Google Developers

Google API...   X | Search

Sign in

Products   Google APIs Client Library for JavaScript (Beta)

Methods and Classes

This page documents all the methods and classes defined in the JavaScript client library.

Contents

Methods

gapi.auth.authorize
gapi.auth.init
gapi.auth.getToken
gapi.auth.setToken
gapi.client.load
gapi.client.register (Deprecated)
gapi.client.request
gapi.client.newBatch
gapi.client.rpcRequest (Deprecated)
gapi.client.setApiKey

Classes

OAuth 2.0 Token Object
gapi.client.Request
gapi.client.RpcRequest (Deprecated)
gapi.client.Batch
gapi.client.RpcBatch (Deprecated)

Methods

gapi.auth.authorize(*params, callback*)

Initiates the OAuth 2.0 authorization process. The browser displays a popup window prompting the user authenticate and authorize. After the user authorizes, the popup closes and the callback function fires.

> **Note:** Popup blockers working with your user's browser may interfere with this process. For a workaround, see this note in the Authentication page.

Arguments:

Name	Type	Description
params	object	A key/value map of parameters for the request. If the key is not one of the expected OAuth 2.0 parameters (see below), it is added to the URI as a query parameter. Valid keys for OAuth 2.0 parameter include:

Name	Type	Description
client_id	string	The application's client ID. Visit the Google API Console to get an OAuth 2.0 client ID.
immediate	boolean	If true, then login uses "immediate mode", which means that the token is refreshed behind the scenes, and no UI is shown to the user.
response_type	string	The OAuth 2.0 response type property. **Default:** token

scope	string | array	The auth scope or scopes to authorize as a space-delimited string or array (deprecated). Auth scopes for individual APIs can be found in their documentation.
callback	function	The function to call once the login process is complete. The function takes an OAuth 2.0 token object as its only parameter.

gapi.auth.init(*callback*)

Initializes the authorization feature. Call this when the client loads to prevent popup blockers from blocking the auth window on `gapi.auth.authorize` calls.

> **Note:** Calling `gapi.auth.authorize({immediate: true})` when the client loads achieves the same effect. If your application does an immediate-mode check, `gapi.auth.init` should be omitted entirely.

Arguments:

Name	Type	Description
callback	function	A callback to execute when the auth feature is ready to make authorization calls.

gapi.auth.getToken()

Retrieves the OAuth 2.0 token for the application.

Returns:

Type	Description
object	The OAuth 2.0 token. See OAuth 2.0 Token Object below.

gapi.auth.setToken(*token*)

Sets the OAuth 2.0 token for the application.

Arguments:

Name	Type	Description
token	object	The token to set. See OAuth 2.0 Token Object for details about this object.

gapi.client.load(*name*, *version*, *callback*)

Loads the client library interface to a particular API. If a callback is not provided, a promise is returned. The loaded API interface will be in the form `gapi.client.`*api.collection.method*. For example, the Moderator API would create methods like `gapi.client.moderator.series.list`.

Arguments:

Name	Type	Description
name	string	The name of the API to load.
version	string	The version of the API to load.
callback	function	(optional) the function that is called once the API interface is loaded. If not provided, a promise is returned.

gapi.client.register(*method*)

> **Important:** This method has been deprecated. Please see `gapi.client.load` or `gapi.client.request` instead.

gapi.client.request(*args*)

Creates a HTTP request for making RESTful requests.

Arguments:

Name	Type	Description
args	object	An object encapsulating the various arguments for this method. The path is required, the rest are optional. The values are described in detail below. | Name	Type	Description
path	string	The URL to handle the request.
method	string	The HTTP request method to use. Default is `GET`.
params	object	URL params in key-value pair form.
headers	object	Additional HTTP request headers.
body	string | object	The HTTP request body (applies to `PUT` or `POST`).	

Returns:

Type	Description
`gapi.client.Request` | undefined	The returned request object is a promise.

gapi.client.newBatch()

Creates a batch object for batching individual requests.

Arguments:

Name	Type	Description

args	object	An object encapsulating the various arguments for this method. The path is required, the rest are optional. The values are described in detail below.

Name	Type	Description
path	string	The URL to handle the request.
method	string	The HTTP request method to use. Default is `GET`.
params	object	URL params in key-value pair form.
headers	object	Additional HTTP request headers.
body	string | object	The HTTP request body (applies to PUT or POST).

Returns:

Type	Description
`gapi.client.Batch` | undefined	The returned request object is a promise.

gapi.client.rpcRequest(*method*, *name*, *rpcParams*)

> **Important:** This method has been deprecated. Please see `gapi.client.load` or `gapi.client.request` instead.

Creates an RPC Request directly. The method name and version identify the method to be executed and the RPC params are provided upon RPC creation.

Arguments:

Name	Type	Description
method	string	The method to be executed. For example, `plus.people.search` from the G+ API.
version	string	The version of the API which defines the method to be executed. Defaults to `v1`.
rpcParams	object	A key-value pair of the params to supply to this RPC.

Returns:

Type	Description
`gapi.client.RpcRequest`	The RPC request

gapi.client.setApiKey(*apiKey*)

Sets the API key for the application, which can be found in the Developer Console. Some APIs require this to be set in order to work.

Arguments:

Name	Type	Description
apiKey	string	The API key to set.

Classes

OAuth 2.0 Token Object

The OAuth 2.0 token object represents the OAuth 2.0 token and any associated data. Properties:

Name	Type	Description
access_token	string	The OAuth 2.0 token. Only present in successful responses.
error	string	Details about the error. Only present in error responses.
expires_in	string	The duration, in seconds, the token is valid for. Only present in successful responses.
state	string	The Google API scopes related to this token.

gapi.client.Request

An object encapsulating an HTTP request. This object is not instantiated directly, rather it is returned by `gapi.client.request`. There are two ways to execute a request. We recommend that you treat the object as a promise and use the `then` method, but you can also use the method and pass in a callback.

`gapi.client.Request.then(`*onFulfilled, onRejected, context*`)`

For more information about using promises, see Using Promises.

`gapi.client.Request.execute(`*callback*`)`

Executes the request and runs the supplied callback on response.

Arguments:

Name	Type	Description
callback(*jsonResp, rawResp*)	function	The callback function which executes when the request succeeds or fails. `jsonResp` contains the response parsed as JSON. If the response is not JSON, this field will be `false`. `rawResp` is the HTTP response. It is JSON, and can be parsed to an object which includes `body`, `headers`, `status`, and `statusText` fields.

gapi.client.RpcRequest

Similar to `gapi.client.Request` except this object encapsulates requests generated by registered methods. It defines one method:

`gapi.client.RpcRequest.execute(`*callback*`)`

Executes the request and runs the supplied callback with the response.

Name	Type	Description
callback(*jsonResp*, *rawResp*)	function	The callback function which executes when the request succeeds or fails. `jsonResp` contains the response parsed as JSON. If the response is not JSON, this field will be `false`. `rawResp` is the same as `jsonResp`, except it is a raw string that has not been parsed. It is typically used when the response is not JSON.

gapi.client.Batch

Represents an HTTP Batch operation. Individual HTTP requests are added with the `add` method and the batch can be executed using `then` or `execute`. We recommend that you treat the batch object as a promise and use `then`. This class defines the following methods:

`gapi.client.Batch.add(`*request*, *opt_params*`)`

Adds a `gapi.client.Request` to the batch.

Arguments:

Name	Type	Description
request	gapi.client.Request	The HTTP request to add to this batch. This parameter is required.
opt_params	Object	Optional extra parameters for this batch entry. Accepted fields are `id` and `callback`: <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>id</td><td>string</td><td>Identifies the response for this request in the map of batch responses. If one is not provided, the system generates a random ID.</td></tr><tr><td>callback(individualResponse, rawBatchResponse)</td><td>function</td><td>individualResponse is the response for this request only. Its format is defined by the API method being called. rawBatchResponse is the raw batch ID-response map as a string. It contains all responses to all requests in the batch.</td></tr></table>

`gapi.client.Batch.then(`*onFulfilled*, *onRejected*, *context*`)`

For more information about using promises, see Using Promises.

`gapi.client.Batch.execute(`*callback*`)`

Executes all requests in the batch. The supplied callback is executed on success or failure.

Name	Type	Description
`callback(responseMap, rawBatchResponse)`	function	The callback to execute when the batch returns. `responseMap` is an ID-response map of each requests response. `rawBatchResponse` is the same response, but as an unparsed JSON-string.

gapi.client.RpcBatch

> **Important:** This class has been deprecated. Please see `gapi.client.Batch` instead.

Represents an RPC Batch operation. Individual RPCs are added with the `add` method and the batch is executed using `execute`. This class defines two methods:

gapi.client.RpcBatch.add(*rpcRequest*, *opt_params*)

Adds a `gapi.client.RpcRequest` to the batch.

Arguments:

Name	Type	Description
rpcRequest	gapi.client.RpcRequest	The RPC request to add to this batch. This parameter is required.
opt_params	Object	Optional extra parameters for this batch entry. Accepted fields are `id` and `callback`: <table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>`id`</td><td>string</td><td>Identifies the response for this request in the map of batch responses. If one is not provided, the system generates a random ID.</td></tr><tr><td>`callback(individualResponse, rawBatchResponse)`</td><td>function</td><td>`individualResponse` is the response for this request only. Its format is defined by the API method being called. `rawBatchResponse` is the raw batch ID-response map as a string. It contains all responses to all requests in the batch.</td></tr></table>

gapi.client.RpcBatch.execute(*callback*)

Executes all requests in the batch. The supplied callback is executed on success or failure.

Name	Type	Description
`callback(responseMap, rawBatchResponse)`	function	The callback to execute when the batch returns. `responseMap` is an ID-response map of each requests response. `rawBatchResponse` is the same response, but as an unparsed JSON-string.