



FREE eBook

LEARNING

google-maps-api-3

Free unaffiliated eBook created from
Stack Overflow contributors.

**#google-
maps-api-3**

Table of Contents

About.....	1
Chapter 1: Getting started with google-maps-api-3.....	2
Remarks.....	2
Official Google Documentation.....	2
About the examples in this topic.....	2
Examples.....	2
Basics.....	2
CSS.....	2
HTML.....	2
JavaScript.....	3
Complete example.....	3
Demo.....	4
More info.....	4
Place the user's pin in the map.....	4
You may remplace YOUR_API_KEY in the code by a google api key. The is a link to get a key.....	5
Now you normally have a basic map on your screen. You can find the complete code on google.....	5
The geolocation's function witch we use here is very simple. You can have the complete doc.....	6
Chapter 2: Google Maps JavaScript API v3 - Advanced.....	9
Remarks.....	9
Official Google Documentation.....	9
About the examples in this topic.....	9
Examples.....	9
Custom Styled Map.....	9
Credits.....	12

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [google-maps-api-3](#)

It is an unofficial and free google-maps-api-3 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-maps-api-3.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with google-maps-api-3

Remarks

Official Google Documentation

- [Google Maps JavaScript API Overview](#)
- [Google Maps JavaScript API Code Samples](#)
- [Google Maps JavaScript API Reference](#)

About the examples in this topic

- `YOUR_API_KEY` needs to be replaced by your own application API key. You can obtain an API key and configure it in the [Google API Console](#).

Examples

Basics

CSS

Here are the minimum CSS rules that Google advises you to use, in a separate CSS file, or within an HTML style tag, e.g. `<style type="text/css">...</style>`.

```
html, body {
  height: 100%;
  margin: 0;
  padding: 0;
}

#map {
  height: 400px;
}
```

HTML

Google recommends that you declare a true `DOCTYPE` within your web application.

```
<!DOCTYPE html>
```

Use the following script tag to load the Google Maps JavaScript API in your application.

```
<script async defer
  src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initialize">
</script>
```

Create an HTML element to hold the map.

```
<div id="map"></div>
```

JavaScript

Here is a very simple example displaying a [Map](#) and a [Marker](#).

```
function initialize() {

  // Create a LatLng object
  // We use this LatLng object to center the map and position the marker
  var center = new google.maps.LatLng(50,0);

  // Declare your map options
  var mapOptions = {
    zoom: 4,
    center: center,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  };

  // Create a map in the #map HTML element, using the declared options
  var map = new google.maps.Map(document.getElementById("map"), mapOptions);

  // Create a marker and place it on the map
  var marker = new google.maps.Marker({
    position: center,
    map: map
  });
}
```

Complete example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Map</title>
    <meta name="viewport" content="initial-scale=1.0">
    <meta charset="utf-8">
    <style>
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }

      #map {
        height: 400px;
      }
    </style>
  </head>
  <body>
    <div id="map">
    </div>
  </body>
</html>
```

```
</style>
</head>
<body>
  <div id="map"></div>
  <script>
    function initialize() {

      // Create a LatLng object
      // We use this LatLng object to center the map and position the marker
      var center = new google.maps.LatLng(50, 0);

      // Declare your map options
      var mapOptions = {
        zoom: 4,
        center: center,
        mapTypeId: google.maps.MapTypeId.ROADMAP
      };

      // Create a map in the #map HTML element, using the declared options
      var map = new google.maps.Map(document.getElementById("map"), mapOptions);

      // Create a marker and place it on the map
      var marker = new google.maps.Marker({
        position: center,
        map: map
      });
    }
  </script>
  <script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initialize" async
defer></script>
  </body>
</html>
```

Demo

[JSFiddle demo](#)

More info

Please read this topic's [Remarks](#) for more information.

Place the user's pin in the map.

Note, if you are not familiar with the google maps api, you may read the precedent example (basics) in order to understand this little example.

- **First, initialize the map.**

You may add an map's element in your HTML code and a bite of CSS like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
  /* Always set the map height explicitly to define the size of the div
   * element that contains the map. */
  #map {
    height: 100%;
  }
  /* Optional: Makes the sample page fill the window. */
  html, body {
    height: 100%;
    margin: 0;
    padding: 0;
  }
</style>
</head>
<body>
<div id="map"></div>
</body>
</html>
```

Now, you have to add the google maps library into your code with a balise script like this:

```
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>
```

You may remplace YOUR_API_KEY in the code by a google api key. The is a [link](#) to get a key.

Next, you have to write in your code a function witch serve as a callback (or a function of initialization) for your map. Here, we just add a small function witch you can find on google [here](#):

```
function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: -34.397, lng: 150.644},
    zoom: 8
  });
}
```

Now you normally have a basic map on your screen. You can find the complete code on [google](#).

- **Second, find the user position.**

To request the user position, there is a very simple function witch is provided by the navigator:

```
navigator.geolocation.getCurrentPosition(showPosition);
```

Note that this function accept a parameter. It is a function to call if the geolocation is successful.

We have to create this function. :)

```
function showPosition(position) {
    alert (position);
}
```

This function is very simple and we will have to update it after in order to plot a marker on the user position.

The geolocation's function witch we use here is very simple. You can have the complete documentation on [w3schools](https://www.w3schools.com/geolocation/).

At his point the code looks like this:

```
<!DOCTYPE html>
<html>

<head>
  <title>Simple Map</title>
  <meta name="viewport" content="initial-scale=1.0">
  <meta charset="utf-8">
  <style>
    /* Always set the map height explicitly to define the size of the div
     * element that contains the map. */

    #map {
      height: 100%;
    }
    /* Optional: Makes the sample page fill the window. */

    html,
    body {
      height: 100%;
      margin: 0;
      padding: 0;
    }
  </style>
</head>

<body>
  <div id="map"></div>
  <script>
    var map;
    navigator.geolocation.getCurrentPosition(showPosition);
    function initMap() {
      map = new google.maps.Map(document.getElementById('map'), {
        center: { lat: -34.397, lng: 150.644 },
        zoom: 8
      });
    }
    function showPosition(position) {
      console.log(position);
    }
  </script>
  <script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>
```



```
</body>

</html>
```

- **And third, display the user's position on the map with a marker.**

In order to display a marker on the map you can use the function in the example 'basics':

```
// Create a marker and place it on the map
var marker = new google.maps.Marker({
  position: position,
  map: map
});
```

I will no details this lines of codes very precisely. You just may to now that when you create a marker with this code: `new google.maps.Marker({});`, you pass the 'marker options' enter the embraces. You can consult the google documentation [here](#).

Also note that you can specify the position of the marker very easily with the position parameter.

Now we have to modify the `showPosition` function.

You can access simply to the lat and lng of the variable position like this:

```
var markerPosition={};
markerPosition.lat=position.coords.latitude;
markerPosition.lng=position.coords.longitude;
```

Like this, google understand how to simply access to the lat and lng value.

Now we add to modify the `showPosition` function to add a marker in the user position.

```
function showPosition(position) {
  var markerPosition={};
  markerPosition.lat=position.coords.latitude;
  markerPosition.lng=position.coords.longitude;
  // Create a marker and place it on the map
  var marker = new google.maps.Marker({
    position: markerPosition,
    map: map
  });
}
```

- **Finally, your code should looks like this:**

```
<!DOCTYPE html>
<html>

<head>
  <title>Simple Map</title>
  <meta name="viewport" content="initial-scale=1.0">
  <meta charset="utf-8">
```

```

<style>
  /* Always set the map height explicitly to define the size of the div
   * element that contains the map. */

  #map {
    height: 100%;
  }
  /* Optional: Makes the sample page fill the window. */

  html,
  body {
    height: 100%;
    margin: 0;
    padding: 0;
  }
</style>
</head>

<body>
  <div id="map"></div>
  <script>
    var map;
    navigator.geolocation.getCurrentPosition(showPosition);
    function initMap() {
      map = new google.maps.Map(document.getElementById('map'), {
        center: { lat: -34.397, lng: 150.644 },
        zoom: 8
      });
    }
    function showPosition(position) {
      var markerPosition={};
      markerPosition.lat=position.coords.latitude;
      markerPosition.lng=position.coords.longitude;

      // Create a marker and place it on the map
      var marker = new google.maps.Marker({
        position: markerPosition,
        map: map
      });
    }
  </script>
  <script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
  async defer></script>
</body>

</html>

```

Read [Getting started with google-maps-api-3](https://riptutorial.com/google-maps-api-3/topic/3504/getting-started-with-google-maps-api-3) online: <https://riptutorial.com/google-maps-api-3/topic/3504/getting-started-with-google-maps-api-3>

Chapter 2: Google Maps JavaScript API v3 - Advanced

Remarks

Official Google Documentation

- [Google Maps JavaScript API Overview](#)
- [Google Maps JavaScript API Code Samples](#)
- [Google Maps JavaScript API Reference](#)

About the examples in this topic

- `YOUR_API_KEY` needs to be replaced by your own application API key. You can obtain an API key and configure it in the [Google API Console](#).

Examples

Custom Styled Map

```
<!DOCTYPE html>
<html>
  <head>
    <title>Styled Maps</title>
    <meta charset="utf-8">
    <style>
      #map {
        height: 100%;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script type="text/javascript">
      function initialize() {

        // Create an array of styles.
        var styles = [{
          stylers: [{
            hue: "#4679BD"
          }, {
            saturation: 100
          }]
        }, {
          featureType: "poi",
          elementType: "labels",
          stylers: [{
            visibility: "off"
          }]
        }
      ]
    </script>
  </body>
</html>
```

```

    }, {
      featureType: "administrative",
      elementType: "labels",
      stylers: [{
        color: "#
      }]
    }, {
      featureType: "road.local",
      elementType: "geometry",
      stylers: [{
        visibility: "off"
      }]
    }, {
      featureType: "road",
      elementType: "labels",
      stylers: [{
        visibility: "off"
      }]
    }, {
      featureType: "land",
      elementType: "geometry",
      stylers: [{
        hue: "#e4cc55",
        saturation: 100
      }]
    }, {
      featureType: "water",
      elementType: "geometry",
      stylers: [{
        color: "#C5E7FF"
      }]
    }, {
      featureType: "transit.station.airport",
      elementType: "geometry",
      stylers: [{
        hue: "#FF00CA"
      }]
    }
  ]];

```

// Create a new StyledMapType object, passing it the array of styles, as well as the name to be displayed on the map type control.

```

var styledMap = new google.maps.StyledMapType(styles, {
  name: "Styled Map"
});

```

// Create a map object, and include the MapTypeId(s) to add to the map type control.

```

var mapOptions = {
  zoom: 6,
  center: new google.maps.LatLng(46.13, 6.14),
  mapTypeControlOptions: {
    mapTypeIds: [google.maps.MapTypeId.TERRAIN, 'custom_map_style']
  }
};

```

// Create the map.

```

var map = new google.maps.Map(document.getElementById('map-canvas'),
  mapOptions);

```

// Associate the styled map with the MapTypeId and set it to display.

```

map.mapTypes.set('custom_map_style', styledMap);

```

```
        map.setMapTypeId('custom_map_style');
    }
</script>
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initialize"
  async defer></script>
</body>
</html>
```

To create your own map style, please refer to the [Style Reference](#) documentation and/or use the great [Styled Maps Wizard](#) tool.

[JSFiddle demo](#)

Read Google Maps JavaScript API v3 - Advanced online: <https://riptutorial.com/google-maps-api-3/topic/6781/google-maps-javascript-api-v3---advanced>

Credits

S. No	Chapters	Contributors
1	Getting started with google-maps-api-3	Abhishek , Abrar Jahin , bamnet , Community , Daniel Nugent , Mr. J , MrUpsidown , Praveen Kumar , Rachel Gallen , RamenChef , S.P.H.I.N.X , Sarah Maddox , Shuvo Habib , Soldeplata Saketos
2	Google Maps JavaScript API v3 - Advanced	MrUpsidown , RamenChef