

Visualization: Map

Overview

The Google Map Chart displays a map using the [Google Maps API](https://developers.google.com/maps/) (<https://developers.google.com/maps/>). Data values are displayed as markers on the map. Data values can be coordinates (lat-long pairs) or addresses. The map will be scaled so that it includes all the identified points.

If you want your maps to be line drawings rather than satellite imagery, use a [geochart](https://developers.google.com/chart/interactive/docs/gallery/geochart) (<https://developers.google.com/chart/interactive/docs/gallery/geochart>) instead.

Named Locations

You can identify the places to put markers by name, as shown below in this map of the top ten countries by population.



When the user selects one of the markers, a tooltip with the name and population of the country is displayed, because we used the `showInfoWindow` option. Also, when the user hovers over one of the markers for a short while, a 'title' tip will show up with the same info, because we used the `showTooltip` option. Here's the full HTML:

```
<html>
<head>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js">
</script>
<script>
  google.charts.load('current', { 'packages': ['map'] });
  google.charts.setOnLoadCallback(drawMap);

  function drawMap() {
    var data = google.visualization.arrayToDataTable([
      ['Country', 'Population'],
      ['China', 'China: 1,363,800,000'],
      ['India', 'India: 1,242,620,000'],
      ['US', 'US: 317,842,000'],
      ['Indonesia', 'Indonesia: 247,424,598'],
      ['Brazil', 'Brazil: 201,032,714'],
      ['Pakistan', 'Pakistan: 186,134,000'],
      ['Nigeria', 'Nigeria: 173,615,000'],
      ['Bangladesh', 'Bangladesh: 152,518,015'],
      ['Russia', 'Russia: 146,019,512'],
      ['Japan', 'Japan: 127,120,000']
    ]);

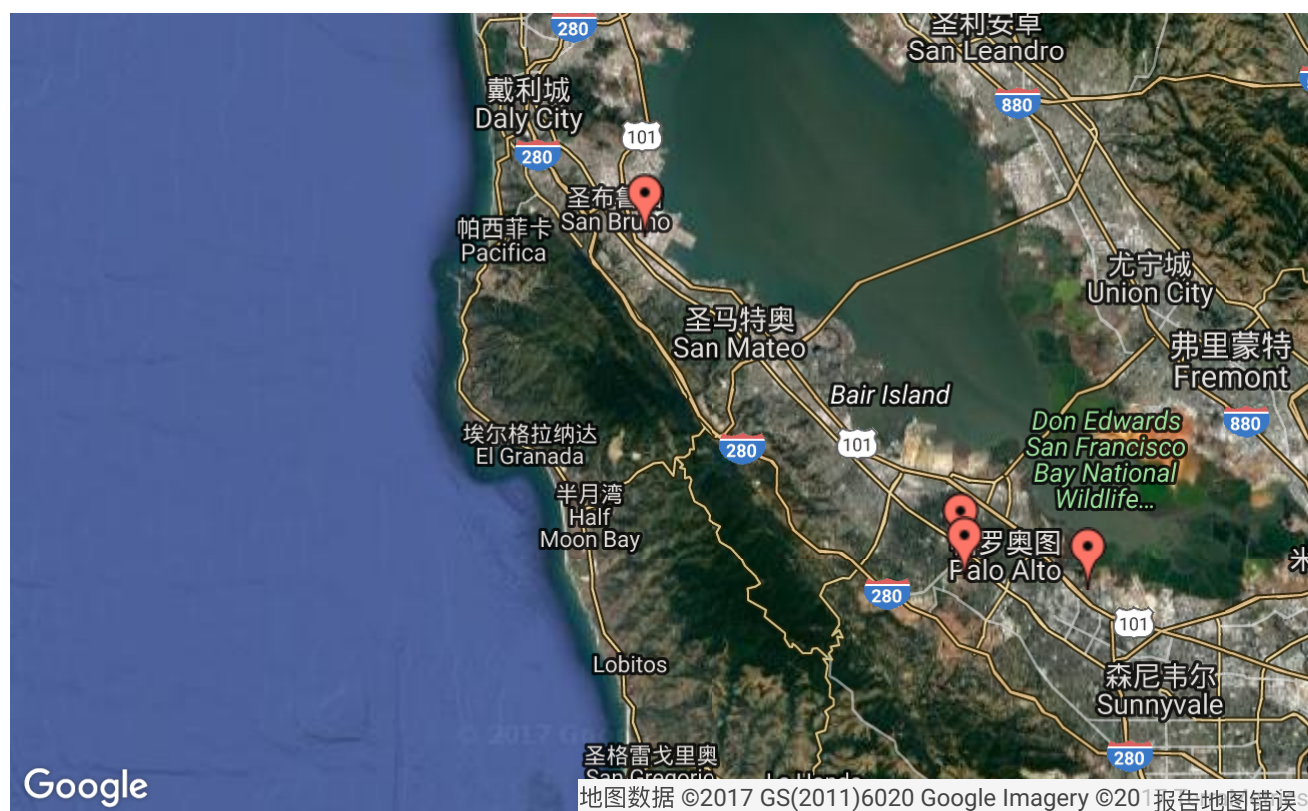
    var options = {
      showTooltip: true,
      showInfoWindow: true
    };

    var map = new google.visualization.Map(document.getElementById('chart_div'));

    map.draw(data, options);
  };
</script>
</head>
<body>
  <div id="chart_div"></div>
</body>
</html>
```

Geocoded Locations

You can also specify locations by latitude and longitude, which loads faster than named locations:



The above chart identifies four locations by latitude and longitude.

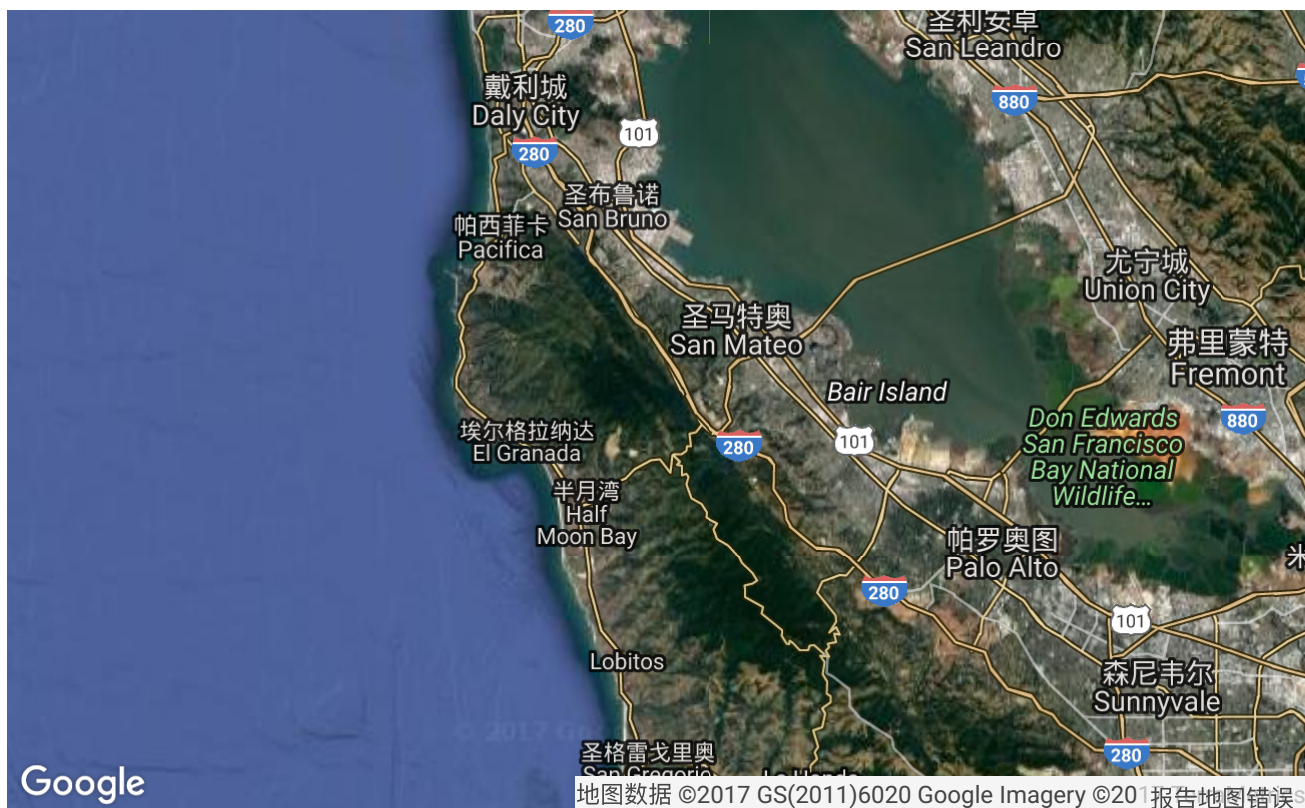
DATA

FULL HTML

```
var data = google.visualization.arrayToDataTable([
  ['Lat', 'Long', 'Name'],
  [37.4232, -122.0853, 'Work'],
  [37.4289, -122.1697, 'University'],
  [37.6153, -122.3900, 'Airport'],
  [37.4422, -122.1731, 'Shopping']
]);
```

Customizing Markers

You can use marker shapes located elsewhere on the web. Here's an example using blue pins from iconarchive.com:



If you select pins in the above chart, they slant. PNGs, GIFs, and JPGs are supported.

DATA

FULL HTML

```

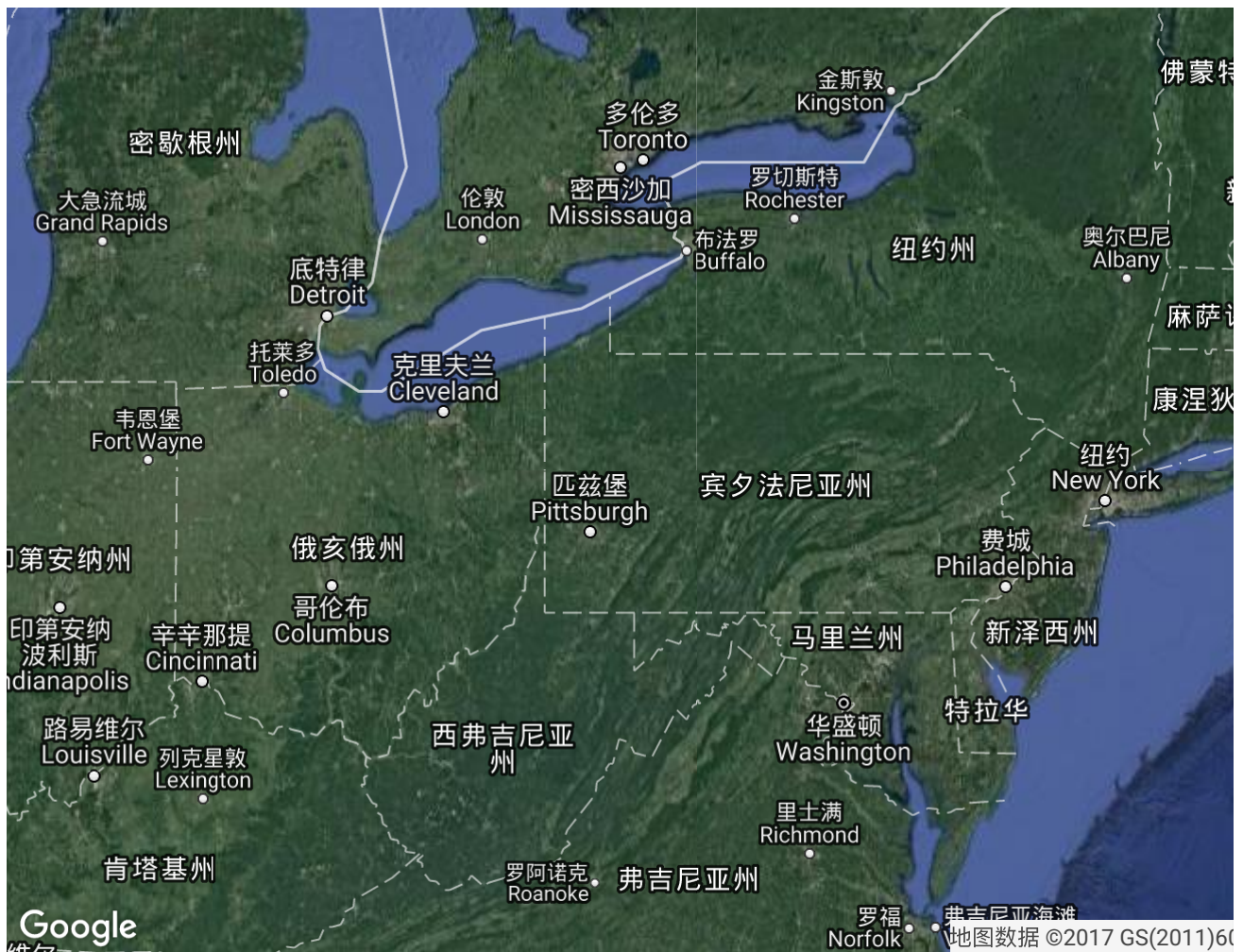
var options = {
  icons: {
    default: {
      normal: 'https://icons.iconarchive.com/icons/iconarchive.com/icons-land/vista-map-ma
      selected: 'https://icons.iconarchive.com/icons/iconarchive.com/icons-land/vista-map-i
    }
  }
};

```

Adding Multiple Marker Sets

In addition to creating a default set of markers, you can also create multiple sets of markers.

To create an additional marker set, add an ID to the `icons` option and set your marker images. Then, add a column to your Data Table, and set the ID of the marker set you wish to use for each row in the Data Table (you can also use `'default'` or `null` to use the default markers).



[CODE IT YOURSELF ON JSFIDDLE](#)

OPTIONS

FULL HTML

```
var url = 'https://icons.iconarchive.com/icons/icons-land/vista-map-ma

var options = {
  zoomLevel: 6,
  showTooltip: true,
  showInfoWindow: true,
  useMapTypeControl: true,
  icons: {
    blue: {
      normal: url + 'Map-Marker-Ball-Azure-icon.png',
      selected: url + 'Map-Marker-Ball-Right-Azure-icon.png'
    },
    green: {
      normal: url + 'Map-Marker-Push-Pin-1-Chartreuse-icon.png',
      selected: url + 'Map-Marker-Push-Pin-1-Right-Chartreuse-icon.png'
    },
    pink: {
```

```
        normal: url + 'Map-Marker-Ball-Pink-icon.png',  
        selected: url + 'Map-Marker-Ball-Right-Pink-icon.png'  
    }  
}  
};
```

Styling Maps

The Map Visualization comes with the ability to set custom styles, allowing you to create one, or several, custom map types.

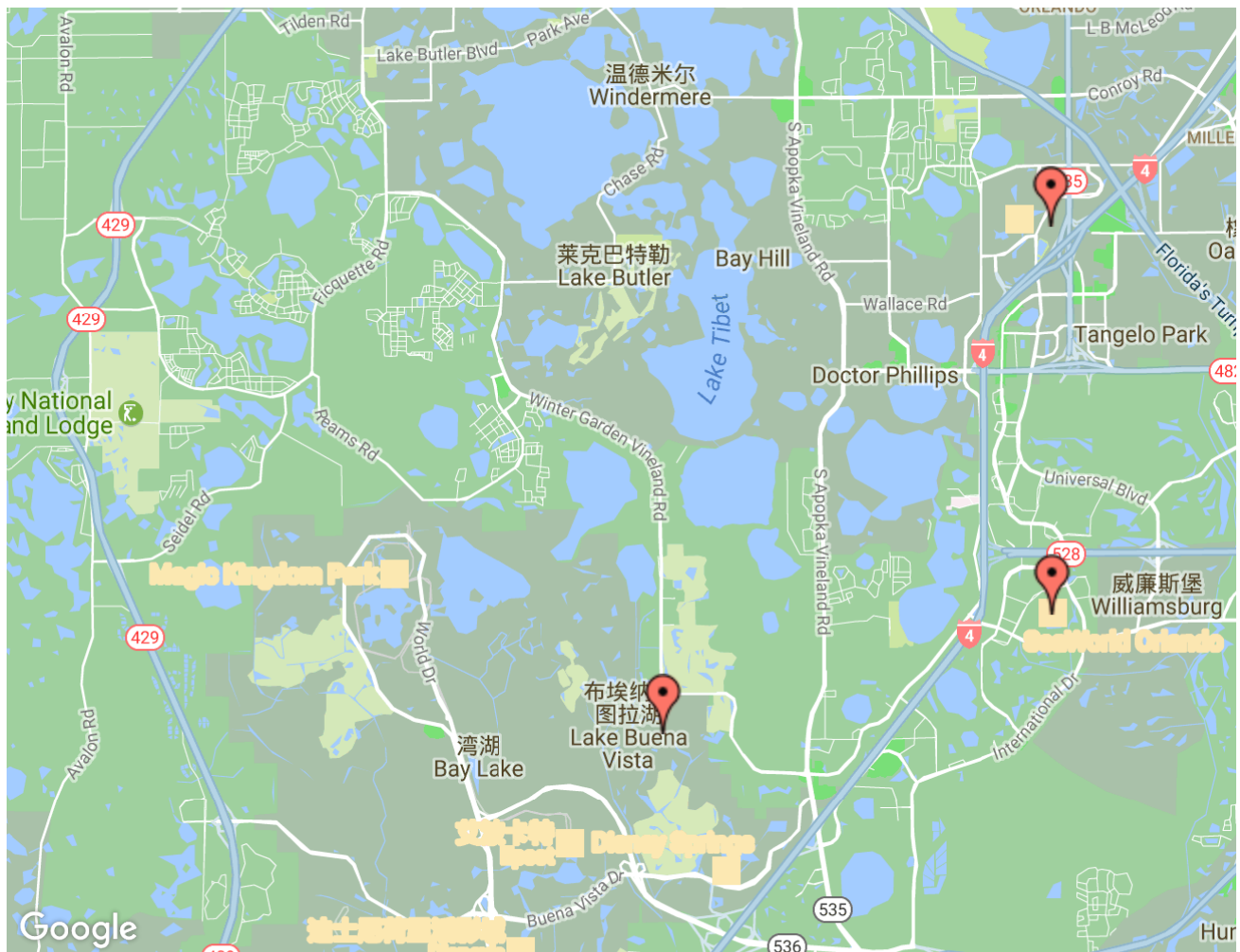
You can define a custom map type by creating a map style object and putting it under its identifier (`mapTypeId`) under the `maps` option. For example:

```
var options = {  
  maps: {  
    <map style ID>: <map style object>  
  }  
};
```

You can later refer to this custom map type by the map style ID you assigned to it.

Your map style object contains a `name`, which will be the display name in the map type control (it does not have to match its `mapTypeId`), and a `styles` array, which will contain style objects for each element you wish to style. The [Google Maps API reference](https://developers.google.com/maps/documentation/javascript/reference#MapTypeStyle) (<https://developers.google.com/maps/documentation/javascript/reference#MapTypeStyle>) contains a list of the different element, feature, and style types with which you can create a custom map type.

Note: Your custom `mapTypeId` must be placed under the `maps` option.



[CODE IT YOURSELF ON JSFIDDLE](#)

OPTIONS

FULL HTML

```
var options = {
  mapType: 'styledMap',
  zoomLevel: 12,
  showTooltip: true,
  showInfoWindow: true,
  useMapTypeControl: true,
  maps: {
    // Your custom mapTypeId holding custom map styles.
    styledMap: {
      name: 'Styled Map', // This name will be displayed in the map type
      styles: [
        {featureType: 'poi.attraction',
          stylers: [{color: '#fce8b2'}]},
        {featureType: 'road.highway',
          stylers: [{hue: '#0277bd'}, {saturation: -50}]},
      ],
    },
  },
}
```

```
        {featureType: 'road.highway',  
          elementType: 'labels.icon',  
          stylers: [{hue: '#000'}, {saturation: 100}, {lightness: 50}]  
        },  
        {featureType: 'landscape',  
          stylers: [{hue: '#259b24'}, {saturation: 10}, {lightness: -22  
        }  
      ]}}  
    };
```

In addition to creating custom map types, you can also specify which maps the user can view in the map type control with the `mapTypeIds` option. By default all map types, including your custom map types, will appear in the map type control. The `mapTypeIds` option accepts an array of strings of the map types you wish to allow the user to view. These strings must refer to either the predefined names for the default map styles (i.e. normal, satellite, terrain, hybrid), or the map style ID of a custom map type you defined. By setting the `mapTypeIds` option, your map will only make available the map types you specify in this array.

You can also use this in conjunction with the `mapType` option to set a map style as the default, but not include it in the `mapTypeIds` array. This will cause that map to only display on the initial load.



Google

[CODE IT YOURSELF ON JSFIDDLE](#)

```
var options = {
  mapType: 'styledMap',
  zoomLevel: 7,
  showTooltip: true,
  showInfoWindow: true,
  useMapTypeControl: true,
  // User will only be able to view/select custom styled maps.
  mapTypeIds: ['styledMap', 'redEverything', 'imBlue'],
  maps: {
    styledMap: {
      name: 'Styled Map',
      styles: [
        {featureType: 'poi.attraction',
          stylers: [{color: '#fce8b2'}]},
        {featureType: 'road.highway',
          stylers: [{hue: '#0277bd'}, {saturation: -50}]},
        {featureType: 'road.highway', elementType: 'labels.icon',
          stylers: [{hue: '#000'}, {saturation: 100}, {lightness: 50}]},
        {featureType: 'landscape',
```

```

        stylers: [{hue: '#259b24'}, {saturation: 10}, {lightness: -22}]
    }},
    redEverything: {
        name: 'Redden All The Things',
        styles: [
            {featureType: 'landscape',
              stylers: [{color: '#fde0dd'}]},
            {featureType: 'road.highway',
              stylers: [{color: '#67000d'}]},
            {featureType: 'road.highway', elementType: 'labels',
              stylers: [{visibility: 'off'}]},
            {featureType: 'poi',
              stylers: [{hue: '#ff0000'}, {saturation: 50}, {lightness: 0}]}
            {featureType: 'water',
              stylers: [{color: '#67000d'}]},
            {featureType: 'transit.station.airport',
              stylers: [{color: '#ff0000'}, {saturation: 50}, {lightness: -50}]}
        ]},
    imBlue: {
        name: 'All Your Blues are Belong to Us',
        styles: [
            {featureType: 'landscape',
              stylers: [{color: '#c5cae9'}]},
            {featureType: 'road.highway',
              stylers: [{color: '#023858'}]},
            {featureType: 'road.highway', elementType: 'labels',
              stylers: [{visibility: 'off'}]},
            {featureType: 'poi',
              stylers: [{hue: '#0000ff'}, {saturation: 50}, {lightness: 0}]}
            {featureType: 'water',
              stylers: [{color: '#0288d1'}]},
            {featureType: 'transit.station.airport',
              stylers: [{color: '#0000ff'}, {saturation: 50}, {lightness: -50}]}
        ]}
    }
};

```

Loading

The `google.charts.load` package name is `"map"`.

```
google.charts.load("current", {packages: ["map"]});
```

The visualization's class name is `google.visualization.Map`.

```
var visualization = new google.visualization.Map(container);
```

Data Format

Two alternative data formats are supported:

1. **Lat-Long pairs** - The first two columns should be numbers designating the latitude and longitude, respectively. An optional third column holds a string that describes the location specified in the first two columns.
2. **String address** - The first column should be a string that contains an address. This address should be as complete as you can make it. An optional second column holds a string that describes the location in the first column. String addresses load more slowly, especially when you have more than 10 addresses.

Note: The Lat-Long pairs option loads maps much faster, especially with large data. We recommend that you use this option for large data sets. Please visit [Google Maps API](https://developers.google.com/maps/documentation/javascript/v2/reference#GClientGeocoder) (<https://developers.google.com/maps/documentation/javascript/v2/reference#GClientGeocoder>) to find out how to transform your addresses to lat-long points. The map can display a maximum of 400 entries; if your data holds more than 400 rows, only the first 400 will be shown.

Configuration Options

Name	
enableScrollWheel	If set to true, enables zooming in and out using the mouse scroll wheel Type: boolean Default: false
icons	Holds set(s) of custom markers. Each marker set can specify a normal image. Can set a default set by setting the default option, or custom unique marker ID. <pre>var options = { icons: { default: { normal: '/path/to/marker/image', selected: '/path/to/marker/image' }, }, }</pre>

	<pre> customMarker: { normal: '/path/to/other/marker/image', selected: '/path/to/other/marker/image' } }; </pre> <p>Type: object Default: null</p>
lineColor	<p>If showLine is true, defines the line color. For example: '#800000'.</p> <p>Type: string Default: default color</p>
lineWidth	<p>If showLine is true, defines the line width (in pixels).</p> <p>Type: number Default: 10</p>
maps.<mapTypeId>	<p>An object containing properties of a custom map type. This custom map type is created by calling <code>maps.mapTypeId</code> with the <code>mapTypeId</code> you specify for the custom map type. A new <code>mapTypeId</code> is created for each custom map type created. Each custom map type should contain two properties:</p> <ul style="list-style-type: none"> name: The display name for the styled map type styles: An array containing the style objects for the styled map type <p>Type: object Default: null</p>
maps.<mapTypeId>.name	<p>The name of the map that will be displayed in the map control if <code>useMapId</code> is <code>true</code>.</p> <p>Type: string Default: null</p>
maps.<mapTypeId>.styles	<p>Holds the style objects for the various elements of a custom map type.</p> <p>Each style object can contain 1 to 3 properties: <code>featureType</code>, <code>elementType</code>, and <code>stylers</code>. If <code>featureType</code> and <code>elementType</code> are omitted, the styles will be applied to all features/elements.</p> <pre> { featureType: 'roadway.highway', elementType: 'labels.text.fill', stylers: [{hue: '#ff0000'}, {saturation: 50}, {lightness: 40}] } </pre>

	<p>See the Maps (https://developers.google.com/maps/documentation/javascript/reference) for more information on the different features, elements and controls.</p> <p>Type: array Default: null</p>
mapType	<p>The type of map to show. Possible values are 'normal', 'terrain', 'satellite' or a custom map type, if any were created.</p> <p>Type: string Default: 'hybrid'</p>
mapTypeIds	<p>If using the map type control (<code>useMapTypeControl: true</code>), the IDs of the only map types displayed in the map type control. If this option is not used, the map type control will default to the standard Google Maps map type control options plus any custom map types that may be available.</p> <p>Type: array Default: null</p>
showInfoWindow	<p>If set to true, shows the location description in a separate window when the user clicks on a point marker by the user. This option used to be called <code>showTip</code> up until version 45.</p> <p>Type: boolean Default: false</p>
showLine	<p>If set to true, shows a Google Maps polyline (https://developers.google.com/maps/documentation/javascript/v2/controls#PolylineControl) through all the points.</p> <p>Type: boolean Default: false</p>
showTooltip	<p>If set to true, shows the location description as a tooltip when the mouse hovers over a point marker. This option used to be called <code>showTip</code> up until version 45. If HTML is not supported, so the tooltip would show raw HTML tags.</p> <p>Type: boolean Default: false</p>
useMapTypeControl	<p>Show a map type selector that enables the viewer to switch between [normal, terrain, satellite] map types. When <code>useMapTypeControl</code> is false (default) no selector is presented and the <code>mapType</code> option is used.</p> <p>Type: boolean Default: false</p>
zoomLevel	<p>An integer indicating the initial zoom level of the map, where 0 is compass view (world) and 19 is the maximum zoom level. (See "Zoom Levels" in the Google Maps Static Maps API documentation (https://developers.google.com/maps/documentation/staticmaps/#ZoomLevels)).</p>

Type: number
Default: automatic

Methods

Method	
<code>draw(data, options)</code>	<p>Draws the map.</p> <p>Return Type: none</p>
<code>getSelection()</code>	<p>Standard getSelection() (https://developers.google.com/chart/interactive/docs/reference#vis) implementation. Selection elements are all row elements. Can return n selected row.</p> <p>Return Type: Array of selection elements</p>
<code>setSelection(selection)</code>	<p>Standard setSelection() (https://developers.google.com/chart/interactive/docs/reference#vis) implementation. Treats every selection entry as a row selection. Support of multiple rows.</p> <p>Return Type: none</p>

Events

Name	
<code>error</code>	<p>Fired when an error occurs when attempting to render the chart.</p> <p>Properties: id, message</p>
<code>select</code>	<p>Standard select event</p> <p>Properties: None</p>

Data Policy

Map are displayed by Google Maps. Please refer to the [Google Maps Terms of Service](https://developers.google.com/maps/terms) (<https://developers.google.com/maps/terms>) for more information on data policy.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

上次更新日期: 二月 23, 2017