



TomTom JavaScript SDK - Migrating from 1.x

For version 2.0

Migrating from 1.x

The JavaScript SDK v2.0 has removed OpenLayers as the underlying mapping framework. It has been replaced by leaflet, which means that some method signatures have changed slightly. More information about leaflet, including examples and plugins, can be found at <http://leafletjs.com/>. This guide will quickly show you how to modify your code to work with the new version of the SDK.

Leaflet LatLng object

The leaflet LatLng object replaces OpenLayers.LonLat. The LatLng object is actually much easier to use. Basically anywhere in the SDK that expects a Latitude and Longitude, you can now pass a LatLng object or an array. Take the tomtom.Marker constructor for example:

```
var marker = new tomtom.Marker([42.332916, -83.047853]);
```

Notice that the coordinates are expected latitude first, then longitude, which is the opposite of the way it was done in 1.x.

Map Projections

Leaflet assumes all coordinates are in EPSG4326. This means you will need to convert anything that is in another coordinate system. Leaflet can transform EPSG3857 (EPSG900913).

To transform from EPSG4326 to EPSG3857:

```
var latlng = new L.LatLng(42.332916, -83.047853);  
var point = L.CRS.EPSG3857.project(latlng);
```

To transform from EPSG3857 to EPSG4326, it takes a little more work:

```
var point = new L.Point(-9244844.707434637, 5210979.837890326);  
var earthRadius = 6378137;  
var latlng = L.Projection.SphericalMercator.unproject(point.divideBy(earthRadius));
```

Due to the minor inconvenience with unprojecting EPSG3857, we've put in a pull request with Leaflet to add a more convenient method. If you're interested, you can follow the progress of the [pull request](#).

Map methods

The basic methods to manipulate the map have changed slightly.

```
// change the map view (center and zoom)
map.setView([42.332916, -83.047853], 12);

// change the center
map.panTo([42.332916, -83.047853]);

// change the zoom level
map.setZoom(12);

// get the bounds
map.getBounds();

// fit the specified map bounds
map.fitBounds([[42.332916, -83.047853], [43.651893, -79.381713]]);
```

Adding and removing markers

The addMarker and removeMarker methods have been removed. Leaflet's Markers are considered a layer, and can be added with the addLayer and removed with the removeLayer methods.

```
var marker = new tomtom.Marker([42.332916, -83.047853]);

// add the marker
map.addLayer(marker);

// remove the marker
map.removeLayer(marker);
```

Moving markers

The method used to move a marker has also been changed. Now, the setLatLng method can be used, which will move the marker.

```
var marker = new tomtom.Marker([42.332916, -83.047853]);

// move the marker
marker.setLatLng([43.651893, -79.381713]);
```

Adding a popup to a marker

Popups have been greatly simplified in the latest SDK. Adding a popup can be done using Leaflet's `bindPopup` method on the marker.

```
var marker = new tomtom.Marker([42.332916, -83.047853]);  
  
// add a popup  
marker.bindPopup("Hello Detroit!");
```

Marker options

Customizing marker options has been simplified as well. See <http://leafletjs.com/reference.html#icon-options> for more details.

```
var iconOptions = {  
    iconUrl: "photo.png",  
    iconSize: [32, 37],  
    iconAnchor: [ 17, 34 ]  
};  
map.addLayer(new tomtom.Marker([43.651893, -79.381713], iconOptions));
```