

Overview:

First, I viewed a random holiday page on the national trust to see the overall layout of the page. I noticed that the holiday page uses google maps to display the location of the holiday house/cottage, and predicted I would find the longitude and latitude from that frame.

This was important as google maps is dynamically loaded content, possibly lazy loaded, hence it won't load onto the webpage until after a delay. I then crawled the webpage to extract the entire webpage data. I made sure to delay the html extraction by a few seconds to ensure that all the JavaScript on the page has fully loaded, especially the google maps segment.

After this I was able to locate where the longitude and latitude values were stored within the webpage using a lookup/search function. From there I was able to locate the element that stores the longitude and latitude values as '___NEXT_DATA___'. I then used this element, and a recursive search to extract the values for longitude and longitude from the page.

Then once I get these values, I alter the API URL with these values to extract data from the API about the location. I then use a HTTP request to the URL to extract the data. Because I have had access to the API website. I then loop over every forecast and extract its details. As the API is a fake mock-up of openweathermap endpoint, I am able to use the icon codes in the fake API to extract the real images from openweathermap.

I then create a widget to store this data by creating new elements, which have minimal reactive CSS styling displaying all the data about forecasts, and I append this onto the page above the calendar prices. This was because, I thought that users would be interested at looking at the weather for a day on the calendar.

I then tested this on 10 different random holiday pages to ensure that it works, and it proved successful for that random sample.

Included error handling for debugging purposes.

Improvements:

To improve instead of displaying all the weather data about all dates, consider having the weather data display dates selected on the check availability widget, this provides a cleaner and simpler look, where users can see directly their potential booking dates weather forecasts. Prevents users having to sift through all the weather data to find specific dates.

Furthermore, incorporate A/B testing by dividing users in half at random, of those who can see the weather widget, and those who can't. Use dev tools to track a variety of

different metrics including time spent on webpage, webpage interactions, whether a holiday was booked rate, etc.