



# TomTom JavaScript SDK - Getting Started

For version 2.0

## Introduction

The TomTom JavaScript SDK makes it simple to embed interactive maps in any web based application. It is based on an OpenSource mapping framework called Leaflet. More information, including examples and plugins, can be found at <http://leafletjs.com/>. This guide will quickly show you how to start using the TomTom JavaScript SDK.

## Including Files

To start using the JavaScript SDK, you will need to include a bit of CSS and JavaScript.

```
<link rel="stylesheet" type="text/css" href="css/tomtom.map.min.css" />
<!--[if lte IE 8]><link rel="stylesheet" href="css/tomtom.map.ie.min.css" /><![endif]-->
<script type="text/javascript" src="js/leaflet.js"></script>
<script type="text/javascript" src="js/tomtom.map.js"></script>
```

These tags should be placed within the `<head>` tag of your page. All of the files can be found in the TomTom JavaScript SDK zip file.

You'll notice that there is a conditional stylesheet being included for IE 7/8. This stylesheet is a part of Leaflet.

## Displaying a map

Displaying a basic map is extremely simple and can be accomplished with just a few lines of JavaScript:

```
// wait for all of the page resources to load
window.onload = function() {

    // create the map
    var map = new tomtom.Map({
        domNode: "map",
        // make sure to fill in your API key!
        apiKey: "YOUR API KEY HERE"
    });

};
```

Also make sure you've created an element on your page for the map to be added to. Note in the above

JavaScript the domNode property passed to the tomtom.Map constructor. This is specifying the ID of the node the map should be added to. Also, fill in the API key you received from <http://developer.tomtom.com/>.

```
<div id="map" style="width: 640px; height: 480px;"><div>
```

## Adding Markers

Adding a default styled marker can be added using the addMarker method on the Map.

```
map.addLayer(new tomtom.Marker([43.67863, -72.26259]));
```

## Displaying Traffic

TomTom's traffic can be displayed using the JavaScript SDK by specifying an additional property in the map constructor, or by calling setDisplayTraffic on the map:

```
window.onload = function() {  
  
    // displaying traffic via the constructor  
    var map = new tomtom.Map({  
        domNode: "map",  
        apiKey: "YOUR API KEY HERE",  
        center: [40.71305, -74.00723],  
        zoom: 12,  
        displayTraffic: true  
    });  
  
    // displaying traffic using setDisplayTraffic  
    map.setDisplayTraffic(true);  
  
};
```

## Adding a route

The JavaScript SDK provides easy routing capabilities via the displayRoute method on the tomtom.Map class. A mixed list of points or addresses can be passed to the displayRoute method to display a route.

```
map.displayRoute([  
    [43.67863, -72.26259],  
    "Boston, MA"  
]);
```

## That's it!

We've covered the basics of building maps with the TomTom JavaScript SDK. For more information, view

the SDK documentation (which is in the documentation folder of the SDK zip file) and check out the examples (in the examples folder of course!).