7.10 Using third-party web APIs (JavaScript)

Introduction

Many organizations have created public web APIs that provide access to the organization's data or the user's data that is stored by the organization. Ex: The Google Maps API provides applications information about geographic locations, and the Instagram API allows applications access to photos shared on Instagram. Public APIs on GitHub.com lists thousands of free, public web APIs.

A ***third-party web API*** is a public web API used by a web application to access data provided by a third party. "Third-party" refers to a person or organization that is neither the web application using the API nor the user using the web application, which are the "first" and "second" parties. Websites rely on third-party web APIs to integrate with social media, obtain maps and weather data, or access collections of data.

To use a third-party web API, a developer usually registers with the third party to obtain an ***API key***. Third parties require API keys for several reasons:

- The API key identifies who or what application is using the web API.
- The API key helps the third party limit the number of requests made to the API in a fixed time period or may be used to charge a developer a fee for additional requests.
- To obtain an API key, developers must agree to restrictions the third party places on data obtained from the web API.

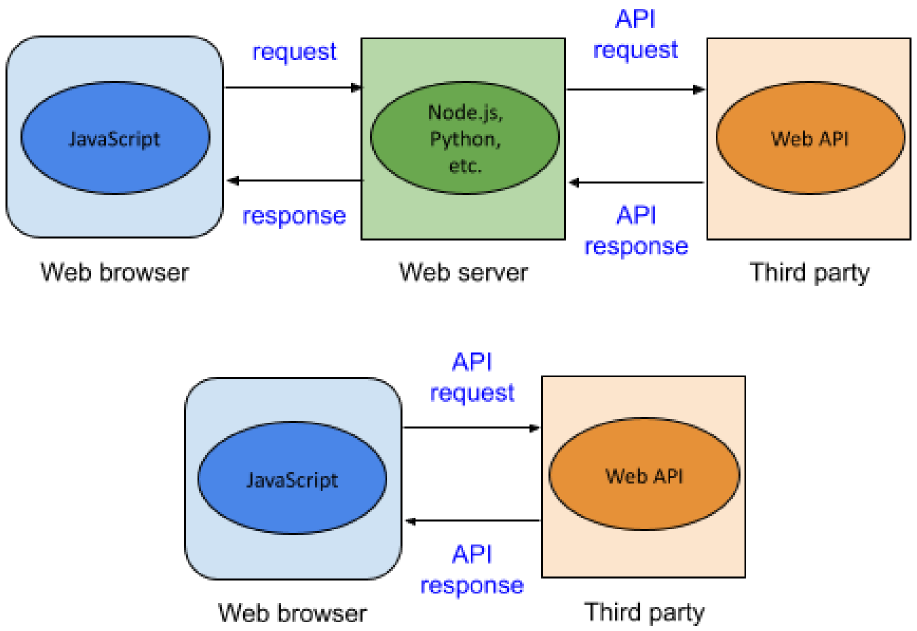
Most third-party web APIs are RESTful. A ***RESTful web API*** is a web API that is called with a URL that specifies API parameters and returns JSON or XML containing the API data. Ex: The URL http://linkedin.com/api/article?id=123 specifies the article ID 123, so the article would be returned formatted in JSON.

Third-party web APIs may be called from the web server or the web browser. This material shows how to call web APIs from the web browser using JavaScript.

SOAP

A ***SOAP-based web API*** *is another type of web API that relies heavily on XML and is in general more complex to use than RESTful web APIs. See the "Exploring further" section for more information on SOAP.*

Figure 7.10.1: Calling third-party web API from the web server or web browser.



Feedback?

PARTICIPATION ACTIVITY 7.10.1: Third-party web APIs. ✓

1) Information from a third-party web API reaches the browser faster if the browser calls the web API directly instead of the web server calling the web API.

● True

○ False

Correct

The web server in between the browser and third party introduces extra time due to the request/response between the browser and web server.

✓

2) For a third-party web API requiring an API key, the API key must be transmitted with every API request.

● True

○ False

Correct

The third party uses the API key to track which application is making the API request.

✓

3) When the browser makes an API request to a third-party web API, the web API key can be kept secret from prying eyes.

○ True

● False

✓

Correct

A user can examine API requests in the web browser's developer tools to discover a web application's web API key. Many developers prefer to make requests to third-party web APIs from the web server to protect APIs keys.

4) Many web APIs charge a fee to the developer after a limited number of requests have been made in a 24-hour period.

● True

○ False

✓

Correct

Google's Custom Search API is an example of an API that provides a limited number of API requests per day for free and charges a fee for additional requests.

5) RESTful web APIs only return XML.

○ True

● False

✓

Correct

RESTful web APIs often return JSON or XML. JSON and XML are two ways to format data in a way that programs can easily parse and extract the data.

Feedback?

Weather API

OpenWeatherMap provides a free Weather API providing current weather data, forecasts, and historical data. Developers must register at openweathermap.org for an API key that must be transmitted in all API requests.

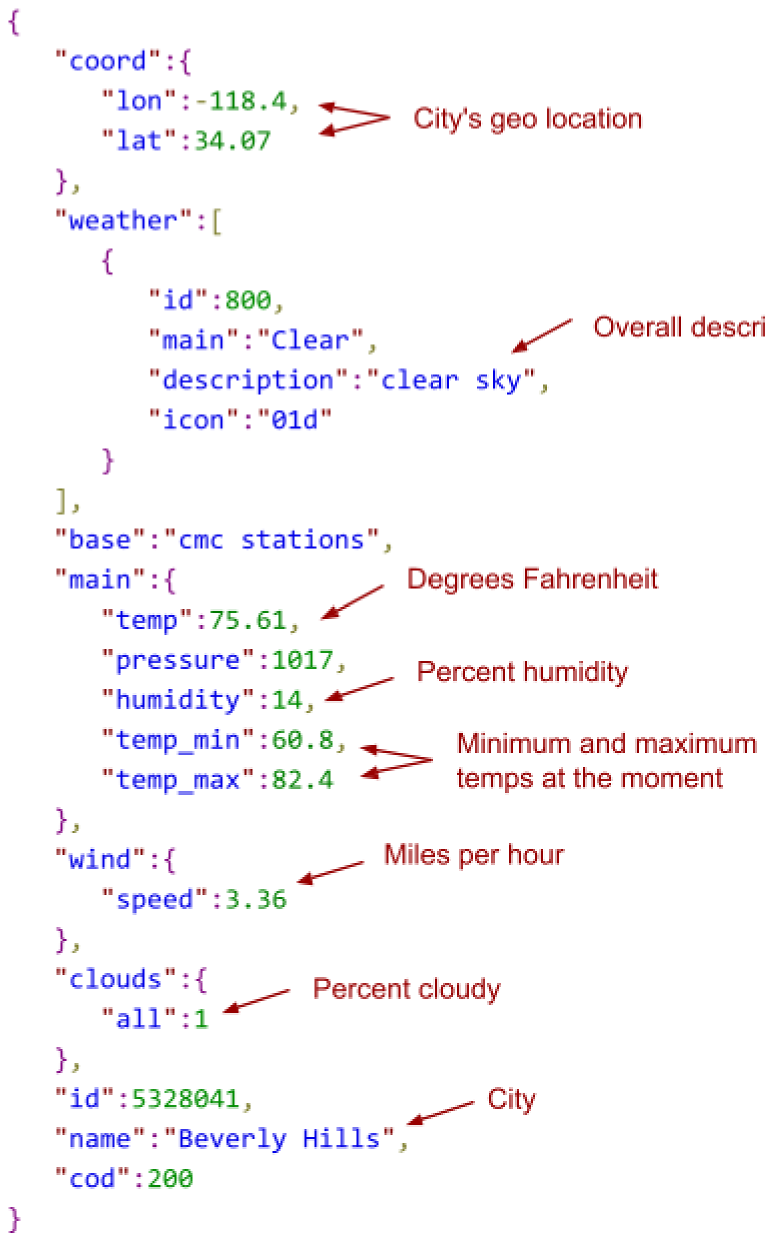
The OpenWeatherMap website provides documentation explaining how to use the Weather API using GET requests with various query string parameters. The API endpoint `http://api.openweathermap.org/data/2.5/weather` returns the current weather based on the following query string parameters:

- zip - Five digit US ZIP code
- units - Standard, metric, or imperial units to use for measurements like temperature and wind speed
- appid - Developer's API key

Other parameters are documented in the OpenWeatherMap website. The Weather API returns weather data in JSON format by default.

Figure 7.10.2: GET request to obtain the current weather for ZIP 90210.

http://api.openweathermap.org/data/2.5/weather?**zip**=90210&**units**=imperial&**appid**=APIKEY



```
{
   "coord":{
      "lon":-118.4,
      "lat":34.07
   },
   "weather":[
      {
         "id":800,
         "main":"Clear",
         "description":"clear sky",
         "icon":"01d"
      }
   ],
   "base":"cmc stations",
   "main":{
      "temp":75.61,
      "pressure":1017,
      "humidity":14,
      "temp_min":60.8,
      "temp_max":82.4
   },
   "wind":{
      "speed":3.36
   },
   "clouds":{
      "all":1
   },
   "id":5328041,
   "name":"Beverly Hills",
   "cod":200
}
```

Feedback?

Try 7.10.1: Try OpenWeatherMap's API in your web browser.

1. Go to openweathermap.org.
2. Sign up for an account to obtain an API key.

3. When your API key is ready, try the link: http://api.openweathermap.org/data/2.5/weather?zip=90210&units=imperial&appid=APIKEY to make an API request for the weather with ZIP 90210. The page should indicate an invalid API key was used.
4. Replace APIKEY in the URL's query string with your API key, and reload the webpage. The JSON-encoded weather information for 90210 should be displayed.
5. Change the ZIP code in the URL's query string to your ZIP code, and reload the URL to see the weather in your ZIP code.

Feedback?

PARTICIPATION ACTIVITY 7.10.2: The Weather API. ✓

1) What does the Weather API return when an invalid API key is used in a request?

○ A blank webpage

○ Weather for the 90210 ZIP

● An error message formatted in JSON

✓ **Correct**

An "Invalid API key" message is returned.

2) In the figure above, what does the Weather API return as the current humidity in the 90210 ZIP code?

○ 75.61

● 14

○ 3.36

✓ **Correct**

14% is the current humidity level.

3) What "units" parameter value would make the Weather API return the temperature in Celsius?

○ imperial

● metric

○ standard

✓

Correct

Metric returns the temperature in Celsius.

4) Does the Weather API support finding the current weather by city name?

● Yes

○ No

✓

Correct

The Weather API may be searched by city name as in this example query string:

```
?q=Denver&appid=APIKEY
```

Feedback?

Cross-origin requests

Calling a third-party web API from the web browser requires a cross-origin HTTP request, since the web API is not hosted on the local website's web server. Two main techniques are used to make cross-origin requests:

- ***Cross-Origin Resource Sharing*** (***CORS***) is a W3C specification for how web browsers and web servers should communicate when making cross-origin requests.
- ***JSON with Padding*** (***JSONP***) is a technique to circumvent cross-origin restrictions by injecting `<script>` elements dynamically into a webpage. Script elements have no cross-origin restrictions.

CORS is the more common of the two techniques and, for the web API user, the easiest to use. CORS requires the web browser to send an `Origin` header in a web API request to indicate the scheme and domain making the API request. If the API accepts the request, the API responds with an `Access-Control-Allow-Origin` header indicating the same value in the `Origin` request header or "`*`", which indicates that requests are allowed from any origin. CORS uses other headers that begin with `Access-Control-*` to support other interactions with the API.

CORS allows the browser to send GET, POST, PUT, and DELETE requests. JSONP limits the browser to sending only GET requests.

Figure 7.10.3: Making a request to the Weather API with CORS.

HTTP request	HTTP response
GET /data/2.5/weather? zip=90210&units=imperial&appid=APIKEY HTTP/1.1 Host: api.openweathermap.org Origin: http://mywebsite.com User-Agent: Mozilla/5.0 Chrome/48.0.2564	HTTP/1.1 200 OK Access-Control-Allow-Origin: * Content-Type: application/json; charset=utf-8 Content-Length: 431 Date: Mon, 28 Mar 2016 16:09:48 Server: openresty {"coord": {"lon":-118.4,"lat":34.07},"weat [{"id":500, "main":"Rain","description":"lig rain","icon":"10d"}], etc...}

Feedback?

PARTICIPATION ACTIVITY 7.10.3: Cross-origin requests. ✓

1) What HTTP header must the web browser send in every CORS request?

○ Access-Control-Allow-Origin

● Origin

○ User-Agent

Correct

The browser must add the Origin header in the request. ✓

2) The web browser knows to send the `Origin` header in the HTTP request when the requested URL's domain name and the requesting script's domain name are _____.

○ the same

● different

Correct

CORS is needed only when making a cross-origin HTTP request. ✓

3) When a third-party web API does not support CORS, what is Access-Control-Allow-Origin set to in the web API's response?

○ Access-Control-Allow-Origin is set to *.

○ Access-Control-Allow-Origin is set to the Origin value.

● Access-Control-Allow-Origin is not present.

Correct

The web browser will not allow access to the response if the Access-Control-Allow-Origin header is missing.

✓

4) Does JSONP support POST or PUT request methods?

○ Yes

● No

Correct

GET is the only request method available to JSONP. CORS supports all request methods.

✓

Feedback?

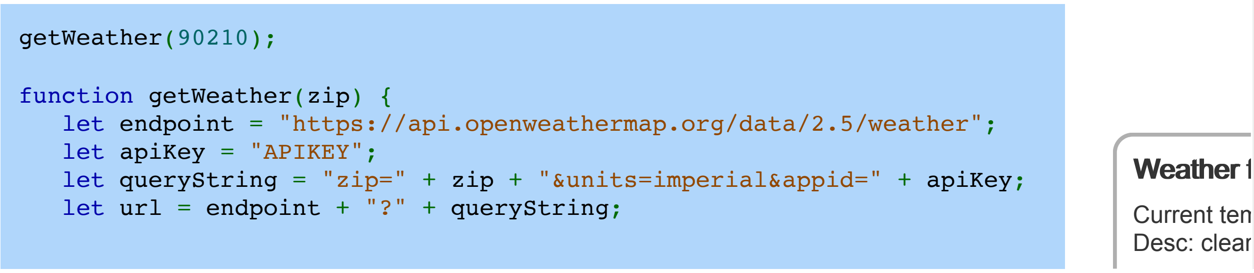
Calling the Weather API from JavaScript

The Weather API may be called from JavaScript using the `XMLHttpRequest` object, which makes asynchronous HTTP requests. The OpenWeatherMap implements CORS, and API requests can come from any origin.

The animation below shows how to retrieve weather information for a given ZIP code. For the JavaScript code to work in a web browser, the "APIKEY" string needs to be replaced with an actual API key.

PARTICIPATION ACTIVITY 7.10.4: Calling the Weather API with JavaScript. ✓

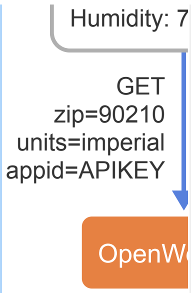
■ 1 2 3 4 5 ◀ ☑ 2x speed



```
getWeather(90210);

function getWeather(zip) {
   let endpoint = "https://api.openweathermap.org/data/2.5/weather";
   let apiKey = "APIKEY";
   let queryString = "zip=" + zip + "&units=imperial&appid=" + apiKey;
   let url = endpoint + "?" + queryString;
   let xhr = new XMLHttpRequest();
   xhr.addEventListener("load", responseReceivedHandler);
   xhr.responseType = "json";
   xhr.open("GET", url);
   xhr.send();
}

function responseReceivedHandler() {
   let weatherInfo = document.getElementById("weather");
   if (this.status === 200) {
      weatherInfo.innerHTML =
       "<p>Current temp: " + this.response.main.temp + " &deg;F</p>" +
       "<p>Desc: " + this.response.weather[0].description + "</p>" +
       "<p>Humidity: " + this.response.main.humidity + "%</p>";
   } else {
      weatherInfo.innerHTML = "Weather data unavailable.";
   }
}
```



Weather information is extracted from this.response and displayed in the webpage.

Captions

1. getWeather() creates a URL to request the current weather for the 90210 ZIP.
2. The XMLHttpRequest object sends a GET request to the Weather API.
3. OpenWeatherMap responds with JSON containing the current weather for ZIP code 90210. responseReceivedHandler() executes when the browser receives the JSON response.
4. this.status is 200 unless the ZIP code is not found.
5. Weather information is extracted from this.response and displayed in the webpage.

Feedback?

PARTICIPATION ACTIVITY 7.10.5: Calling the Weather API from JavaScript. ✓

Refer to the animation above.

1) What JavaScript variable must be modified for the webpage to correctly access the Weather API?

○ `endpoint`

● `apiKey`

○ `queryString`

✓

Correct

`apiKey` must be initialized with the developer's API key. Otherwise the Weather API returns a 401 response.

2) What JavaScript variable must be modified if the webpage is to display the temperature in Celsius instead of Fahrenheit?

○ `endpoint`

○ `apiKey`

● `queryString`

Correct

The query string parameter `units` must be changed from `imperial` to `metric`.

✓

3) What is the expected output for the call below?

```
getWeather("test");
```

● "Weather data unavailable."

○ Temperature for "test" ZIP code.

○ Exception is thrown.

Correct

The Weather API returns a 404 status code when the ZIP code cannot be found, as "test" is not a valid ZIP code.

✓

4) According to the figure above that shows the Weather API's JSON response, what variable in `responseReceivedHandler()` contains the wind speed?

○ `this.status.speed`

○ `this.response.speed`

● `this.response.wind.speed`

Correct

The JSON response shows "speed" under "wind", so the `this.response` object is given a `wind` property that has a `speed` property.

✓

Feedback?

Exploring further:

- Public APIs on GitHub.com
- Understanding SOAP and REST Basics And Differences

How was this section? 👍 👎 **Provide section feedback**