# Google Map Lightning Component

By Sam Man (sman@salesforce.com)

Visualforce Page works well with Google Map. It allows to integrate Google Map API to include the Google Map in the Visualforce Page.

When the Google Map is loaded in the web page, it will add its <script> tag. This will cause an issue to include the Google Map in the Lightning Component due to the security restrictions of the locker service enabled – that prevent components from XSS and similar security issues.

<https://developer.salesforce.com/blogs/developer-relations/2016/04/introducing-lockerservice-lightning-components.html>

Referring to the blog of [Jaswinder Rattanpal](https://www.rattanpal.com/author/jrattanpal/), we can make the Google Map by implementing the map in the Visualforce page and embed in Lightning component as iFrame. The Lightning component and Visualforce can communicate by sending/receiving messages via window.postMessage().

<https://www.rattanpal.com/2017/03/19/salesforce-implement-google-maps-lightning-components/>

This is a sample custom Lightning Component that integrate the Google Map in the Lightning Component following the iFrame approach.

Use Case:

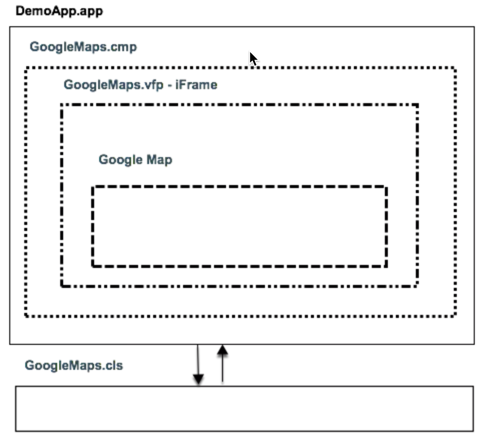
The Lightning component can display the object (object type as specified as attribute) on the Google Map based on an object field (specified as attribute).

The Lightning component has the following attributes:

* objectType – specify the Object Type
* nameField – specify the name field of the object. The component uses the name as the maker text.
* geoLocField – specify the name field of the geolocation field that specify the latitude and longitude. For custom field, please specify the field name without “\_\_c”
* nameValue – the name value to locate the object (e.g. account name of the account object or the case number of case object)

## Design:

Architecture:

 From the blog of Jaswinder Rattanpal

GoogleMap.apxc

The server side controller that have functions

* load the object according to the attributes of the component

Query:

strQuery = 'select id,' + nameField + ',' + latFieldName + ','+ lngFieldName + ' from ' + objectType

+ ' where ' + nameField + ' = ' + '\'' + key + '\'' + ' LIMIT 1';

* update the object if the geolocation

GoogleMap.vfp

This is the Visualforce page the load the Google Map and display the marker based on the latitude and longitude passed from Lightning Component.

script.src = "https://maps.googleapis.com/maps/api/js?key*={Insert the Google Map API Key}*&callback=initMap";

This page will receive the object data by receiving the post message from the Lightning Component.

Once the map is loaded, user can click the map to pin the new location of the object. Pressing the “Update” control will update the object by sending the message back to Lightning component.

GoogleMap.cmp

This is the Lightning Component that load the data from the object according to the attributes set and include the iFrame to load the GoogleMap Visualforce page.

DemoApp.app

This is a Lightning Application used as the container to include the GoogleMap Lightning Component for testing and demo purpose.

<aura:application controller="GoogleMap">

<!-- link to the Account Object -->

<c:GoogleMap objectType='Account' nameField='Name' geoLocField='Location' nameValue='GenePoint'/>

</aura:application>

To access the demo app:

https:// *{domain of the instance}*/c/DemoApp.app

Flow Diagram:

GoogleMap.apxc GoogleMap.cmp GoogleMap.vfp

updateGeoLocation ()

User click the map to pinpoint the new location

User click the ‘Update’ control to update the location

Post message (state = ‘UPDATE)

Post message (state = ‘CHANGED’)

Post message with object data and load the Google map

Post message (state = ‘LOADED’)

Load iFrame include the Visualforce page

getMapObject ()

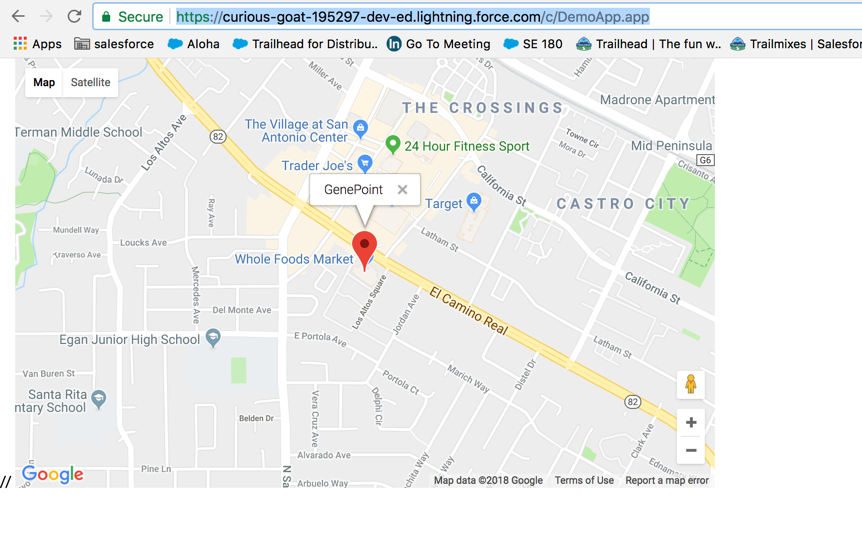
## Demo Flow:

1. Open the DemoApp.app to configure the component attribute.

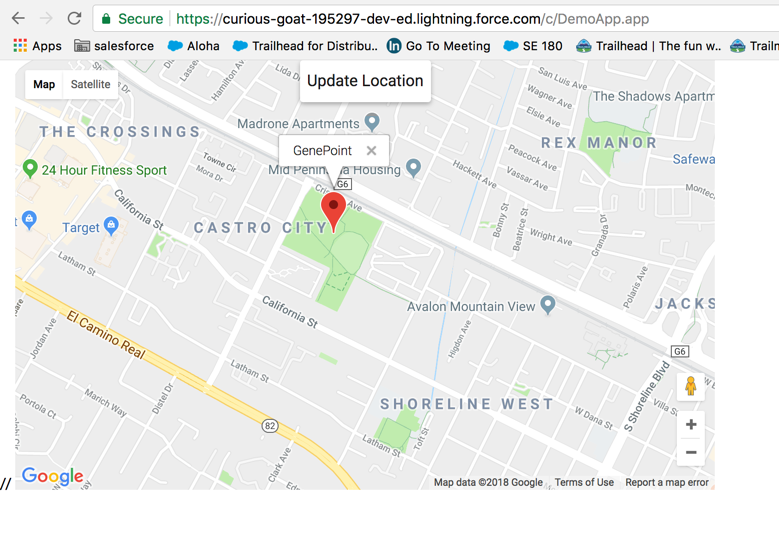
Display the ‘GenePoint’ account according to the geolocation field - *Location*



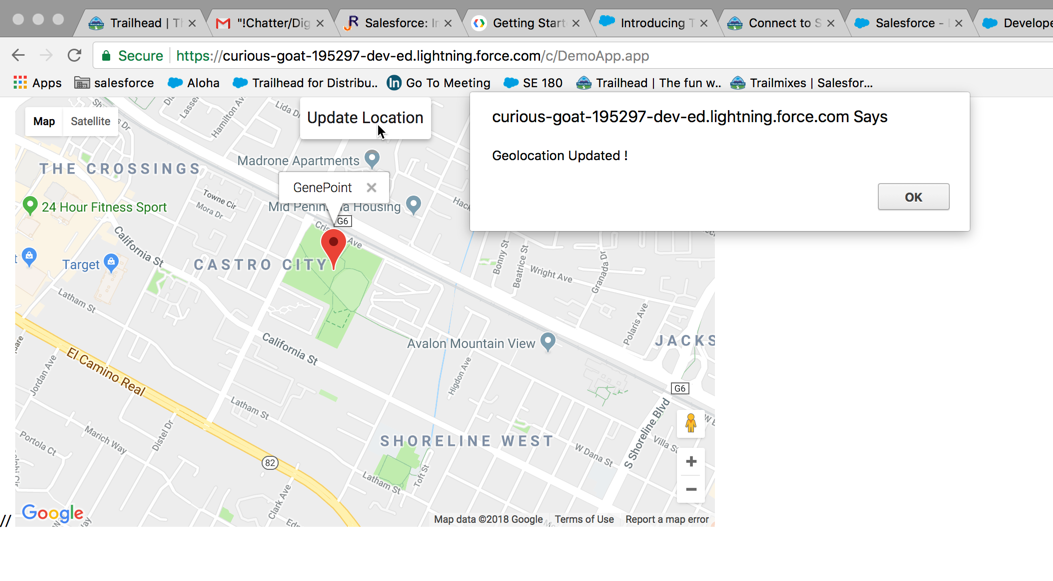
1. Run the DemoApp.app



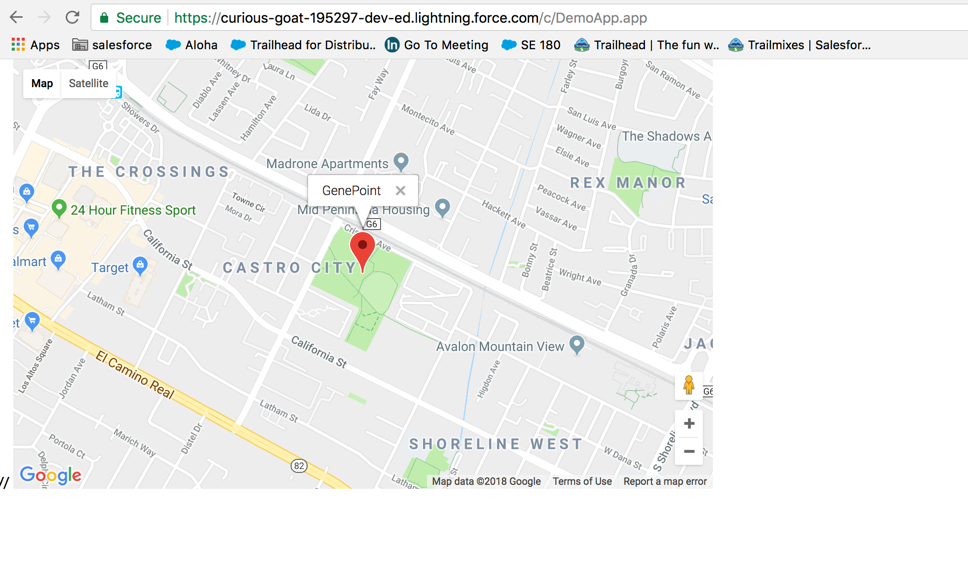
1. Change the location by clicking the map and the “Update Location” button is shown in the top middle of the map



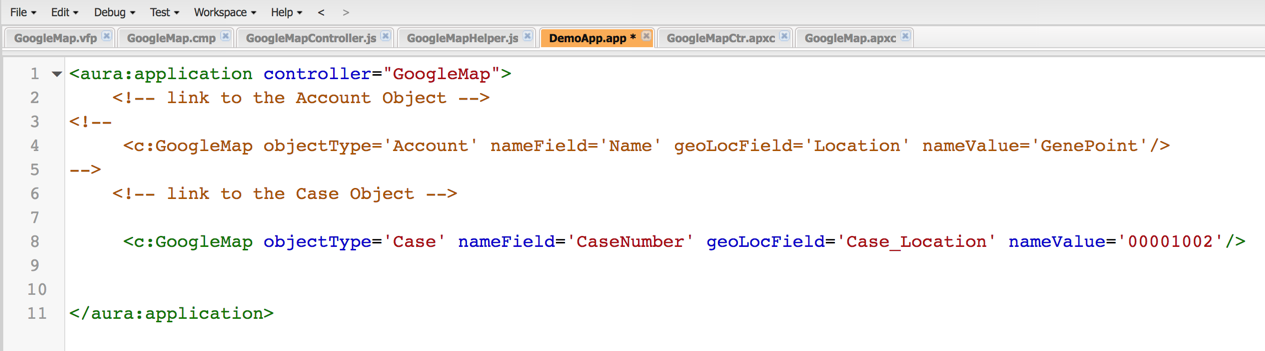
1. Click the button to update the geolocation field of the object



1. Reload the DemoApp.app and the marker is set according to the new geolocation



1. Link to the Case object. Open the DemoApp.app code and change the attributes as below:



Case\_Location is a custom field of Case that store the geolocation.

00001002 is the case number of the object

1. Reload the DemoApp.app to display the marker according to the related case object.

Display the Case *‘00001002’* according to the geolocation field – *Case\_Location*

