <div class="chanceRain">30</div>
     <div class="trust">80</div>
</div>
```

`trust` is a measure of confidence in the forecast. You can read more about each return value under the "Output parameters" section below.

What are the inputs and outputs of getWeather?

You must pass in 4 input parameters to the GetMyWeather API, and you will receive 4 output parameters that specify the forecast.

Input parameters

There are 4 input parameters. There are no default input values. - `key` - `location` - `specificity` - `time`

key

The `key` is your API key which you receive after you register to use the API. You must pass in your `key` as a parameter each time you call getWeather.

location

The `location` value is specified as a latitude and longitude, in ISO 6709 format.

In this format, you write "(+/-) latitudeValue(+/-)longitudeValue/" where the latitudeValue is the degrees, minutes, and seconds away from the equator and longitudeValue is the degrees, minutes, and seconds away from the prime meridian. The sign in front of the latitudeValue denotes north as positive or south as negative. Similarly for the longitudeValue, east is positive and west is negative.

specificity

The `specificity` is an integer value between 10 and 100,000 (meters) inclusive.

`specificity` expresses the radius in meters of the desired area around the location for which you'd like the forecast.

time

The `time` is a positive float, expressing the number of hours in the future for which you would like the forecast.

Output parameters

There are 4 output parameters or values. - `temperature` - `windspeed` - `chanceRain` - `trust`

temperature

The `temperature` is a numeric value in Fahrenheit.

windspeed

The `windspeed` is a numeric value in miles per hour (mph).

chanceRain

The `chanceRain` is a numeric, percentage value ranging from 0 and 100 and representing the likelihood of it raining.

trust

The `trust` is a numeric, percentage value between 1 and 100 (where 1 is very unreliable and 100 is completely reliable).

This " `trust` " value is a level of confidence in the forecast. It may decrease for large areas (which you control through the input variable `specificity`); the weather at a specific moment in time on one side of a city could be very different from the weather on the opposite side of the city. Additionally, the `trust` value may decrease for forecasts further in the future (which you control through the input variable `time`); it's much harder to forecast the weather accurately two weeks in advance than just a few days. Generally, you will receive a low `trust` value if you get a forecast more than 240 hours in advance.

What errors might you encounter?

Some standard error messages that you might run into are " `invalid key` ", " `invalid parameterformat` ", " `parameter out of range` ", and " `server unavailable` ". - `Invalid key` - you may have mistyped the key, have forgotten to input one, or have an out-of-date key. - `Invalid parameterformat` - you may have inputted the value in the wrong format. Reference the "Input parameters" or "Invoke getWeather" sections above for the proper format. - `Parameter out of` `range` - you may have inputted an illegal value to a parameter. Reference the "Input parameters" section above for the legal values for each parameter. - `Server unavailable` - the internet, browser, or GetMyWeather server is likely down.

Browser issues

getWeather should run on all modern browsers. If you have any issues with registering for or calling getWeather, you might not have JavaScript activated; you must have **JavaScript activated** to use GetMyWeather.

Forecast not updating

The return time should be negligible. If you are waiting for a response from GetMyWeather, there might be a connection or browser issue.

However, you should also keep in mind that the return package is **static**. You must call getWeather every time you want to update the forecast.