ITS 485 Lab: 2 Leaflet Routing Machine

Objective

- To get familiar with routing engines with real world map
- To understand how to make routing by Leaflet Routing Library based on OSRM
- To learn how to adapt this method with real world project

Introduction

Leaflet Routing Machine is an easy, flexible and extensible way to add routing to a Leaflet map. Using the default is just a few lines of code to add fully functional routing, but you can still customize almost every aspect of the user interface and interactions [1].

The leaflet routing machine has many features such as leaflet control, add/edit way point, routing from sources and destination. The routing part can be based on many routing. For example, OSRM (Open Source Routing Machine), Map Box Direction API, Graph Hopper, Mapzen Valhalla, and TomTom Online Routing API [1].

In this case, we going to use OSRM as our routing based engine. To get started, the template file for this lab is in "Lab2-Leaflet-OSRM" folder. The method of this template can be divide into 6 steps which are:

First Step: Initialize leaflet and leaflet routing machine library

The code will be modified to initialize leaflet and leaflet routing machine library script as the following code. The highlighted texts are depended on your machine local path.

```
<html>
   <head>
      <title>Leaflet Routing Machine Lab</title>
         <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
         <link rel="stylesheet" href="./dist/leaflet-routing-machine.css" />
         <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
         <script src="./dist/leaflet-routing-machine.js"></script>
   </head>
<body>
</body>
</html>
```

Second Step: Set up div element and map style

- add div element with certain ID where you want to put the map

```
<div id="map" class="map"></div>
```

- add style to map. The map ID must be the same as define above unless the style will not appear.

```
<style>
      #map {
         width: 100%;
         height: 100%;
      }
</style>
```

Example of the code after adding library and map

```
<html>
<head>
      <title>Leaflet Routing Machine Lab</title>
   <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
      <link rel="stylesheet" href="./dist/leaflet-routing-machine.css" />
      <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
   <script src="./dist/leaflet-routing-machine.js"></script>

   <style>
      .map {
         position: absolute;
         width: 100%;
         height: 100%;
      }
   </style>

</head>
<body>
   <div id="map" class="map"></div>
</body>
</html>
```

Third Step: Initialize the map and routing control

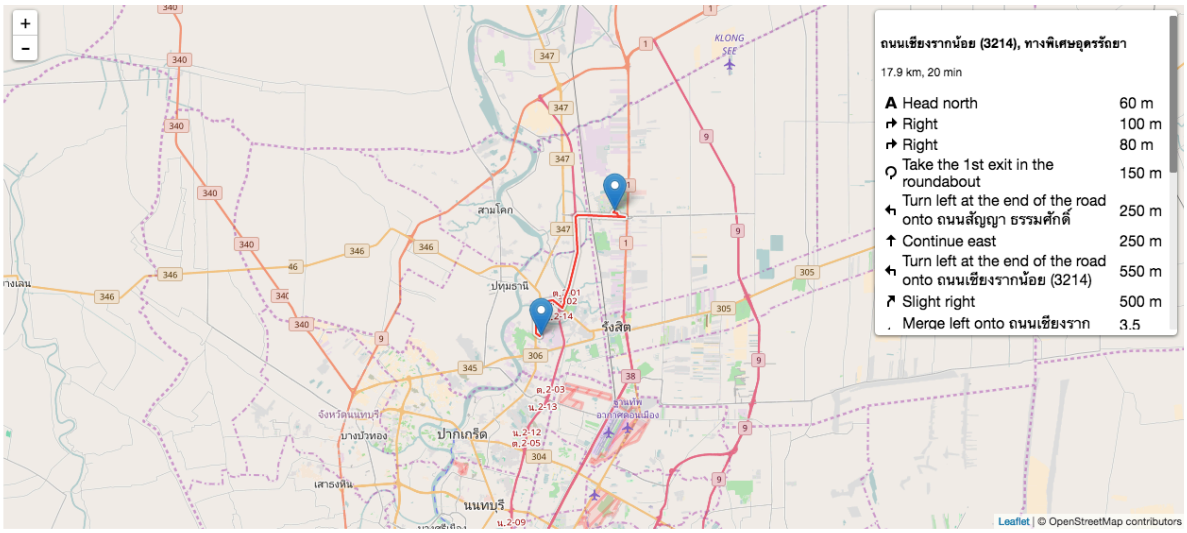
- Defining variable of the map with ID “map”, then add tile layer to it.

```
var map = L.map('map').setView([13.751908, 100.539029], 13);
//initialize tile layer and add it to map
   L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
   attribution: '© OpenStreetMap contributors'
   }).addTo(map);
```

- **Add routing control to map**
  - **a.** The way point is used to initialize the starting and ending point at first when load the map
  - **b.** RoutingWhileDragging means you can alter the location of a marker to make new routing direction.
  - **c.** “serverURL” is use to set your routing engine server. For another server setting, please take a look at http://www.liedman.net/leaflet-routing-machine/tutorials/alternative-routers/ for more detail.

```
var control =  L.Routing.control({
  waypoints: [
     L.latLng(14.068860, 100.607822), //starting point
     L.latLng(13.980708, 100.554199) //end point
  ],
  serviceUrl: ' http://203.131.209.91:5000/route/v1',
  routeWhileDragging: true
   }).addTo(map);
```

After modify the code, the result is shown as follow:



Forth Step: Address and geocoder

- Add geocoder script and style to your code in the <head> section

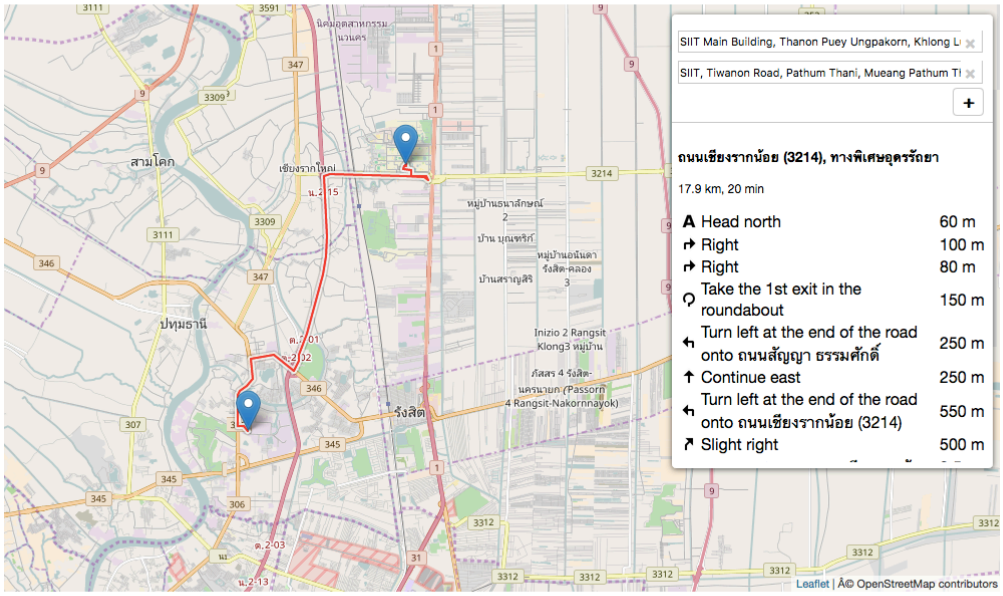
```
<link rel="stylesheet" href="./dist/Control.Geocoder.css" />

<script src="./dist/Control.Geocoder.js"></script>
```

- modify the code of step 3 to be as follow.

By adding "geocoder:" is the place where you specify a geocoder provider. In this lab, we used "nominatim", it is a free service geocoding provider. There is another service provider such Bing, Google, etc. but you need to have a token to get access to it [4].

```
var control = L.Routing.control({
  waypoints: [
     L.latLng(14.068860, 100.607822), //starting point
     L.latLng(13.980708, 100.554199) //end point
  ],
  serviceUrl: ' http://203.131.209.91:5000/route/v1',
  routeWhileDragging: true,
  geocoder: L.Control.Geocoder.nominatim()
   }).addTo(map);
```

After modifying the code, the result is shown as follow:



Fifth Step: Interaction

- Adding new way point by click on the map
    a. Add pop up event when the map is clicked

```
function createButton(label, container) {
    var btn = L.DomUtil.create('button', '', container);
    btn.setAttribute('type', 'button');
    btn.innerHTML = label;
    return btn;
}

map.on('click', function(e) {
    var container = L.DomUtil.create('div'),
        startBtn = createButton('Start from this location', container),
        destBtn = createButton('Go to this location', container);

    L.popup()
        .setContent(container)
        .setLatLng(e.latlng)
        .openOn(map);
});
```

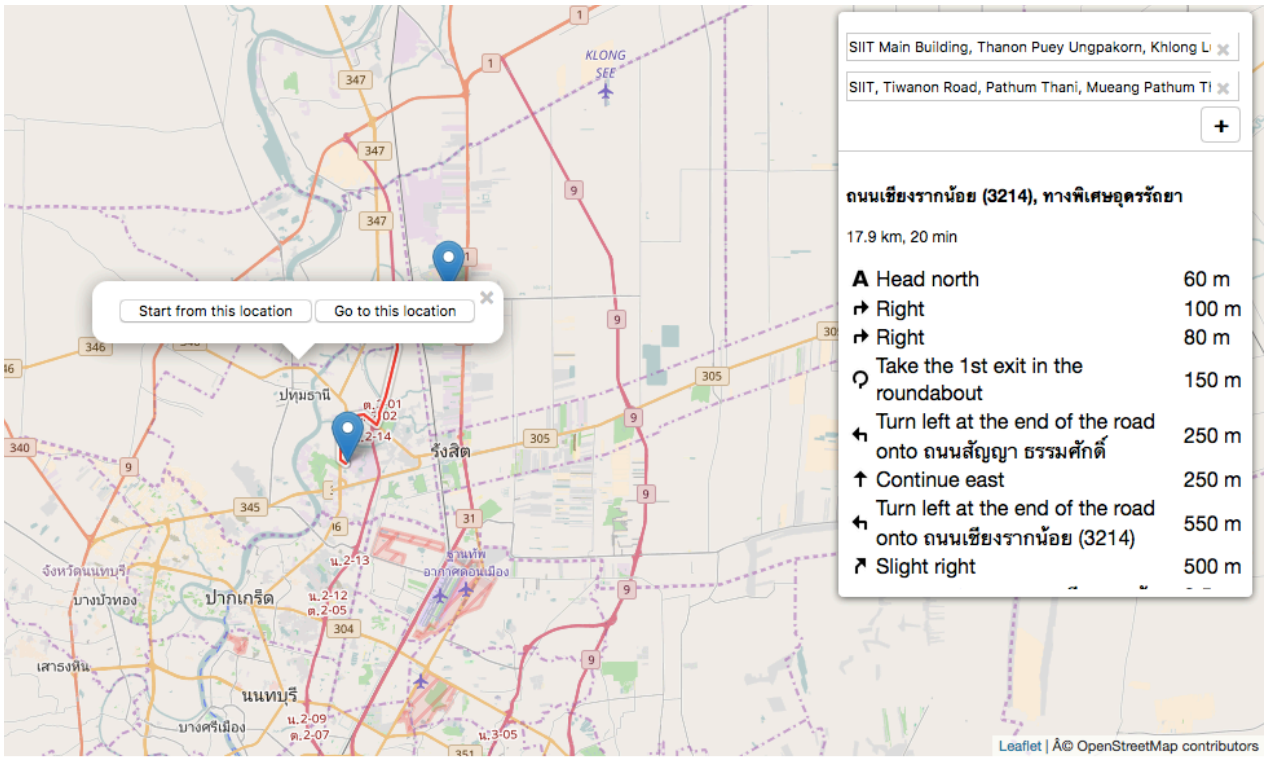
- Add event to the button of a pop up when it has clicked
    a. The starting Button (Replace the current starting point with the new location)

```
L.DomEvent.on(startBtn, 'click', function() {
        control.spliceWaypoints(0, 1, e.latlng);
        map.closePopup();
    });
```

    b. Destination Button (Replace the current destination point with the new location)

```
 L.DomEvent.on(destBtn, 'click', function() {
        control.spliceWaypoints(control.getWaypoints().length - 1, 1, e.latlng);
        map.closePopup();
     });
```

The result should be as follow:



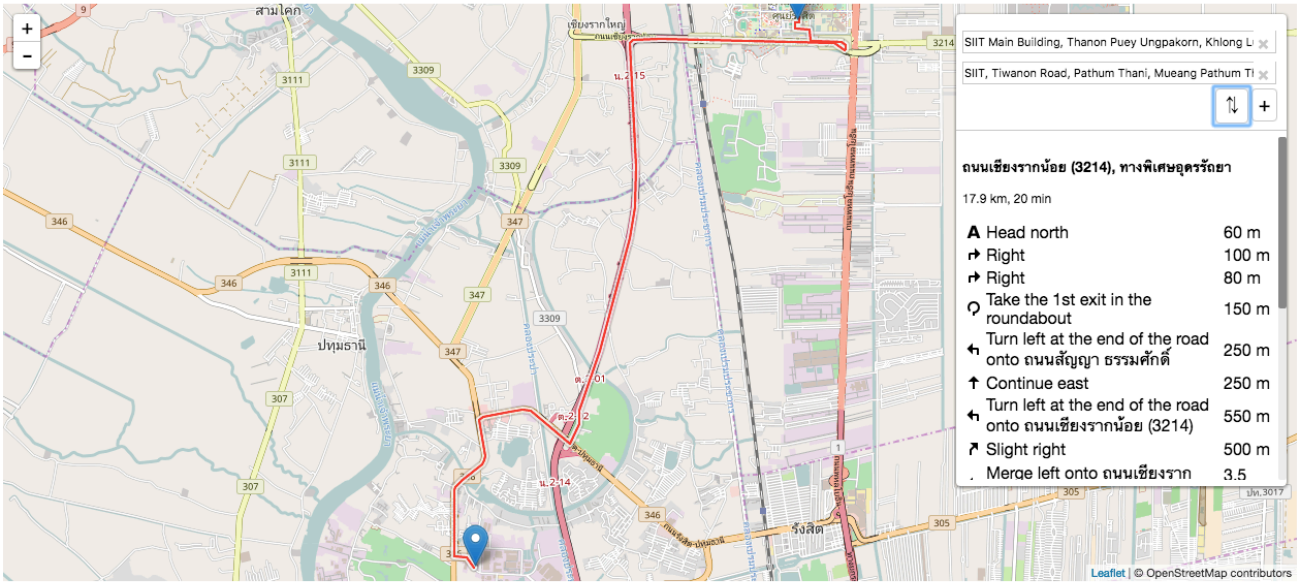
Sixth Step: Interaction (Reversing the route)

In order to make routing to be in both direction, we need to enable the "reverseWaypoints" function to make the route reversible. The procedure is as follow:

- we need to modify the code of step 3 then there will be a button for reverse the routing direction as following code.

```
……
var control = L.Routing.control({
    waypoints: [
        L.latLng(14.068860, 100.607822), //starting point
        L.latLng(13.980708, 100.554199) //end point
    ],
    serviceUrl: ' http://203.131.209.91:5000/route/v1',
    routeWhileDragging: true,
    reverseWaypoints: true,
    geocoder: L.Control.Geocoder.nominatim()

}).addTo(map)
```

The result shown as follow:



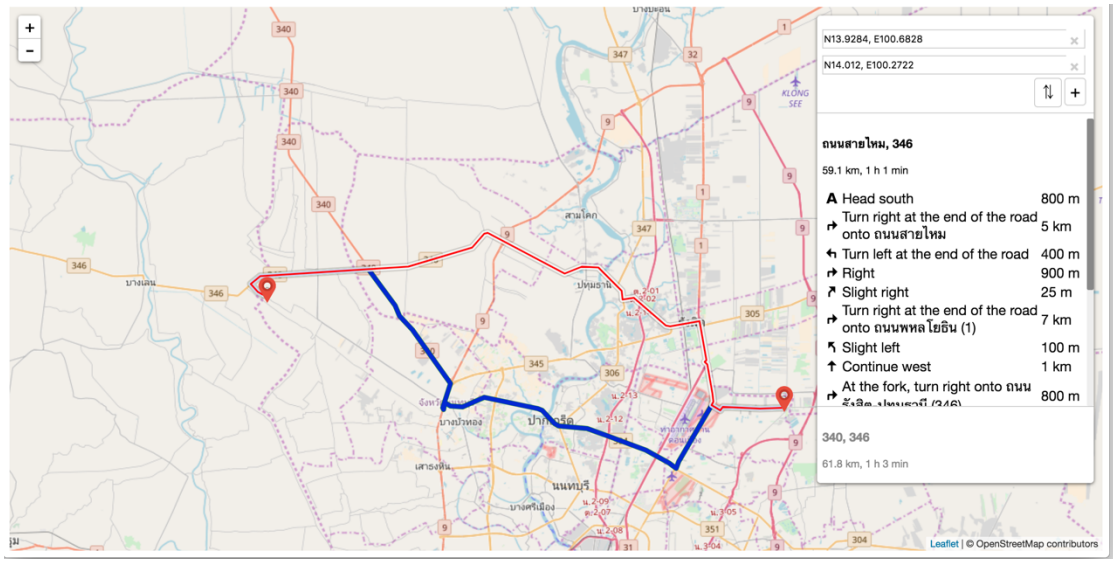
Exercise

1. Modify the code to make the program show alternative routing on the screen
   a. Hint: Use "showAlternatives" [2] option
2. Modify the code to make re-routing when screen has zoom in and out
   a. Hint: Use "useZoomParameter" [2] option
3. Try to modify the code in step 3 and customize marker icon, polyline transparent, color, thickness……etc.
4. Try to get summary Distance and Time from the service (The initial unit is meter for "distance" and second for "Time")

Hint for Exercise 3:

```
…….
CreateMarker: function(i,wp){
        return L.marker(wp.latLng,{
                draggable: true,
                icon: L.icon({……..})
                })
        ]);
},
……..
   showAlternatives: true,
   altLineOption:{
        styles: [{color: 'blue', opacity: 0.15, weight: 9}]
}
```

Example from code above:



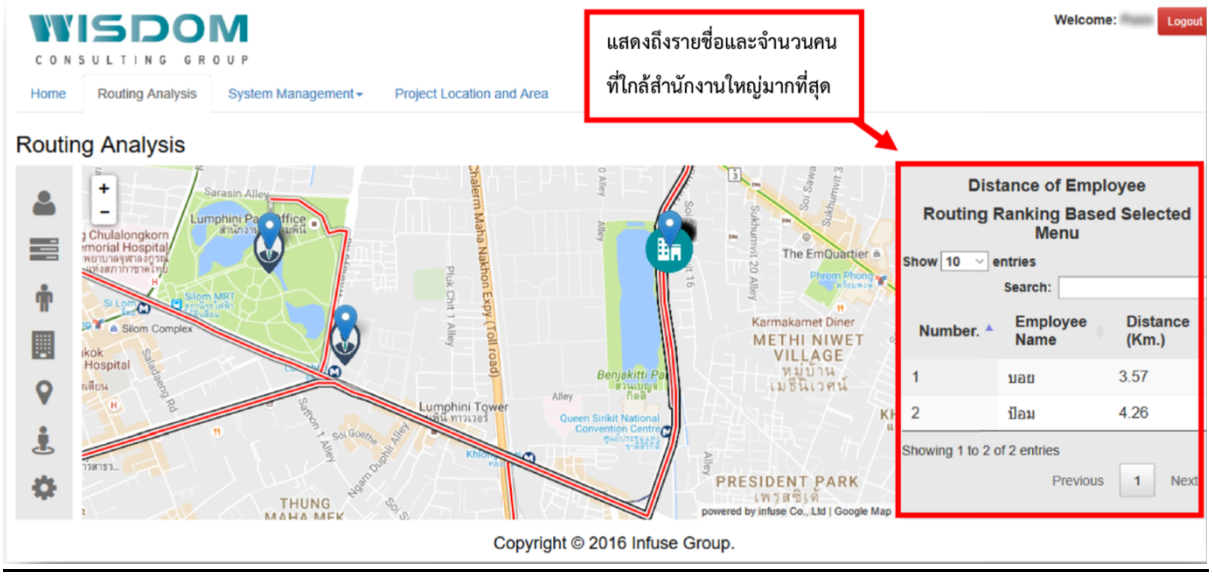
Hint for Exercise 4:

```
……..

control.on('routesfound', function(e){
        var distance = e.routes[0].summary.totalDistance;
        var time = e.routes[0].summary.totalTime;
        ……..
});

……..
```

Assignment (Due date: October 17, 2017)

Create a web application which calculate routing from your current location to your friend location by using procedure from step 1 to step 6. **Hint:** May make use of leaflet's marker "onClick" to call routing function for each pair of Origin and Destination and code from Exercise 4.

Example of web application which apply from "Leaflet Routing Machine"



Reference

1. http://www.liedman.net/leaflet-routing-machine/#about
2. http://www.liedman.net/leaflet-routing-machine/api/
3. http://www.liedman.net/leaflet-routing-machine/tutorials/
4. http://www.liedman.net/leaflet-routing-machine/tutorials/geocoders/