The Google Maps API and OpenLayers Javascript Framework: a NMGIC Workshop

Karl Benedict Director, Earth Data Analysis Center
Associate Professor, College of University Libraries & Learning Sciences
University of New Mexico kbene@unm.edu

May 1, 2014

Contents

Google Maps API

Introduction

Outline

- What is an API
- The Google Maps API
  - Version
  - Reference Information
  - Key Components
  - Examples

What is an API

- API Stands for Application Programming Interface

  An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers. – From Wikipedia: http://en.wikipedia.org/wiki/Api

- The Google Maps API provides an interface for interacting with Google's mapping services from external web applications

Google Maps API Version

- The version of the Google Maps API used in this session is v3 of the Javascript API
  - Freely usable for free applications
  - Subject to Google's Terms of Service
  - No longer requires a Google API key, but one is recommended for tracking usage
- Key capabilities in v3
  - Interactive maps based on Google's mapping engine (contrast w. static maps API)
  - Optimized for desktop and mobile platforms and applications

Reference Information

Google Maps API Family http://code.google.com/apis/maps/

Javascript API Home Page http://code.google.com/apis/maps/documentation/javascript/

Javascript Basics Entry Page http://code.google.com/apis/maps/documentation/javascript/basics.html

Javascript API v3 Tutorial Page http://code.google.com/apis/maps/documentation/javascript/tutorial.html

Key Components

Map object options

Types (required) ROADMAP
SATELLITE
HYBRID
TERRAIN

Latitude and Longitude (required) specification of where the map should initially be centered

Zoom Level (required) 0=global, higher values increasingly local. Limited by map type

Controls

- Available Controls (enabled through map options) default controls
  - Zoom Control
  - Pan Control
  - Scale Control
  - MapType Control
  - Street View Control
- Different control styles may be defined
- Controls may be positioned positioning options
- Custom controls may be defined and attached to fixed location in the map

Overlays

Overlay Types documentation

Marker points depicted by specified or defined icons at locations within the map

Polyline linear features defined by multiple points with a defined style for the line

Polygon closed features defined by multiple points. Supports multi-polygons, and donuts. Line and fill styles may be specified.

(Ground) Overlay Maps Image-based map layers that replace or overlay Google layers - registered to the map coordinates

Info Windows floating content windows for displaying content defined as HTML, a DOM element, or text string

Layers Grouped display content assigned to a specific layer: KmlLayer, FusionTablesLayer, TrafficLayer, BicyclingLayer

Custom Overlays definition of programmatically controlled layers

Services

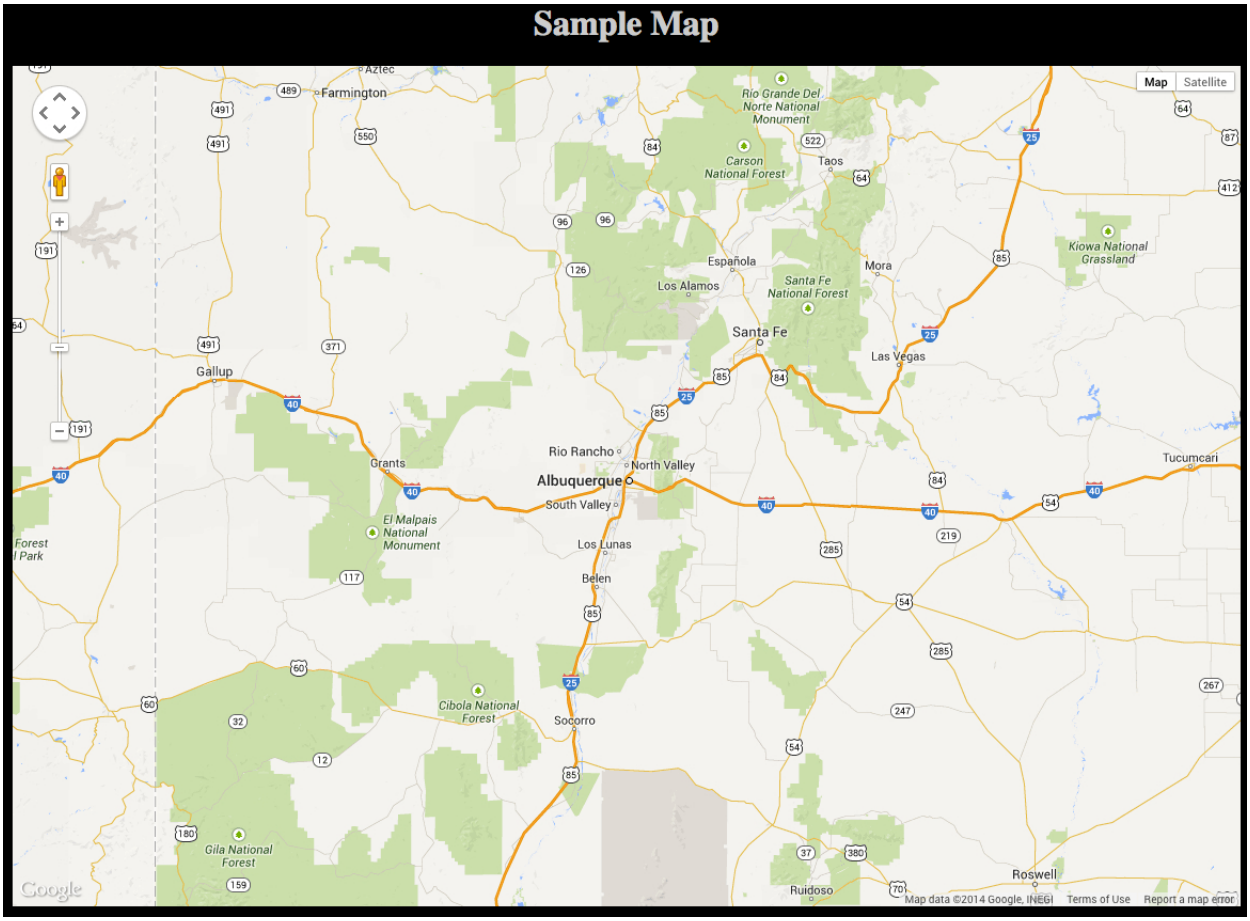
- Geocoding Service
  - Forward and reverse geocoding:
    - address to LatLon
    - LatLon to Nearest Address
  - May be biased to current viewport, region
- Directions
  - Based upon an origin, destination, and a variety of additional options
  - Available directions and rendered route
- Elevation
  - Delivery of elevation data for locations or paths
- Streetview
  - Integration of Google Streetview within a DOM element
- Maximum Zoom
  - Provides information about the maximum available zoom level

Events

- Events provide the ability to attach custom behaviors to events in the interface. For example:
  - Changing items in the interface as the user zooms in on a map
  - Displaying additional information outside the map when the user clicks a location in the map
  - Synchronizing the behavior of multiple maps as the user interacts with one map
- Requires higher-level Javascript that we will cover here

Examples

Simple - Roadmap

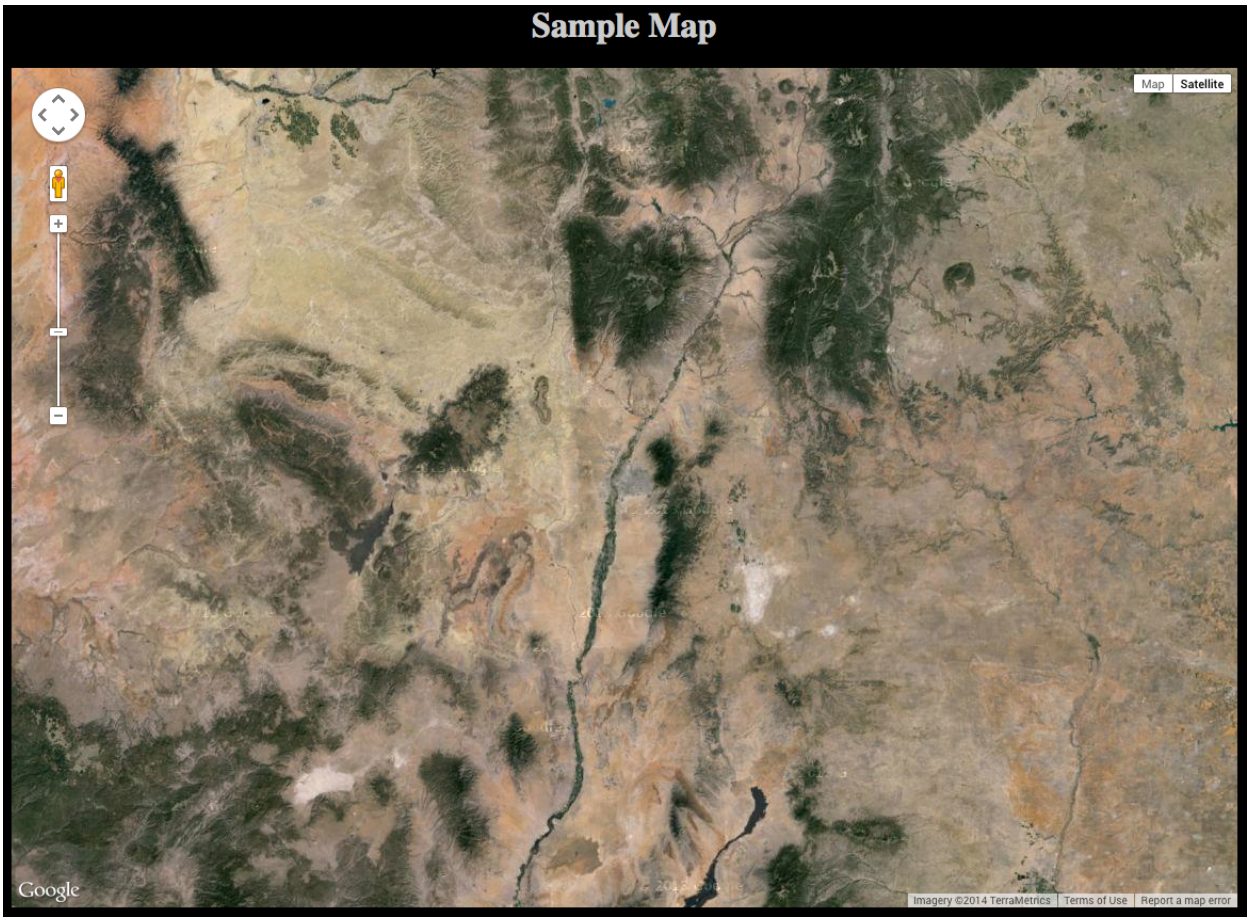


http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps01.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
            html { height: 100% }
            body { height: 100%;
                margin: 0px;
                padding: 0px;
                background-color: black;
                color: #CCCCCC;
                text-align: center}
            #map_canvas { width:90%;
                height:80%;
                margin-left:auto;
                margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
```

```
        </script>
        <script type="text/javascript">
            function initialize() {
            var classroom = new google.maps.LatLng(35.084280,-106.624073)
            var myOptions = {
                zoom: 8,
                center: classroom,
                mapTypeId: google.maps.MapTypeId.ROADMAP
            };
            var map = new google.maps.Map(
                document.getElementById("map_canvas"),
                myOptions);
            }
        </script>
    </head>

    <body onload="initialize()">
        <h1>Sample Map</h1>
        <div id="map_canvas"></div>
    </body>
</html>
```

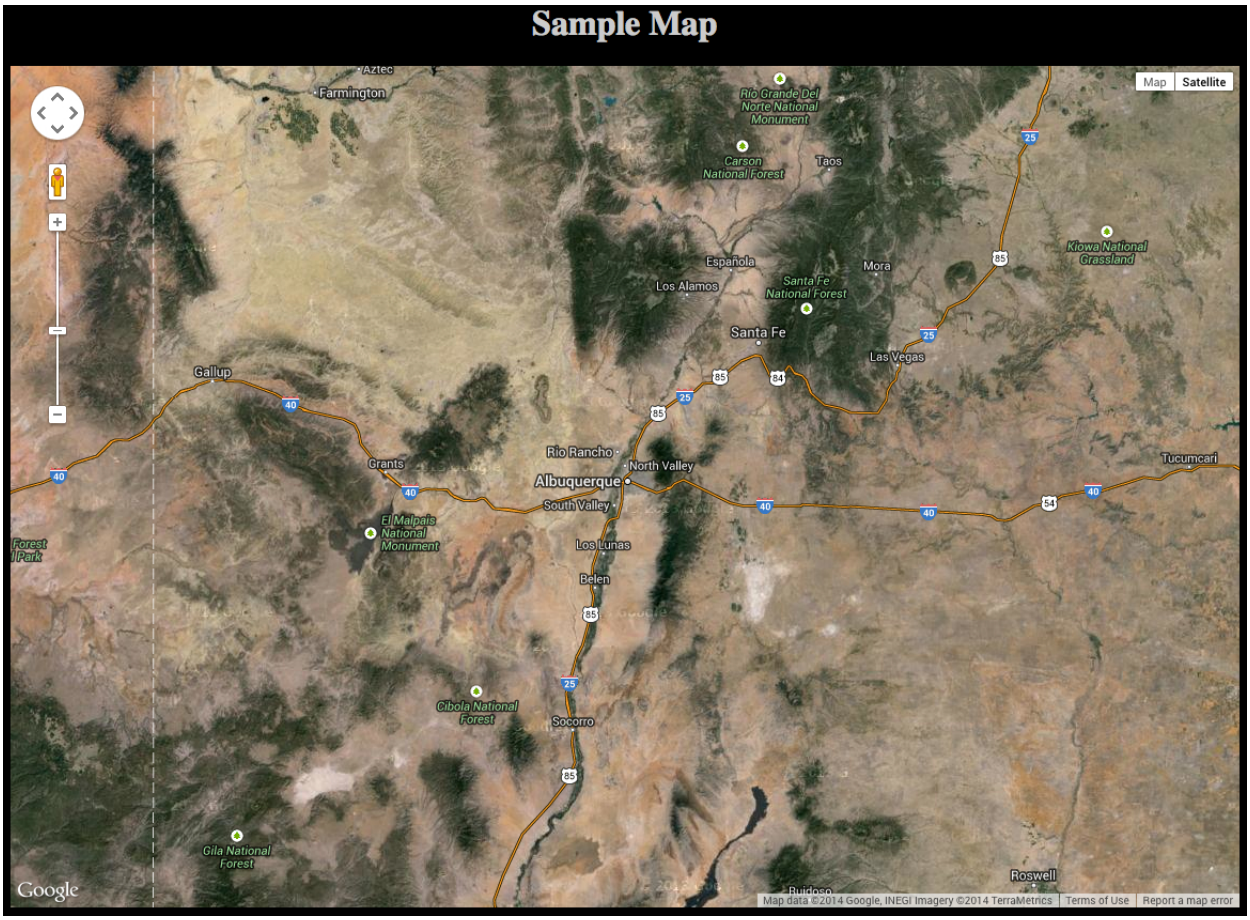
Simple - Satellite



http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps02.html

```
<!DOCTYPE html>

<html>
    <head>
        <style type="text/css">
            html { height: 100% }
            body { height: 100%;
                margin: 0px;
                padding: 0px;
                background-color: black;
                color: #CCCCCC;
                text-align: center}
            #map_canvas { width:90%;
                height:80%;
                margin-left: auto;
                margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
            function initialize() {
                var classroom = new google.maps.LatLng(35.084280,-106.624073)
                var myOptions = {
                    zoom: 8,
                    center: classroom,
                    mapTypeId: google.maps.MapTypeId.SATELLITE
                };
                var map = new google.maps.Map(
                    document.getElementById("map_canvas"),
                    myOptions);
            }
        </script>
    </head>

    <body onload="initialize()">
        <h1>Sample Map</h1>
        <div id="map_canvas"></div>
    </body>
</html>
```

Simple - Hybrid

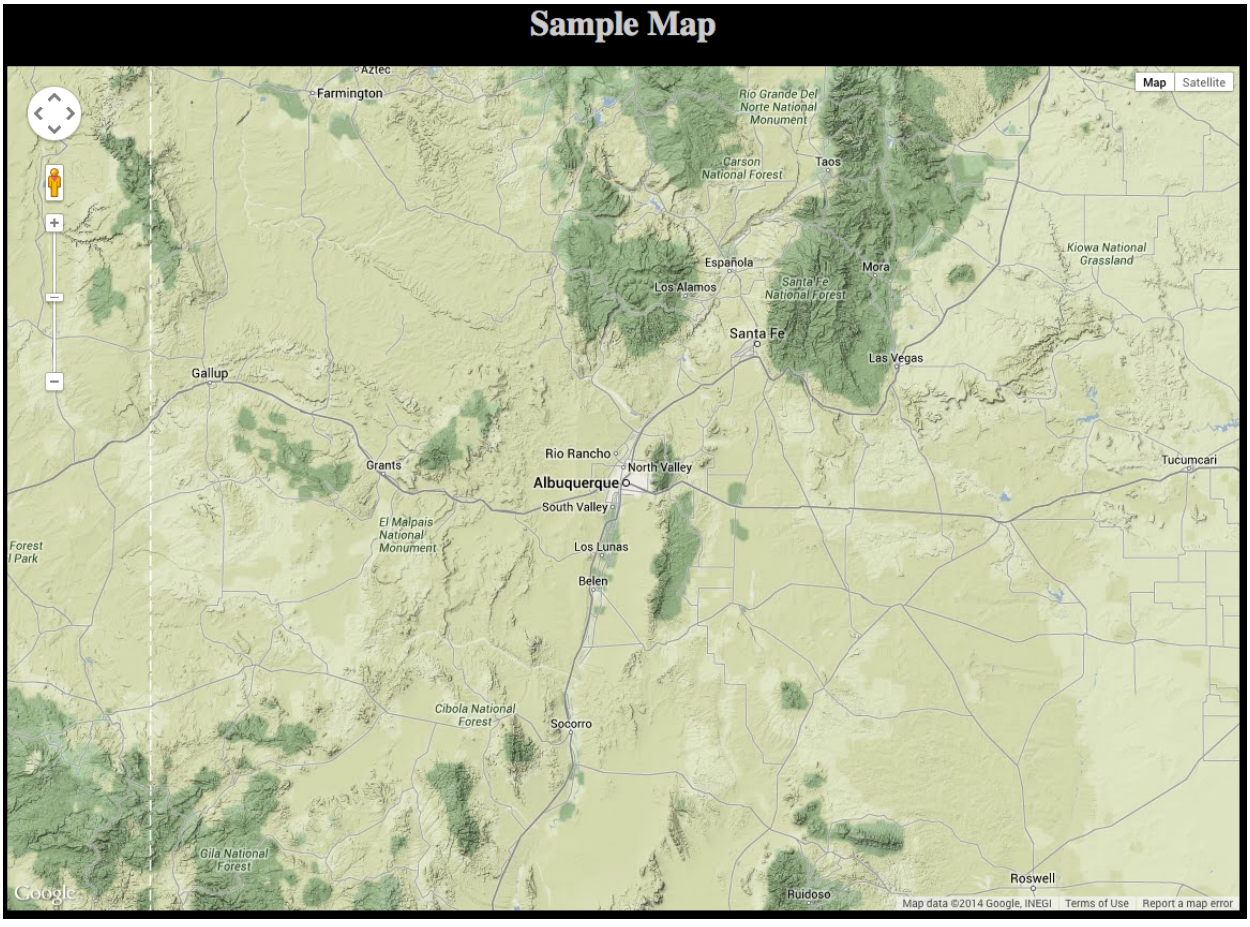


http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps03.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
```

```
21            function initialize() {
22              var classroom = new google.maps.LatLng(35.084280,-106.624073)
23              var myOptions = {
24                zoom: 8,
25                center: classroom,
26                mapTypeId: google.maps.MapTypeId.HYBRID
27              };
28              var map = new google.maps.Map(
29                  document.getElementById("map_canvas"),
30                  myOptions);
31            }
32          </script>
33      </head>
34
35      <body onload="initialize()">
36        <h1>Sample Map</h1>
37        <div id="map_canvas"></div>
38      </body>
39
40  </html>
```

Simple - Terrain



http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps04.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
          function initialize() {
            var classroom = new google.maps.LatLng(35.084280,-106.624073)
            var myOptions = {
              zoom: 8,
              center: classroom,
              mapTypeId: google.maps.MapTypeId.TERRAIN
            };
            var map = new google.maps.Map(
                document.getElementById("map_canvas"),
                myOptions);
          }
        </script>
    </head>

    <body onload="initialize()">
      <h1>Sample Map</h1>
      <div id="map_canvas"></div>
    </body>

</html>
```

Simple - Hybrid - Zoomed

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps05.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
```

```
21            function initialize() {
22              var classroom = new google.maps.LatLng(35.084280,-106.624073)
23              var myOptions = {
24                zoom: 18,
25                center: classroom,
26                mapTypeId: google.maps.MapTypeId.HYBRID
27              };
28              var map = new google.maps.Map(
29                  document.getElementById("map_canvas"),
30                  myOptions);
31            }
32          </script>
33      </head>
34
35      <body onload="initialize()">
36        <h1>Sample Map</h1>
37        <div id="map_canvas"></div>
38      </body>
39
40  </html>
```

Simple - Zoomed - Modified Controls

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps06.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
          function initialize() {
            var classroom = new google.maps.LatLng(35.084280,-106.624073)
            var myOptions = {
              zoom: 18,
              center: classroom,
              mapTypeId: google.maps.MapTypeId.HYBRID,
              zoomControl: true,
              zoomControlOptions: {style: google.maps.ZoomControlStyle.SMALL},
              mapTypeControl: true,
              mapTypeControlOptions: {
                style: google.maps.MapTypeControlStyle.DROPDOWN_MENU},
              streetViewControl: false
            };
            var map = new google.maps.Map(
                document.getElementById("map_canvas"),
                myOptions);
          }
        </script>
    </head>

    <body onload="initialize()">
      <h1>Sample Map</h1>
      <div id="map_canvas"></div>
    </body>

</html>
```

Markers

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps07.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
          function initialize() {
```

```
            var classroom = new google.maps.LatLng(35.084280,-106.624073)
            var office = new google.maps.LatLng(35.084506,-106.624899)
            var myOptions = {
              zoom: 18,
              center: classroom,
              mapTypeId: google.maps.MapTypeId.HYBRID
              };
            var map = new google.maps.Map(
              document.getElementById("map_canvas"),
              myOptions);

            var classroomMarker = new google.maps.Marker({
              position: classroom,
              title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
              });
            classroomMarker.setMap(map);

            var officeMarker = new google.maps.Marker({
              position: office,
              title:"Office, Bandelier West, Room 107"
              });
            officeMarker.setMap(map);
          }
        </script>
    </head>

    <body onload="initialize()">
      <h1>Sample Map</h1>
      <div id="map_canvas"></div>
    </body>

</html>
```

Polyline

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps08.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left:
            auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
```

```
        <script type="text/javascript">
          function initialize() {
            var classroom = new google.maps.LatLng(35.084280,-106.624073)
            var office = new google.maps.LatLng(35.084506,-106.624899)
            var myOptions = {
              zoom: 18,
              center: classroom,
              mapTypeId: google.maps.MapTypeId.HYBRID
              };
            var map = new google.maps.Map(
              document.getElementById("map_canvas"),
              myOptions);

            var classroomMarker = new google.maps.Marker({
              position: classroom,
              title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
              });
            classroomMarker.setMap(map);

            var officeMarker = new google.maps.Marker({
              position: office,
              title:"Office, Bandelier West, Room 107"
              });
            officeMarker.setMap(map);

            var officeVisitCoordinates = [
              office,
              new google.maps.LatLng(35.084445,-106.624327),
              new google.maps.LatLng(35.084309,-106.624308),
              classroom
              ];
            var officePath = new google.maps.Polyline({
              path: officeVisitCoordinates,
              strokeColor: "#FF0000",
              strokeOpacity: 1.0,
              strokeWeight: 2
            });
            officePath.setMap(map)
          }
        </script>
    </head>

    <body onload="initialize()">
      <h1>Sample Map</h1>
      <div id="map_canvas"></div>
    </body>

</html>
```

Polygon

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps09.html

```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
        <script type="text/javascript">
```

```
21            function initialize() {
22              var classroom = new google.maps.LatLng(35.084280,-106.624073)
23              var office = new google.maps.LatLng(35.084506,-106.624899)
24              var myOptions = {
25                zoom: 18,
26                center: classroom,
27                mapTypeId: google.maps.MapTypeId.HYBRID
28                };
29              var map = new google.maps.Map(
30                document.getElementById("map_canvas"),
31                myOptions);
32              var classroomMarker = new google.maps.Marker({
33                position: classroom,
34                title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
35                });
36              classroomMarker.setMap(map);
37              var officeMarker = new google.maps.Marker({
38                position: office,
39                title:"Office, Bandelier West, Room 107"
40                });
41              officeMarker.setMap(map);
42              var buildingCoordinates = [
43                new google.maps.LatLng(35.084498,-106.624921),
44                new google.maps.LatLng(35.084558,-106.624911),
45                new google.maps.LatLng(35.084566,-106.624970),
46                new google.maps.LatLng(35.084609,-106.624966),
47                new google.maps.LatLng(35.084544,-106.624383),
48                new google.maps.LatLng(35.084438,-106.624317),
49                new google.maps.LatLng(35.084384,-106.623922),
50                new google.maps.LatLng(35.084164,-106.623970),
51                new google.maps.LatLng(35.084214,-106.624324),
52                new google.maps.LatLng(35.084214,-106.624324),
53                new google.maps.LatLng(35.084391,-106.624284)
54                ];
55              var bldgPoly = new google.maps.Polygon({
56                paths: buildingCoordinates,
57                strokeColor: "#FF0000",
58                strokeOpacity: 0.8,
59                strokeWeight: 2,
60                fillColor: "#FF0000",
61                fillOpacity: 0.35
62              });
63              bldgPoly.setMap(map)
64            }
65          </script>
66      </head>
67
68      <body onload="initialize()">
69        <h1>Sample Map</h1>
70        <div id="map_canvas"></div>
71      </body>
72
73  </html>
```

Adding an Info Window

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps10.html

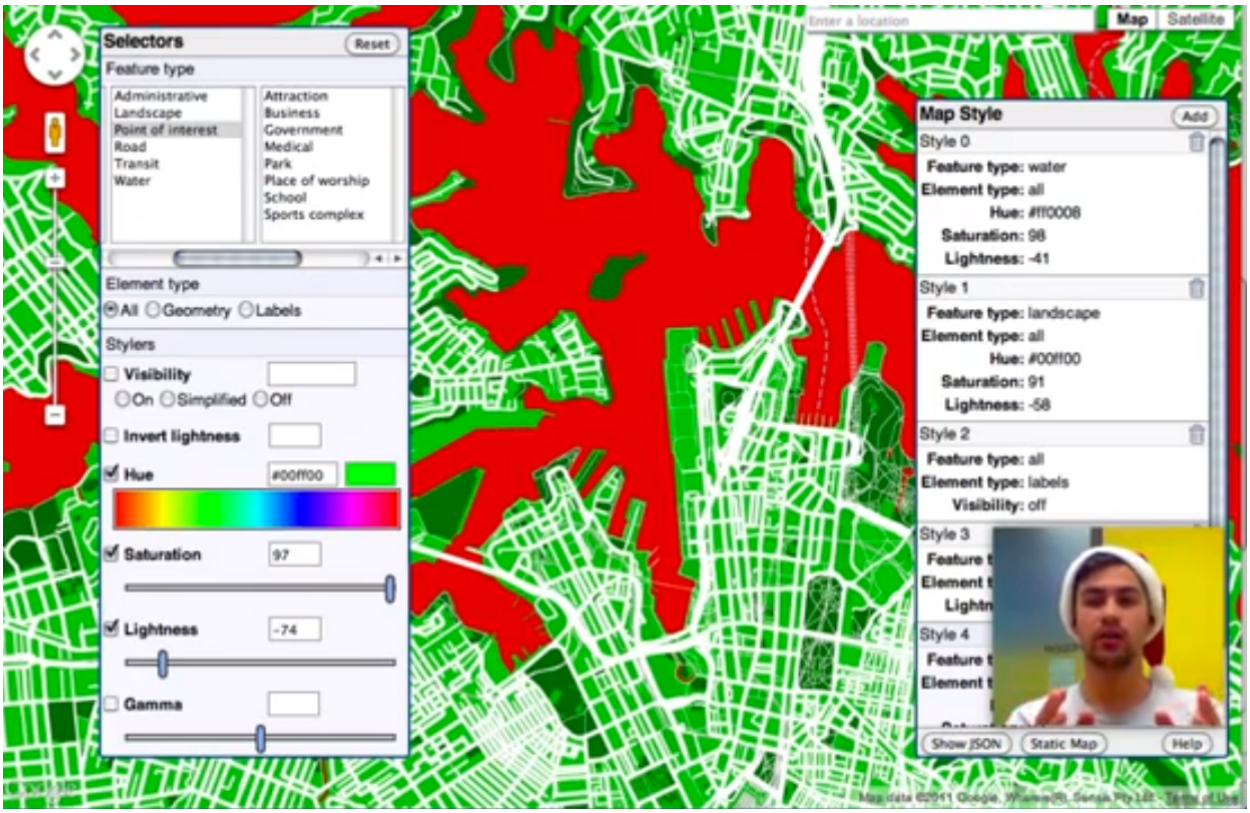
```
<!DOCTYPE html>
<html>
    <head>
        <style type="text/css">
          html { height: 100% }
          body { height: 100%;
            margin: 0px;
            padding: 0px;
            background-color: black;
            color: #CCCCCC;
            text-align: center}
          #map_canvas { width:90%;
            height:80%;
            margin-left: auto;
            margin-right: auto }
          .infoBox { color:black }
        </style>
        <script type="text/javascript"
            src="http://maps.google.com/maps/api/js?sensor=false">
        </script>
```

```
<script type="text/javascript">
  function initialize() {
    var classroom = new google.maps.LatLng(35.084280,-106.624073)
    var office = new google.maps.LatLng(35.084506,-106.624899)
    var myOptions = {
      zoom: 18,
      center: classroom,
      mapTypeId: google.maps.MapTypeId.HYBRID
      };
    var map = new google.maps.Map(
      document.getElementById("map_canvas"),
      myOptions);
    var classroomMarker = new google.maps.Marker({
      position: classroom,
      title:"Geography 485L/585L Classroom, Bandelier East, Room 106"
      });
    classroomMarker.setMap(map);
    var officeMarker = new google.maps.Marker({
      position: office,
      title:"Office, Bandelier West, Room 107"
      });
    officeMarker.setMap(map);
    var buildingCoordinates = [
      new google.maps.LatLng(35.084498,-106.624921),
      new google.maps.LatLng(35.084558,-106.624911),
      new google.maps.LatLng(35.084566,-106.624970),
      new google.maps.LatLng(35.084609,-106.624966),
      new google.maps.LatLng(35.084544,-106.624383),
      new google.maps.LatLng(35.084438,-106.624317),
      new google.maps.LatLng(35.084384,-106.623922),
      new google.maps.LatLng(35.084164,-106.623970),
      new google.maps.LatLng(35.084214,-106.624324),
      new google.maps.LatLng(35.084214,-106.624324),
      new google.maps.LatLng(35.084391,-106.624284)
      ];
    var bldgPoly = new google.maps.Polygon({
      paths: buildingCoordinates,
      strokeColor: "#FF0000",
      strokeOpacity: 0.8,
      strokeWeight: 2,
      fillColor: "#FF0000",
      fillOpacity: 0.35
    });
    bldgPoly.setMap(map);
    var classInfoContent = '<div class="infoBox">' +
      '<p>This is the location for the Geography 485L/585L class</p>' +
      '</div>'
    var classInfoWindow = new google.maps.InfoWindow({
      content: classInfoContent
      });
    google.maps.event.addListener(classroomMarker, 'click', function() {
      classInfoWindow.open(map,classroomMarker);
      });
  }
```

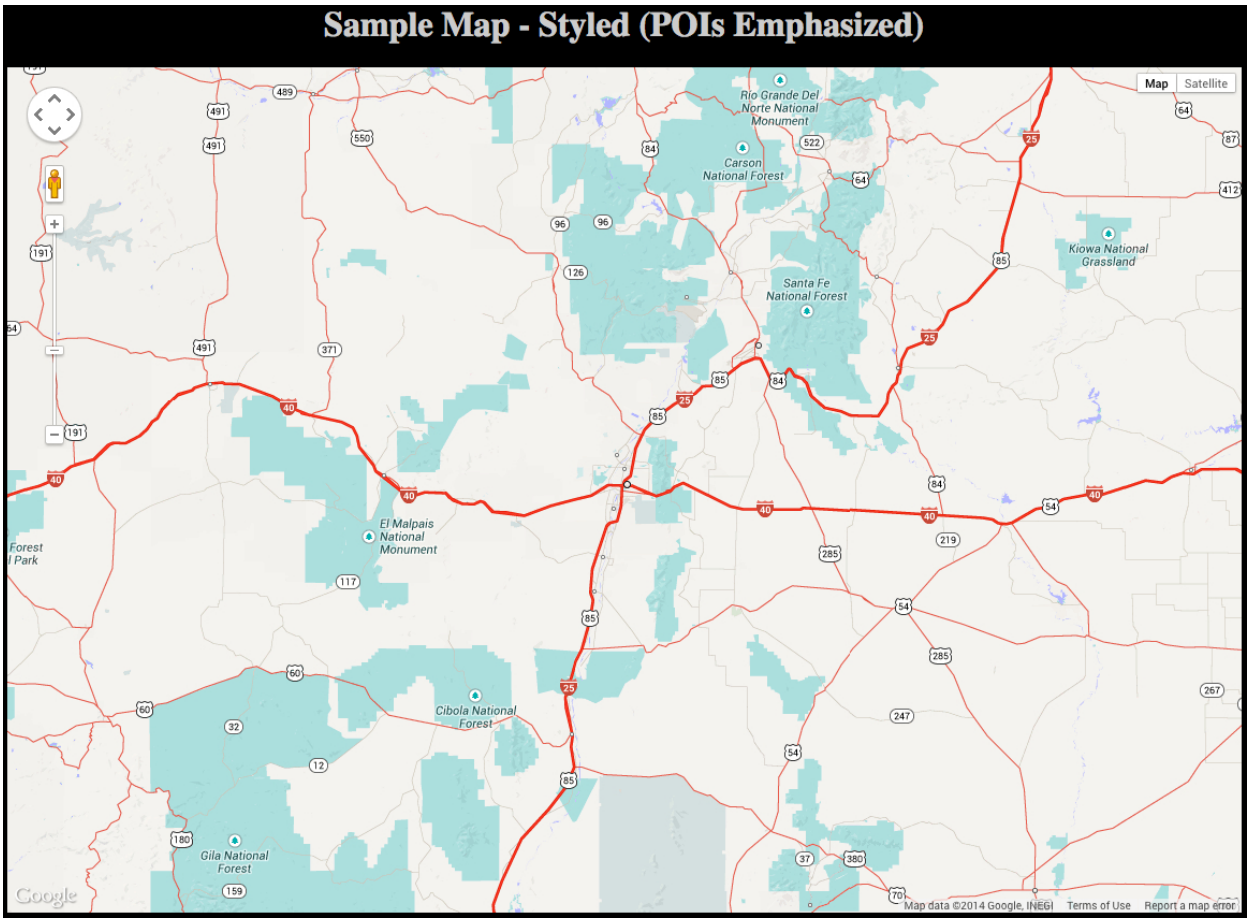
```
        </script>
    </head>

    <body onload="initialize()">
      <h1>Sample Map</h1>
      <div id="map_canvas"></div>
    </body>

</html>
```

Getting Started with Styled Maps - Video

Styled Maps Documentation | Styled Maps Wizard | YouTube Introductory Video



Map Example: Simple - Styled



http://karlbenedict.com/presentations/2014-04-NMGIC/examples/gmaps_styled.html

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
  html { height: 100% }
  body { height: 100%;
    margin: 0px;
    padding: 0px;
    background-color: black;
    color: #CCCCCC;
    text-align: center}
  #map_canvas { width:90%;
    height:80%;
    margin-left:
    auto;
    margin-right: auto }
</style>
<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?v=3.2&sensor=false">
</script>
```

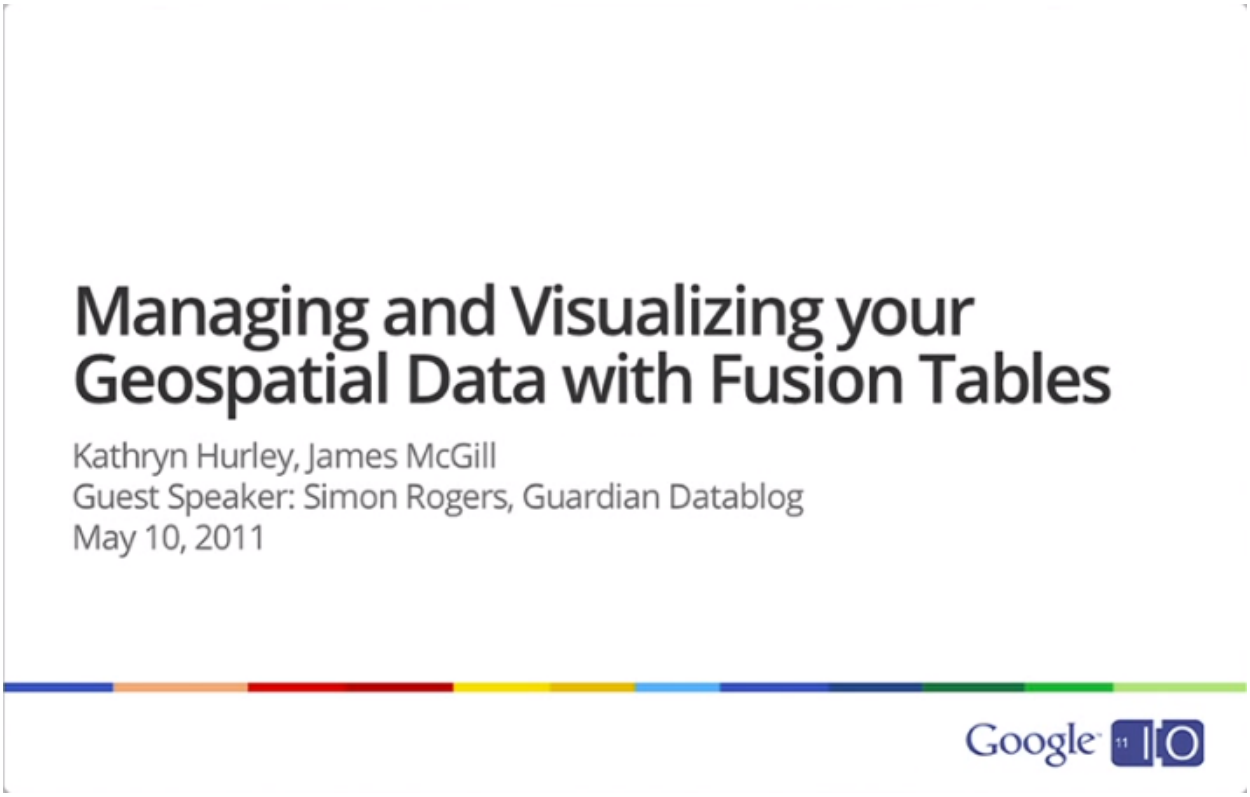
```
<script type="text/javascript">
  function initialize() {
    var classroom = new google.maps.LatLng(35.084280,-106.624073)
    var myOptions = {
      zoom: 8,
      center: classroom,
      mapTypeId: google.maps.MapTypeId.ROADMAP,
      styles: [
              {
                featureType: "water",
                stylers: [
                  { visibility: "on" },
                  { hue: "#0008ff" }
                ]
              },{
                featureType: "road.highway",
                stylers: [
                  { hue: "#ff1a00" }
                ]
              },{
                featureType: "road.arterial",
                stylers: [
                  { hue: "#ffa200" },
                  { visibility: "simplified" }
                ]
              },{
                featureType: "road.local",
                stylers: [
                  { visibility: "off" }
                ]
              },{
                featureType: "administrative",
                stylers: [
                  { visibility: "simplified" }
                ]
              },{
                featureType: "poi",
                stylers: [
                  { visibility: "on" },
                  { hue: "#00ffff" }
                ]
              },{
                featureType: "poi",
                stylers: [
                  { visibility: "on" }
                ]
              }
            ]
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"),
        myOptions);
  }
</script>
</head>
```

```
75
76  <body onload="initialize()">
77    <h1>Sample Map - Styled (POIs Emphasized)</h1>
78    <div id="map_canvas"></div>
79  </body>
80
81  </html>
```

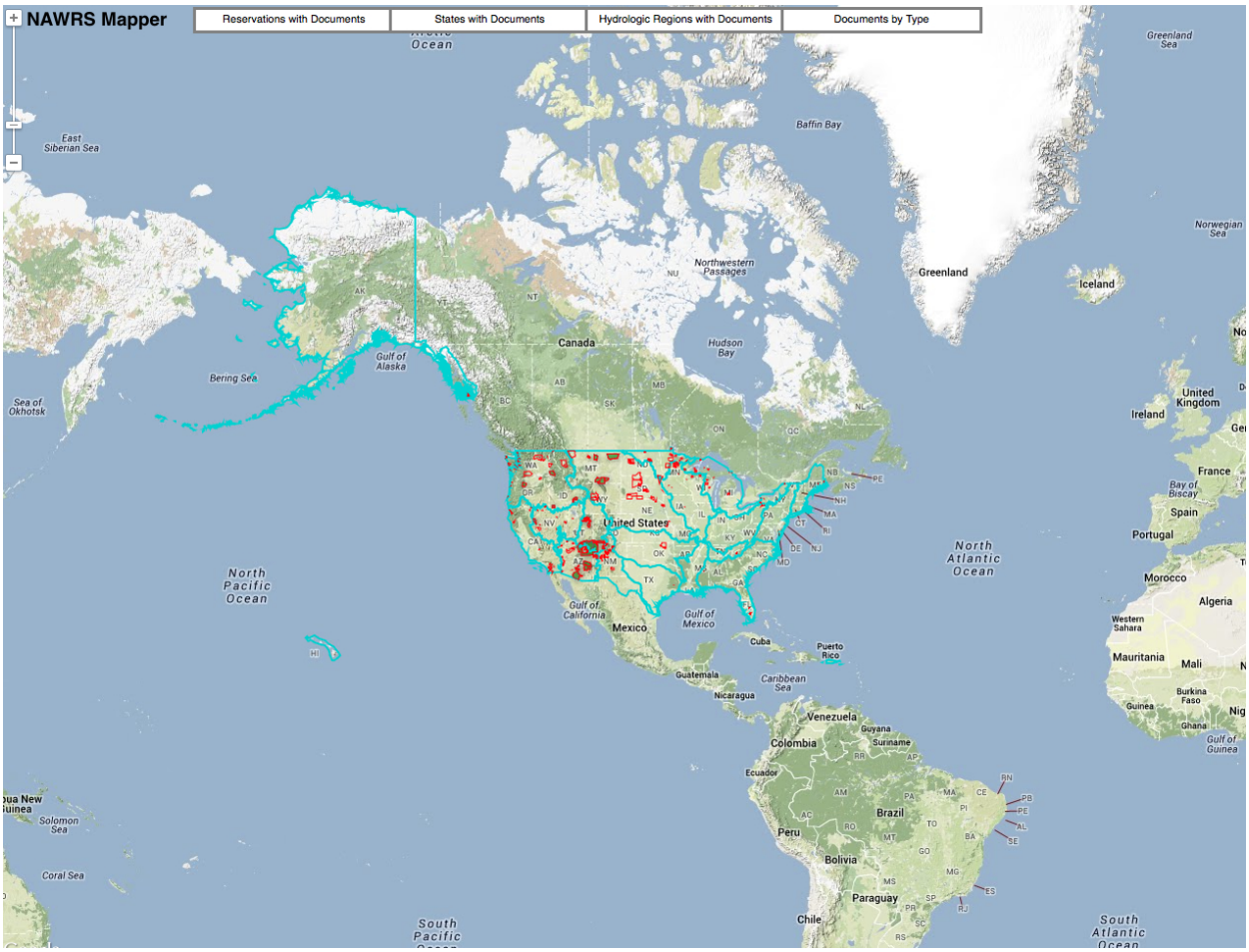
***Google I/O 2011: Managing and visualizing your geospatial data with Fusion Tables* - Video**

Fusion Tables Introduction Video - Some particularly relevant sections: Introduction (0:00 - 10:30) | Google Maps API Integration (21:40 - 34:42) | Summary and Links (52:00 - 52:40)

Fusion Tables Documentation/Help



Bringing It All Together



http://karlbenedict.com/nawrs/

Fusion Tables: Merged document info, State bounding boxes, HUC bounding boxes

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles.css">

<script type="text/javascript"
    src="http://maps.google.com/maps/api/js?v=3.2&sensor=false"></script>
<script type="text/javascript"
    src="//ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
<script type="text/javascript" src="http://www.google.com/jsapi"></script>
<script type="text/javascript" charset="utf-8" src="./core.js"></script>

<!-- DataTables and DataTables CSS -->
<link rel="stylesheet" type="text/css"
  href="http://kkb-projects.s3.amazonaws.com/nawrs/js/DataTables-1.9.4/
  media/css/jquery.dataTables.css">
<link rel="stylesheet" type="text/css"
  href="http://kkb-projects.s3.amazonaws.com/nawrs/js/DataTables-1.9.4/extras/TableTools/
```

```
  media/css/TableTools.css">
<script type="text/javascript" charset="utf8"
  src="http://kkb-projects.s3.amazonaws.com/nawrs/js/DataTables-1.9.4/media/js/
  jquery.dataTables.min.js"></script>
<script type="text/javascript" charset="utf8"
  src="http://kkb-projects.s3.amazonaws.com/nawrs/js/DataTables-1.9.4/extras/TableTools/media/js/
  TableTools.min.js"></script>

</head>

<body onload="initialize()">
  <h1>NAWRS Mapper</h1>
  <div id="docsReservations">Reservations with Documents</div>
  <div id="docsReservationsPopUp"><ul id="docsReservationsList"></ul></div>
  <div id="docsStates">States with Documents</div>
  <div id="docsStatesPopUp"><ul id="docsStatesList"></ul></div>
  <div id="docsHucs">Hydrologic Regions with Documents</div>
  <div id="docsHucsPopUp"><ul id="docsHucsList"></ul></div>
  <div id="docsType">Documents by Type</div>
  <div id="docsTypePopUp"><ul id="docsTypeList"></ul></div>
  <div id="map_canvas"></div>
  <div id="docListHandle">Document List</div>
  <div id="docList">
    <table id="docListTable">
  </table>
  </div>

</body>

</html>
```

OpenLayers Javascript Framework

Outline

- Capabilities
- OpenLayers = Javascript (by example)

OpenLayers Capabilities

- Support for Multiple basemaps: *Google*, *Yahoo*, *Bing*, *OpenStreetMap*
- Model for interaction with multiple map server platforms: *ArcGIS* (REST & cache), *ArcIMS*, *KaMap*, *MapServer*
- Support for key OGC standards: *WMS*, *WMTS*, *WFS*, *GML*, *KML*, *SLD*
- Multiple control types: *Navigation*, *Pan*, *Zoom*, *Overview*, *Scale*, *Feature Creation & Editing*, *Graticle*, *Layer Switcher*
- Custom styled features with associated attributes: *Curve*, *LinearRing*, *LineString*, *MultiLineString*, *MultiPoint*, *MultiPolygon*, *Point*, *Polygon*, *Rectangle*
- Support for many formats for data read and write: *ArcXML*, *ATOM*, *GeoRSS*, *GPX*, *KML*, *WKT*, any many others
- Open Source, enabling modification and integration into other systems (e.g. GeoExt)

Distinguishing Characteristics Between OpenLayers and Google Maps

- Greater emphasis on client-side processing - Client access and rendering of data files that Google's servers otherwise take care of (pros & cons to this approach)
- Integrated support for OGC services and their products
- Support for different projections (adds complexity)
- API more rich in options ==> more complexity

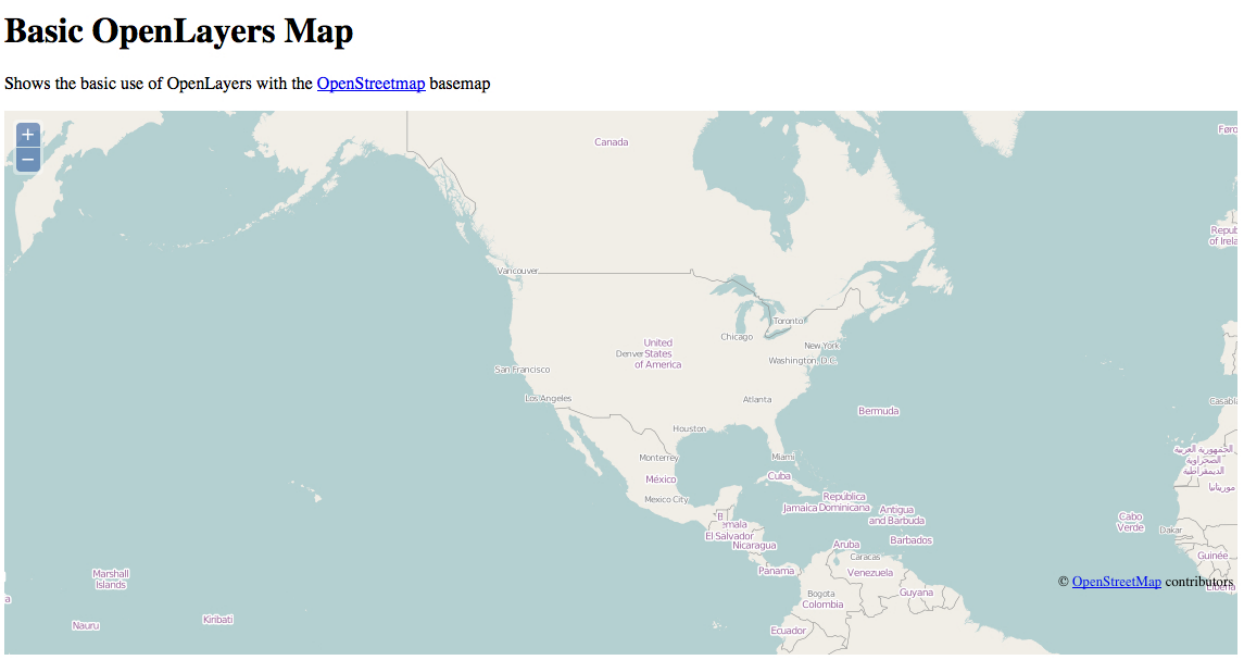
Resources

OpenLayers Home Page

Application Programming Interface (API) Reference

Examples

Demonstrations and Examples



- Basic Mapper (with OpenStreetMaps [OSM] base map)

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script type="text/javascript" src="http://openlayers.org/api/OpenLayers.js"></script>
    <script type="text/javascript">
        // define global variables
        var lon = -106.5;
        var lat = 36;
        var zoom = 3;
        var map;
        var layer;
```

```
        // =============== Initialization function ==================
        function init(){
            map = new OpenLayers.Map( 'map' );

            // ========== OSM Map ====================
            layer = new OpenLayers.Layer.OSM( "Open Street Map");
            map.addLayer(layer);

            map.setCenter(
                new OpenLayers.LonLat(lon, lat).transform(
                    new OpenLayers.Projection("EPSG:4326"),
                    map.getProjectionObject()
                ), zoom
            );
        }
        // =============== End of Initialization Function ============

    </script>
    <style type="text/css">
        #map {width:90%; height:500px}
    </style>
  </head>
  <body onload="init()">
    <h1>Basic OpenLayers Map</h1>
    <p>Shows the basic use of OpenLayers with the
    <a href="http://www.openstreetmap.org/">OpenStreetmap</a> basemap</p>
    <!-- Map DIV -->
    <div id="map"></div>
  </body>
</html>
```

Demonstration and Examples - Online Resources

- Mapper with a variety of base maps (Google, Bing, Yahoo, OSM)
- Basic Mapper with Controls: No Controls, Layer Switcher, Control Array, Overlay Map, Scale Information
- Positioning Controls with the `moveTo` function: two controls moved

Map Object Options

Map Object Options API Reference

Two methods for constructing a new `OpenLayers.Map` object

```
    // create a map with default options in an element with the id "map1"
    var map = new OpenLayers.Map("map1");

    // create a map with non-default options in an element with id "map2"
    var options = {
        maxExtent: new OpenLayers.Bounds(-200000, -200000, 200000, 200000),
        maxResolution: 156543,
        units: 'm',
```

```
        projection: "EPSG:41001"
    };
    var map = new OpenLayers.Map("map2", options);

    // map with non-default options - same as above but with a single argument
    var map = new OpenLayers.Map({
        div: "map_id",
        maxExtent: new OpenLayers.Bounds(-200000, -200000, 200000, 200000),
        maxResolution: 156543,
        units: 'm',
        projection: "EPSG:41001"
    });
```

Excerpts from the API documentation

allOverlays {Boolean} Allow the map to function with "overlays" only. Defaults to false. If true, the lowest layer in the draw order will act as the base layer. In addition, if set to true, all layers will have isBaseLayer set to false when they are added to the map.

div {DOMElement|String} The element that contains the map (or an id for that element). If the OpenLayers.Map constructor is called with two arguments, this should be provided as the first argument. Alternatively, the map constructor can be called with the options object as the only argument. In this case (one argument), a div property may or may not be provided. If the div property is not provided, the map can be rendered to a container later using the render method.

layers {Array(OpenLayers.Layer)} Ordered list of layers in the map

tileSize {OpenLayers.Size} Set in the map options to override the default tile size for this map.

projection {String} Set in the map options to override the default projection string this map - also set **maxExtent**, **maxResolution**, and **units** if appropriate. Default is "EPSG:4326".

units {String} The map units. Defaults to 'degrees'. Possible values are 'degrees' (or 'dd'), 'm', 'ft', 'km', 'mi', 'inches'.

resolutions {Array(Float)} A list of map resolutions (map units per pixel) in descending order. If this is not set in the layer constructor, it will be set based on other resolution related properties (maxExtent, maxResolution, maxScale, etc.).

maxResolution {Float} Default max is 360 deg / 256 px, which corresponds to zoom level 0 on gmaps. Specify a different value in the map options if you are not using a geographic projection and displaying the whole world.

minResolution {Float}

maxScale {Float}

minScale {Float}

maxExtent {OpenLayers.Bounds} The maximum extent for the map. Defaults to the whole world in decimal degrees (-180, -90, 180, 90). Specify a different extent in the map options if you are not using a geographic projection and displaying the whole world.

minExtent {OpenLayers.Bounds}

restrictedExtent {OpenLayers.Bounds} Limit map navigation to this extent where possible. If a non-null restrictedExtent is set, panning will be restricted to the given bounds. In addition, zooming to a resolution that displays more than the restricted extent will center the map on the restricted extent. If you wish to limit the zoom level or resolution, use maxResolution.

numZoomLevels {Integer} Number of zoom levels for the map. Defaults to 16. Set a different value in the map options if needed.

Layer Object Options

Layer Object Options API Reference

Common Pattern of Layer Object Creation (varies some depending upon the specific layer type)

```
    new OpenLayers.Layer.*** (
        'layer name',
        'layer URL',
        {server-related options},
        {OpenLayers Layer Object options}
    )
```

id {String}

name {String}

isBaseLayer {Boolean} Whether or not the layer is a base layer. This should be set individually by all subclasses. Default is false

displayInLayerSwitcher {Boolean} Display the layer's name in the layer switcher. Default is true.

visibility {Boolean} The layer should be displayed in the map. Default is true.

attribution {String} Attribution string, displayed when an OpenLayers.Control.Attribution has been added to the map.

projection {OpenLayers.Projection} or {String} Set in the layer options to override the default projection string this layer - also set maxExtent, maxResolution, and units if appropriate. Can be either a string or an OpenLayers.Projection object when created – will be converted to an object when setMap is called if a string is passed.

units {String} The layer map units. Defaults to 'degrees'. Possible values are 'degrees" (or 'dd'), 'm', 'ft', 'km', 'mi', 'inches'.

scales {Array} An array of map scales in descending order. The values in the array correspond to the map scale denominator. Note that these values only make sense if the display (monitor) resolution of the client is correctly guessed by whomever is configuring the application. In addition, the units property must also be set. Use resolutions instead wherever possible.

resolutions {Array} A list of map resolutions (map units per pixel) in descending order. If this is not set in the layer constructor, it will be set based on other resolution related properties (maxExtent, maxResolution, maxScale, etc.).

maxExtent {OpenLayers.Bounds} The center of these bounds will not stray outside of the viewport extent during panning. In addition, if displayOutsideMaxExtent is set to false, data will not be requested that falls completely outside of these bounds.

minExtent {OpenLayers.Bounds}

maxResolution {Float} Default max is 360 deg / 256 px, which corresponds to zoom level 0 on gmaps. Specify a different value in the layer options if you are not using a geographic projection and displaying the whole world.

minResolution {Float}

`numZoomLevels` {Integer}

`minScale` {Float}

`maxScale` {Float}

`displayOutsideMaxExtent` {Boolean} Request map tiles that are completely outside of the max extent for this layer. Defaults to false.

`transitionEffect` {String} The transition effect to use when the map is panned or zoomed.

There are currently two supported values `null` No transition effect (the default).

`resize` Existing tiles are resized on zoom to provide a visual effect of the zoom having taken place immediately. As the new tiles become available, they are drawn over top of the resized tiles.

Additional Map and Layer Object Functions & Events

Both Map and Layer Objects have a number of associated functions as well

- Retrieving object properties programmatically with `Get` functions.
- Modifying existing object properties with `Set` functions
- Map destruction, and reconfiguration
- Linkage of object events with Javascript functions

WMS Layer Configuration

Some key issues to be aware of when using the WMS Layer Class:

- The *projection* of the map object must be supported by the included WMS service (review the WMS GetCapabilities response to see what projections are supported by the service)
- The *layers* parameter/property must be provided as part of the server-related property list (the layer names also come from the GetCapabilities response)
- Other WMS parameters may be provided as well to "adjust" the request automatically generated by OpenLayers

Sample WMS Layer Object Creation

Basic OpenLayers Map

Shows the basic use of OpenLayers with a set of WMS layers

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/openLayers10_wms01.html

```
    countiesLayer = new OpenLayers.Layer.WMS(
        "US Counties",
        "http://webservices.nationalatlas.gov/wms?",
        {layers: "counties", version: '1.3.0', transparent: 'TRUE'},
        {isBaseLayer: false, visibility: false, opacity: .8}
    );
    map.addLayer(countiesLayer);
```

Vector Layer Configuration

Vector layers support

- External Data in a Variety of supported formats for both *reading* and *writing* (just a sample): ArcXML.Features, GeoJSON, GeoRSS, GPX, JSON, KML, WFS, WKT
- Directly encoded [geometries][OpenLayers.Geometry API Link]: Collection, Curve, LinearRing, LineString, MultiLineString, MultiPoint, MultiPolygon, Point, Polygon, Rectangle
- User created features, including support for interactive editing of features
- Styling of Vector features

Vector Layer Objects are Typically Defined using three OpenLayers classes

`Protocol` Connection protocol for requesting the data that would be provided from an external source

`Format` The OpenLayers supported format of the vector data object

`Strategy` A specification of how OpenLayers should request the data from the server, and also handle the data within the client (browser).

Basic OpenLayers Map

Shows the basic use of OpenLayers with a set of WMS layers with vector overlays and KML layers

http://karlbenedict.com/presentations/2014-04-NMGIC/examples/openLayers11_vectorData_KML.html

Sample Point Feature Object creation

```
    var Coord_classroom = new OpenLayers.Geometry.Point(-106.624073,35.084280);
    var Point_classroom = new OpenLayers.Feature.Vector(Coord_classroom);
    Layers["localFeatures"].addFeatures([Point_classroom])
```

Sample KML Layer Object creation

```
    Layers.counties = new OpenLayers.Layer.Vector("KML - Counties", {
        projection: map.displayProjection,
        strategies: [new OpenLayers.Strategy.Fixed()],
        protocol: new OpenLayers.Protocol.HTTP({
            url: "NMCounties.kml",
            format: new OpenLayers.Format.KML({
                extractAttributes: true
```

```
            })
        })
    });
    map.addLayer(Layers.counties)
```

Questions?