

Explanatory file

about the **open data API** usage in my project

In this project, I made a single-page responsive weather website.

I have used the **OpenWeatherMap API** for the weather information.

To make it work, we need to call the weather API and load the dynamic data from there.

I have followed several steps during the implementation of this project:

First, I have created my html file, after I've styled it with css properties and finally to make it dynamic I've created a javascript file where I have used the API from the OpenWeatherMap.

How I have inserted the API into my project?

In the javascript file I have loaded the html elements:

```
const currentweatheritemsEl = document.getElementById('current-weather-items');
const timezone = document.getElementById('time-zone');
const countryEl = document.getElementById('country');
const weathercastEl = document.getElementById('weather-forecast');
const currentTempEl = document.getElementById('current-temp');
```

After that in the website of the openweathermap...

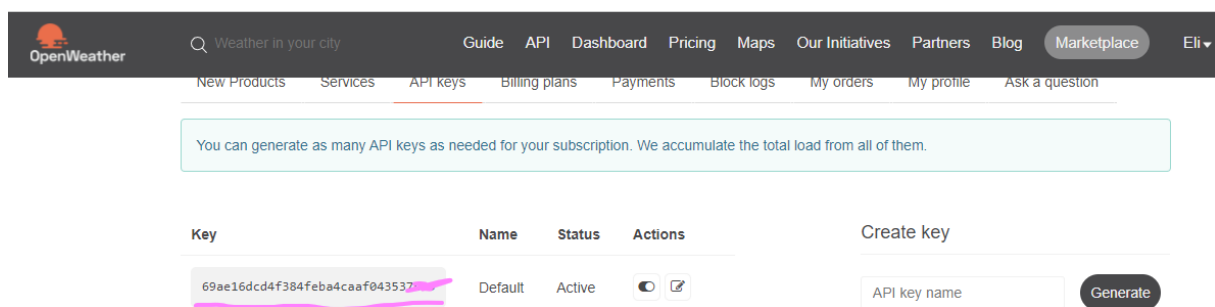
<https://openweathermap.org> > [api](#) ▾ [Превод на страницата](#)

Weather API - OpenWeatherMap



Simple and fast and free **weather API** from OpenWeatherMap you have access to c
weather data, hourly, 5- and 16-day forecasts. Historical weather data ...

[Current weather data](#) · [Historical weather API](#) · [One Call API](#) · [5 day weather forecast](#)

After signing up with an email we have got our **API key**, we just copy and paste it into our javascript file.



The screenshot shows the OpenWeatherMap dashboard for API keys. At the top, there's a navigation bar with the OpenWeather logo and a search bar. Below it, a secondary navigation bar lists various options: New Products, Services, API keys (highlighted with a red underline), Billing plans, Payments, Block logs, My orders, My profile, Ask a question, and a Marketplace button. A light blue box contains the text: "You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them." Below this is a table with columns: Key, Name, Status, and Actions. The first row shows a key value (partially obscured by a pink highlight), the name "Default", and the status "Active". To the right of the table is a "Create key" section with a text input field labeled "API key name" and a "Generate" button.

| Key | Name | Status | Actions |
|----------------------------------|---------|--------|---|
| 69ae16dcd4f384feba4caaf043537... | Default | Active |   |

I set the API value in a variable, that I will need for all the calls I will do in my project.

```
const API_KEY = "6366e0ddd90f543bc07b185cd8fc14e9";
```

In the API portion, we can see all the documentation corresponding to the Weather API.

The one I have used is One Call API. This API has the daily forecast, the current forecast, ect..a lot of data just in one single API.

One Call API

[API doc](#)[Subscribe](#)

- 1 days
- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **National weather alerts**
- JSON format
- Included in both free and paid subscriptions

The URL to call the API we copy and paste in our javascript file.

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/onecall?lat={lat}&lon={lon}&exclude={part}&appid={API key}
```

Parameters

| | | |
|----------|----------|---|
| lat, lon | required | Geographical coordinates (latitude, longitude) |
| appid | required | Your unique API key (you can always find it on your account page under the "API key" tab) |
| exclude | optional | By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: <ul style="list-style-type: none">current<u>minutely</u><u>hourly</u>dailyalerts |

We need the **latitude**{lat}, the **longitude**{lon} and in the {part} I will exclude the minutely and hourly data that I don't need.

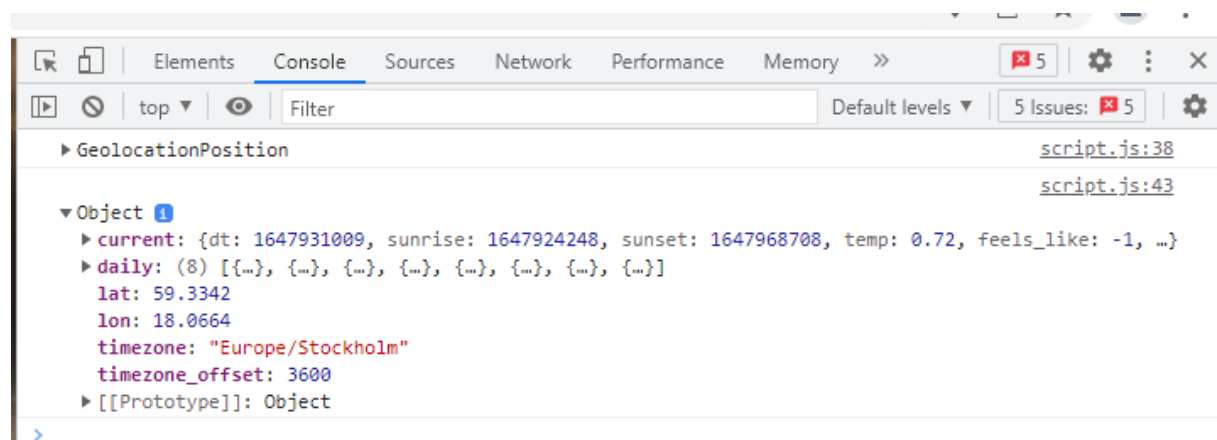
To call the API I used a function and to get the longitude and latitude I used the navigator to get the geolocation and based on the geolocation I have done the API call.

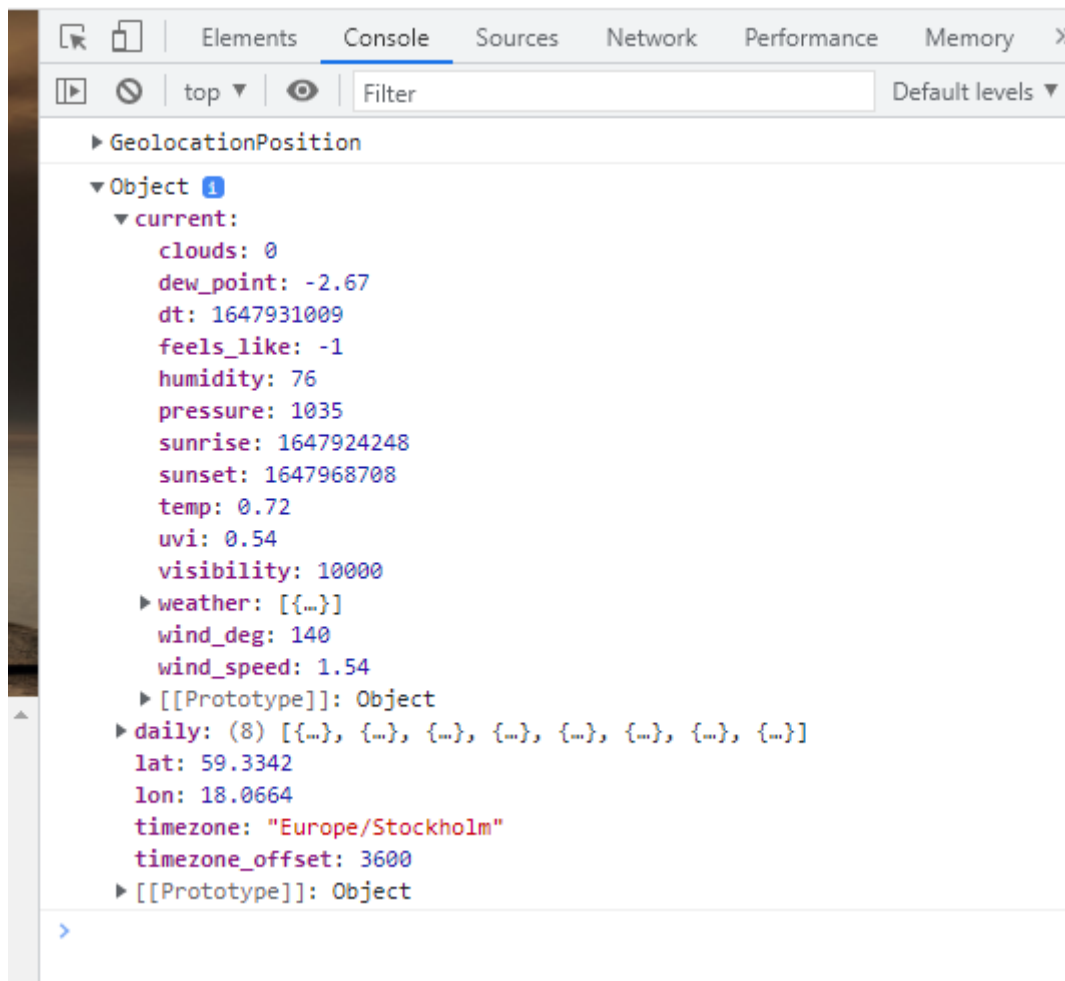
```
getWeatherData();
function getWeatherData () {
  navigator.geolocation.getCurrentPosition((success) => {
    console.log(success);
    let {latitude, longitude} = success.coords;

    fetch(`https://api.openweathermap.org/data/2.5/onecall?lat=${latitude}&lon=${longitude}&exclude=hourly,minutely&units=metric&appid=${API_KEY}`).then(res => res.json()).then(data => {

      console.log(data);
      showWeatherData(data);
    })
  })
}
```

After this API call, we have got a response that we can inspect:





We have got the timezone, the daily data, the current data, etc..

I have just used this data to load all the data to my project. I have passed all the data to my javascript file using a function:

```
showWeatherData(data);  
  })  
})  
}
```

To show the weather data for this section,



I used this function:

```
function showWeatherData(data) {
  let {humidity, pressure, sunrise, sunset, wind_speed} =
data.current;
  timezone.innerHTML = data.timezone;
  countryEl.innerHTML = data.lat + 'N' + ' ' + data.lon + 'E';

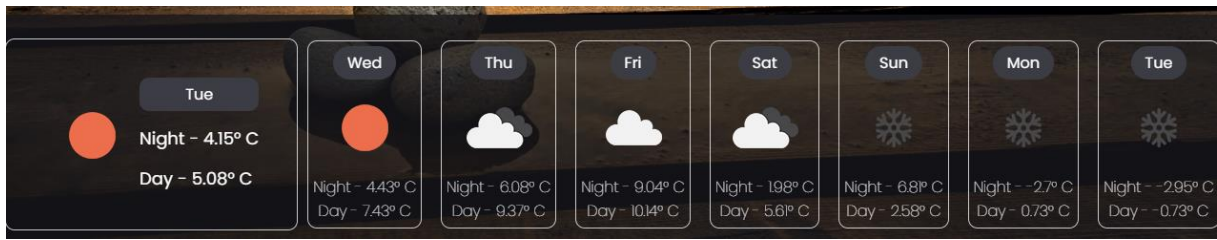
  timezone.innerHTML = data.timezone;
  countryEl.innerHTML = data.lat + 'N' + data.lon + 'E'

  currentweatheritemsEl.innerHTML =

`<div class="weather-item">
  <div>Humidity</div>
  <div>${humidity}%</div>
</div>
<div class="weather-item">
  <div>Pressure</div>
  <div>${pressure}</div>
</div>
<div class="weather-item">
  <div>Wind speed</div>
  <div>${wind_speed}</div>
</div>
<div class="weather-item">
  <div>Sunrise</div>
  <div>${window.moment(sunrise* 1000).format('HH:mm a')}</div>
</div>
<div class="weather-item">
  <div>Sunset</div>
  <div>${window.moment(sunset* 1000).format('HH:mm a')}</div>
</div>`;
}
```

!!! To format the sunrise and sunset values I have used the moment package

For this part of my website,



I have used this function:

```
let otherDayForecast = ''
data.daily.forEach((day, i) => {
  if(i == 0){
    currentTempEl.innerHTML = `

<div class="other">
<div class="day">${window.moment(day.dt* 1000).format('ddd')}</div>
<div class="temp">Night - ${day.temp.night}&#176 C</div>
<div class="temp">Day - ${day.temp.day}&#176 C</div>
</div>
`

  }else{

    otherDayForecast += `
<div class="weather-forecast-item">
<div class="day">${window.moment(day.dt* 1000).format('ddd')}</div>

<div class="temp">Night - ${day.temp.night}&#176 C</div>
<div class="temp">Day - ${day.temp.day}&#176 C</div>
</div> `

  }
})

weathercastEl.innerHTML = otherDayForecast;
}
```

Developers use an open API to gain access to certain features of a software program that would be difficult to access without taking the time to develop a large amount of code.