

CORNELL UNIVERSITY LIBRARY

---

# Creating Maps

Geographic Visualization in Drupal 8

Presented by

Liz Woods, Cornell University Library

# First, we need some data to map.

- Free dataset of world cities and populations from SimpleMaps: <http://simplemaps.com/data/world-cities>
- It's in CSV format – we will import this into Drupal after shuffling the format a bit.
- Requirements:
  - First column needs to be a sequential numerical index
  - First row needs to be field names surrounded by single quotes
  - Data columns must line up with content type fields
  - CSV file needs to be in Unicode-8. Experiments show: do NOT edit in Excel! Use a CSV editor, if you have one, or try OpenRefine (<http://openrefine.org/>).

# It's a little bit convoluted.

- Create a project in OpenRefine – imports your CSV file into a database-like editing tool.
- In this sample case, we sort by population. Then, we'll remove columns of data that we don't want to import.
- Add a column and set its value to `row.index` to get our unique ID, name it "id" and put it at the start of the data table.
- For the sake of this demo, we will use just the most populous cities. Now that we have a numeric ID column, we can facet on that ID and take the number of data points we want.
- Lastly, export data back to CSV. Ready to import!

# Next, set up Drupal to receive data.

- Modules to enable:
  - Migrate \*
- Modules to install and enable:
  - Migrate CSV \*
  - Migrate Tools \*
  - Migrate Plus \*
  - Libraries API

\* You won't want to leave these modules enabled, so remember to turn them off when you're done with the migration.

# Watch out for the pitfalls...

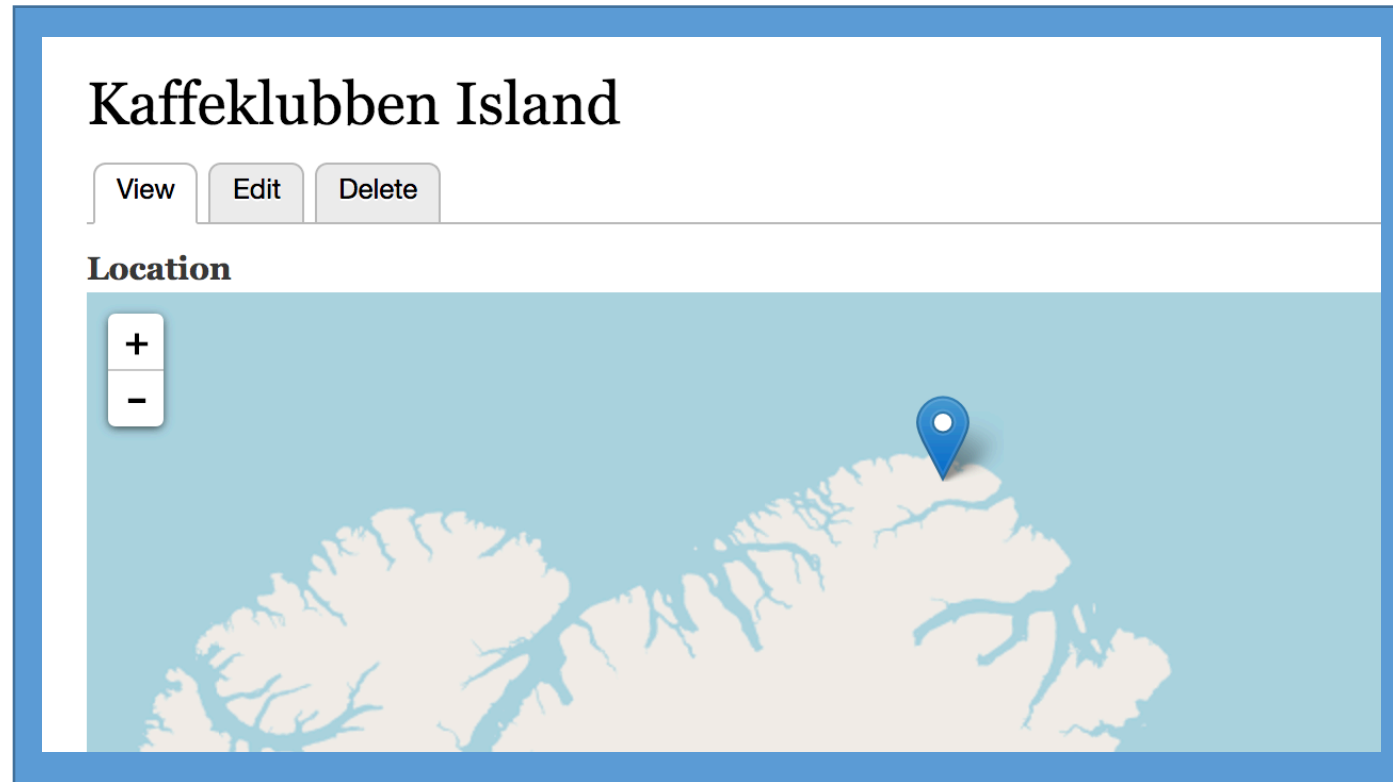
- Modules to install (do NOT enable in the same step):
  - Geofield
  - Leaflet
  - Leaflet More Maps
- Geofield requires geoPHP, managed as a dependency via composer. The install instructions on the module page are incorrect (as of 10/17/17) – use this instead as a reference: <https://www.drupal.org/node/2728709>
- Leaflet requires a small patch until the next version comes out – apply manually for best results. <https://www.drupal.org/node/2839538>
- Leaflet also requires Leaflet Library (download from <http://leafletjs.com/download.html>) and install to /code/libraries/)

# Create a content type.

- Our new content type needs to have fields that map to all of the columns the CSV we just created, except for the ID key column.
- When you create a field in the content type for latitude and longitude, you will create ONE field of type Geofield – here, let's call it 'Location'. Both latitude and longitude will map to that field.
- In the Manage Display section for your new content type, choose "Leaflet Map" for your Geofield.
- Create one node of data in your new content type, so that the database fields are created prior to starting your import. You can delete this later.

# Confirm that everything is working so far.

- Look at your new node and bask in the success of your mini map!



# Okay, time for that data migration.

- You'll need to write a migration profile in YAML that maps your CSV data onto the content type fields you created. For this, you need the **machine names** of each field in the content type. The profile will map column numbers to field names – so put the field mappings in the left-to-right order in which they appear in the CSV, and not the order in which they are displayed in the content type.
- References: <https://www.mtech-llc.com/blog/lucas-hedding/migrating-using-csv> and <https://evolvingweb.ca/blog/drupal-8-migration-migrating-basic-data-part-1>
- We will use the Configuration Manager to set the migration parameters, and then CLI drush commands (provided from the Migrate Tools module) to actually set and complete the migration.
- The magic incantation: `$ drush migrate-import [[profile-id]] --update`



But wait, I thought this session  
was about **maps**.

# Yep. The map is easy once you have data.

- Create a view with a page display of content type [[City]]
- Set the page display type to Leaflet Map (old)
- There are settings for the map as a whole in the Geofield field-specific settings, but they won't work right. Make sure you set the map type and other settings also in the view FORMAT settings.
- To get your supporting information to appear, use <node entity> as the format for Description (in view format settings). Then, you can adjust field display through the content type settings if needed and style via CSS.

# Fiddle with your view, style, and icon settings.

- Tip: **use the Teaser view and omit the geofield from that display** if you want to have node-based popups on your data points.
- Use the custom icon settings on the view format, not the geofield. The origin point fields are tricky – use a known location against which to calibrate.
- If you are seeing too many repeats of the world on your map, adjust the height settings or zoom level.

Enjoy your mapping odyssey!

Questions? Comments?

Thank you!!