



## Migration Guide

### Getting Started

The MapQuest JavaScript SDK v7 and MapQuest.js both require a key. The same key will work in both, as well as all of the APIs. Get a key free by signing up for an account on our Developer Network.

It is recommended to set the doctype of the page and each SDK requires a div for the map. A basic map ready html page may look like this.

```
<!DOCTYPE html>
<html>
  <head>
    <title>MapQuest.js sdk</title>
    <script src="https://api.mqcdn.com/sdk/
mapquest-js/v1.0.0/mapquest.js"></script>
    <link type="text/css" rel="stylesheet"
href="https://api.mqcdn.com/sdk/mapquest-js/v1.0.0/
mapquest.css"/>
    <script type="text/javascript">
      window.onload = function() {
        // map magic goes here
      }
    </script>
    <style>
      #map {
        width: 800px;
        height: 600px;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

### JavaScript (and style) Include Tag(s)

Both SDKs require a JavaScript download before creating a map. MapQuest JavaScript SDK v7 required the key in the download tag but MapQuest.js does not.

#### JSv7

```
<script src="http://www.mapquestapi.com/sdk/js/v7.2.s/mqa.
toolkit.js?key=KEY"></script>
```

#### MapQuest.js

The MapQuest.js version includes both a js and css file.

```
<script src="https://api.mqcdn.com/sdk/mapquest-js/v1.0.0/
mapquest.js"></script>
```

```
<link type="text/css" rel="stylesheet" href="https://api.mqcdn.
com/sdk/mapquest-js/v1.0.0/mapquest.css"/>
```

### Map Constructor

MapQuest JavaScript SDK v7 required a key in the download tag but MapQuest.js requires the key to be set after the download before mapping functionality is used.

#### JSv7

```
var map = new MQA.TileMap({
  elt: document.getElementById('map'),
  latLng: [lat: 39.738453, lng: -104.984853],
  zoom: 10
});
```

#### MapQuest.js

The MapQuest.js version includes both a js and css file.

```
L.mapquest.key = 'KEY';
var map = L.mapquest.map('map', {
  center: [39.738453, -104.984853],
  layers: L.mapquest.tileLayer('map'),
  zoom: 10
});
```

### Controls

Adding controls to the map are a little different process. MapQuest JavaScript SDK v7 maps require the zoom control be put on specifically and all available layers are shown in the view control. A control like the one found on MapQuest.com can easily be added to the MapQuest.js map.

## MapQuest JavaScript SDK

```
map.addControl(new MQA.LargeZoom());
map.addControl(new MQA.ViewOptions());
```

## MapQuest.js

```
map.addControl(L.mapquest.control());
```

## Traffic Control

### MapQuest JavaScript SDK

```
MQA.withModule('traffictoggle', 'mousewheel', function() {
    map.addControl(new MQA.TrafficToggle());
});
```

## MapQuest.js

```
map.addControl(L.mapquest.control());
```

## More Controls

MapQuest.js can also show Leaflet style controls. The zoom control will appear by default if the MapQuest.com style control is not used. A layer control can also be added. The MapQuest.com style controls can not be mixed with the Leaflet style controls.

## MapQuest.js

```
var baseLayer = L.mapquest.tileLayer('map');
var map = L.mapquest.map('map', {
    center: [39.738453, -104.984853],
    layers: baseLayer,
    zoom: 10
});
L.control.layers({
    'Map': baseLayer,
    'Hybrid': L.mapquest.tileLayer('hybrid'),
    'Satellite': L.mapquest.tileLayer('satellite'),
    'Light': L.mapquest.tileLayer('light'),
    'Dark': L.mapquest.tileLayer('dark')
}).addTo(map);
```

## Markers, Popups, Rectangles

You can still put markers on the map and use custom icons, but the code is a little different.

## MapQuest JavaScript SDK

```
map.addShape(new MQA.Poi({lat: 39.738453, lng:
-104.984853}));
```

## MapQuest.js

```
map.addLayer(L.marker([39.738453, -104.984853]));
```

## Custom Icon

### MapQuest JavaScript SDK

```
custom = new MQA.Poi({lat: 39.738453, lng: -104.984853});
icon = new MQA.Icon('http://www.mapquestapi.com/
staticmap/geticon?uri=poi-blue.png', 20, 29);
custom.setIcon(icon);
map.addShape(custom);
```

## MapQuest.js

```
icon = L.Icon({
    iconSize: [20, 29],
    iconUrl: 'http://www.mapquestapi.com/staticmap/
geticon?uri=poi-blue.png'
});
map.addLayer(L.marker([39.738453, -104.984853],{icon: icon}));
```

Or even easier with an icon from MapQuest's icon api.

```
L.marker([39.7392, -104.9903], {
    icon: L.mapquest.icons.marker()
}).addTo(map);
```

## Information Window

MapQuest's POI information window was a two stage process (rollover and click) and Leaflet's popup is a single stage (click) and can be connected to just about anything on the map.

### MapQuest JavaScript SDK

```
var info = new MQA.Poi({lat: 39.743943, lng: -105.020089});
info.setRolloverContent('Sports Authority Field at Mile High');
info.setInfoContentHTML('Home of the Denver Broncos');
map.addShape(info);
```

## MapQuest.js

```
map.addLayer(L.marker([39.743943, -105.020089]).
bindPopup("Sports Authority Field at Mile High, Home of the
Denver Broncos"));
```

## Overlays

Let's take a look at a basic rectangle on the map. The format of the shape points are different.

### MapQuest JavaScript SDK

```
var rectangle = new MQA.RectangleOverlay();
rectangle.setShapePoints([[39.847136, -105.362437, 39.641389,
-104.682833]]);
map.addShape(rectangle);
```

### MapQuest.js

```
map.addLayer(L.rectangle([[39.847136, -105.362437],[39.641389,
-104.682833]]));
```

## Events

Events are also handled slightly differently by Leaflet.

### MapQuest JavaScript SDK

```
MQA.EventManager.addListener(map, 'click', eventRaised);
```

### MapQuest.js

```
map.on('click', eventRaised);
```

## Geocoding

Geocoding is a little easier now.

### MapQuest JavaScript SDK

```
MQA.withModule('geocoder', function() {
    map.geocodeAndAddLocations('Denver CO');
});
```

### MapQuest.js

```
L.mapquest.geocoding().geocode('Denver CO');
```

## Routing

Routing is easier too.

### MapQuest JavaScript SDK

```
MQA.withModule('new-route', function() {
    map.addRoute({
        request: {
            locations:[ '80127', 'Steamboat
Springs, CO' ]
        }
    });
});
```

### MapQuest.js

```
L.mapquest.directions().route({
    start: '80127',
    end: 'Steamboat Springs, CO'
});
```

## ...And Beyond

The MapQuest.js sdk also includes a handler for a Search Ahead control to help geocoding.

### MapQuest.js

```
L.mapquest.geocodingControl().addTo(map);
```

Any questions, issues, or suggestions can be sent on to the MapQuest support team or posted in the Developer Network forum. Good luck and have fun!