

# Assignment: Module #8 - Geolocated Images and Maps

## Objective

By the end of this assignment, you will be able to:

- Implement a map view of elements related to selected item
- Build on the skills from modules 5-7
  - Design and implement query access to a resource collection
  - Design and implement API access to a resource that contains information, geographic, and image properties
  - Design and implement a synchronized page of display components
  - Implement a list view of selectable of items

This assignment has a few options.

- All assignment options are intended to extend the course example developed in the lectures. It is not intended that you start from scratch. By extending the course example covered in the lectures, you will not only have a decent starting point – you will also have a base that is familiar to the peer grader to be able to more easily follow the details of your solution. You may augment or modify the course examples as needed.
- All assignment options have a minimal set of requirements that could easily be far short of what a full capability would contain. You are to limit your implementations to only what is required to provide query/index of a resource collection, selection of a specific resource, and display related element information on a map. This is similar to the course example, but one key difference is that only only a single item (group of elements) is displayed at a time on the map.
- All assignment options are notional. You may use your imagination to fill in the details of the technical solution that will meet the overall goals of the assignment.
- All mandatory capabilities of this assignment are read-only with no access restrictions required. All data made available to the user will be ingested by you at deployment time by a `db/seeds.rb` or rakefile script. No user input should be required.

## Requirements (5hr 45min)

1. (0min – assumes you have been following along with lectures) You are required to start with the `map-assignment` tag from the course example and make you changes from there. If you are not using Asset Pipeline, then you may copy the course examples from the Asset Pipeline area into your development area of choice and work from that point (from a Git perspective, it is just a movement of files).

The following steps are notional of how you get started.

```
```shell
$ git clone https://github.com/jhu-ep-coursera/capstone_demoapp.git map-assignment
$ cd map-assignment
$ git checkout map-assignment
You are in 'detached HEAD' state ...
$ git checkout -b getting-started
Switched to a new branch 'getting-started'
```

```shell
$ git remote add staging https://git.heroku.com/abc-123.git #your staging repository
$ git push --force staging getting-started:master
$ heroku run rake db:migrate --remote staging
$ heroku run rake ptourist:reset_all --remote staging
$ heroku restart --remote staging
```
```

2. (15min) Pick an assignment option. You do not have to implement the entire capability, but what you implement should be within the scope of the option.

3. (30min) Your solution must provide data for the user to query and view. The data can be generated (e.g., using the Faker gem) and can leverage the ptourist content as well. It is assumed this will be automated in the form of a `seeds.rb` or `rakefile` script.
4. (60min) Design and implement API index and show access to database-backed resource(s) on a Rails server. The backend can be either ActiveRecord, Mongoid, or both. This resource must have related information, geographic coordinates, and images.
5. (30min) Design and implement API query access to the resource(s). The query must be able to select all or a sub-set of items and can be one that you choose.
6. (90min) Implement a new page that contains at least a list component, a map, and a means to query for items.
7. (60min) When the user selects an item from the list, the map should display a marker for each element associated with that item.
8. (60min) When the user selects a marker on the map, the marker should display something about that specific item element.

## Options

You are to chose *one* of the following options to implement in this assignment. You do not need to implement the entire capability to receive full credit. Just query, list, select items and show related elements on a map is the key task. Anything else should be added outside of the scope of the assignment.

Use your imagination. There is nothing firm that any of these options require except that the solution must include the key technical demonstration aspects mentioned.

There is no requirement to provide functional access to the legacy course example pages or to remove them. The focus will be on the new page implemented.

### Option: Types of Things

Implement a means to identify and search for a type of Thing. You may design the types to be static (e.g., a museum is always a museum), or use dynamic tags (e.g., Things having easy wheelchair access)

Example Solution Choices:

- Query
  - Identify desired Thing by types (e.g., museum)
- List
  - Show Things that match provided query
- Select Item
  - Place a marker on the map to represent related Images to the selected Thing
- Select Marker
  - Display information about the Image selected

Note: This is similar to but different from the course example in that you must implement a new page, a query for type, and only display the Images for one Thing at a time on the map.

### Option: Trip Logs

Implement a display of trips with stops.

Example Solution Choices:

- Query
  - Identify an origin and distance criteria to locate trips with stops located within that criteria
- List
  - Show trips that have a stop within a geographic bounds
- Select Item

- Place a marker on the map to represent each stop on the trip.
- Select Marker
  - Display information and Image(s) related to the stop selected.

Note: You could use Thing (with at least a primary Image) to represent a stop and a Trip to be a collection of Things.

#### Rubric

1. Identify which assignment option you implemented and describe what the peer grader will be able to do with the enhancement relative to the rubric requirements. Feel free to boast but focus on what the peer grader needs to know first.
2. Supply the base, public URL of your web client that hosts your solution for this assignment.
  - (10) Were you able to access a new page that implemented the display of information, images, and maps of the items and elements of the selected option?
  - (10) Was there a list of items you could select from?
  - (10) Could you supply a query inputs that resulted in a subset of items shown in the list?
  - (20) When you selected one of the items in the list, did only the related elements of that item get displayed on the map using markers?
  - (20) When you clicked on a new item in the list, did the map clear of the old markers and display new markers representing the elements of the new selection?
  - (20) When you clicked on one of the markers, did it display information and an image related to that element?
3. Supply the URL of a branch or tag in your public Git repository that represents the version of source code deployed for the Rails portion of your solution. This may be the same URL as the SPA version if you are using a single source tree.
  - (5) Was the URL to the Rails source repository branch deployed to the Internet provided for you to review the solution?
4. Supply the URL of a branch or tag in your public Git repository that represents the version of source code deployed for the SPA portion of your solution. This may be the same URL as the Rails version if you are using a single source tree.
  - (5) Was the URL to the SPA source repository branch deployed to the Internet provided for you to review the solution?

**Last Updated: 2017-04-13**