<html ng-app="EmployeeModule">

<head>

<script src="js/jquery-3.0.0.js"></script>

<script>

$(document).ready(function(){

$("#getData").click(function(){

});

$.ajax({

   url: '<https://maps.googleapis.com/maps/api/geocode/json>',

   data: {

      address: 'Guindy Chennai'

   },

   error: function() {

      $('result').html('<p>An error has occurred</p>');

   },

   dataType: 'jsonp',

   success: function(data) {

    $("result").html(data);

   },

   type: 'GET'

});

});

</script>

</head>

<body>

<input type="button" value="GetData" id="getData"/>

<div id="result"></div>

</body>

</html>

This routine calculates the distance between two points (given the :\*/

/\*:: latitude/longitude of those points). It is being used to calculate :\*/

/\*:: the distance between two locations using GeoDataSource (TM) prodducts :\*/

/\*:: :\*/

/\*:: Definitions: :\*/

/\*:: South latitudes are negative, east longitudes are positive :\*/

/\*:: :\*/

/\*:: Passed to function: :\*/

/\*:: lat1, lon1 = Latitude and Longitude of point 1 (in decimal degrees) :\*/

/\*:: lat2, lon2 = Latitude and Longitude of point 2 (in decimal degrees) :\*/

/\*:: unit = the unit you desire for results :\*/

/\*:: where: 'M' is statute miles (default) :\*/

/\*:: 'K' is kilometers :\*/

/\*:: 'N' is nautical miles :\*/

/\*:: Worldwide cities and other features databases with latitude longitude :\*/

/\*:: are available at http://www.geodatasource.com :\*/

/\*:: :\*/

/\*:: For enquiries, please contact sales@geodatasource.com :\*/

/\*:: :\*/

/\*:: Official Web site: http://www.geodatasource.com :\*/

/\*:: :\*/

/\*:: GeoDataSource.com (C) All Rights Reserved 2015 :\*/

/\*:: :\*/

/\*::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\*/

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class DistanceCalculator

{

public static void main (String[] args) throws java.lang.Exception

{

System.out.println(distance(32.9697, -96.80322, 29.46786, -98.53506, "M") + " Miles\n");

System.out.println(distance(32.9697, -96.80322, 29.46786, -98.53506, "K") + " Kilometers\n");

System.out.println(distance(32.9697, -96.80322, 29.46786, -98.53506, "N") + " Nautical Miles\n");

}

private static double distance(double lat1, double lon1, double lat2, double lon2, String unit) {

double theta = lon1 - lon2;

double dist = Math.sin(deg2rad(lat1)) \* Math.sin(deg2rad(lat2)) + Math.cos(deg2rad(lat1)) \* Math.cos(deg2rad(lat2)) \* Math.cos(deg2rad(theta));

dist = Math.acos(dist);

dist = rad2deg(dist);

dist = dist \* 60 \* 1.1515;

if (unit == "K") {

dist = dist \* 1.609344;

} else if (unit == "N") {

dist = dist \* 0.8684;

}

return (dist);

}

/\*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\*/

/\*:: This function converts decimal degrees to radians :\*/

/\*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\*/

private static double deg2rad(double deg) {

return (deg \* Math.PI / 180.0);

}

/\*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\*/

/\*:: This function converts radians to decimal degrees :\*/

/\*:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::\*/

private static double rad2deg(double rad) {

return (rad \* 180 / Math.PI);

}

}