PHP Coding Test: Weather API

Introduction

The goal is to build a small application using PHP, HTML, JS and CSS. The application should allow a user to enter a location (e.g. city or country) to retrieve the current weather forecast.

You will be given a boilerplate that contains:

- HTML file with an input element
- CSS file
- JS file that listens for input changes

The boilerplate does not contain any business logic. In this exercise you will have to implement the business logic and do styling based on rudimentary wireframes. The business logic should be written entirely in PHP. You can use your framework of choice, but make sure to read the objectives first before you decide on which one.

The interviewer will introduce you to the application. A short walk-through will be conducted.

Objectives

The objectives are separated in business logic & styling. The goal is not to finish everything. Prioritize your work. Do the things you are most familiar with first.

Business Logic

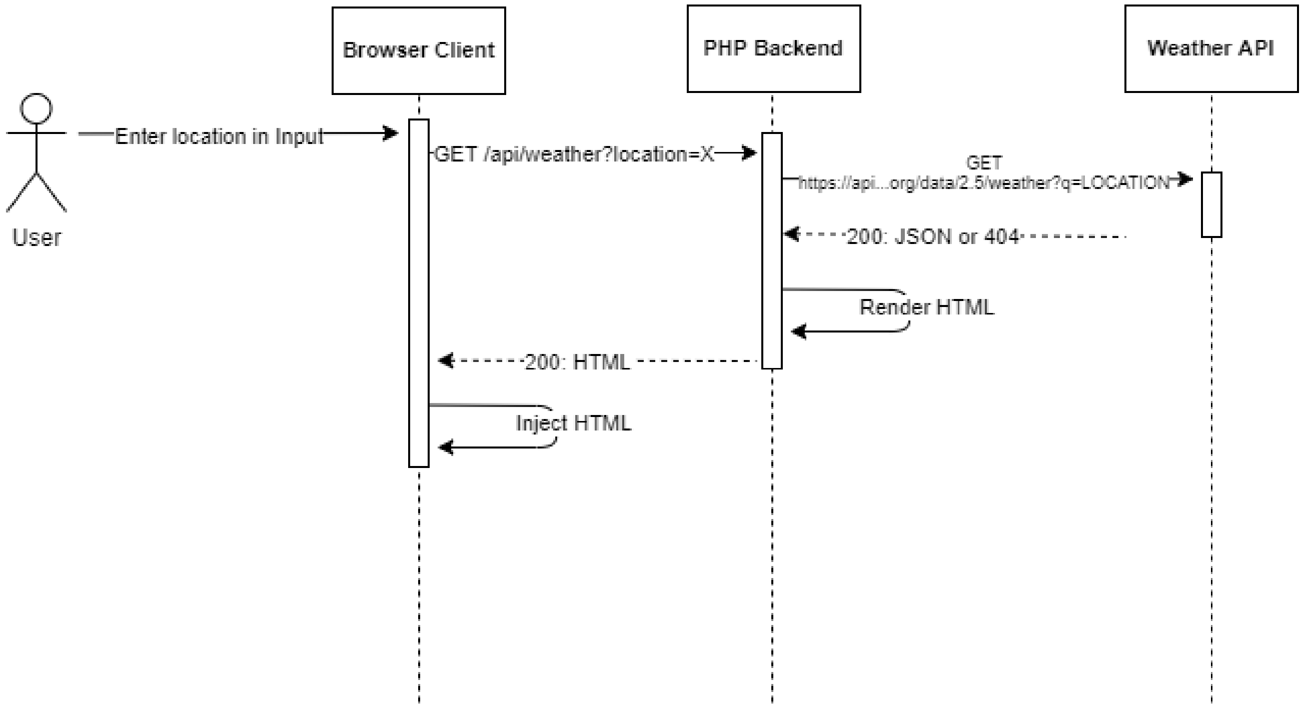
- Create an API endpoint using PHP which fetches the current forecast from a weather API.
  - An example of the call to the weather API is given below (see chapter Weather API).
  - It is possible that the request to the weather API will return a 404. In that case, respond with “Nothing found 😞” (see wireframes).
- Your API endpoint **should not** return any JSON. Use a templating engine of your choice (e.g. Twig) to construct the HTML markup and send that as a response.
- The prepared JS file must connect to your PHP endpoint and inject the received response into the browser. You can change whatever you think is necessary in the JS file.
- Hint: Depending on your setup you might encounter CORS issues when contacting your endpoint. Be sure your endpoint handles CORS requests.

Styling

- Use the CSS file to style the markup returned by the API endpoint, so it looks like the wireframes you see below.
- The styling should include a mobile-friendly and desktop solution (see chapter Wireframes).
  - On desktop the entire application should be vertically centered.

- It is important that the application does not jump around if the user is presented with a result and no result.
- If you are aware of the BEM standard to name classes, apply the standard best to your knowledge.

The following UML sequence diagram outlines the flow of the application



Weather API

URL to API: `https://api.openweathermap.org/data/2.5/weather?q=YOUR_ENTERED_LOCATION&units=metric&appid=042559fc8f13bd0e86e557aa02965a24`

Make sure you use the provided URL. It already includes the API key.

API Spec: https://openweathermap.org/current

You should use the endpoint *By city name*.

The example below shows the response when using *London* as the `q` parameter. The attributes of interest to you are highlighted with comments `//`. You can assume that there is only one entry in the `weather` array:

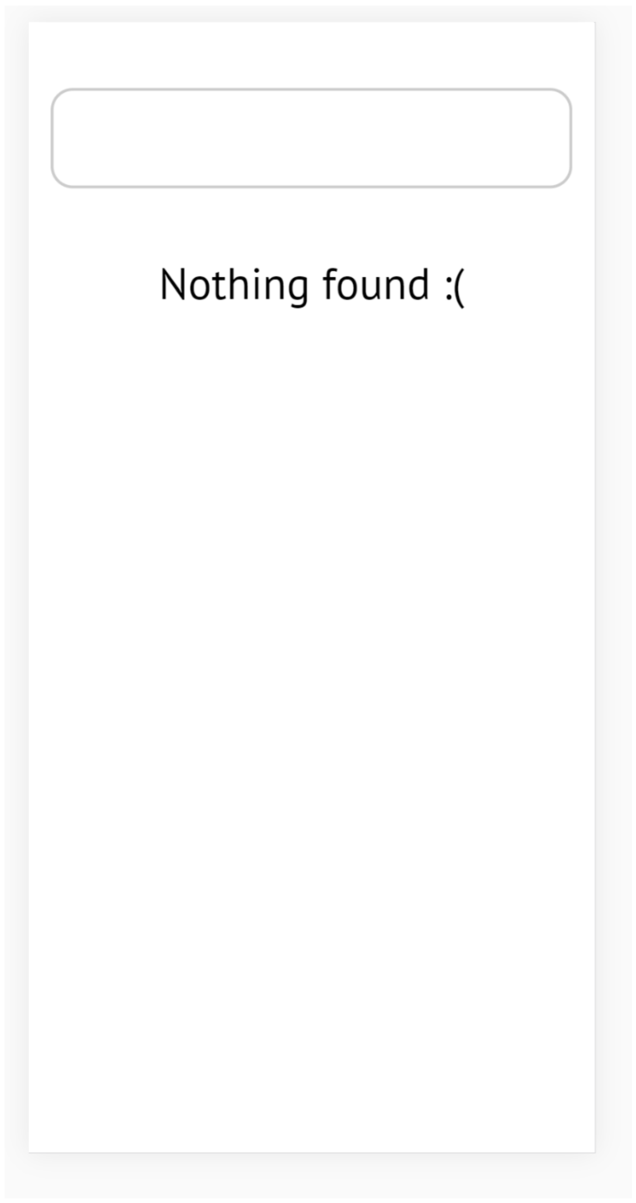
```
{
        "coord": {
                "lon": -0.1257,
                "lat": 51.5085
        },
        "weather": [{
                "id": 804,
                "main": "Clouds",                          // Useful
                "description": "overcast clouds",
                "icon": "04d"                              // Useful
        }],
        "base": "stations",
        "main": {
                "temp": 4.99,                              // Useful
                "feels_like": 1.15,
                "temp_min": 3.33,                          // Useful
                "temp_max": 6,                             // Useful
                "pressure": 1033,
                "humidity": 81
        },
        "visibility": 10000,
        "wind": {
                "speed": 3.09,
                "deg": 300
        },
        "clouds": {
                "all": 100
        },
        "dt": 1615964870,
        "sys": {
                "type": 1,
                "id": 1414,
                "country": "GB",
                "sunrise": 1615961393,
                "sunset": 1616004470
        },
        "timezone": 0,
        "id": 2643743,
        "name": "London",                                  // Useful
        "cod": 200
}
```

Wireframes

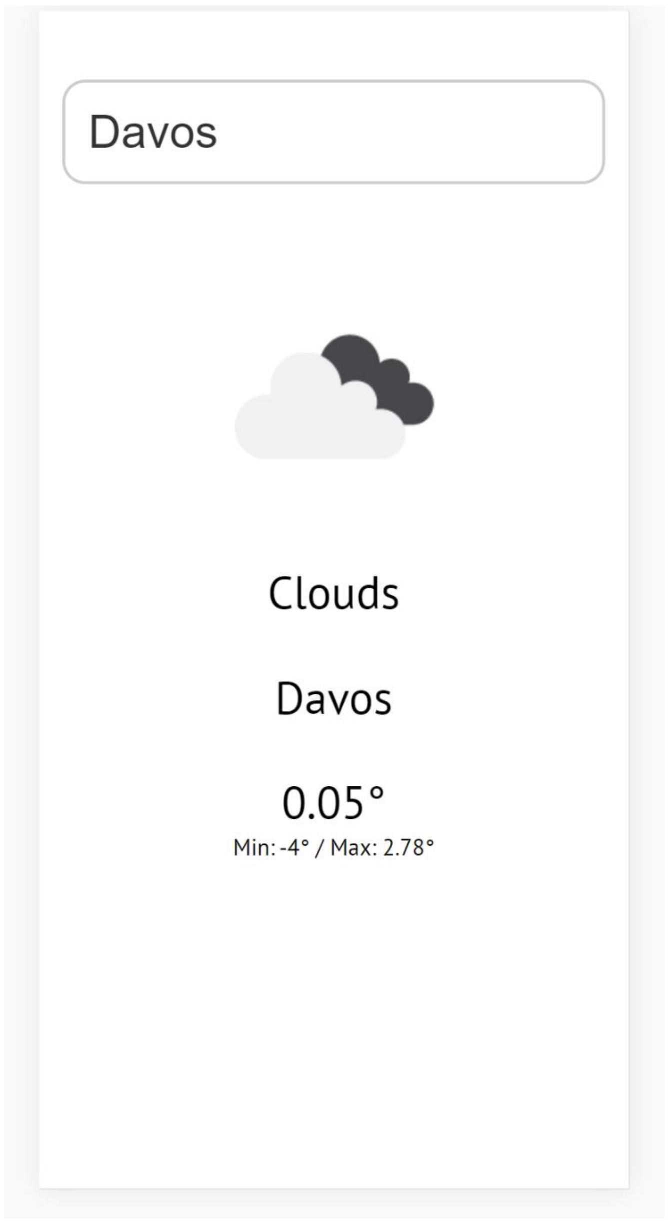
For the weather icon you can use the following URL:
`http://openweathermap.org/img/wn/ICON_FROM_RESPONSE@4x.png`

Make sure to use the icon attribute from the response (see example above) for the URL of the icon.

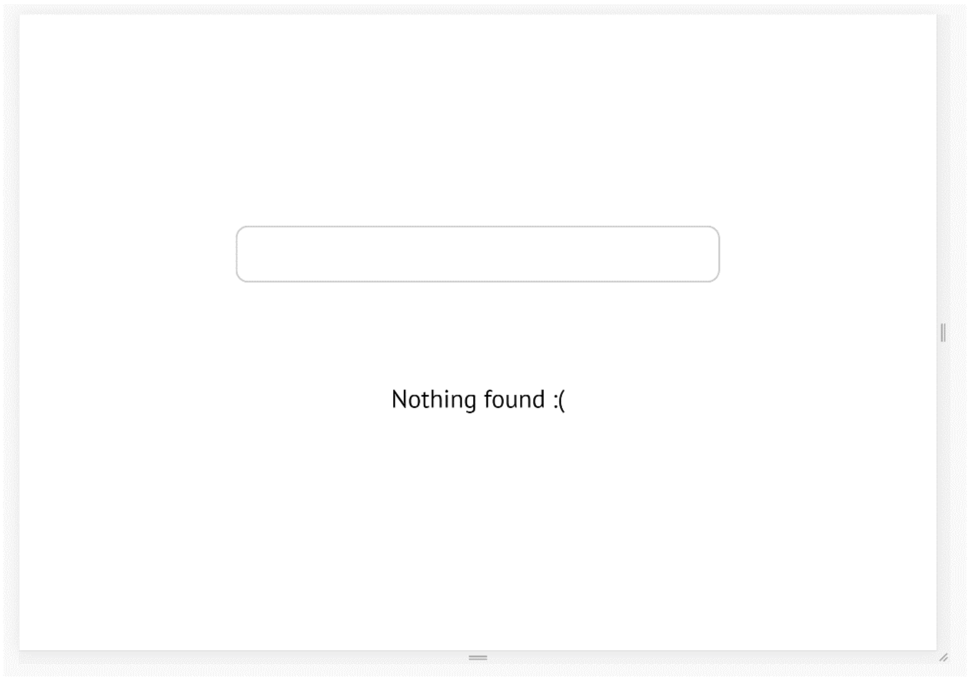
Mobile: Nothing found



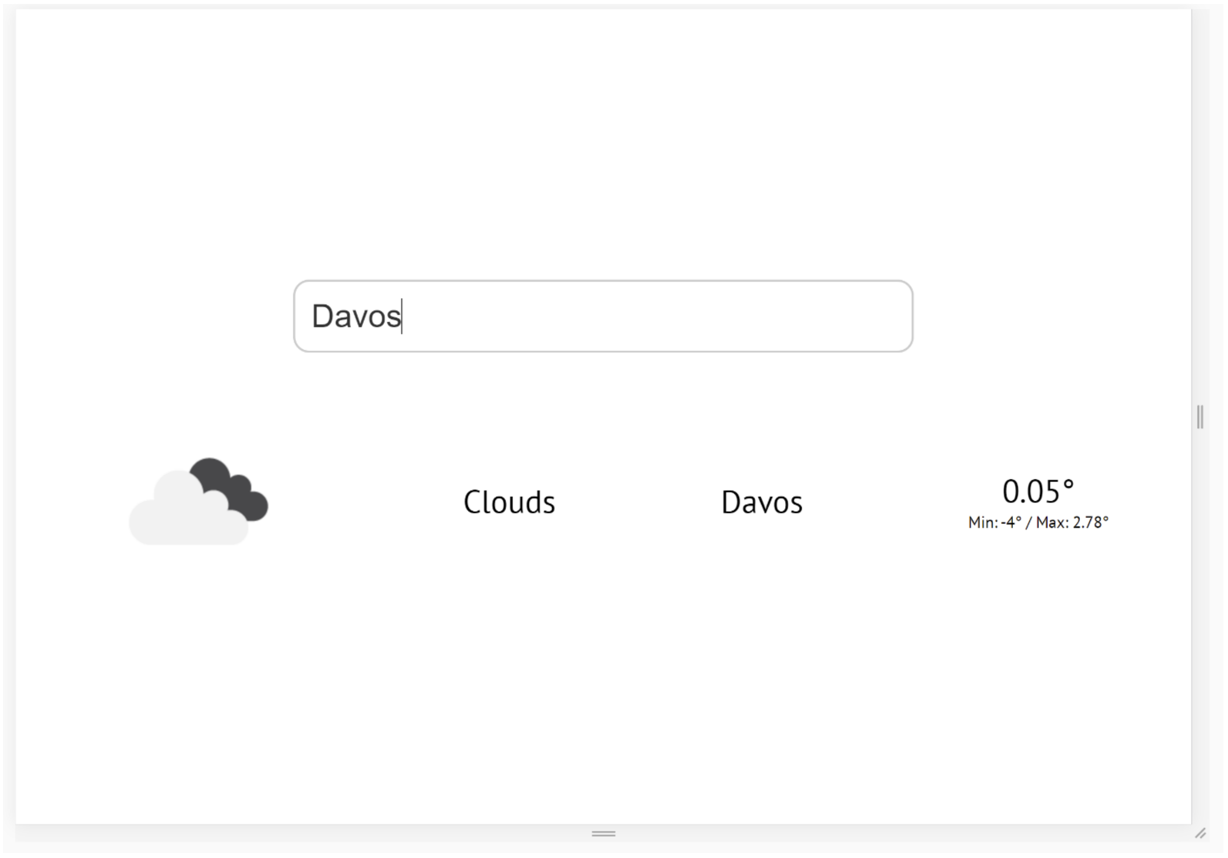
Mobile: Weather found for a location (Davos)



Desktop: Nothing found



Desktop: Weather found for a location (Davos)


Davos
Clouds
Davos
0.05°
Min: -4° / Max: 2.78°