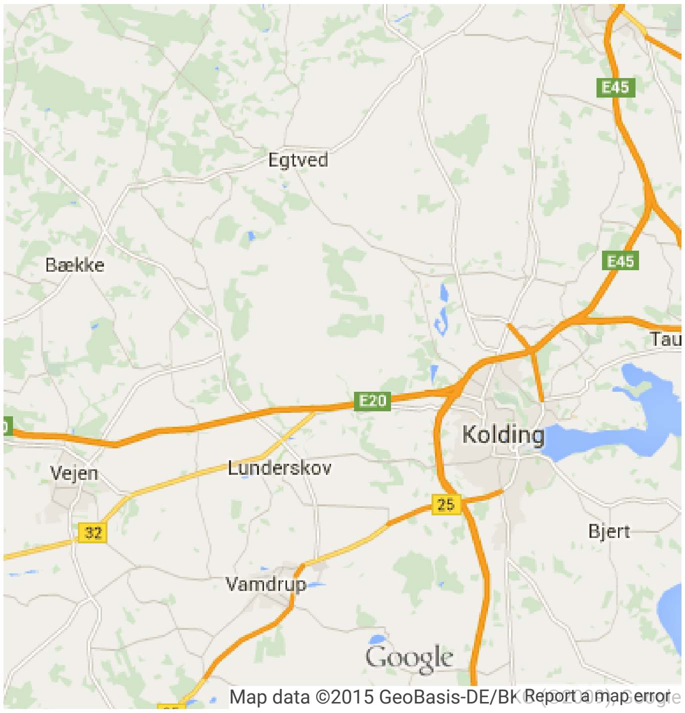
Distance Matrix service



```
var origin1 = new google.maps.LatLng(55.930, -3.118)
var origin2 = 'Greenwich, England';
var destinationA = 'Stockholm, Sweden';
var destinationB = new google.maps.LatLng(50.087, 14
```

Calculate distances

View this example full screen.

JavaScript JavaScript + HTML

```
var map;
var geocoder;
var bounds = new google.maps.LatLngBounds();
var markersArray = [];

var origin1 = new google.maps.LatLng(55.930, -3.118);
var origin2 = 'Greenwich, England';
var destinationA = 'Stockholm, Sweden';
var destinationB = new google.maps.LatLng(50.087, 14.421);

var destinationIcon = 'https://chart.googleapis.com/chart?chst=d_map_pin_letter&chld=D|FF0000|000000'
;
var originIcon = 'https://chart.googleapis.com/chart?chst=d_map_pin_letter&chld=O|FFFF00|000000';

function initialize() {
  var opts = {
    center: new google.maps.LatLng(55.53, 9.4),
    zoom: 10
  };
  map = new google.maps.Map(document.getElementById('map-canvas'), opts);
  geocoder = new google.maps.Geocoder();
}

function calculateDistances() {
  var service = new google.maps.DistanceMatrixService();
  service.getDistanceMatrix(
    {
      origins: [origin1, origin2],
      destinations: [destinationA, destinationB],
      travelMode: google.maps.TravelMode.DRIVING,
      unitSystem: google.maps.UnitSystem.METRIC,
      avoidHighways: false,
      avoidTolls: false
    }, callback);
}

function callback(response, status) {
  if (status != google.maps.DistanceMatrixStatus.OK) {
    alert('Error was: ' + status);
  } else {
    var origins = response.originAddresses;
    var destinations = response.destinationAddresses;
    var outputDiv = document.getElementById('outputDiv');
    outputDiv.innerHTML = '';
    deleteOverlays();

    for (var i = 0; i < origins.length; i++) {
      var results = response.rows[i].elements;
      addMarker(origins[i], false);
      for (var j = 0; j < results.length; j++) {
        addMarker(destinations[j], true);
        outputDiv.innerHTML += origins[i] + ' to ' + destinations[j]
            + ': ' + results[j].distance.text + ' in '
            + results[j].duration.text + '<br>';
      }
    }
  }
}

function addMarker(location, isDestination) {
  var icon;
  if (isDestination) {
    icon = destinationIcon;
  } else {
    icon = originIcon;
  }
  geocoder.geocode({'address': location}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
      bounds.extend(results[0].geometry.location);
      map.fitBounds(bounds);
      var marker = new google.maps.Marker({
        map: map,
        position: results[0].geometry.location,
        icon: icon
      });
      markersArray.push(marker);
    } else {
      alert('Geocode was not successful for the following reason: '
        + status);
    }
  });
}

function deleteOverlays() {
  for (var i = 0; i < markersArray.length; i++) {
    markersArray[i].setMap(null);
  }
  markersArray = [];
}

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

Last updated March 17, 2015.