Assignment 4 AJAX Using Open Data

“Open data is the idea that some data should be freely available to everyone to use and re-publish as they wish, without restrictions from copyright, patents or other mechanisms of control.” - Wikipedia

Instructions

Use these starting files. Failure to do so will result in a mark of 0 for this assignment.

For this assignment you can pick a new JSON dataset or use the same dataset you used in the Open Data practice exercise. The dataset you use for this part of the assignment **must** be from the City of Winnipeg Open Data portal. A list of potential datasets can be found below or you could explore the open data portal to find your own dataset.

Your task is to build an AJAX search form for this dataset that queries the City of Winnipeg Open Data API and displays search results below the form. Your form should include one or more inputs that correspond to fields within your selected JSON dataset. When the form is submitted you should use the form data to query the Open Data API. Your search should allow users to find a subset of records from a specific dataset. To do this you will be using the WHERE, ORDER and LIMIT capabilities of the Open Data API explained below. Your search form should return anywhere from 0 to a maximum of 100 results. If zero results are found, display a message letting the user know that nothing was found.

Only HTML, CSS and Javascript should be used for this portion of the assignment. *No PHP code should be used. (Your form will request JSON data from the open data API.)*

Demo Version

A demo version can be found here. This demo uses the city's tree inventory dataset. Please note that you cannot replicate this example for your assignment.

The demo version includes a form with a single input. When submitted it searches the tree inventory for the top 100 largest trees that include the searched for name in their common name.

This is done by crafting a special kind of URL request to the Open Data API:

```
let commonName = 'Ash'; // This would actually be coming from the
form input.
const apiUrl = 'https://data.winnipeg.ca/resource/d3jk-hb6j.json?'
+
                `$where=common_name LIKE '%${commonName}%'` +
                '&$order=diameter_at_breast_height DESC' +
                '&$limit=100';
const encodedURL = encodeURI(apiUrl);
```

Note the use of a back-ticked (`` ` ``) template literal in the above code to inject the `commonName` variable into the URL. Also note that we must encode the URL before we use it with `fetch` to properly encode spaces and other restricted characters.

Querying the Open Data API

Notice how in the example URL above there are SQL-like keywords being used like WHERE, ORDER and LIMIT. The City's Open Data API is built on the Socrata platform and this style of querying is called the "Socrata Query Language" or "SoQL". More information about SoQL can be found here.

You should use at least WHERE and ORDER clauses.

If you want your WHERE clause to be case insensitive you can use the `lower` function. For example:

```
`$where=lower(common_name) LIKE lower('%${commonName}%')`
```

Some Open Data Datasets

- Locations and Amenities of City Recreation Complexes: https://data.winnipeg.ca/Recreation/Recreation-Complex/xuqw-wemm
- City Air Quality: https://data.winnipeg.ca/Organizational-Support-Services/Air-Quality/f58p-2ju3
- Parks and Open Spaces: https://data.winnipeg.ca/Parks/Parks-and-Open-Space/tx3d-pfxq
- Daily Adult Mosquito Trap Data: https://data.winnipeg.ca/Insect-Control/Daily-Adult-Mosquito-Trap-Data/du7c-8488
- Building Permits Issues since 2010: https://data.winnipeg.ca/Development-Approvals-Building-Permits-Inspections/Detailed-Building-Permit-Data/it4w-cpf4
- City Council Member Expenses: https://data.winnipeg.ca/Council-Services/Council-Member-Expenses/mgde-4fua
- 311 Service Request: https://data.winnipeg.ca/Contact-Centre-311/311-Service-Request/4her-3th5

- Winnipeg Election Results back to 1966: https://data.winnipeg.ca/Council-Services/Winnipeg-Election-Results/7753-3fjc
- Parking tickets since 2010: https://data.winnipeg.ca/Parking/Parking-Contravention-Citations-/bhrt-29rb
- Current lane and street closures: https://data.winnipeg.ca/Transportation-Planning-Traffic-Management/Lane-Closure/h367-iifg

To find the JSON endpoint for these datasets:

- If it's a map go to "Export" and then "SODA API" and copying the API Endpoint URL.
- If it's a regular dataset just click on the API button and grab the endpoint URL.

Rubric

Your assignment will be marked out of 10 possible points.

Starting with a mark of 10, use the following rubric to determine how many (if any) points would be deducted from your mark.

IMPORTANT: Your work should not look or work just like the demo; come up with an original idea!

Also, your solutions to this assignment should not be the same as another student's. (Everything should be your own work, of course.)

	Ineffective (3 mark deduction)	**Adequate (2 mark deduction)**	**Effective (0 deductions)**

Form Submission	Submitting the form submits the data to a separate script, rather than resulting in an AJAX request.		Submitting the search form results in an AJAX fetch. The form inputs are not submitted to another URL.
Open Data Dataset	The filtered dataset was either not retrieved or the retrieval resulted in an error.		The dataset was properly filtered using the supplied form data with a SOQL request to the API; it should include WHERE and ORDER clauses.
Search Results	The retrieved filtered dataset is not added to the page at all, or an error occurs during this step.	The retrieved filtered dataset is properly added to the page, but there is no custom message for empty search results.	The retrieved filtered dataset is added to the page using the DOM. Past results are cleared.