# How to create custom styling for react native maps.

This post will teach you how to change the **style (colors and other visual elements)** of a react native map.  
You will also learn how to use **callouts** to add a **search box** to a map.

# Get a map

If you haven’t done this already, get yourself a nice basic map. If you have a map already, skip to the next section. If you don’t, slap this into your terminal:

npm install react-native-maps --save

And follow these [instructions](https://github.com/react-community/react-native-maps/blob/master/docs/installation.md). After you have installed react native maps, you are going to want to import react native maps and add a MapView component to the render of your app.

import MapView from 'react-native-maps';// declare this outside of render  
var region = {  
latitude: 37.78825,  
longitude: -122.4324,  
latitudeDelta: 0.0922,  
longitudeDelta: 0.0421,  
};// add this to render  
<MapView  
initialRegion={region}  
/>

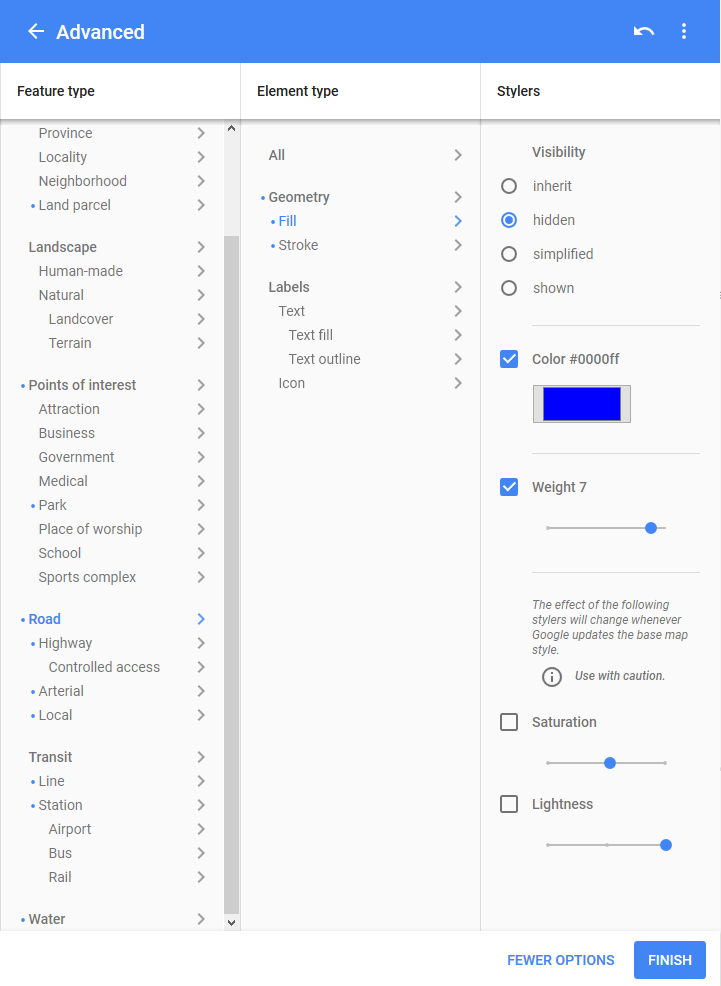
Verify that the map works and move on.

# Choosing the style

According to the [MapView documentation](https://github.com/react-community/react-native-maps/blob/master/docs/mapview.md) you can add a custom style to a map with the prop **customMapStyle**. This prop accepts an **array** which contains the custom style. To generate a custom style go to the [Google Maps Styling Wizard](https://mapstyle.withgoogle.com/).

With the Google Maps Styling Wizard you can create a unique style for your react native map. Using the basic editor you can choose from six themes and change the density of roads, landmarks, and labels. Click “More Options” and you will be presented with the ultimate Styling Wizard for Google Maps.

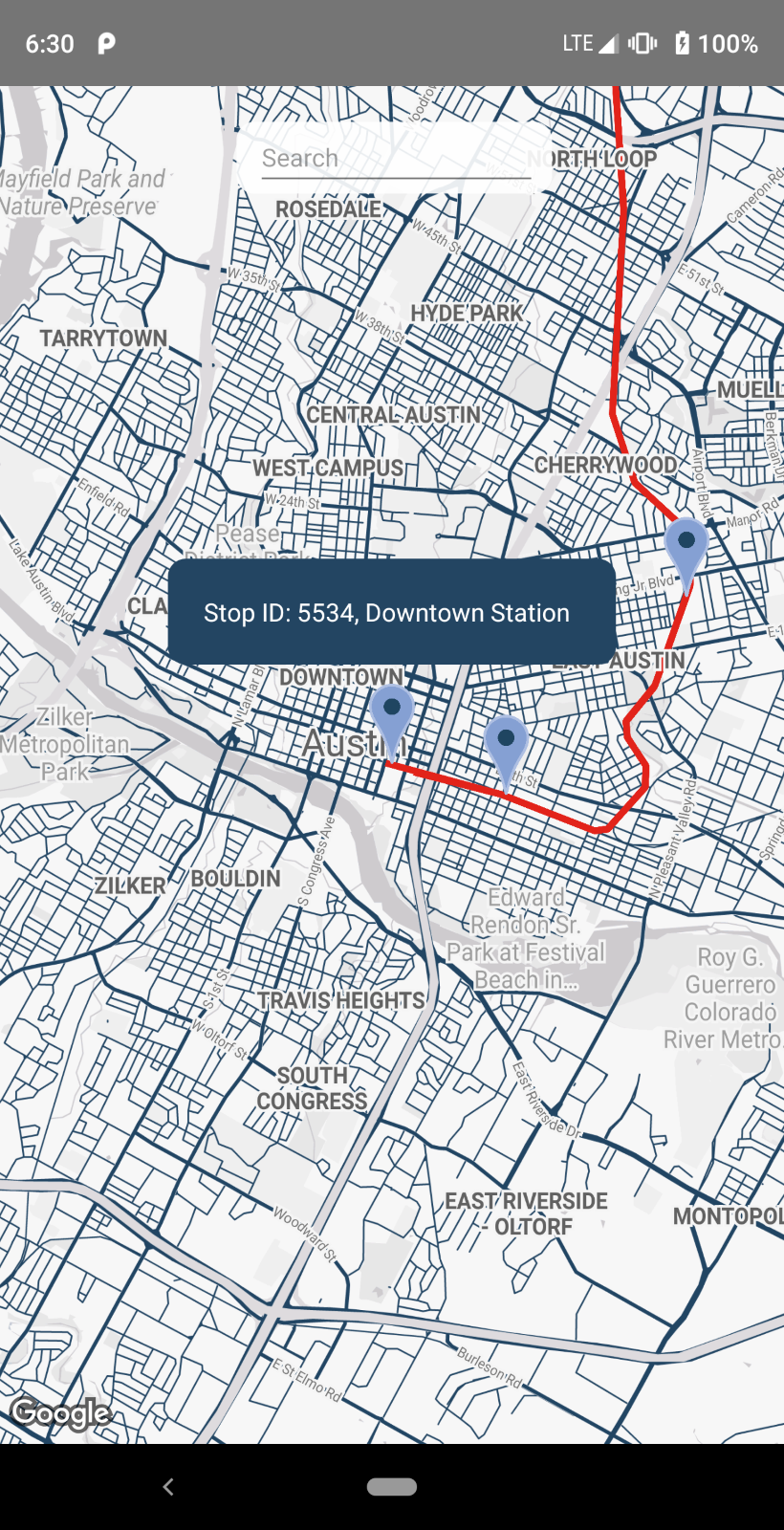
If will briefly go over the editor. A good feature for experimenting with the editor options is the Road since any changes that you make will be immediately visible. Click on Road and then click on Fill. Now you can change the color, weight, saturation, and lightness of the different elements that make up the Road such as labels and geometry.



Once you have spent countless agonizing days creating the perfect style for your map click **FINISH and copy the JSON**. Or just select one of the basic themes and click finish. Now, put this in a variable. And finally **set the customMapStyle prop** of your MapView to this variable.

var mapStyle = [  
{ "elementType": "geometry", "stylers": [ { "color": "#f5f5f5" } ] },  
{ "elementType": "labels.icon", "stylers": [ { "visibility": "off" } ] },  
// and it goes on and on<MapView  
initialRegion={region}  
customMapStyle={mapStyle}  
/>

Here is an example of a custom map style:



# Creating a search box with callouts

You can use the Callout component to add components to the map. **First** place a callout next to the MapView. By next I mean inside the same view.

<View>  
<MapView  
initialRegion={region}  
customMapStyle={mapStyle}  
/>  
<Callout>  
// put some components inside the map  
</Callout>  
</View>

You can use a callout to put whatever you want inside the map. This next example is a search box.

// on render  
<View>  
<MapView  
initialRegion={region}  
customMapStyle={mapStyle}  
/>  
<Callout>  
<View style={styles.calloutView} >  
<TextInput style={styles.calloutSearch}  
placeholder={"Search"}  
/>  
</View>  
</Callout>  
</View>// on the style sheet  
calloutView: {  
flexDirection: "row",  
backgroundColor: "rgba(255, 255, 255, 0.9)",  
borderRadius: 10,  
width: "40%",  
marginLeft: "30%",  
marginRight: "30%",  
marginTop: 20  
},  
calloutSearch: {  
borderColor: "transparent",  
marginleft: 10,  
width: "90%",  
marginRight: 10,  
height: 40,  
borderWidth: 0.0   
}

This search box will look like this:

