



Putting Location Data on the Map in Drupal 8

March 10, 2018



Overview

1. Background
2. Requirements
3. Core, Contributed, and Custom Modules
4. Importing Data from Dynamic and Static Sources
5. Processing and Formatting Imported Data
6. Page Display - Lists, Maps, and Detail Pages
7. Making Location Data Available via RESTful API
8. Problems Encountered
9. Questions and Answers

Background

Genero

Matthew Messmer

Director, International Business / Genero

D.O PROFILE / [machostache](#)

- Acquia Certified Drupal Grand Master
- Over seven years of Drupal experience
- Not socially awkward

Background

Clarity Partners

Robert Shell

Senior Consultant / Clarity Partners

D.O PROFILE / [robertshell22](#)

- Acquia Certified Developer
- Acquia Certified Front End Specialist
- Over nine years of Drupal experience

Background

Chicago Park District



-
- 600 park locations
 - Over 2,000 facility locations
 - 26 miles of lakefront, including 23 swimming beaches
 - Thousands of events and programs every year
 - Serving ALL of Chicago

Background

Chicago Park District

Website Stats

- **Website:** <https://www.chicagoparkdistrict.com>
- **Pageviews:** 100,000 – 500,000 per day
- Popular Content:
 - Programs
 - Parks and Facilities
 - Events
 - Schedules
- Highly Location-based

Requirements

IMPORT DATA FROM MULTIPLE SOURCES

- ◆ Basic park data from an unsupported SOAP service
- ◆ Park and facility data from old website using CSV files
- ◆ Daily updates of data via SOAP service

DISPLAY DATA ON THE PAGE

- ◆ Lists of parks and facilities
- ◆ Maps of parks and facilities
- ◆ Park and facility data on detail pages

OUTPUT DATA TO EXISTING MOBILE APP

- ◆ Specific RESTful API Feeds for multiple uses

	abbreviatedName	address1	city	state	zipCode	count
3	A98646	5873 N. Nina Ave	Chicago	IL	60631	US
1	A103841	3315 W. Carmen Ave.	Chicago	IL	60625	US
2	A103842	3315 W. Carmen Ave.	Chicago	IL	60625	US
3	A103843	3315 W. Carmen Ave.	Chicago	IL	60625	US
1	A111801	2941 N. Washtenaw Ave.	Chicago	IL	60618	US
3	A111813	2133 N. McVicker Ave	Chicago	IL	60639	US
2	A121982	4320 W. 5th Ave.	Chicago	IL	60624	US
3	A121983	4320 W. 5th Ave.	Chicago	IL	60624	US
4	A121984	4320 W. 5th Ave.	Chicago	IL	60624	US
3	A125813	2645 N. Sheffield Ave.	Chicago	IL	60614	US
2	A136122	2228 N. Oakley Ave	Chicago	IL	60647	US
3	A136123	1200 South Lake Shore Drive	Chicago	IL	60647	US
3	A139873	N. Western Ave. and Logan Blvd.	Chicago	IL	60647	US
4	A139874	1200 South Lake Shore Drive	Chicago	IL	60647	US
1	A140111	1200 South Lake Shore Drive	Chicago	IL	60614	US
4	A141514	1200 South Lake Shore Drive	Chicago	IL	60646	US
5	A141585	1410 S. Museum Campus	Chicago	IL	60605	US
5	A141586	1800 S. Lake Shore Dr.	Chicago	IL	60605	US
0	A141770	1200 South Lake Shore Drive	Chicago	IL	60643	US
1	A141771	1200 South Lake Shore Drive	Chicago	IL	60643	US
4	A141784	7705 N. Eastlake Terrace	Chicago	IL	60626	US
9	A144079	11400 S. Stony Island Ave.	Chicago	IL	60633	US
0	A144080	13200-13298 S. Torrence Ave.	Chicago	IL	60633	US
1	A144081	2200 E. 103rd St.	Chicago	IL	60633	US
2	A144082	13600 S. Crandon Ave.	Chicago	IL	60633	US
4	A145054	3501 N. Kilbourn Ave.	Chicago	IL	60641	US
5	A145055	3501 N. Kilbourn	Chicago	IL	60641	US
2	A145482	1208 W. Jarvis Ave.	Chicago	IL	60626	US
3	A145483	1050 W. Pratt Ave.	Chicago	IL	60626	US
4	A145484	1200 South Lake Shore Drive	Chicago	IL	60626	US
7	A153389	7442 N. Ashland Ave.	Chicago	IL	60626	US
1	A160601	4807 W Gunnison St	Chicago	IL	60630	US
3	A160663	1725 N. Springfield Ave	Chicago	IL	60647	US

Modules and Methods

CONTRIBUTED MODULES

- ◆ Geolocation
- ◆ Geocoder
- ◆ Migrate Tools
- ◆ Migrate Plus
- ◆ Ultimate Cron

KEY CORE MODULES

- ◆ Migrate
- ◆ RESTful Web Services

CUSTOM MODULES

- ◆ General import functions, submodules for updates
- ◆ Custom blocks for maps and listings. Custom field formatters.

performing arts, photography, and ceramics. Late night basketball leagues began playing and hold five gymnasias. The park district purchased the armory in 1998, after the state decided the purposes.

Parking Directions

Metered parking is available in the lot on site. [Get rates and additional information.](#)

For directions using public transportation visit www.transitchicago.com.

Map & Facilities

The screenshot shows a map application interface. At the top, there's a search bar labeled 'Park/Facility Name' with a search icon and a 'Filter' button below it. A search result popup is visible, showing 'Broadway Armory Fieldhouse' with the address '5917 N. Broadway Chicago, IL 60660' and a distance of '7.6 miles away from you'. The map below shows a street grid with labels for W Thorndale Ave, W Rosedale Ave, and N Magnolia Ave. Various points of interest are marked, including 'Cellars Bar and Grill', 'North Broadway Dental', and 'Broadway Armory Park'. Below the map, there's a list of facilities, each with a small image, a title, address, and distance. The first three items are: 'Building the Heart of Edgewater', 'Broadway Armory Mini Climbing Wall', and 'Broadway Armory Gymnasium'. Each item includes the address '5917 N Broadway Chicago, IL 60660' and '7.6 miles away from you' with a 'Show on map' link. To the right of the list, there's a vertical column of facility images and a circular logo for 'CHICAGO PARK DISTRICT ORGANIZED 1837'.

Importing Data

- **Sources:** Receiving dynamic and static data
- **Processing:** Getting the data mapped and into Drupal
- **Formatting:** Converting and transforming data for use

Importing Data Sources

1

Dynamic

- **APIs** - SOAP, REST, JSON, Services

2

Static

- **Documents** – CSV, TXT, XML, etc.

Importing Data Sources

Dynamic Sources

- Often little or no control over the data received
- Migrate module assumes one to one field mapping
- Service availability and authentication varies
- **Benefits:** Data is always current, automated processes
- **Challenges:** Extracting correct and properly formatted data

```

/**
 * {@inheritdoc}
 */
public function initializeIterator() {
    $wsdl_uri = \Drupal::service('file_system')->realpath(drupal_get_path('type', 'module', 'activenet_import') . '/wsdl/activenet.wsdl');

    // Create the soap client based on our wsdl files.
    $client = new \SoapClient($wsdl_uri);
    // allow the instance to set the operation we should use.
    $operation = $this->configuration['soap_operation'];
    $response = $client->$operation($this->params);
    $response_object = simplexml_load_string($response->return);
    $response_array = json_decode(json_encode($response_object), assoc: TRUE);
    $entries = $response_array['entries'];

    foreach ($entries as $entries_key => $entries_value) {
        if (is_array($entries_value)) {
            foreach ($entries_value as $value_key => $value_value) {
                if (is_array($value_value) && empty($value_value)) {
                    $entries[$entries_key][$value_key] = '';
                }
            }
        }
    }

    return new \ArrayIterator($entries);
}

/**
 * {@inheritdoc}
 */
public function fields() {
    $fields = array();
    if (!empty($this->configuration['fields'])) {
        $fields = $this->configuration['fields'];
    }

    return $fields;
}

```

SOAP Source Plugin

Importing Data Sources

Static Sources

- Complete control over the data that you are importing
- Guarantees one to one field mapping and pre-formatting
- Very large imports can be completed more efficiently
- **Benefits:** Complete control of data, no service required
- **Challenges:** Chance of stale data, coordinating vendor exports

121984	A121984	4320 W. 5th Ave.	Chicago	IL	60624	US	772
125813	A125813	2645 N. Sheffield Ave.	Chicago	IL	60614	US	772
136122	A136122	2228 N. Oakley Ave	Chicago	IL	60647	US	763
136123	A136123	1200 South Lake Shore Drive	Chicago	IL	60647	US	769
139873	A139873	N. Western Ave. and Logan Blvd.	Chicago	IL	60647	US	763
139874	A139874	1200 South Lake Shore Drive	Chicago	IL	60647	US	769
140111	A140111	1200 South Lake Shore Drive	Chicago	IL	60614	US	769
141514	A141514	1200 South Lake Shore Drive	Chicago	IL	60646	US	763
141585	A141585	1410 S. Museum Campus	Chicago	IL	60605	US	763
141586	A141586	1800 S. Lake Shore Dr.	Chicago	IL	60605	US	770
141770	A141770	1200 South Lake Shore Drive	Chicago	IL	60643	US	769
141771	A141771	1200 South Lake Shore Drive	Chicago	IL	60643	US	769
141784	A141784	7705 N. Eastlake Terrace	Chicago	IL	60626	US	771
144079	A144079	11400 S. Stony Island Ave.	Chicago	IL	60633	US	770
144080	A144080	13200-13298 S. Torrence Ave.	Chicago	IL	60633	US	770

Importing Data

Processing

- Transformation of data to different formats (Full Text, Date)
- Values can be mapped to one or many Drupal fields (Address)
- Lookups can be performed on migrate map table
- Able to match data without keys using custom plugins
- Values can be split using custom plugins (explode, dashes)

Migration YML

```
status: true
dependencies:
  enforced:
    module:
      - migrate_plus
id: activenet_parks
class: null
field_plugin_method: null
cck_plugin_method: null
migration_tags: null
migration_group: activenet
label: 'Activenet Parks'
source:
  plugin: activenet_soap
  soap_operation: getCenters
  fields:
    ID: 'Unique id and park id'
    abbreviatedName: 'name and title of park'
    address1: 'street address'
    city: 'city the park is in'
    state: 'state the park is in'
    zipCode: 'Zipcode of the park'
    country: 'The country the park is in'
    siteID: 'Not sure what this maps to yet'
    geographicAreaID: 'Chicago Community'
    retired: 'if the park has been retired'
    phone1: 'primary phone number'
    phone2: 'secondary phone number'
    parkingCapacity: 'number of parking spots at the park'
    name: 'full name of park'
    hideOnInternet: 'if this park should be published'
    centerPhone1/rawPhoneNumber: 'is this the real phone number'
    description: 'Park description'
  idfield: ID
process:
  uid:
    plugin: default_value
    default_value: 1
```

```
process:
  uid:
    plugin: default_value
    default_value: 1
  type:
    plugin: default_value
    default_value: location
  field_location_type:
    plugin: default_value
    default_value:
      tid: 381
  title:
    plugin: callback
    callable: html_entity_decode
    source: abbreviatedName
  field_location_active_centerid: ID
  field_location_address/address_line1: address1
  field_location_address/locality: city
  field_location_address/administrative_area: state
  field_location_address/postal_code: zipCode
  field_location_address/country_code: country
  field_location_parking_direction/format:
    plugin: default_value
    default_value: basic_html
  field_location_parking_direction/value: parkingCapacity
  field_description/format:
    plugin: default_value
    default_value: basic_html
  field_description/value: description
  field_chicago_community:
    plugin: migration
    migration: activenet_geographic_areas
    source: geographicAreaID
  moderation_state:
    plugin: default_value
    default_value: published
  destination:
    plugin: 'entity:node'
  migration_dependencies:
    required:
      - activenet geographic areas
```

Importing Data

Formatting

- **Address Field** stores basic location data
- **Geocoder** calculates coordinates based on address via service
- **Geolocation** stores coordinates and allows map selection
- Custom module helper traits calculate values for broad use
- Polyline text field stores boundary data
- **Challenges:** Large data requires paid geocoding API (Google)

Importing Data Formatting

▼ ADDRESS

If a specific address is available, enter it here. If no proper address is available change the country to 'none' and use a dropped pin on the map below.

Country

United States ▼

Street address *

541 N Fairbanks

City *

Chicago

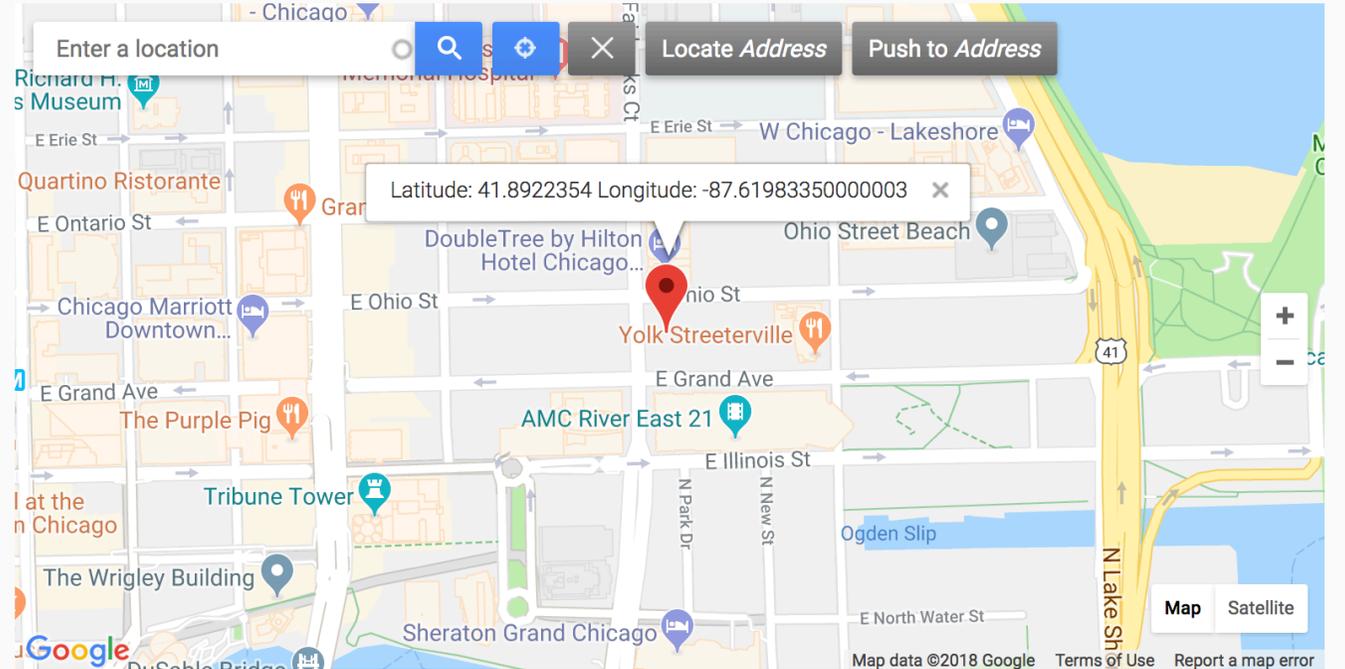
State *

Illinois ▼

Zip code *

60611

LOCATION COORDS



bundle	deleted	entity_id	revision_id	langcode	delta	field_location_coords_lat	field_location_coords_lng
location	0	1111	1111	en	0	41.8922354	-87.61983350000003

Importing Data

Daily Imports

- **Custom Submodule** for cron job and settings storage
- Uses Ultimate Cron module for separate cron job config
- Cron hook in .module file with update functions and error output
- Submodule for each import, separate cron jobs due to timing
- Allows custom error messages for troubleshooting
- Store the migrate YML file in the config directory as a master

Submodule / Cron Hook

```
function activenet_geographic_areas_cron() {  
  
  $migration_id = 'activenet_geographic_areas';  
  $migration_name = 'Activenet Geographic Areas Import';  
  
  /** @var |Drupal|Migrate|Plugin|Migration.$migration.*/  
  $migration = \Drupal::service('plugin.manager.migration')  
    ->createInstance($migration_id);  
  $migration->getIdMap()->prepareUpdate();  
  $executable = new MigrateExecutable($migration, new MigrateMessage());  
  
  $executable->import();  
  
  $map = $migration->getIdMap();  
  $imported = $map->importedCount();  
  $source_plugin = $migration->getSourcePlugin();  
  $source_rows = $source_plugin->count();  
  // -1 indicates uncountable sources.  
  if ($source_rows == -1) {  
    $source_rows = dt('N/A');  
    $unprocessed = dt('N/A');  
  }  
  else {  
    $unprocessed = $source_rows - $map->processedCount();  
  }  
  
  $migrate_last_imported_store = \Drupal::keyValue('migrate_last_imported');  
  $last_imported = $migrate_last_imported_store->get($migration->id(), default: FALSE);  
  if ($last_imported) {  
    /** @var |Drupal|Core|Datetime|DateFormatter.$date_formatter.*/  
    $date_formatter = \Drupal::service('date.formatter');  
    $last_imported = $date_formatter->format(timestamp: $last_imported / 1000,  
      type: 'custom', format: 'Y-m-d.H:i:s');  
  }  
  else {
```

```
▼ activenet_import  
  ► config  
  ▼ modules  
    ► activenet_activity_categories  
    ► activenet_activity_type  
    ► activenet_event_age_categories  
    ► activenet_events  
    ► activenet_geographic_areas  
    ▼ activenet_parks  
      ▼ config  
        ▼ install  
          migrate_plus.migration.activenet_parks.yml  
        activenet_parks.info.yml  
        activenet_parks.module
```

Page Display

1

Lists

- Display locations in filtered lists using View Modes

2

Maps

- Show custom markers and tooltips for locations

3

Detail Pages

- Show pre-filtered related locations on the page

Page Display

1

Lists

- **View Modes** - Able to create sidebar lists for map pages
- **Formatting** - Custom field formatter for distance from user
- **Related Content** - Pull in other location info for display

Page Display Lists

3885 Parks & Facilities [Filter by Facility Type](#)

Sort By:



Bessemer Nature Garden

8930 S Muskegon Ave
Chicago, IL 60617

📍 14 miles away from you



Bessemer Park - Bena Marie Jackson

8930 S. Muskegon
Chicago, IL 60617

(773) 656-3659

📍 14.1 miles away from you



Bessemer Paths and Trails

8930 S. Muskegon Ave.
Chicago, IL 60617

📍 14.1 miles away from you



Bessemer Playground

8930 S. Muskegon Ave.
Chicago, IL 60617

📍 14.1 miles away from you



Bessemer Park

8930 S. Muskegon Ave.
Chicago, IL 60617

(312) 747-6023

📍 14.1 miles away from you



Bessemer Park Pool

8930 S. Muskegon Ave.
Chicago, IL 60617

(312) 747-6023

📍 14.1 miles away from you



Bessemer Picnic Area

8930 S. Muskegon Ave.
Chicago, IL 60617

📍 14.1 miles away from you



Bessemer Running Track

8930 S. Muskegon Ave.
Chicago, IL 60617

📍 14.1 miles away from you

Page Display

2

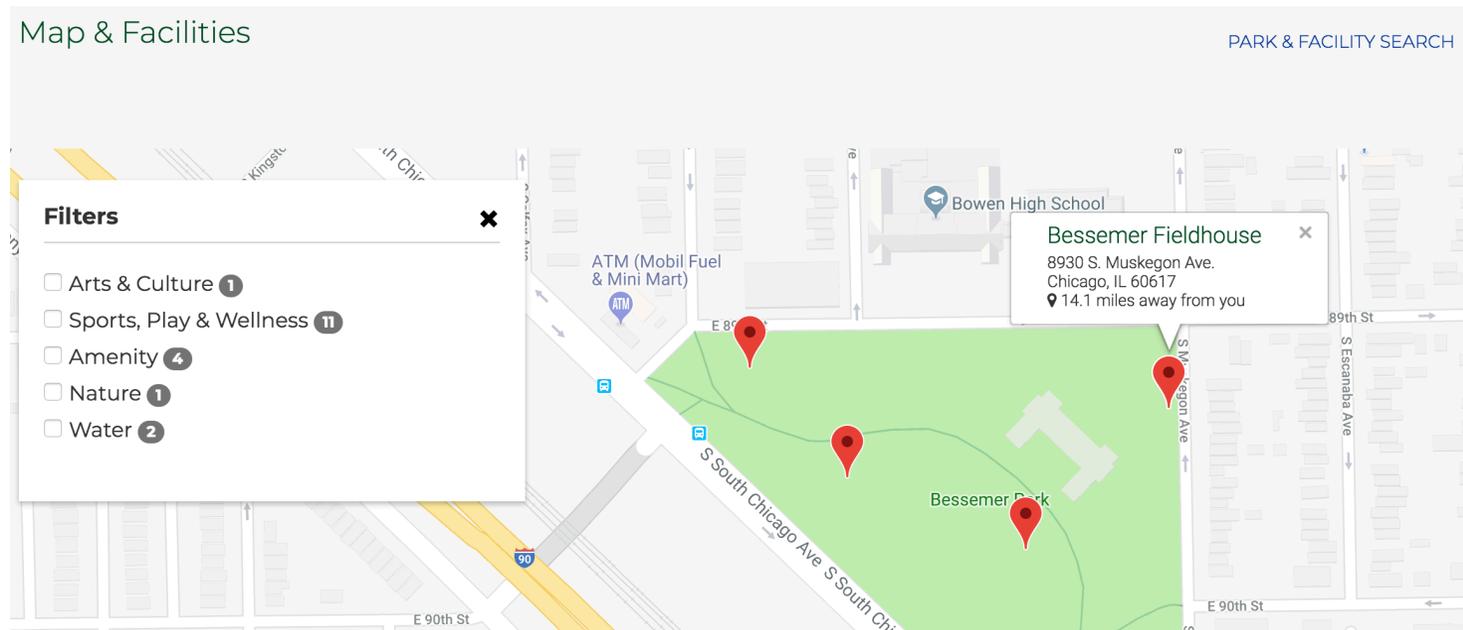
Maps

- **View Mode** - Show formatted info in tooltip on click or hover
- **Marker Image** - Show custom marker using taxonomy value
- **Organization** - Move specific location types to top of stack

Page Display Maps

Map & Facilities

PARK & FACILITY SEARCH



Page Display

3

Detail Page

- **Related Locations** - Show all child locations using field value
- **AJAX Search** - Narrow child locations by Location Type
- **Custom Block** - Place map display in any position on any page

Page Display Detail Page

Parks & Facilities

Bessemer Park

[Hours](#) [Description](#) [History](#) [Parking](#) [Map & Facilities](#) [News](#) [Programs](#) [Related Locations](#)



A great Chicago Plays! Ribbon Cutting celebration

Location:

8930 S. Muskegon Ave.
Chicago, IL 60617

Supervisor: Alonzo Dunlap

Phone Numbers:

Main (312) 747-6023



Feeds

RESTful API Output

- Very easy out of the box with Views in Drupal 8
- If output needs to be very specific, custom resource needed
- Custom REST resource functions create exact matches
- Easy to create additional feeds with different output
- Polyline field used for park boundaries to facilitate check-ins

Feeds

RESTful API

```
<:php
namespace Drupal\cpd_feeds\Plugin\rest\resource;

use ...

/**
 * Provides a resource to get view modes by entity and bundle.
 *
 * @RestResource(
 *   id = "cpd_rest_resource",
 *   label = @Translation("CPD Rest Resource"),
 *   uri_paths = {
 *     "canonical" = "/api/v1/{type}"
 *   }
 * )
 */
class CPDRestResource extends ResourceBase {

    /**
     * A current user instance.
     *
     * @var \Drupal\Core\Session\AccountProxyInterface
     */
    protected $currentUser;

    /**
     * Drupal\Core\Database\Driver\mysql\Connection definition.
     *
     * @var \Drupal\Core\Database\Driver\mysql\Connection
     */
    protected $database;

    /**
     * The entity type manager interface.
     *
     * @var \Drupal\Core\Entity\EntityTypeManagerInterface
     */
    protected $entityTypeManager;
```

Feeds

RESTful API

```
private function getPark() {
    $query = $this->database->select('node_field_data', 'n');
    $query->fields('n', ['nid']);
    $query->addTag('node_access');
    $query->condition('n.status', NODE_PUBLISHED);
    $query->condition('n.type', 'location');
    $query->leftJoin('node__field_parent_location', 'parent', 'parent.entity_id = n.nid');
    $query->isNull('parent.entity_id');
    $results = $query->execute()->fetchCol();
    $nodes = Node::loadMultiple($results);
    $data = [];
    // Process node data and structure return array for encoding.
    /** @var Node $node */
    foreach ($nodes as $nid => $node) {
        $node_data = [];
        $node_data['ParkId'] = $node->id();
        $node_data['ParkName'] = $node->getTitle();
        $create_time = new \Datetime();
        $create_time->setTimestamp($node->getCreatedTime());
        $node_data['CreateDate'] = $create_time->format('Y-m-d\TH:i:s');
        $modify_date = new \Datetime();
        $modify_date->setTimestamp($node->getRevisionCreationTime());
        $node_data['ModifyDate'] = $modify_date->format('Y-m-d\TH:i:s');
        $address_field = $node->get('field_location_address')->get(0)->getValue();
        $address_string = $address_field ? $address_field['address_line1'] . ' ' . $address_field['locality'] . ', ' . $address_field['administrative_area'] . ' ' . $address
        $node_data['Address'] = $address_string;
        $node_data['Address1'] = $address_field['address_line1'];
        $node_data['Address2'] = $address_field['address_line2'];
        $node_data['City'] = $address_field['locality'];
        $node_data['State'] = $address_field['administrative_area'];
        $node_data['Zip'] = $address_field['postal_code'];
        $node_data['Latitude'] = $node->field_location_coords->lat;
        $node_data['Longitude'] = $node->field_location_coords->lng;
        $node_data['ParkRegion'] = $node->field_location_region->entity ? $node->field_location_region->entity->getName() : null;
        $node_data['Polylines'] = $node->field_polygon_data->value;
        $node_data['PhoneNumber'] = $node->field_phone_numbers->entity ? $node->field_phone_numbers->entity->field_phone_number->value : null;
        $hours = $node->field_hours->view();
        if (isset($hours[0]['#office_hours'])) {
            $office_hours = $hours[0]['#office_hours'];
            $node_data['TimeJson'] = ['ParkHours' => [
                'Sunday' => [
                    'Open' => $this->human_readable_hours($office_hours[0]['slots'][0]['start']),
                    'Close' => $this->human_readable_hours($office_hours[0]['slots'][0]['end']),
                ],
                'Monday' => [...],
                'Tuesday' => [...],
                'Wednesday' => [...],
                'Thursday' => [...],
                'Friday' => [...],
                'Saturday' => [...],
            ]];
        }
        $data[$nid] = $node_data;
    }
    return new Response(json_encode($data));
}
```

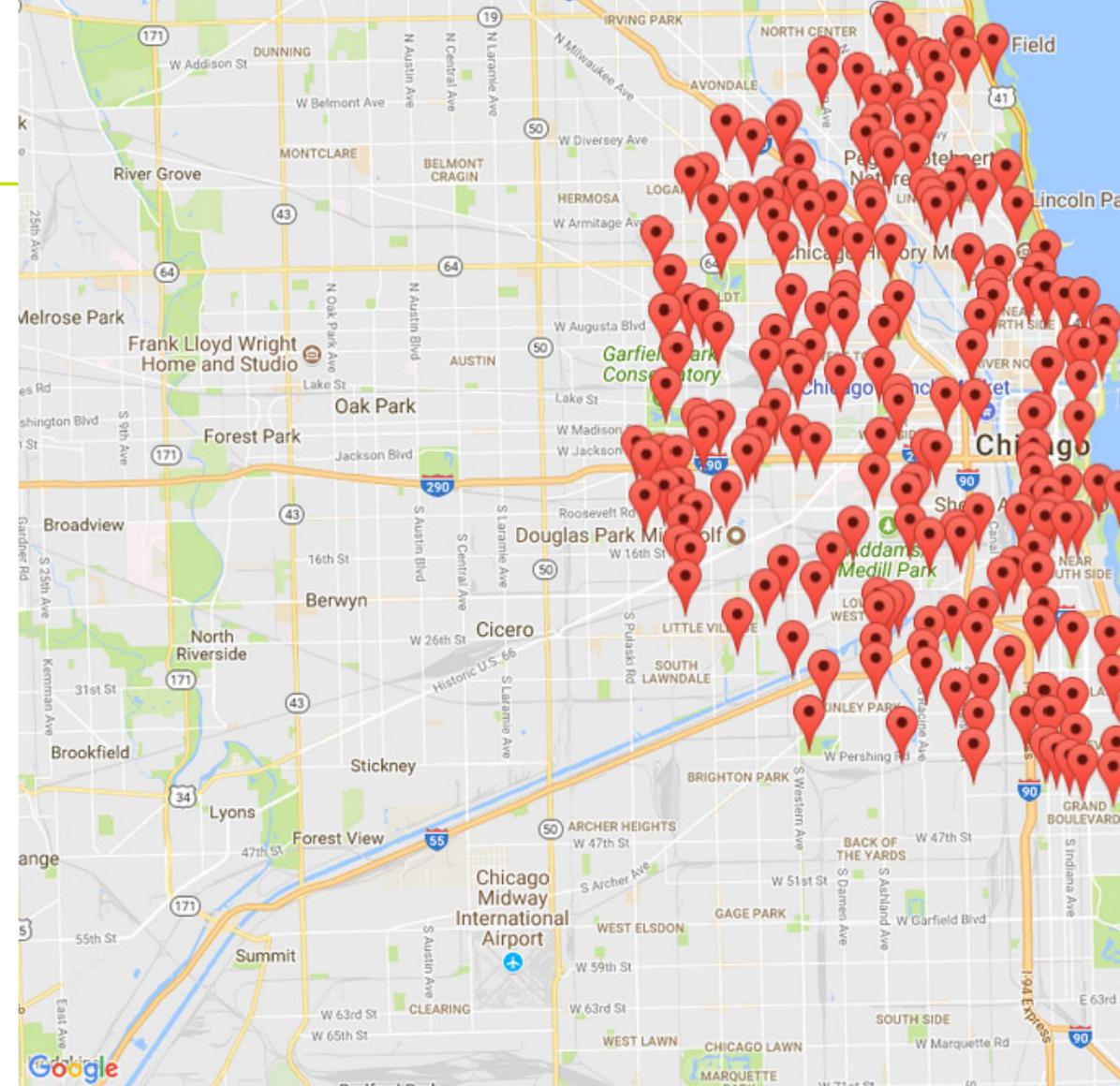
Problems Encountered

MIGRATION

- ◆ Empty checks on values
- ◆ Formatting checks on values
- ◆ Server/ API timeouts with large volume requests

GEOCODING

- ◆ Needed paid Google API key for geocoding volume
- ◆ Geocoding does not work over HTTP, HTTPS only
- ◆ Inconsistencies with browser location security approval





Q

A

Thank you!

Putting Location Data on the Map in Drupal 8

Presented by:

Matthew Messmer

Director, International Business Division, Genero

matthew@genero.jp

<http://genero.jp/en/>

Robert Shell

Senior Consultant, Clarity Partners

r.shell@claritypartners.com

www.claritypartners.com