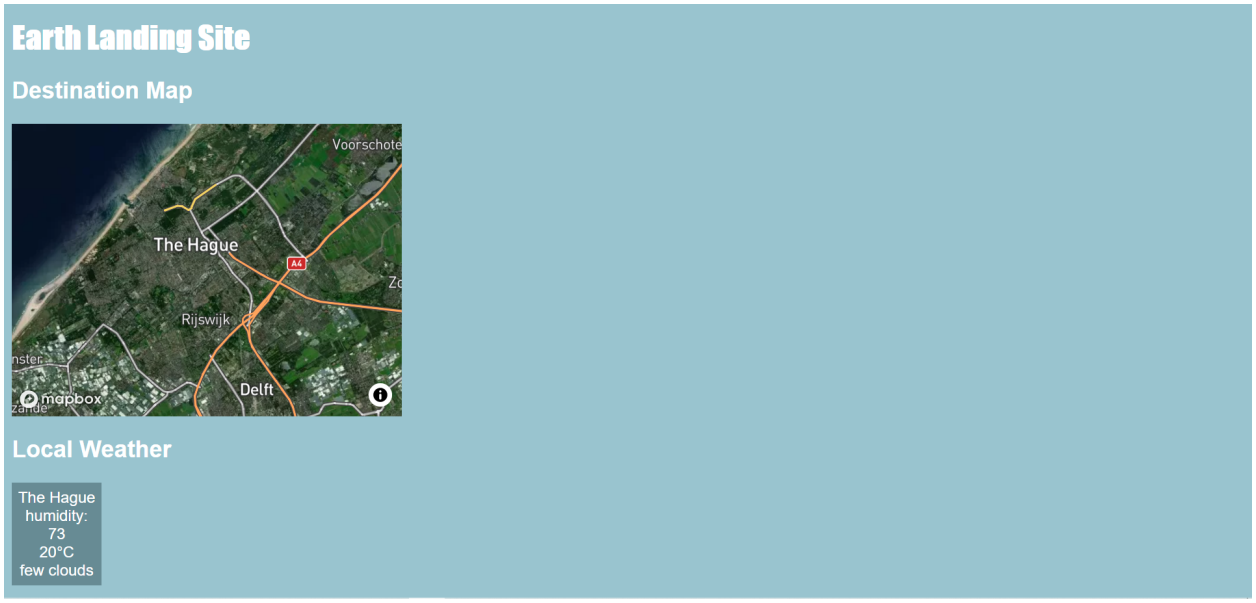
Challenge 3 Description

API (Application Programming Interface) is an interface between applications. The server online that provides the API lets you program two interfaces between said applications. It is different from GUI (Graphical User Interface) that lets an interaction between a user and an interface. With Javascript, we can create a website that asks for information from API website providers like Openweather and Mapbox and display it on the website.

API works with JSON (Javascript Object Notation), which is a language to store and transport data. API data has to be converted to JSON in order to work.

In the third challenge, I used Openweather and Mapbox API. This is how the website looks like:



The codes are below:

```
mapboxgl.accessToken = "pk.eyJ1IjoiYWtpbGFhbWFsYSIsImEiOiJja3FwMjVtaHUwcTgwMnZwNXJvNHB0N2N0In0.MS09WBDgtZz8NM02hZam-A"

var map = new mapboxgl.Map({
container: 'map',
style: 'mapbox://styles/mapbox/satellite-streets-v11',

center: [4.322840, 52.067101],
zoom: 10,
});
```

In the .js file, I put the mapbox API token and put the style to satellite mode to emulate a space station interface. I wanted the map to show The Hague and its surrounding cities, so I set the zoom to 10.

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Earth Landing Site</title>

    <script src='https://api.mapbox.com/mapbox-gl-js/v2.3.1/mapbox-gl.js'></script>
    <link href='https://api.mapbox.com/mapbox-gl-js/v2.3.1/mapbox-gl.css' rel='stylesheet' />
    <link rel="stylesheet" href="styles.css">

</head>
<body>
    <header>
        <h1 class="title">Earth Landing Site</h1>
    </header>
  <script src="app.js" defer></script>

<h2 class="map"> Destination Map</h2>
<div id='map' style='width: 400px; height: 300px;'></div>
<h2 class="map"> Local Weather</h2>
<article id='weather' style='width: 900px height: 700px'></div>

</body>

```

This is how the HTML looks like. In order for the map and the weather box to show up, I had to type in the width and height of the div section.

```javascript

function getAPIdata() {

    var url = 'https://api.openweathermap.org/data/2.5/weather';
    var apiKey ='fbdf505db145dd956f16c39df1d95ff8';
    var city = 'den haag';
    var request = url + '?' + 'appid=' + apiKey + '&' + 'q=' + city;

    fetch(request)

    .then(function(response) {
        if(!response.ok) throw Error(response.statusText);
        return response.json();
    })

    .then(function(response) {
        onAPISucces(response);
    })

}

function onAPISucces(response) {

    console.log(response);

    var city = response.name;

    var humid = response.main.humidity;

    var celsius = Math.floor(response.main.temp - 273.15);

    var weather = response.weather[0].description;


    var localWeather = document.getElementById('weather');
    localWeather.innerHTML = city + '<br>' + 'humidity: <br>'+ humid + '<br>' + celsius + '&#176;C <br>' + weather +'<br>';

}

getAPIdata();

```

For the local weather section, I decided to put the city name, humidity, temperature, and current weather condition. I set the city's weather to Den Haag in order to correlate it with the satellite map above.