Google Maps API

Google Maps API Reference

The Google Maps API is now integrated with the Google AJAX API loader, which creates a common namespace for loading and using multiple Google AJAX APIs. This framework allows you to use the optional `google.maps.*` namespace for all classes, methods and properties you currently use in the Google Maps API, replacing the normal `G` prefix with this namespace. Don't worry: the existing `G` namespace will continue to be supported.

For example, the `GMap2` object within the Google Maps API can also be defined as `google.maps.Map2`.

Note that this reference documentation refers only to the existing `G` namespace.

Core Class:

GMap2

This is the most important class within the Maps API. The other classes in this reference are grouped by their purpose.

Base Classes:

GBounds
GBrowserIsCompatible
GDraggableObject
GDraggableObjectOptions
GInfoWindow
GInfoWindowOptions
GInfoWindowTab
GKeyboardHandler
GLanguage
GLatLng
GLatLngBounds
GLog
GMapOptions
GMapPane
GPoint
GSize
GUnload
G_API_VERSION

Event Classes:

GEvent
GEventListener

Control Classes:

GControl
GControlAnchor
GControl
GControlPosition
GHierarchicalMapTypeControl
GMapType
GMapTypeControl
GMapTypeOptions
GMapUIOptions
GMenuMapTypeControl
GNavLabelControl

Overlay Classes:

GCopyright
GCopyrightCollection
GGroundOverlay
GIcon
GLayer
GMarker
GMarkerManager
GMarkerManagerOptions
GMarkerOptions
GMercatorProjection
GObliqueMercator
GOverlay
GPolyEditingOptions
GPolyStyleOptions
GPolygon
GPolygonOptions
GPolyline
GPolylineOptions
GProjection
GScreenOverlay
GScreenPoint
GScreenSize
GTileLayer
GTileLayerOptions
GTileLayerOverlay
GTileLayerOverlayOptions

Service Classes:

GAdsManager
GAdsManagerOptions
GAdsManagerStyle
GClientGeocoder
GDirections
GDirectionsOptions
GDownloadUrl
GFactualGeocodeCache
GGeoAddressAccuracy
GGeoStatusCode
GGeoXml
GGeocodeCache
GGoogleBar
GGoogleBarAdsOptions
GGoogleBarLinkTarget
GGoogleBarListingTypes
GGoogleBarOptions
GGoogleBarResultList
GPhotoSpec
GPov
GRoute
GStep
GStreetviewClient
GStreetviewClient.ReturnValues
GStreetviewData
GStreetviewFeatures
GStreetviewLink
GStreetviewLocation
GStreetviewOverlay
GStreetviewPanorama
GStreetviewPanorama.ErrorValues
GStreetviewPanoramaOptions
GStreetviewUserPhotosOptions
GTrafficOverlay
GTrafficOverlayOptions
GTravelModes
GXml
GXmlHttp
GXslt

class GMap2

Instantiate class `GMap2` in order to create a map. This is the central class in the API. Everything else is auxiliary.

Constructor

Constructor	Description
`GMap2(container:Node, opts?:GMapOptions)`	Creates a new map inside of the given HTML container, which is typically a `DIV` element. If no set of map types is given in the optional argument `opts.mapTypes`, the default set `G_DEFAULT_MAP_TYPES` is used. If no size is given in the optional argument `opts.size`, then the size of the `container` is used. If `opts.size` is given, then the container element of the map is resized accordingly. See `class GMapOptions`. Note: a Map needs to be centered before it can be used. You should immediately call `GMap2.setCenter()` to initialize a map created with this constructor.

Methods

Configuration

Method	Return Value	Description
`enableDragging()`	`None`	Enables the dragging of the map (enabled by default).
`disableDragging()`	`None`	Disables the dragging of the map.
`draggingEnabled()`	`Boolean`	Returns `true` iff the dragging of the map is enabled.
`enableInfoWindow()`	`None`	Enables info window operations on the map (enabled by default).
`disableInfoWindow()`	`None`	Closes the info window, if it is open, and disables the opening of a new info window.
`infoWindowEnabled()`	`Boolean`	Returns `true` iff the info window is enabled.
`enableDoubleClickZoom()`	`None`	Enables double click to zoom in and out (enabled by default). (Since 2.58)
`disableDoubleClickZoom()`	`None`	Disables double click to zoom in and out. (Since 2.58)
`doubleClickZoomEnabled()`	`Boolean`	Returns `true` iff double click to zoom is enabled. (Since 2.58)

`enableContinuousZoom()`	`None`	Enables continuous smooth zooming for select browsers (disabled by default). (Since 2.58)
`disableContinuousZoom()`	`None`	Disables continuous smooth zooming. (Since 2.58)
`continuousZoomEnabled()`	`Boolean`	Returns `true` if continuous smooth zooming is enabled. (Since 2.58)
`enableGoogleBar()`	`None`	Enables the **GoogleBar**, an integrated search control, to the map. When enabled, this control takes the place of the default *Powered By Google* logo. Note that this control is not enabled by default. Note: The GoogleBar is currently not compatible with the Google Earth plugin, used by map type `GMapType.G_SATELLITE_3D_MAP`, and will be disabled while the Earth plugin is shown. (Since 2.92)
`disableGoogleBar()`	`None`	Disables the **GoogleBar** integrated search control. When disabled, the default *Powered by Google* logo occupies the position formerly containing this control. Note that this control is already disabled by default. (Since 2.92)
`enableScrollWheelZoom()`	`None`	Enables zooming using a mouse's scroll wheel. Note: scroll wheel zoom is disabled by default. (Since 2.78)
`disableScrollWheelZoom()`	`None`	Disables zooming using a mouse's scroll wheel. Note: scroll wheel zoom is disabled by default. (Since 2.78)
`scrollWheelZoomEnabled()`	`Boolean`	Returns a Boolean indicating whether scroll wheel zooming is enabled. (Since 2.78)
`enablePinchToZoom()`	`None`	Enables pinching to zoom on an iPhone or iPod touch. Note: pinch to zoom is enabled by default. (Since 2.143)
`disablePinchToZoom()`	`None`	Disables pinching to zoom on an iPhone or iPod touch. Note: pinch to zoom is enabled by default. (Since 2.143)
`pinchToZoomEnabled()`	`Boolean`	Returns a Boolean indicating whether pinch to zoom is enabled. (Since 2.143)
`getDefaultUI()`	`Object`	Returns a `GMapUIOptions` object specifying default behaviour and UI elements for the Map, based on the UI of `maps.google.com`. (Since 2.147)
`setUIToDefault()`	`None`	Adds the default behaviour and UI elements specified in `getDefaultUI()` to the Map. (Since 2.147)

`setUI(ui:GMapUIOptions)`	`None`	Adds behaviour and UI elements specified in the `ui` parameter, which can be a modified version of the object returned from `getDefaultUI()`. (Since 2.147)

Controls

Method	Return Value	Description
`addControl(control:GControl, position?:GControlPosition)`	`None`	Adds the control to the map. The position on the map is determined by the optional `position` argument. If this argument is absent, the default position of the control is used, as determined by the `GControl.getDefaultPosition()` method. A control instance must not be added more than once to the map.
`removeControl(control:GControl)`	`None`	Removes the control from the map. It does nothing if the control was never added to the map.
`getContainer()`	`Node`	Returns the DOM object that contains the map. Used by `GControl.initialize()`.

Map Types

Method	Return Value	Description
`getMapTypes()`	`GMapType[]`	Returns the array of map types registered with this map.
`getCurrentMapType()`	`GMapType`	Returns the currently selected map type.
`setMapType(type:GMapType)`	`None`	Selects the given new map type. The type must be known to the map. See the constructor, and the method `addMapType()`.
`addMapType(type:GMapType)`	`None`	Adds a new map type to the map. See section `GMapType` for how to define custom map types.
`removeMapType(type:GMapType)`	`None`	Removes the map type from the map. Will update the set of buttons displayed by the `GMapTypeControl` or `GHierarchicalMapTypeControl` and fire the `removemaptype` event.

Map State

Method	Return Value	Description
`isLoaded()`	`Boolean`	Returns `true` iff the map was initialized by `setCenter()` since it was created.
`getCenter()`	`GLatLng`	Returns the geographical coordinates of the center point of the map view.
`getBounds()`	`GLatLngBounds`	Returns the the visible rectangular region of the map view in geographical coordinates.
`getBoundsZoomLevel(bounds:GLatLngBounds)`	`Number`	Returns the zoom level at which the given rectangular region fits in the map view. The zoom level is computed for the currently selected map type. If no map type is selected yet, the first on the list of map types is used.
`getSize()`	`GSize`	Returns the size of the map view in pixels.
`getZoom()`	`Number`	Returns the current zoom level.
`getDragObject()`	`GDraggableObject`	Returns the draggable object used by this map. (Since 2.93)
`getEarthInstance(callback:Function(instance: GEPlugin))`	`None`	Retrieves the instance of the Google Earth Browser Plugin attached to this map, and calls the passed callback function once the instance is ready to receive commands, passing the earth instance (a `GEPlugin` object) as a parameter. This callback function may fire immediately if the Earth instance had previously been instantiated by setting the Google Earth map type via `GMap2.setMapType(G_SATELLITE_3D_MAP)`. If the instance if not yet created, calling `getEarthInstance` will initialize the Google Earth Plug-in. (This asynchronous behavior is why this method requires a callback.) Note: if the initialization encounters an error, the callback function will pass a `null` parameter. Note that that while this method will create an Earth instance, it will **not** set the Earth map type as the current view. To so so, call `GMap2.setMapType(G_SATELLITE_3D_MAP)` explicitly, or allow the user to select this map type via the `MapTypeControl`. See the Google Earth API Developer's Guide for details on how to use the `GEPlugin` object. (Since 2.113)

Modify the Map State

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.