# Google Maps

Category	Requires	Version
Google	API 19, Android 4.4 - 4.4.4 KitKat	4

### Overview

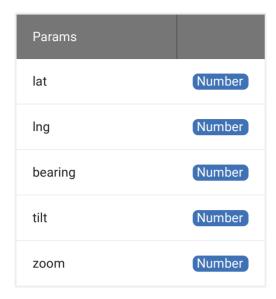
A visible component that shows a map on the screen powered by Google's Maps service.



# **Events**

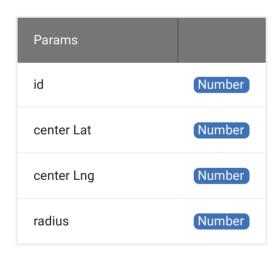
# Camera Position Changed

Called after the camera position of a map has changed.



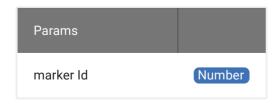
## Finished Dragging Circle

Event been raised after the action of moving a draggable circle is finished. Possible a user drag the center of the circle or drag the radius marker of the circle



### Info Window Clicked

When the marker's infowindow is clicked, returning marker's id



## Map Is Ready

Indicates that the map has been rendered and ready for adding markers or changing other settings. Please add or updating markers within this event

## On Location Changed

Triggers this event when user location has changed. Only works when EnableMylocation is set to true



## On Map Click

Called when the user makes a tap gesture on the map



## On Map Long Click

Called when the user makes a long-press gesture on the map



### On Marker Click

When a marker is clicked



# On Marker Drag

When a marker is been dragged



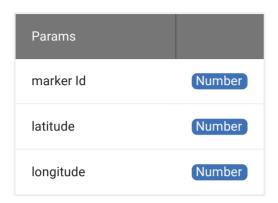
# On Marker Drag End

When the user drags a marker and finish the action, returning marker's id and it's latest position



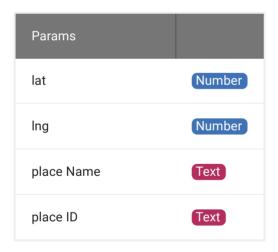
## On Marker Drag Start

When a marker starts been dragged



#### On Point Of Interest Click

This event will be invoked when a user clicks on a point of interest. This can be a shop, coffee-bar or else.



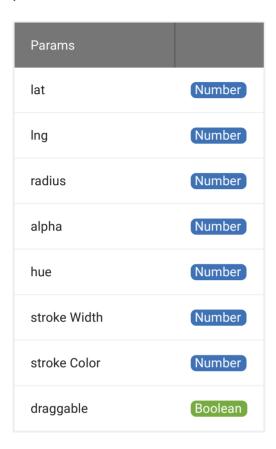
# Methods

### Add Circle

#### Returns: Number

Create a circle overlay on the map UI with specified latitude and longitude for center. "hue" (min 0, max 360) and "alpha" (min 0, max 255) are used to set color and transparency level of the circle, "strokeWidth" and "strokeColor" are for the

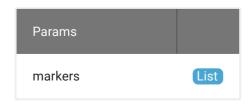
perimeter of the circle. Returning a unique id of the circle for future reference to events raised by moving this circle. If the circle isset to be draggable, two default markers will appear on the map: one in the center of the circle, another on the perimeter.



#### Add Markers

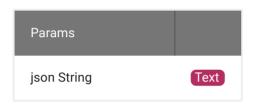
#### Returns: List

Adding a list of YailLists for markers. The representation of a maker in the inner YailList is composed of: lat(double) [required], long(double) [required], Color, title (String), snippet(String), draggable(boolean). Return a list of unquie ids for the added markers. Note that the markers ids are not meant to persist after the app is closed, but for temporary references to the markers within the program only. Return an empty list if any error happen in the input



### Add Markers From Json

Adding a list of markers that are represented as JsonArray. The inner JsonObject represents a markerand is composed of name-value pairs. Name fields for a marker are: "lat" (type double) [required], "lng"(type double) [required], "color"(type int)[in hue value ranging from 0~360], "title"(type String), "snippet"(type String), "draggable"(type boolean)



#### Add Markers Hue

#### Returns: List

Adding a list of YailList for markers. The inner YailList represents a marker and is composed of lat(Double) [required], long(Double) [required], color(int)[in hue value ranging from 0-360], title(String), snippet(String), draggable(boolean). Return a list of unique ids for the markers that are added

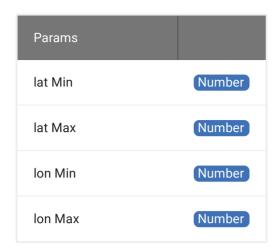


## Add Overlay

Add overlay.

## Add Polygon

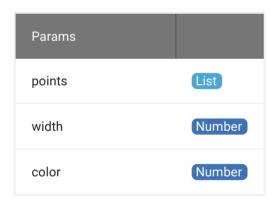
A Polygon is an enclosed shape that can be used to mark areas on the map.



## Add Polyline

#### Returns: Number

This block will return the unique id of the new added polyline. Create a new polyline on the map. Use for 'points' a list of lat, lng pairs. A integer for the 'width' (in pixel) and a valid color for the 'color' parameter.

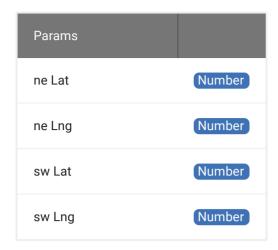


## Add Tile Overlay

Add title overlay.

#### **Bound Camera**

Transforms the camera such that the specified latitude/longitude bounds are centered on screen at the greatest possible zoom level. Need to specify both latitudes and longitudes for both northeast location and southwest location of the bounding box



# Clear All Polygons

Clear all Polygons.

### Draw Central Square

Draw central square.

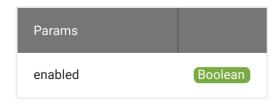
## **Enable Compass**

Enables/disables the compass widget on the map's ui. Call this only after event "MapIsReady" is received



## Enable Map Camera Pos Change Listener

Enable/Disable to listen to map's camera position changed event



## Enable Map Click Listener

Enable/Disable to listen to map's click event



### Enable Map Long Click Listener

Enable/disable to listen to map's long click event



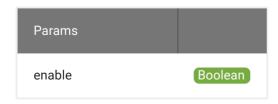
### **Enable My Location**

Enable or disable my location widget control for Google Map. One can call GetMyLocation() to obtain the current location after enable this."



#### **Enable Rotate**

Enables/disables the capability to rotate a map on the ui. Call this only after the event "MapIsReady" is received.



#### **Enable Scroll**

Enables/disables the capability to scroll a map on the ui. Call this only after the event "MapIsReady" is received



#### **Enable Zoom Control**

Enables/disables the zoom widget on the map's ui. Call this only after the event "MapIsReady" is received



### **Enable Zoom Gesture**

Enables/disables zoom gesture on the map ui. Call this only after the event "MapIsReady" is received.



### Get All Circle IDs

### Returns: List

Get all circles Ids. A short cut to get all the references for the eixisting circles

### Get All Marker IDs

Returns: List

Get all the existing markers's Ids

## Get All Polyline Ids

Returns: List

This will return a list with all available polyline id's.

### Get Bounding Box

Returns: Text

Get bounding box.



## Get Map Center

Returns: Text

Get map center. If a error occures the output will be '-999'.

Get Markers

Returns: List

Add a list of markers composed of name-value pairs. Name fields for a marker are: "lat" (type double) [required], "lng"(type double) [required], "color"(type int)[in hue value ranging from 0~360], "title"(type String), "snippet"(type String), "draggable"(type boolean)

### Get My Location

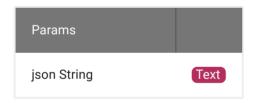
#### Returns: List

Get current location using Google Map Service. Return a YailList with first item beingthe latitude, the second item being the longitude, and last time being the accuracy of the reading.

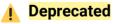
### Get Points From |son

#### Returns: List

Convert a JsonArray of points (lat, lng pairs) to a list.



#### Get Zoom Level Info



### Returns: Number

Deprecated block! Don't use this anymore. Use instead 'Camera Zoom Level'.

#### Move Camera

Move the map's camera to the specified position and zoom level



### Remove Circle

#### Returns: Boolean

Remove a circle for the map. Returns true if successfully removed, false if the circle does not exist with the specified id



### Remove Marker

Remove a marker from the map



# Remove Polyline

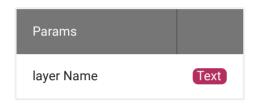
### Returns: Boolean

Use this block to remove a polyline from the map. It will return true if it was successful.



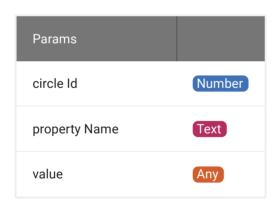
## Set Map Type

Set the layer of Google map. Default layer is "normal", other choices including "hybrid", "satellite", and "terrain"



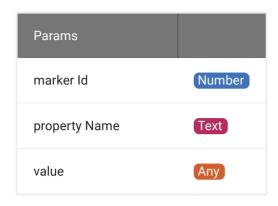
## **Update Circle**

Set the property of an existing circle. Properties include: "alpha" (number, value ranging from  $0\sim255$ ), "color" (nimber, hue value ranging  $0\sim360$ ), "radius" (number in meters)



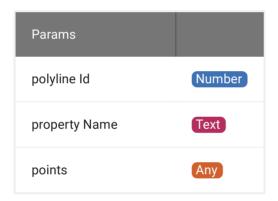
# Update Marker

Set the property of a marker, note that the marker has to be added first or else will throw an exception! Properties include: "color" (hue value ranging from  $0\sim360$ ), "title", "snippet", "draggable" (give either true or false as the value).



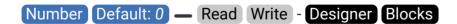
### **Update Polyline**

Update any polyline with the given id. You can change the property values for 'width' (in pixel), 'color' or 'points (a list of lat, lng pairs).



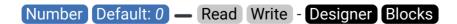
# **Properties**

## Camera Angle



Move the map's camera to the specified tilt, the angle (in degrees) from the nadir (directly facing the Earth). Must be a value between 0.0 and 90.0

#### Camera Rotation



Move the map's camera to the specified bearing, the direction that the camera is pointing in (in degrees clockwise from north).

#### Camera Zoom Level

Move the map's camera to the specified zoom level.

### Compass Enabled

Indicates whether the compass widget is currently enabled in the map ui

### Height

Specifies the component's vertical height, measured in pixels.

### **Height Percent**

Specifies the component's vertical height as a percentage of the height of its parent Component.

Map Camera Changed Listener Enabled

Indicates if the map camera's position changed listener is currently enabled

Map Click Listener Enabled

Indicates if the mapClick event listener is currently enabled

### Map Long Click Listener Enabled

Indicates if the map longClick listener is currently enabled

### Map Type

Indicates the current map type

#### My Location Enabled

Indicates whether my locaiton UI control is currently enabled for the Google map.

#### Rotate Enabled

Indicates whether the capability to rotate a map on the ui is currently enabled

#### Scroll Enabled

Indicates whether the capability to scroll a map on the ui is currently enabled

### Style

Sets the style of the map from json. Just use a text field and paste there the json data. Create a custom map style at https://mapstyle.withgoogle.com/. Set the theme to "standard" to clear the style json.

#### Theme

Sets the theme of the map. The choices are "standard" (default), "silver", "retro", "dark", "night", "aubergine", "vintage", "kodular" and "roads-only".

#### Visible

Returns true iff the component is visible.

#### Width

Specifies the component's horizontal width, measured in pixels.

#### Width Percent

Specifies the component's horizontal width as a percentage of the Width of its parent Component.

#### Zoom Control Enabled

Indicates whether the zoom widget on the map ui is currently enabled

#### Zoom Gesture Enabled

Indicates whether the zoom gesture is currently enabled

Last update: January 25, 2020