Google Developers

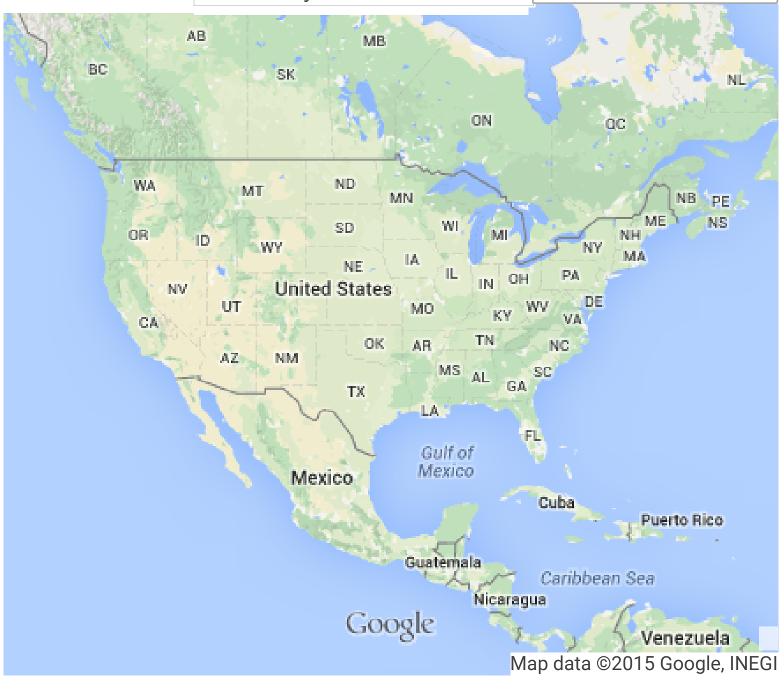
Googl... X Search

Sign in

Products Google Maps API Google Maps JavaScript API v3

Place Autocomplete Hotel Search

Find hotels in: Enter a city U.S.A. ▼



View this example full screen.

JavaScript JavaScript + HTML

```
// This example uses the autocomplete feature of the Google Places API.
// It allows the user to find all hotels in a given place, within a given
// country. It then displays markers for all the hotels returned,
// with on-click details for each hotel.

var map, places, infoWindow;
var markers = [];
var autocomplete;
var countryRestrict = { 'country': 'us' };
var MARKER_PATH = 'https://maps.gstatic.com/intl/en_us/mapfiles/marker_green';
var hostnameRegexp = new RegExp('^https?://.+?/');

var countries = {
  'au': {
    center: new google.maps.LatLng(-25.3, 133.8),
    zoom: 4
  },
  'br': {
    center: new google.maps.LatLng(-14.2, -51.9),
    zoom: 3
  },
  'ca': {
    center: new google.maps.LatLng(62, -110.0),
    zoom: 3
  },
  'fr': {
    center: new google.maps.LatLng(46.2, 2.2),
    zoom: 5
  },
  'de': {
    center: new google.maps.LatLng(51.2, 10.4),
    zoom: 5
  },
  'mx': {
    center: new google.maps.LatLng(23.6, -102.5),
    zoom: 4
  },
  'nz': {
    center: new google.maps.LatLng(-40.9, 174.9),
    zoom: 5
  },
  'it': {
    center: new google.maps.LatLng(41.9, 12.6),
    zoom: 5
  },
  'za': {
    center: new google.maps.LatLng(-30.6, 22.9),
    zoom: 5
  },
  'es': {
    center: new google.maps.LatLng(40.5, -3.7),
    zoom: 5
  },
  'pt': {
    center: new google.maps.LatLng(39.4, -8.2),
    zoom: 6
  },
  'us': {
    center: new google.maps.LatLng(37.1, -95.7),
    zoom: 3
  },
  'uk': {
    center: new google.maps.LatLng(54.8, -4.6),
    zoom: 5
  }
};

function initialize() {
  var myOptions = {
    zoom: countries['us'].zoom,
    center: countries['us'].center,
    mapTypeControl: false,
    panControl: false,
    zoomControl: false,
    streetViewControl: false
  };

  map = new google.maps.Map(document.getElementById('map-canvas'), myOptions);
  infoWindow = new google.maps.InfoWindow({
      content: document.getElementById('info-content')
      });

  // Create the autocomplete object and associate it with the UI input control.
  // Restrict the search to the default country, and to place type "cities".
  autocomplete = new google.maps.places.Autocomplete(
      /** @type {HTMLInputElement} */(document.getElementById('autocomplete')),
      {
        types: ['(cities)'],
        componentRestrictions: countryRestrict
      });
  places = new google.maps.places.PlacesService(map);

  google.maps.event.addListener(autocomplete, 'place_changed', onPlaceChanged);

  // Add a DOM event listener to react when the user selects a country.
  google.maps.event.addDomListener(document.getElementById('country'), 'change',
      setAutocompleteCountry);
}

// When the user selects a city, get the place details for the city and
// zoom the map in on the city.
function onPlaceChanged() {
  var place = autocomplete.getPlace();
  if (place.geometry) {
    map.panTo(place.geometry.location);
    map.setZoom(15);
    search();
  } else {
    document.getElementById('autocomplete').placeholder = 'Enter a city';
  }

}

// Search for hotels in the selected city, within the viewport of the map.
function search() {
  var search = {
    bounds: map.getBounds(),
    types: ['lodging']
  };

  places.nearbySearch(search, function(results, status) {
    if (status == google.maps.places.PlacesServiceStatus.OK) {
      clearResults();
      clearMarkers();
      // Create a marker for each hotel found, and
      // assign a letter of the alphabetic to each marker icon.
      for (var i = 0; i < results.length; i++) {
        var markerLetter = String.fromCharCode('A'.charCodeAt(0) + i);
        var markerIcon = MARKER_PATH + markerLetter + '.png';
        // Use marker animation to drop the icons incrementally on the map.
        markers[i] = new google.maps.Marker({
          position: results[i].geometry.location,
          animation: google.maps.Animation.DROP,
          icon: markerIcon
        });
        // If the user clicks a hotel marker, show the details of that hotel
        // in an info window.
        markers[i].placeResult = results[i];
        google.maps.event.addListener(markers[i], 'click', showInfoWindow);
        setTimeout(dropMarker(i), i * 100);
        addResult(results[i], i);
      }
    }
  });
}

function clearMarkers() {
  for (var i = 0; i < markers.length; i++) {
    if (markers[i]) {
      markers[i].setMap(null);
    }
  }
  markers = [];
}

// Set the country restriction based on user input.
// Also center and zoom the map on the given country.
function setAutocompleteCountry() {
  var country = document.getElementById('country').value;
  if (country == 'all') {
    autocomplete.setComponentRestrictions([]);
    map.setCenter(new google.maps.LatLng(15, 0));
    map.setZoom(2);
  } else {
    autocomplete.setComponentRestrictions({ 'country': country });
    map.setCenter(countries[country].center);
    map.setZoom(countries[country].zoom);
  }
  clearResults();
  clearMarkers();
}

function dropMarker(i) {
  return function() {
    markers[i].setMap(map);
  };
}

function addResult(result, i) {
  var results = document.getElementById('results');
  var markerLetter = String.fromCharCode('A'.charCodeAt(0) + i);
  var markerIcon = MARKER_PATH + markerLetter + '.png';

  var tr = document.createElement('tr');
  tr.style.backgroundColor = (i % 2 == 0 ? '#F0F0F0' : '#FFFFFF');
  tr.onclick = function() {
    google.maps.event.trigger(markers[i], 'click');
  };

  var iconTd = document.createElement('td');
  var nameTd = document.createElement('td');
  var icon = document.createElement('img');
  icon.src = markerIcon;
  icon.setAttribute('class', 'placeIcon');
  icon.setAttribute('className', 'placeIcon');
  var name = document.createTextNode(result.name);
  iconTd.appendChild(icon);
  nameTd.appendChild(name);
  tr.appendChild(iconTd);
  tr.appendChild(nameTd);
  results.appendChild(tr);
}

function clearResults() {
  var results = document.getElementById('results');
  while (results.childNodes[0]) {
    results.removeChild(results.childNodes[0]);
  }
}

// Get the place details for a hotel. Show the information in an info window,
// anchored on the marker for the hotel that the user selected.
function showInfoWindow() {
  var marker = this;
  places.getDetails({placeId: marker.placeResult.place_id},
      function(place, status) {
        if (status != google.maps.places.PlacesServiceStatus.OK) {
          return;
        }
        infoWindow.open(map, marker);
        buildIWContent(place);
      });
}

// Load the place information into the HTML elements used by the info window.
function buildIWContent(place) {
  document.getElementById('iw-icon').innerHTML = '<img class="hotelIcon" ' +
      'src="' + place.icon + '"/>';
  document.getElementById('iw-url').innerHTML = '<b><a href="' + place.url +
      '">' + place.name + '</a></b>';
  document.getElementById('iw-address').textContent = place.vicinity;

  if (place.formatted_phone_number) {
    document.getElementById('iw-phone-row').style.display = '';
    document.getElementById('iw-phone').textContent =
        place.formatted_phone_number;
  } else {
    document.getElementById('iw-phone-row').style.display = 'none';
  }

  // Assign a five-star rating to the hotel, using a black star ('&#10029;')
  // to indicate the rating the hotel has earned, and a white star ('&#10025;')
  // for the rating points not achieved.
  if (place.rating) {
    var ratingHtml = '';
    for (var i = 0; i < 5; i++) {
      if (place.rating < (i + 0.5)) {
        ratingHtml += '&#10025;';
      } else {
        ratingHtml += '&#10029;';
      }
    document.getElementById('iw-rating-row').style.display = '';
    document.getElementById('iw-rating').innerHTML = ratingHtml;
    }
  } else {
    document.getElementById('iw-rating-row').style.display = 'none';
  }

  // The regexp isolates the first part of the URL (domain plus subdomain)
  // to give a short URL for displaying in the info window.
  if (place.website) {
    var fullUrl = place.website;
    var website = hostnameRegexp.exec(place.website);
    if (website == null) {
      website = 'http://' + place.website + '/';
      fullUrl = website;
    }
    document.getElementById('iw-website-row').style.display = '';
    document.getElementById('iw-website').textContent = website;
  } else {
    document.getElementById('iw-website-row').style.display = 'none';
  }
}
```

Last updated March 17, 2015.