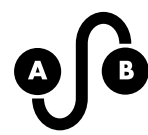


# Leaflet Routing Machine



Created by [Alexey Ivanov](#)  
from the [Neon Project](#)

Easy, flexible routing for Leaflet

[« Back to tutorials](#)

## Basics

This tutorial will show you how to download, install and get started with using Leaflet Routing Machine.

## Downloading and installing

You can [download a ZIP](#) containing the necessary files, or you can install it with NPM if you plan to bundle your scripts with Browserify or a similar tool

```
npm install --save leaflet-routing-machine
```

Once you have the files, you need to include three files so that they are accessible from your page:

- `leaflet-routing-machine.js` - this is the JavaScript containing the plugin code. It needs to be loaded *after* Leaflet itself has been loaded, since it relies on classes in Leaflet.
- `leaflet-routing-machine.css` - contains the stylesheet for the default look of the plugin.
- `leaflet.routing.icons.png` - sprites that contain the icons used to give turn directions in the itinerary.

## Including the plugin in a page

The minimal HTML to create a page with Leaflet Routing Machine looks pretty much like the minimal Leaflet map page, but with the stylesheet and code mentioned above included in it:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Leaflet Routing Machine Example</title>
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.2.0/dist/leaflet.css" />
  <link rel="stylesheet" href="leaflet-routing-machine.css" />
  <style>
    .map {
      position: absolute;
      width: 100%;
      height: 100%;
    }
  </style>
</head>
<body>
  <div id="map" class="map"></div>
  <script src="https://unpkg.com/leaflet@1.2.0/dist/leaflet.js"></script>
  <script src="leaflet-routing-machine.js"></script>
</body>
</html>
```

# Initialize the plugin

The main class of the plugin is [L.Routing.Control](#) . As you might guess from the name, it is a control, and by default it will add a panel on the right side of the map that will contain the itinerary of the route. As with any Leaflet control, the location of the control's interface can be changed by using the option [position](#) .

In its most basic form, the user will have no means of actually adding waypoints to the route. This means you will have to provide the initial waypoints to the control <sup>1</sup>.

```
var map = L.map('map');

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '@ OpenStreetMap contributors'
}).addTo(map);

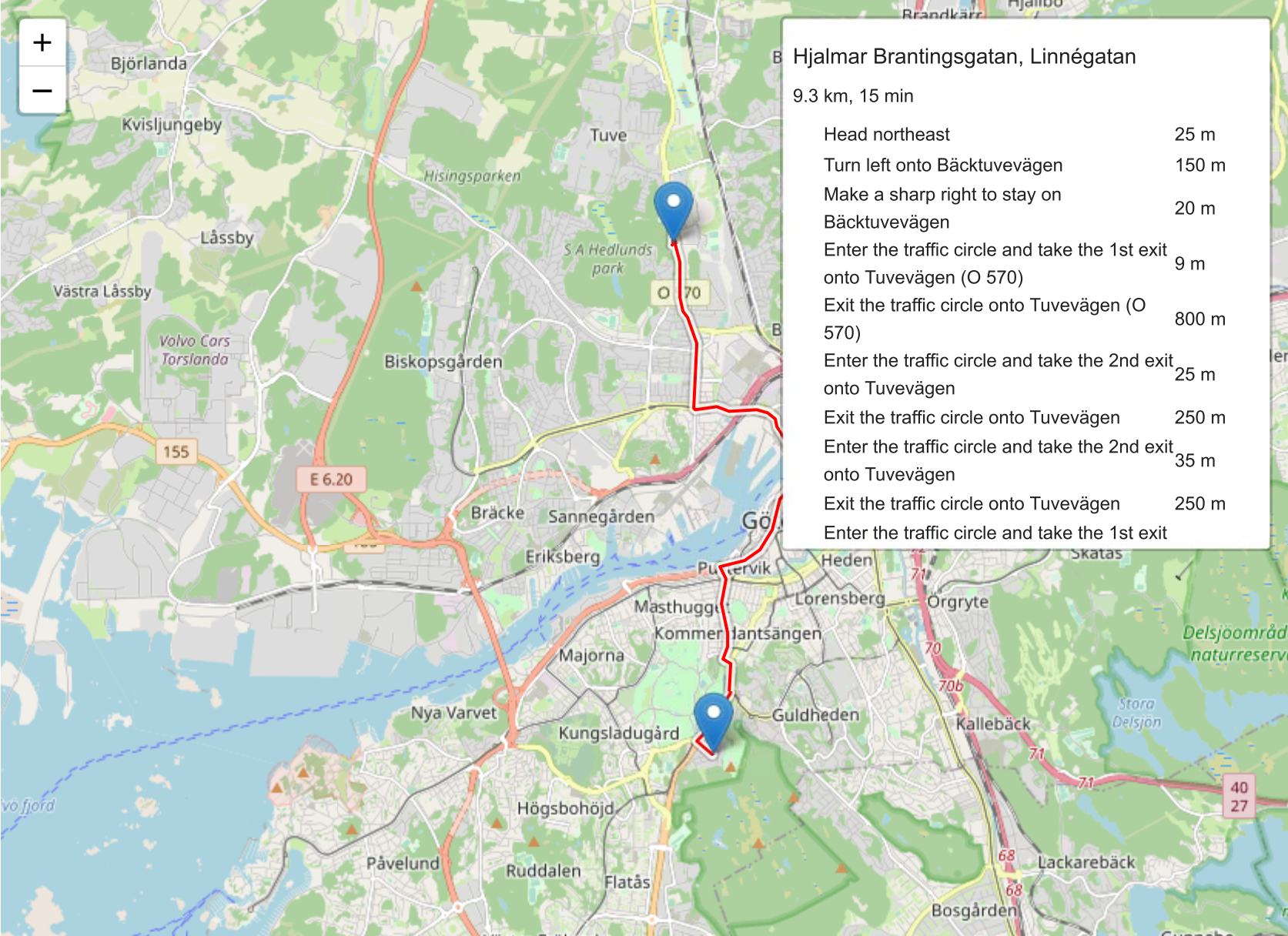
L.Routing.control({
  waypoints: [
    L.latLng(57.74, 11.94),
    L.latLng(57.6792, 11.949)
  ],
  routeWhileDragging: true
}).addTo(map);
```

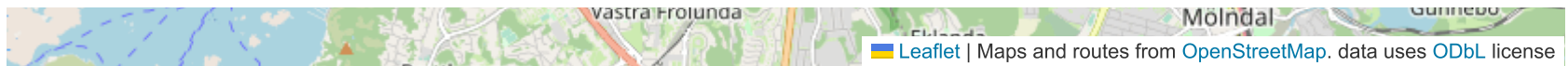
Lines 1 to 5 contain the normal Leaflet initialization and setting up OpenStreetMap's tile layer.

Lines 7 to 13 initializes the control, and adds it to the map. As can be seen, the only argument when creating the control is a set of options. Let's go through the used options and explain them one by one:

- `waypoints` - defines which locations the route must visit. At least two waypoints, the starting point and the destination, must be provided to be able to create a route. A waypoint can either be simply an `L.LatLng` object, or an [L.Routing.Waypoint](#) object for more advanced usage.
- `routeWhileDragging` - allows the plugin to recalculate the route while a waypoint is dragged on the map. This consumes more network bandwidth and load on the route calculation backend, but gives nice feedback to the user on reasonably fast networks.

The result should be a fullscreen version of this <sup>2</sup>:





# Next steps

This is the most basic setup for Leaflet Routing Machine. Continue with one of the other tutorials, for example [geocoders](#) to add support for address lookup, or [interaction](#) for some examples on how to connect map interactions to the route plugin.

For TypeScript users, there are [definitions for Leaflet Routing Machine](#) provided by Chanaka Rathnayaka.

---

<sup>1</sup> While this might seem awkward, it is for good reason: any real application will want to customize how the user interacts with the map; a common method is to bring up a popup when the map is clicked, from which the user can choose the use the clicked location as start or destination. Also, by default Leaflet Routing Machine cannot lookup addresses (geocode), since there are too many different services that provide this functionality, and the plugin shouldn't decide which ones to use.

<sup>2</sup> In my examples, I use maps from [Mapbox](#) instead of the default OpenStreetMap rendering. They look much better, in my opinion, and comes very reasonably priced. Also, Mapbox does a lot of good work for both OpenStreetMap and open source, so I feel pretty good about giving them a small amount of my money.

---

Copyright © 2015 Per Liedman, released under [ISC License](#). Logo by [Alexey Ivanov](#), released under Creative Commons CC-BY 3.0.