Practical examples: use the geolocation API altogether with Google Maps

In this section, we give some examples that shows how to get a static map (a picture), using the Google Static Map API, how to display an interactive map using the Google Map JavaScript API and even how to get an estimation of the surface address from the longitude and latitude, using the Google Reverse Geocoding JavaScript API.

These three examples have an increasing complexity, but most of the code is reused and adapted without even reading the Google documentation about the different APIs.

EXAMPLE 1 (EASY): HOW TO GET A STATIC IMAGE MAP CENTERED ON YOUR LONGITUDE AND LATITUDE

Online example available on JS Bin, or try it here in your browser:

Click the button to get your position:

Try It

It also illustrates the use of the error callback from the previous section. The Google Map API is used to get an image centered at the longitude and latitude collected with the HTML5 Geolocation API.

Source code extract:

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Click the button to get your position:</p>
<button onclick="getLocation()">Try It</button>
<div id="mapholder"></div>
<script>
var x=document.getElementById("demo");
```

```
10.
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition,showError);
  } else{
    x.innerHTML="Geolocation is not supported by this browser.";
  }
}

19. function showPosition(position) {
  // Google map API needs the latitude and longitude separated by a comma
  varlatlon=position.coords.latitude+","+position.coords.longitude;
  // Google map API URL that returns an image centered on the longitude and
latitude
  varimg_url="http://maps.googleapis.com/maps/api/staticmap?center="
          +latlon+"&zoom=14&size=400x300&sensor=false";
  document.getElementById("mapholder").innerHTML="<img src='"+img_url+"'
/>";
 }

 function showError(error) {
   ...
 }
</script>
</body>
35. </html>
```

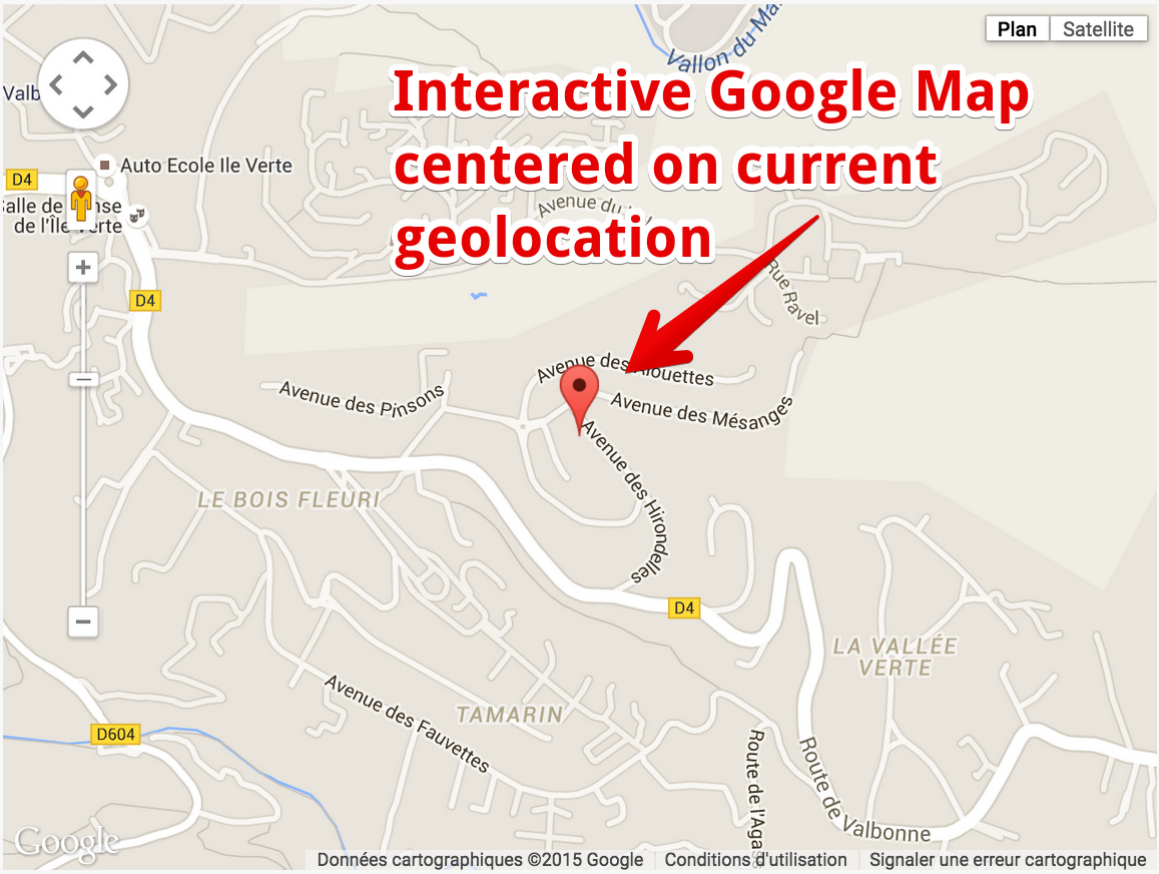
The magic occurs at line 23, where we use the Google Static Map API.

EXAMPLE 2 (A BIT MORE COMPLICATED...) THAT SHOWS HOW TO DISPLAY AN INTERACTIVE GOOGLE MAP CENTERED ON THE CURRENT POSITION

This example is just given as is, as there are so many possibilities for rendering a map with the Google Map API. However, we think having such a basic example might be useful.

Online example at JS Bin

Got position :
Latitude : 43.63989420000001
Longitude: 7.0606103
Altitude : null



Source code of the example:

```
<!doctype html>
<html>
<head>
</head>
<body>
<!-- for position display -->
<div id="myposition"></div>

<!-- for gmap display -->
<div id="map" style="width:640px;height:480px"></div>

<!-- get gmap API -->
<script src="http://maps.google.com/maps/api/js?sensor=false"></script>
```

```
<script>
// Default position
var centerpos = newgoogle.maps.LatLng(48.579400,7.7519);

// default options for the google map
var optionsGmaps = {
   center:centerpos,

navigationControlOptions: {style:google.maps.NavigationControlStyle.SMALL},
   mapTypeId: google.maps.MapTypeId.ROADMAP,
   zoom: 15
};

// Init map object
var map = newgoogle.maps.Map(document.getElementById("map"),optionsGmaps);

if(navigator.geolocation) {

   // callback function, called by getCurrentPosition() in case of success
   function drawPosition(position) {
      var infopos = "Got position : <br>";
      infopos += "Latitude : "+position.coords.latitude+"<br>";
      infopos += "Longitude: "+position.coords.longitude+"<br>";
      infopos += "Altitude : "+position.coords.altitude+"<br>";
      document.getElementById("myposition").innerHTML= infopos;

      // Make new object LatLng for Google Maps
      var latlng = newgoogle.maps.LatLng(position.coords.latitude,
                                         position.coords.longitude);

      // Add a marker at position
      var marker = new google.maps.Marker({
                           position: latlng,
                           map: map,
                           title:"You are here"
      });

      // center map on longitude and latitude
      map.panTo(latlng);
   }
```

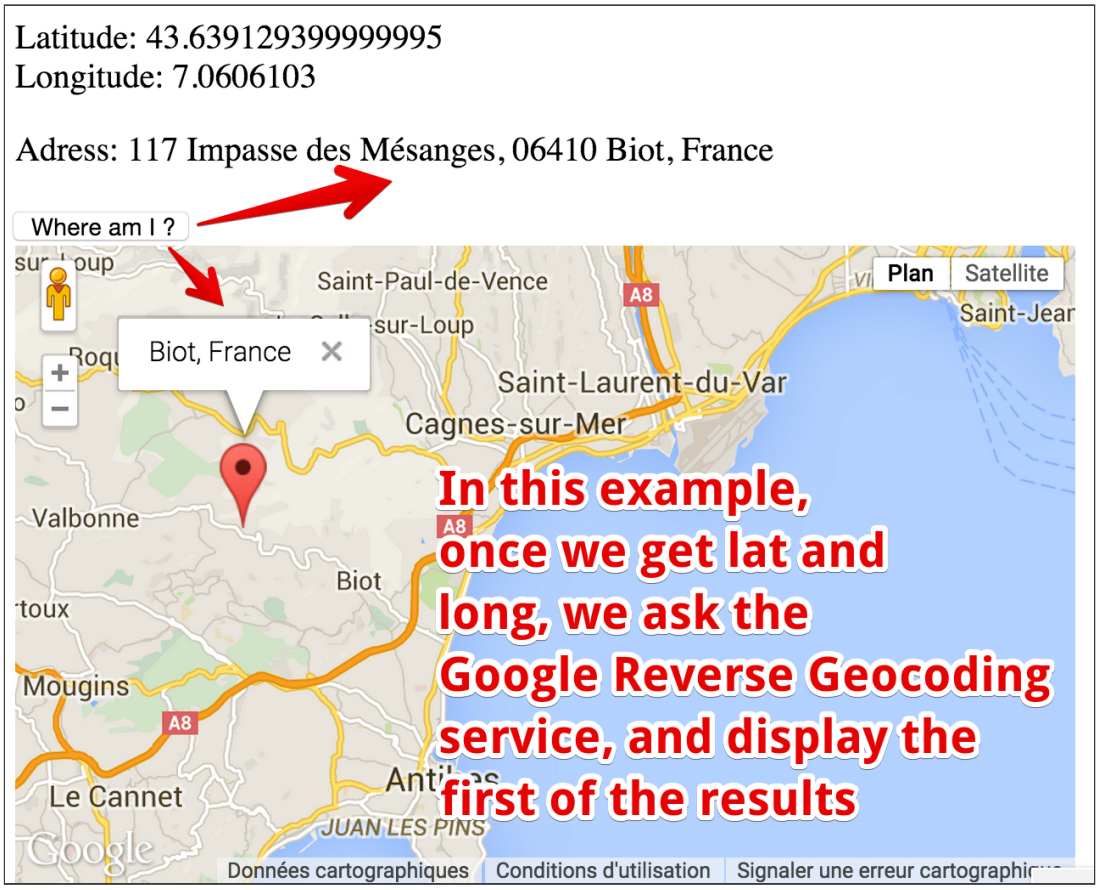
```
    // callback function, called by getCurrentPosition() in case of error
    function errorPosition(error) {
      ...
    }
    navigator.geolocation.getCurrentPosition(drawPosition,errorPosition);
  } else {
    alert("Geolocation API not supported by your browser");
  }
  </script>
</body>
</html>
```

EXAMPLE 3 (ADVANCED) SHOWS HOW TO GET A SURFACE ADDRESS FROM THE LONGITUDE AND LATITUDE

This is another example that gets the surface address from longitude and latitude. It uses the Google Reverse Geocoding JavaScript API. For those of you really interested to know how this API works, please read the Google documentation and tutorials.

Without going into detail, the below example might be useful to copy/paste/adapt for trying to pre-fill a form where one is asked for an address. Geolocation is useful for guessing the country, city, zip code, street, etc. Some examples that use this feature will be given in the next section of the course.

Online example at JS Bin.



Source code of the example:

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <scriptsrc="https://maps.googleapis.com/maps/api/js?
v=3.exp&sensor=false"></script>
 <script>
 // p elements for displaying lat / long and address
 var displayCoords, myAddress;
 // used with the google apis
10. var geocoder;
 var map;
 var infowindow = new google.maps.InfoWindow();
 var marker;
 // Called when the page is loaded
 function init() {
   displayCoords=document.getElementById("msg");
   myAddress =document.getElementById("address");
20. geocoder = new google.maps.Geocoder();
```

```
  // In order to show something even before a user clicked on the button
  var latlng = new google.maps.LatLng(34.0144,-6.83);
  var mapOptions = {
    zoom: 8,
    center: latlng,
    mapTypeId: 'roadmap'
  }
30.
map = newgoogle.maps.Map(document.getElementById('map_canvas'),mapOptions);
} // end of init()
// Called when the button is clicked
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    displayCoords.innerHTML="Geolocation API not supported by your
browser.";
  }
40. }
// Called when a position is available
function showPosition(position) {
  displayCoords.innerHTML="Latitude: " +position.coords.latitude +
                    "<br />Longitude: " +position.coords.longitude;
  // Display the map
  showOnGoogleMap(newgoogle.maps.LatLng(position.coords.latitude,
                                  position.coords.longitude));
}
function showOnGoogleMap(latlng) {
 // Ask google geocoder for a surface address once we get a longitude and
 // a latitude. In fact the reverse geocoder sends back an array of "guesses"
 // i.e. not only one address object, but several. Each entry in this array
 // has several properties like street, city, etc. We use the "formatted_address"
 // one here, but it might be interesting to get the detailed properties in other
 // applications like a form with street, city, zip code etc.
58. geocoder.geocode({'latLng':latlng},reverseGeocoderSuccess);
 function reverseGeocoderSuccess(results, status) {
  if (status == google.maps.GeocoderStatus.OK) {
    if (results[1]) {
      map.setZoom(11);
      marker = new google.maps.Marker({
```

```
                        position: latlng,
                        map: map
                    });
          infowindow.setContent(results[1].formatted_address);
          infowindow.open(map, marker);
          // Display address as text in the page
          myAddress.innerHTML="Adress: " +results[0].formatted_address;
        } else {
          alert('No surface address found');
        }
      } else {
        alert('Geocoder failed due to: ' + status);
      }
    } // end of reverseGeocoderSuccess
} // end of showOnGoogleMap
</script>
</head>
<body onload="init()">
<title>HTML5 + Geolocalisation + Google Maps API Reverse Geocoding</title>
<p id="msg">Click the button to get your coordinates:</p>
<p id="address"></p>
<button onclick="getLocation()">Where am I ?</button>
<div id="map_canvas" style="width: 500px; height:300px"></div>
</body>
</html>
```