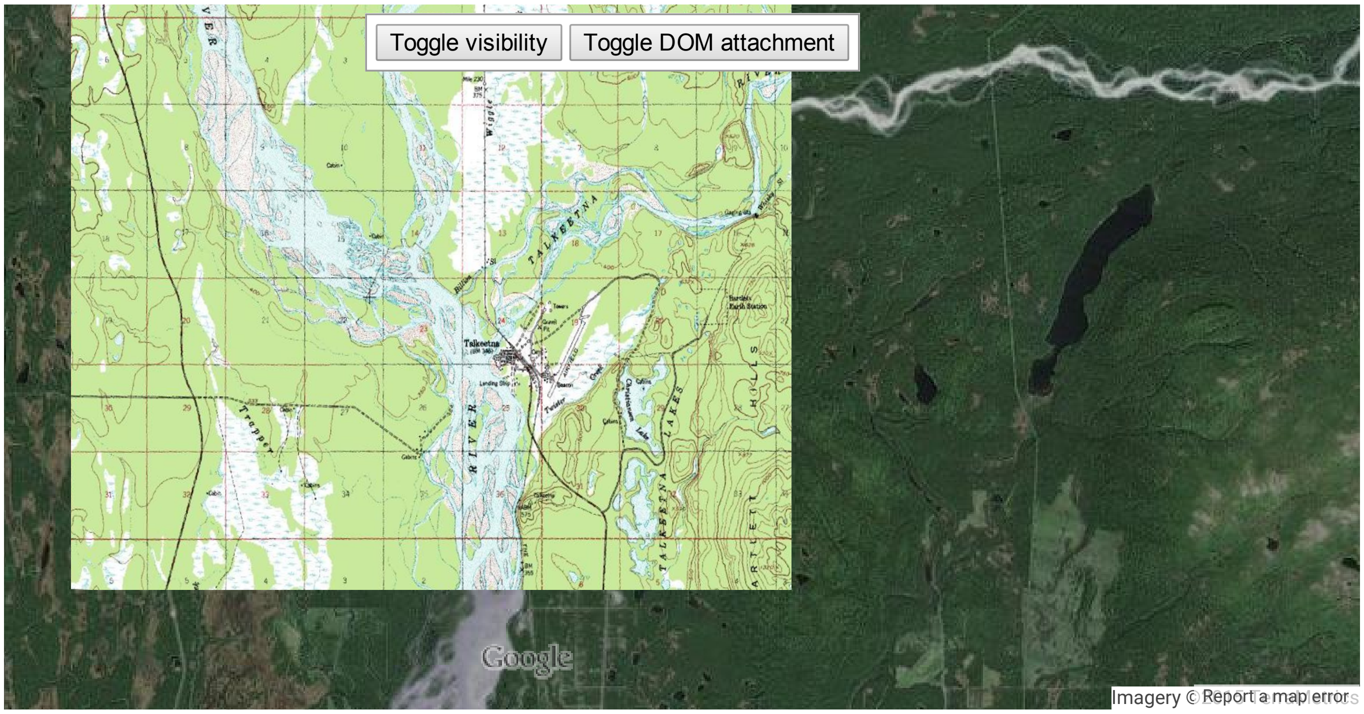
Google Developers

Googl... X Search

Sign in

Products Google Maps API Google Maps JavaScript API v3

Showing/Hiding overlays



View this example full screen.

JavaScript JavaScript + HTML

```
// This example adds hide() and show() methods to a custom overlay's prototype.
// These methods toggle the visibility of the container <div>.
// Additionally, we add a toggleDOM() method, which attaches or detaches the
// overlay to or from the map.

var overlay;

USGSOverlay.prototype = new google.maps.OverlayView();

function initialize() {
  var myLatLng = new google.maps.LatLng(62.323907, -150.109291);
  var mapOptions = {
    zoom: 11,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.SATELLITE
  };

  var map = new google.maps.Map(document.getElementById('map-canvas'),
      mapOptions);
  var swBound = new google.maps.LatLng(62.281819, -150.287132);
  var neBound = new google.maps.LatLng(62.400471, -150.005608);
  var bounds = new google.maps.LatLngBounds(swBound, neBound);

  // The photograph is courtesy of the U.S. Geological Survey.
  var srcImage = 'https://developers.google.com/maps/documentation/javascript/';
  srcImage += 'examples/full/images/talkeetna.png';

  overlay = new USGSOverlay(bounds, srcImage, map);
}

/** @constructor */
function USGSOverlay(bounds, image, map) {

  // Now initialize all properties.
  this.bounds_ = bounds;
  this.image_ = image;
  this.map_ = map;

  // Define a property to hold the image's div. We'll
  // actually create this div upon receipt of the onAdd()
  // method so we'll leave it null for now.
  this.div_ = null;

  // Explicitly call setMap on this overlay
  this.setMap(map);
}

/**
 * onAdd is called when the map's panes are ready and the overlay has been
 * added to the map.
 */
USGSOverlay.prototype.onAdd = function() {

  var div = document.createElement('div');
  div.style.border = 'none';
  div.style.borderWidth = '0px';
  div.style.position = 'absolute';

  // Create the img element and attach it to the div.
  var img = document.createElement('img');
  img.src = this.image_;
  img.style.width = '100%';
  img.style.height = '100%';
  div.appendChild(img);

  this.div_ = div;

  // Add the element to the "overlayImage" pane.
  var panes = this.getPanes();
  panes.overlayImage.appendChild(this.div_);
};

USGSOverlay.prototype.draw = function() {

  // We use the south-west and north-east
  // coordinates of the overlay to peg it to the correct position and size.
  // To do this, we need to retrieve the projection from the overlay.
  var overlayProjection = this.getProjection();

  // Retrieve the south-west and north-east coordinates of this overlay
  // in LatLngs and convert them to pixel coordinates.
  // We'll use these coordinates to resize the div.
  var sw = overlayProjection.fromLatLngToDivPixel(this.bounds_.getSouthWest());
  var ne = overlayProjection.fromLatLngToDivPixel(this.bounds_.getNorthEast());

  // Resize the image's div to fit the indicated dimensions.
  var div = this.div_;
  div.style.left = sw.x + 'px';
  div.style.top = ne.y + 'px';
  div.style.width = (ne.x - sw.x) + 'px';
  div.style.height = (sw.y - ne.y) + 'px';
};

USGSOverlay.prototype.onRemove = function() {
  this.div_.parentNode.removeChild(this.div_);
};

// Set the visibility to 'hidden' or 'visible'.
USGSOverlay.prototype.hide = function() {
  if (this.div_) {
    // The visibility property must be a string enclosed in quotes.
    this.div_.style.visibility = 'hidden';
  }
};

USGSOverlay.prototype.show = function() {
  if (this.div_) {
    this.div_.style.visibility = 'visible';
  }
};

USGSOverlay.prototype.toggle = function() {
  if (this.div_) {
    if (this.div_.style.visibility == 'hidden') {
      this.show();
    } else {
      this.hide();
    }
  }
};

// Detach the map from the DOM via toggleDOM().
// Note that if we later reattach the map, it will be visible again,
// because the containing <div> is recreated in the overlay's onAdd() method.
USGSOverlay.prototype.toggleDOM = function() {
  if (this.getMap()) {
    // Note: setMap(null) calls OverlayView.onRemove()
    this.setMap(null);
  } else {
    this.setMap(this.map_);
  }
};

google.maps.event.addDomListener(window, 'load', initialize);
```

Last updated March 17, 2015.