Front-end developer Assignment

Time to complete: 7 days.

Summary:

In this assignment, we are going to have a little bit of fun with our weather API. You should create one clickable web map. The user should be able to click on the map in order to display information for any geographical point in Greece.

Our goal here is to display, in a meaningful way, information regarding weather forecasts on a 24 hour span. Special care should be taken in order for the resulting work to be elegant and responsive.

As an extra step, you should create a custom paginated list, which will be sortable by the variable of temperature on a 24 hour span.

The web app should do the following:

- Have a fully responsive working map
- The user should be able to view the weather forecast **for the next 24hours** upon clicking on any location in Greece. Feel free to present the data as you see fit.
  - Specifically, they should be able to view the values of the following two variables:
    - Temperature (**Variable name**: *temperature2m*)
    - Humidity (**Variable name**: *rh2m*)
  - Finally, they should be able to view a list of temperatures (24hrs, variable name: *temperature2m*) and they will be able to sort that list in an ascending and descending order. Also, a pagination should be applied to display 4 hours of data per page (i.e. it will have 6 pages). No external library should be used in this step. **This step is optional but we would love to see it in the final product.**

Guidelines on how to develop this

- Fetch the response from the Draxis API (See below API Documentation) for the two requested variables.
- For every call to the weather API, you have to use the following key as a parameter *apikey=4181a631-652a-40a2-a57f-e8338074cc5a*

Example :

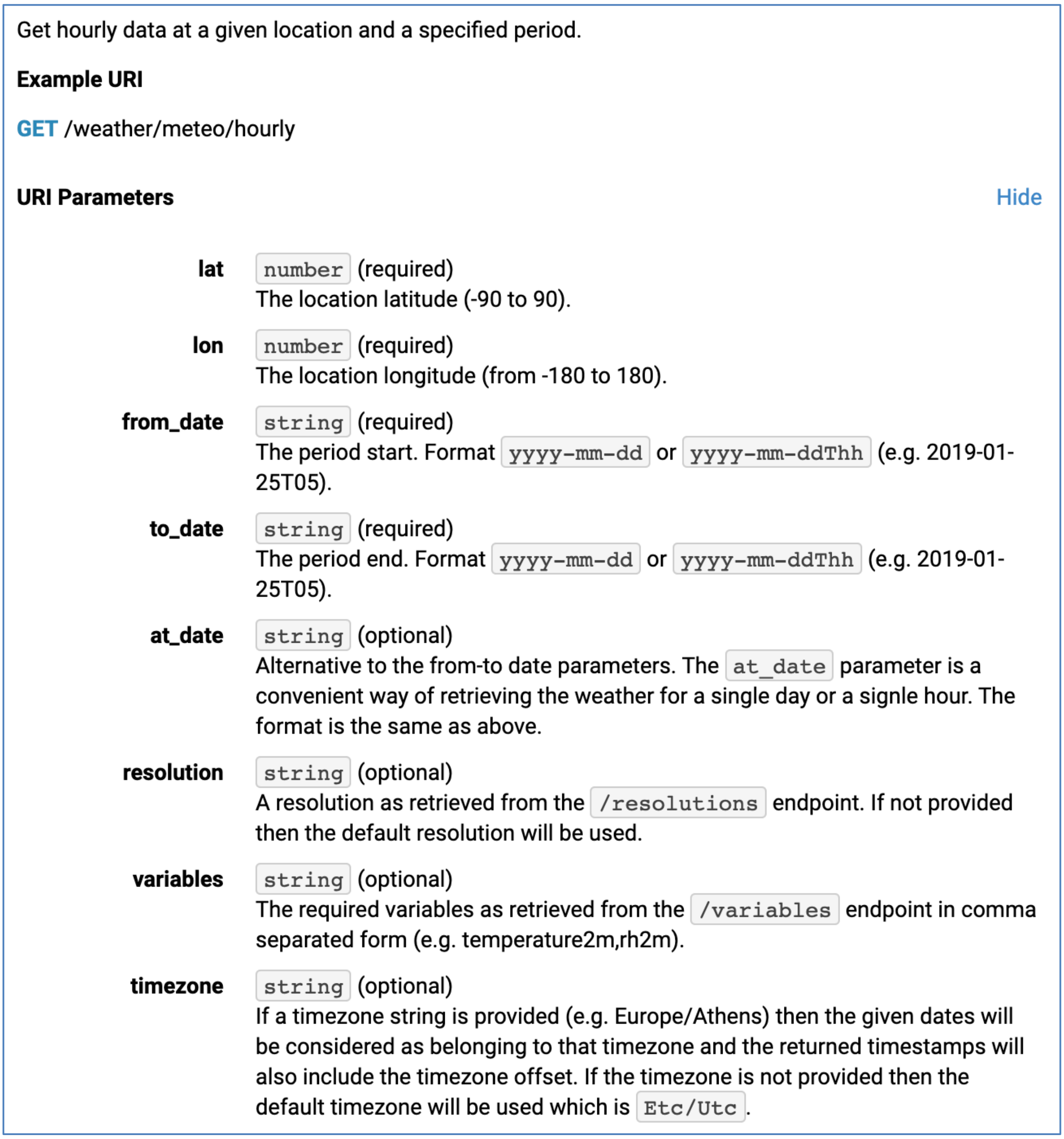
https://api.draxis.gr/weather/meteo/hourly?apikey=4181a631-652a-40a2-a57f-e8338074cc5a&lat=37.983810&lon=23.727539&at_date=2020-05-24&variables=temperature2m,rh2m

- Keep in mind that the endpoint will not work partially or entirely, every day from 6am to 7:15am.
- You will have to utilize your API call through a proxy server, in order to bypass the CORS limitations of it. You can use this https://github.com/Rob--W/cors-anywhere/, but feel free to use any other solution that will solve the issue.

Minimum Requirements

- Write clean, well-structured and maintainable code
- Component design pattern
- Usage of any modern JS Framework (vuejs,react, angular)
- Usage of HTML5, CSS3
- Usage of any UI library (optional)
- Responsive design
- Elegant appearance
- Your deployed application should be working in all major browsers (Google Chrome, Firefox, Edge, Safari)
- Creation of a GitHub repository with a unique name and a README describing the project and the installation process

API documentation for the hourly forecast.



Get hourly data at a given location and a specified period.

Example URI

GET /weather/meteo/hourly

URI Parameters

lat `number` (required)
The location latitude (-90 to 90).

lon `number` (required)
The location longitude (from -180 to 180).

from_date `string` (required)
The period start. Format `yyyy-mm-dd` or `yyyy-mm-ddThh` (e.g. 2019-01-25T05).

to_date `string` (required)
The period end. Format `yyyy-mm-dd` or `yyyy-mm-ddThh` (e.g. 2019-01-25T05).

at_date `string` (optional)
Alternative to the from-to date parameters. The `at_date` parameter is a convenient way of retrieving the weather for a single day or a signle hour. The format is the same as above.

resolution `string` (optional)
A resolution as retrieved from the `/resolutions` endpoint. If not provided then the default resolution will be used.

variables `string` (optional)
The required variables as retrieved from the `/variables` endpoint in comma separated form (e.g. temperature2m,rh2m).

timezone `string` (optional)
If a timezone string is provided (e.g. Europe/Athens) then the given dates will be considered as belonging to that timezone and the returned timestamps will also include the timezone offset. If the timezone is not provided then the default timezone will be used which is `Etc/Utc`.