**INTERNET PROGRAMMING**

**Task 1 Code Explanation**

/\***The function is used to dynamically load a json file containing weather data**

**across cities in the UK.**

**window.onload() function is used here to load contents of the data to the webpage**

\*/

window.onload = function() {

const cityWeatherFile = './weather.json'; **//** **declare a constant variable to hold the json file**

/\***The fetch method is used here to access the resources in the json file.**

**The fetch method handles request and response to the web browser. First it makes a request to the server, then it returns a promise that resolves into a response object. The then function handles the data while the catch function handles error.**

\*/

**//the fetch method takes in a compulsory argument which is the weather.json file stored in a variable called cityWeatherFile in this case**

fetch(cityWeatherFile).then((response) => {

**//the response is accessed here and it is important to ensure that the response status is 200 using an if statement**

if (response.status !== 200) {

console.log("Error occured: " + response.status); // display the error on the console if response status is not 200

} else {

return response.json(); // return the json object of the output if response status is satisfied

}

}).then((data) => {

/\***The text in the response is being examined here. The getWeatherInfo() method is called here and takes in data as an argument. The data displayed consists of the defined attributes in the getWeatherInfo() method and it is matched with attributes in the weather.json file.**

\*/

getWeatherInfo(data); **// function to display weather details of cities**

/\* **The setTimeout function is used here to set interval at which the data in the cityWeatherFile will be uploaded. The data will update every 5000 milliseconds which is equivalent to 5seconds.**

\*/

setTimeout(function() {

getWeatherInfo(data); **// the getWeatherInfo() method is called here again to keep displaying every 5 seconds**

window.onload(); **// window.onload() method is being called here again so that whenever the page refreshes, the content are loaded immediately**

console.log(data); **//display the data retrieved from the cityWeatherFile (weather.json file) on the console**

}, 5000);

}).catch((error) => {

var div = document.getElementById('update-status-message'); **// declare variable to access the html element where status message will be displayed using document.getElementById() method**

div.append('Error Loading Data!!!'); **// append method is used to join the error message to the variable holding the status message**

console.log('Error loading data: ' + error); **//display error message to the console**

});

/\* **This function is used to generate icons depending on the current condition of the city weather the function takes in an argument called description that is used to check different conditions of the weather. The icons have been placed in a folder called weather\_icons**

\*/

const generateIcons = (description) => {

if (description == 'Snow') {

**//If the current condition is Snow, snow icon is displayed from weather-icons folder**

return "../weather\_icons/snow.png";

} else if (description == 'Cloud') {

**//If the current condition is Cloud, cloud icon is displayed from weather-icons folder**

return "../weather\_icons/cloud.png";

} else if (description == 'Sleet') {

**//If the current condition is Sleet, sleet icon is displayed from weather-icons folder**

return "../weather\_icons/sleet.png";

} else if (description == 'Sun') {

**//If the current condition is Sun, sun icon is displayed from weather-icons folder**

return "../weather\_icons/sun.png";

} else if (description == 'Hail') {

**//If the current condition is Hail, hail icon is displayed from weather-icons folder**

return "../weather\_icons/hail.png";

} else if (description == 'Heavy cloud') {

**//If the current condition is Heavy cloud, heavy cloud icon is displayed from weather-icons folder**

return "../weather\_icons/heavycloud.png";

} else if (description == 'Heavy rain') {

**//If the current condition is Heavy rain, heavy rain icon is displayed from weather-icons folder**

return "../weather\_icons/heavyrain.png";

} else if (description == 'Mist') {

**//If the current condition is Mist, mist icon is displayed from weather-icons folder**

return "../weather\_icons/mist.png";

} else if (description == 'Rain') {

**//If the current condition is rain, rain icon is displayed from weather-icons folder**

return "../weather\_icons/rain.png";

} else if (description == 'Sun and cloud') {

**//If the current condition is Sun and cloud, sun and cloud icon is displayed from weather-icons folder**

return "../weather\_icons/sun\_and\_cloud.png";

} else if (description == 'Thunderstorm') {

**//If the current condition is Thunderstorm, thunderstorm icon is displayed from weather-icons folder**

return "../weather\_icons/thunderstorm.png";

}

};

/\***This function handles how the weather data is displayed on the webpage.**

**The getWeatherInfo() method takes in an argument called data.**

**The data is accessed depending on the attribute of the data needed to be displayed.**

**Table format is used to display the data.**

\*/

const getWeatherInfo = (data) => {

let table = document.getElementById('table-display'); **// declare a variable that stores the html content to be displayed. This variable accesses the html element by Id in the html folder using document.getElementById()**

table.innerHTML = ''; **// set the innerHTML content of the table to an empty string to ensure no item is currently displayed on the table**

let row = document.createElement("tr"); **// create a table row element and store in a variable called row**

**//In the innerHTML of the table row created, create table headers, and define their contents**

row.innerHTML = "<th>City Id</th>" + **//table header for the city id column**

"<th>City Name</th>" + **//table header for the city name column**

"<th>Current Condition</th>" + **//table header for the current condition column**

"<th>Cloud Icon</th>" + **//table header for the cloud icon column**

"<th>Temperature</th>" + **//table header for the temperature column**

"<th>Wind Speed</th>" + **//table header for the wind speed column**

"<th>Wind Direction</th>" + **//table header for the win direction column**

"<th>Wind Chill Factor</th>"; **//table header for the wind chill factor column**

table.append(row); **//use the append method to add the defined table head content in the table row to the table**

/\***This for-loop is implemented to loop through the array of the data and adds the content to a table column also called table data.**

**i is a counter in the for loop that loops through array of element in the data and increments until the last element in the data is reached.**

\*/

for (let i = 0; i < data.cities.length; i++) {

let row = document.createElement('tr'); **//create a table row element and store it in a variable called row**

row.setAttribute('class', "active-row"); **// this is used to set a class attribute called active-row to the row element to check for the rows that are active and currently updating**

/\***The elements in the data which includes the id, name, current condition, weather icon, temperature, wind speed, wind direction, windchill factor, of each cities in the data. They are added to the innerHTML of the row element in the table. The generateIcons() method is called in the weather icon column and adds the appropriate icon to each row.**

**It takes in an argument which is the current condition and adds the icon that matches the current condition.**

\*/

row.innerHTML =

'<td>' +

data.cities[i].id + **//dynamically add the city id to the city id column**

'</td>' +

'<td>' +

data.cities[i].name + **// dynamically add the cities name to the city name column**

'</td>' +

'<td>' +

data.cities[i].currentCondition + **//dynamically load the current condition of each city to the current condition column**

'</td>' +

`<td><img src = ${generateIcons(data.cities[i].currentCondition)} alt = "" width= 35px height = 35px></td>` +**// dynamically load icons to the table using the generateIcons() method and parsing the current condition as an argument to determine which icon to generate**

'<td>' +

data.cities[i].temperature + "°C" + **//dynamically add the temperature of each city to the temperature column and use string concatenation to concatenate the unit which is degree Celsius**

'</td>' +

'<td>' +

data.cities[i].windSpeed + "mph" + **//dynamically add the wind speed of each city to the wind speed column and use string concatenation to concatenate the unit which is miles per hour**

'</td>' +

'<td>' +

data.cities[i].windDirection + "°" + **//dynamically add the wind direction of each city to the wind direction column and use string concatenation to concatenate the unit which is degrees**

'</td>' +

'<td>' +

data.cities[i].windChillFactor + "°" + **//dynamically add the wind chill factor of each city to the wind chill factor column and use string concatenation to concatenate the unit which is degrees**

'</td>';

table.appendChild(row); **// The data in the row element containing the cities weather details are added to the table created in the html using the appendChild() method**

}

};

};

**HTML Code Explanation**

<!DOCTYPE html>**//html DOCTYPE declaration which informs the browser what kind of document it is expecting.**

<html lang="en">

**//this indicates the language of the contents in the html. It is English in the case.**

**//Head tag**

<head>

<meta charset="UTF-8" /> **UTF-8 character encoding is used here.**

<meta http-equiv="X-UA-Compatible" content="IE=edge" /> **//this is used to select the IE=edge as the the version of internet explorer when the html content will be rendered.**

<meta name="viewport" content="width=device-width, initial-scale=1.0" />**//this is used to set the page width depending on the device on which it is loaded.**

<title>Uk Weather</title> **//set page title**

<link rel="stylesheet" href="style.css" />**//this links the style.css file to the html page**

</head>

<body>

<!-- nav bar-->

<div class="nav-bar"> **//div element containing the navbar**

<h1>

<a href="weather.html"> **//this link reloads the html page when clicked**

<h1>UK weather forecast</h1>**//nav-bar text set with an h1 tag**

</a>

</h1>

</div>

<!--end of nav bar-->

<!--set table content-->

<table id="table-display"> **//set table element that will be dynamically loaded from the script file**

</table>

<!--end of table content-->

<!--set error message-->

<div id="update-status-message"></div> **//set div element that will be automatically updated if error occurs while loading data**

<!--end of error message-->

<script src="weather.js"></script>**// script tag that links the JavaScript file to the HTML**

</body>**//end of body tag**

</html> **//end of html tag**

**CSS Code Explanation**

**/\*Style for the body of html\*/**

body {

background: linear-gradient(rgba(0, 0, 0, 0.7), rgba(0, 0, 0, 0.7)), url("../img/forest.jpg") center/cover;

background-attachment: fixed;

margin: 0;

--white: #ffffff;

--red: rgb(190, 29, 29);

--blue: #272759;

--dark: #333;

--gray: lightgray;

font-family: Arial, Helvetica, sans-serif;

text-transform: capitalize;

}

**/\*style for the nav bar\*/**

.nav-bar {

top: -10px;

width: 100%;

margin-top: 0px;

overflow: hidden;

position: fixed;

background-color: var(--dark);

color: var(--white);

}

**/\*style for the text content in the navbar\*/**

.nav-bar h1 a {

color: var(--white);

text-decoration: none;

text-align: center;

font-size: 1.2rem;

}

**/\*style for the hover effect on the text in the navbar\*/**

.nav-bar h1 a:hover {

font-style: italic;

text-decoration: underline;

}

**/\*style for the table in the html page\*/**

#table-display {

margin-top: 9rem;

margin-left: 20%;

border-collapse: collapse;

min-width: 400px;

text-align: center;

border-radius: 5px 5px 0 0;

overflow: hidden;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);

}

**/\*style for the table header row\*/**

#table-display th {

background-color: var(--blue);

color: var(--white);

text-align: left;

font-size: 0.9rem;

font-weight: bold;

}

**/\*set padding for the table row and table data\*/**

#table-display th,

#table-display td {

padding: 12px 15px;

}

**/\*style for the border bottom of the table\*/**

#table-display tr {

border-bottom: 1px solid var(--white);

}

**/\*style to target all even rows in the table and set background colour\*/**

#table-display tr:nth-of-type(even) {

background-color: var(--gray);

}

**/\*style to target all odd rows in the table and set background colour\*/**

#table-display tr:nth-of-type(odd) {

background-color: var(--white);

}

**/\*style to target the last row in the table and set border bottom\*/**

#table-display tr:last-of-type {

border-bottom: 2px solid var(--blue);

}

**/\*style to indicate the rows that are active in the table\*/**

#table-display tr.active-row {

font-weight: bold;

color: var(--blue);

}

**/\*style for the update message in case there is an error in loading table data\*/**

#update-status-message {

margin-top: 4rem;

margin-left: 30%;

color: var(--red);

font-weight: 900;

font-size: 40px;

}

**JSON Code**

{

"cities": [{

"id": 1,

"name": "Bath",

"currentCondition": "Cloud",

"temperature": 7,

"windSpeed": 4.60,

"windDirection": 90,

"windChillFactor": 3.78

},

{

"id": 2,

"name": "Bristol",

"currentCondition": "Sun",

"temperature": 15,

"windSpeed": 4.20,

"windDirection": 180,

"windChillFactor": 12.06

},

{

"id": 3,

"name": "Birmingham",

"currentCondition": "Sleet",

"temperature": 8,

"windSpeed": 3.25,

"windDirection": 210,

"windChillFactor": 5.73

},

{

"id": 4,

"name": "Bradford",

"currentCondition": "Sun and cloud",

"temperature": 10,

"windSpeed": 2.98,

"windDirection": 195,

"windChillFactor": 7.91

},

{

"id": 5,

"name": "Bournemouth",

"currentCondition": "Heavy rain",

"temperature": 5,

"windSpeed": 5.23,

"windDirection": 270,

"windChillFactor": 1.34

},

{

"id": 6,

"name": "Cambridge",

"currentCondition": "Rain",

"temperature": 6,

"windSpeed": 4.68,

"windDirection": 220,

"windChillFactor": 2.72

},

{

"id": 7,

"name": "Canterbury",

"currentCondition": "Thunderstorm",

"temperature": 5,

"windSpeed": 4.56,

"windDirection": 310,

"windChillFactor": 1.81

},

{

"id": 8,

"name": "Chester",

"currentCondition": "Snow",

"temperature": 1,

"windSpeed": 0.89,

"windDirection": 305,

"windChillFactor": 0.38

},

{

"id": 9,

"name": "Derby",

"currentCondition": "Heavy cloud",

"temperature": 11,

"windSpeed": 3.47,

"windDirection": 185,

"windChillFactor": 8.57

},

{

"id": 10,

"name": "Exeter",

"currentCondition": "Mist",

"temperature": 13,

"windSpeed": 3.34,

"windDirection": 230,

"windChillFactor": 10.66

},

{

"id": 11,

"name": "Armagh",

"currentCondition": "Hail",

"temperature": 10,

"windSpeed": 4.00,

"windDirection": 95,

"windChillFactor": 7.2

},

{

"id": 12,

"name": "Belfast",

"currentCondition": "Sun",

"temperature": 20,

"windSpeed": 2.29,

"windDirection": 200,

"windChillFactor": 18.34

},

{

"id": 13,

"name": "Londonderry",

"currentCondition": "Sun and cloud",

"temperature": 14,

"windSpeed": 2.99,

"windDirection": 45,

"windChillFactor": 11.91

},

{

"id": 14,

"name": "Aberdeen",

"currentCondition": "Rain",

"temperature": 8,

"windSpeed": 4.12,

"windDirection": 80,

"windChillFactor": 5.12

},

{

"id": 15,

"name": "Glasgow",

"currentCondition": "Thunderstorm",

"temperature": 4,

"windSpeed": 4.89,

"windDirection": 50,

"windChillFactor": 0.58

},

{

"id": 16,

"name": "Cardiff",

"currentCondition": "Cloud",

"temperature": 9,

"windSpeed": 2.99,

"windDirection": 85,

"windChillFactor": 6.91

},

{

"id": 17,

"name": "Swansea",

"currentCondition": "Sun",

"temperature": 22,

"windSpeed": 1.99,

"windDirection": 315,

"windChillFactor": 20.61

},

{

"id": 18,

"name": "Newport",

"currentCondition": "Sun and cloud",

"temperature": 12,

"windSpeed": 2.64,

"windDirection": 240,

"windChillFactor": 10.15

},

{

"id": 19,

"name": "Bangor",

"currentCondition": "Rain",

"temperature": 10,

"windSpeed": 3.59,

"windDirection": 290,

"windChillFactor": 7.49

},

{

"id": 20,

"name": "Dundee",

"currentCondition": "Mist",

"temperature": 8,

"windSpeed": 2.50,

"windDirection": 110,

"windChillFactor": 6.25

}

]

}

**JSONSchema Code**

{

"$schema": "http://json-schema.org/draft-04/schema#",

"type": "object",

"properties": {

"cities": {

"type": "array",

"items": [{

"type": "object",

"properties": {

"id": {

"type": "integer",

"minimum": 0

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string",

"enum": [

"Cloud",

"Rain",

"Heavy rain",

"Mist",

"Thunderstorm",

"Snow",

"Sun and cloud",

"Sun",

"Hail",

"Heavy cloud",

"Sleet"

]

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

},

{

"type": "object",

"properties": {

"id": {

"type": "integer"

},

"name": {

"type": "string"

},

"currentCondition": {

"type": "string"

},

"temperature": {

"type": "integer"

},

"windSpeed": {

"type": "number"

},

"windDirection": {

"type": "integer"

},

"windChillFactor": {

"type": "number"

}

},

"required": [

"id",

"name",

"currentCondition",

"temperature",

"windSpeed",

"windDirection",

"windChillFactor"

]

}

]

}

},

"required": [

"cities"

]

}