

Drupal Module Requirements

Analysis Map v1.08.21

Overview

The purpose of this document is to explicitly define the requirements for Version 1.08.22 “Analysis Map”, a drupal module.

The purpose of the drupal module is to allow an administrator user to save their changes to a map so that a second, non-admin user can view the map and post notes. The map currently has all functionality needed to edit the map on from within the browser. This is done with use of Google Maps JavaScript API, CSS and jQuery.

To help aid in the development process, the requirements are defined as three different types within this document. The requirements are explicitly defined but there is room for interpretation in how to best implement a specific requirement. For example, if a requirement states that a data type must be stored within its own database table, it is expected that proper usage of Drupal hooks within the module will allow for the creation of the database on installation of the module and data that is saved will be done with a secondary script within the module.

The requirements will be defined as three different types.

1. User Interface
2. Drupal Module
3. Development Testing

The next pages explicitly define the requirements.

Requirements: User Interface

The map seen below is the browser-side, Javascript/HTML version of the module. The final version of the browser side functionality is located in the directory: `simplemap/v1.08/v0.21/v0.21.php`.

The screenshot displays the 'Simple GeoJSON Editor' web application. On the left, a blue sidebar contains an 'Upload Image' section with buttons for 'Choose File' and 'Upload File', and a 'Simple GeoJSON Editor' section with 'Select All', 'Download', and 'Save GeoJSON' buttons. Below these is a text area showing a JSON structure for a 'FeatureCollection' with a 'LineString' geometry. The main area is a map of a coastal region with several red location pins and black line shapes. A 'Delete Selected Shape' button is at the top center. On the right, a control panel includes input fields for 'Latitude: 40.7116', 'Longitude: -74.0132', and 'Zoom Level: 18', along with a 'Center Map' button and checkboxes for 'Image Overlay', 'Property Boundary', and 'Drawing Layer'. The bottom right corner shows a Google logo and map data information.

Upload Image

Select image to upload:

Choose File No file chosen

Upload File

Simple GeoJSON Editor

Select All Download

Save GeoJSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            -150.2489697933197,
            62.33590397485731
          ],
          [
            -150.3574597835408,
            62.33590397485731
          ],
          [
            -150.35608649253845,
            62.29921951545998
          ],
          [
            -150.2544629573822,
            62.33590397485731
          ]
        ]
      }
    }
  ]
}
```

Map Satellite

Delete Selected Shape

Latitude: 40.7116

Longitude: -74.0132

Zoom Level: 18

Center Map

☐ Image Overlay

☒ Property Boundary

☒ Drawing Layer

Talkeetna

Trapper Creek

Map data ©2016 Google Terms of Use Report a map error

Requirements: User Interface: Two Maps

Map I: Administrative User

Has full functionality to edit, update and save map.

Saving map populates database tables.

Map II: Non-admin user:

Limited editing abilities. Views map that admin has saved

Editing Abilities:

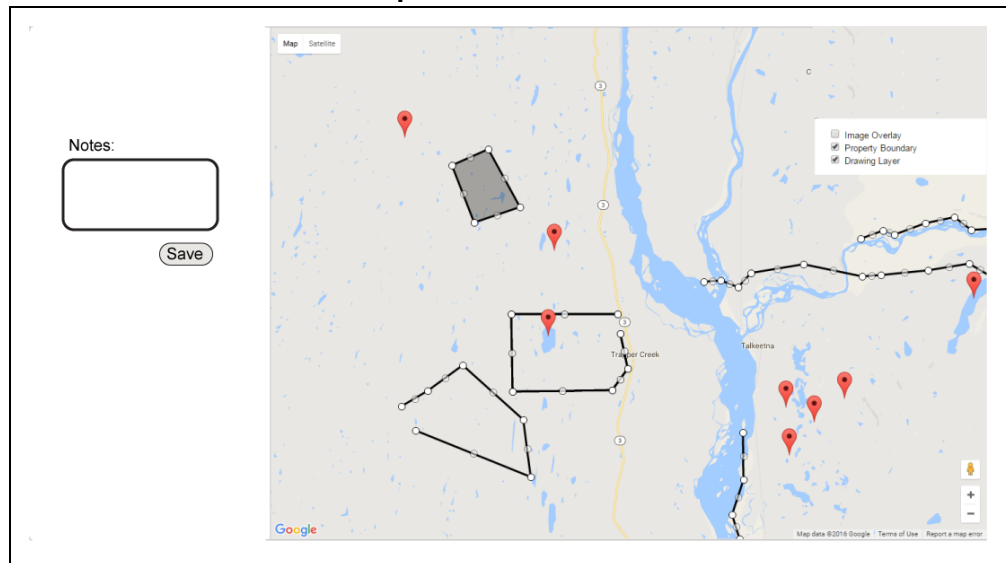
Can view Map, Terrain or Satellite (top left corner)

Can check radio boxes to view overlays (Image Overlay, Property Boundary or Drawing Layer)

Can hit plus or minus to zoom in and zoom out (Bottom right corner)

Can click-drag map to scroll map view (north, east, south, west)

Non Admin User view of map:



Requirements: Drupal Module: Database Tables

When user clicks the “Save Map” button, database tables are populated:

- A. Auto-Generated: Table ID (Primary Key) (Required)
- B. Auto-Generated: Date/Time Stamp formatted yyyy-mm-dd-hh-mm (Required)
- C. Auto-Generated: Map ID (Required)
- D. Data from browser (refer to map on next page):
 - Number 1:** if images is uploaded and changes are made to uploaded image (opacity, rotate, height and width)
 - Number 2:** link to uploaded image
 - Number 3:** as user draws on map using drawing tools, data is passed into from on left as GeoJSON data. Save GeoJSON data
 - Number 4:** latitude, longitude and zoom level and map center must be saved
 - Number 5:** This data is drawn from previously created GeoJSON linestrings. Sample data will be provided.

Map View: Data Sources to Save to Database

The screenshot displays a web-based map application interface with several key components highlighted by numbered red boxes:

- 1** **Properties Panel:** A panel on the left containing four blue buttons labeled **OPACITY**, **ROTATE**, **HEIGHT**, and **WIDTH**.
- 2** **Upload Image:** A section below the properties panel with the text "Select image to upload:", a "Choose File" button, and an "Upload File" button.
- 3** **Map Controls:** A toolbar at the top left of the map area containing icons for Map, Satellite, and other navigation functions.
- 3** **Map Area:** The central map area showing a satellite view with red overlays, including a polygon and several red location pins. A "Delete Selected Shape" button is located at the top right of the map area.
- 4** **Metadata Panel:** A panel on the right containing input fields for **Latitude** (40.7116), **Longitude** (-74.0132), and **Zoom Level** (18), along with a "Center Map" button.
- 5** **Property Boundary GeoJSON Linestring:** A red arrow pointing from the map area to the "Property Boundary" checkbox in the layer control panel.
- 6** **Save Map Button:** A button labeled "Save Map" located below the "Upload Image" section.
- 3** **Simple GeoJSON Editor:** A panel at the bottom left with a text area containing a JSON snippet, and buttons for "Select All", "Download", and "Save GeoJSON".

The JSON snippet in the Simple GeoJSON Editor is as follows:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [
            -150.2489697933197,
            62.33590397485731
          ],
          [
            -150.35745978355408,
            62.33590397485731
          ],
          [
            -150.35608649253845,
            62.29921951545998
          ],
          [
            -150.2544629573822,
            62.33590397485731
          ]
        ]
      }
    }
  ]
}
```

Requirements: Development Testing

1. Completed module should meet all previously defined requirements without error.
2. PHP.ini errors will be turned on to look for errors.
3. Module should be compatible with Drupal Version 7.x
4. Module should be “stand alone” and easily installed on any Drupal Version 7.x system.
5. Module should be compatible with installation database tables in MySQL or PostgreSQL
6. Drupal Coding standards, as defined here: <https://www.drupal.org/docs/develop/standards> must be adhered to.