**Mopsi markerClustering API for Google Maps V3**

The API provides clustering of markers on map in order to remove overlap of markers. Currently, grid-based clustering algorithm is used. It supports thumbnail markers of photos and also icons as follow:

|  |  |  |
| --- | --- | --- |
|  | Single object | Cluster |
| Thumbnail | thumb2.jpg | cluster.jpg |
| Icon | marker1_single_green.png |  |

**Functionalities:**

1. Click on single object: opens information window on map
2. Click on main icon of cluster: zoom in to cluster
3. Click on circle icon of cluster thumbnail: opens information window on map and by repeating click it displays the next object’s information

**Files:**

**clusteringLogic.js** includes clustering functions

**clusteringInterface.js** includes functions to handle information on map

**mapFunctions.js** includes some extra functions related to map and handling markers on it

**markerFunctions.js** includes marker creation and handling click and information window

**Methods:**

**mopsiMarkerClustering(map, options)** constructor

**addObject(object)** adds the input json object to the array **markersData** of cluseringObj

**apply()** performs clustering and displays markers on map

**remoteClick(i)** opens information window of selected object i from all data

**clean()** removes all marker clusters and also listeners from map

**Notes:**

1. If the thumbnail is selected and the photo does not exist, the following default photo is used:



1. You can design, the content of information window in the function “createInfoWindow” of markerX class in markerFunctions.js
2. To apply a new clustering using the same object that has already been used, first use clean() function to destroy the previous clustering on map.

**Usage example:**

var options = {};

options.clusteringMethod = "gridBased";

options.markerStyle = markerType; // “thumbnail”, “marker1”

options.markerColor = “yellow”; // “yellow”, “green”, “red”, “blue”

options.representativeType = "mean"; // “mean”, “first”, “middleCell”

options.autoUpdate = 0; // updates only changed clusters on map if =1

map: Google map object that exists

var cluseringObj = new mopsiMarkerClustering(map, options); // constructor

if (cluseringObj.validParams == "YES" ) { // validParams: “YES” or “NO”

 // add data objects

 // supposing your data is in the array data

 for ( var i = 0 ; i < data.length ; i++ ) {

 **//** creating objects one by one and adding tocluseringObjusing **addObject**

 obj = {};

 obj.lat = data[i].lat;

 obj.lon = data[i].lon;

 // optional,

 obj.photourl = data[i].photourl; // needed for “thumbnail”, photourl: full path of a photo

 // on server

 // needed for displaying objects information in infoWindow

 obj.name = data[i].name;

 obj.address = data[i].address;

 obj.date = data[i].date;

 obj.time = data[i].time;

 obj.author = data[i].author;

 cluseringObj.addObject(obj); // adds object to the array **markersData** of cluseringObj

 }

 cluseringObj.apply(); // performing clustering algorithm and displaying markers

}