

Weather API Server

Simple Flash weather API

[Github link](#)

- [HTTP Get request](#)
 - [Usage](#)
 - [List of measurements](#)
- [Websockets](#)
 - [Usage](#)

HTTP Get request

Get data using http GET request

HTTP Get request

Usage

Access weather data in certain time range by sending GET request to https address of:
`https://dataapi.vaclavlepic.com/weather/{measurement_name}?start_time={YYYY-MM-DDTHH:MM:SS}Z&end_time={YYYY-MM-DDTHH:MM:SS}Z`

Sample return for

`https://dataapi.vaclavlepic.com/weather/outdoor_temperature?start_time=2024-10-25T09:00:00Z&end_time=2024-10-25T11:09:59Z:`

```
{
  "measurement_values": [5.9, 5.8, 5.7, 5.8, 5.9, 6, 5.9],
  "time_values": [
    "2024-10-25T11:00:58.544697+02:00",
    "2024-10-25T11:03:33.577063+02:00",
    "2024-10-25T11:04:35.708740+02:00",
    "2024-10-25T11:05:06.715888+02:00",
    "2024-10-25T11:06:08.731456+02:00",
    "2024-10-25T11:08:43.742152+02:00",
    "2024-10-25T11:10:47.752677+02:00"
  ]
}
```

Access latest weather data by sending GET request to https address of:
`https://dataapi.vaclavlepic.com/weather/{measurement_name}/latest`

Sample return for latest outdoor_temperature:

```
{
  "measurement_values": [5.8],
  "time_values": [
    "2024-10-25T10:58:54.535371+02:00"
  ]
}
```

List of measurements

1. **outdoor_temperature**
 1. Time range
 2. latest
2. **humidity**
 1. Time range
 2. latest
3. **dewpoint**
 1. Time range
 2. latest
4. **absolute_pressure**
 1. Time range
 2. latest
5. **relative_pressure**
 1. Time range
 2. latest
6. **wind_gust**
 1. Time range
 2. latest
7. **wind_speed**
 1. Time range
 2. latest
8. **hourly_rain**
 1. Time range
 2. latest
9. **daily_rain**
 1. Time range
 2. latest
10. **yearly_rain**
 1. Time range
 2. latest
11. **rain_rate**
 1. Time range
 2. latest

Websockets

Access data using websockets

Usage

Sample websockets usage in javascript:

```
const socket = io.connect('https://dataapi.vaclavlepic.com');

// Subscribe to temperature updates
socket.emit('subscribe', { measurement: 'outdoor_temperature' });

// Handle temperature updates
socket.on('outdoor_temperature_update', function(data) {
  console.log(`Teplota: ${data.measurement_values[0]} °C at ${data.time_values[0]}`);
});

// Unsubscribe from temperature updates
function unsubscribe() {
  socket.emit('unsubscribe', { measurement: 'outdoor_temperature' });
}

// Example: Call unsubscribe after 10 seconds
setTimeout(unsubscribe, 10000);
```