Push notifications

This document describes how to use push notifications that inform your application when a resource changes.

Overview

The Google Calendar API provides push notifications that let you monitor changes in resources. You can use this feature to improve the performance of your application. It lets you eliminate the extra network and compute costs involved with polling resources to determine if they have changed. Whenever a watched resource changes, the Google Calendar API notifies your application.

To use push notifications, you must do two things:

- Set up your receiving URL or "webhook" callback receiver.

  This is an HTTPS server that handles the API notification messages that are triggered when a resource changes.

- Set up a (notification channel (https://cloud.google.com/monitoring/support/notification-options)) for each resource endpoint you want to watch.

  A channel specifies routing information for notification messages. As part of the channel setup, you must identify the specific URL where you want to receive notifications. Whenever a channel's resource changes, the Google Calendar API sends a notification message as a `POST` request to that URL.

Currently, the Google Calendar API supports notifications for changes to the Acl (/workspace/calendar/v3/reference/acl/watch), CalendarList (/workspace/calendar/v3/reference/calendarList/watch), Events (/workspace/calendar/v3/reference/events/watch), and Settings (/workspace/calendar/v3/reference/settings/watch) resources.

Create notification channels

To request push notifications, you must set up a notification channel for each resource you want to monitor. After your notification channels are set up, the Google Calendar API informs your application when any watched resource changes.

Make watch requests

Each watchable Google Calendar API resource has an associated `watch` method at a URI of the following form:

```
https://www.googleapis.com/API_NAME/API_VERSION/RESOURCE_PATH/watch
```

To set up a notification channel for messages about changes to a particular resource, send a `POST` request to the `watch` method for the resource.

Each notification channel is associated both with a particular user and a particular resource (or set of resources). A `watch` request won't be successful unless the current user owns or has permission to access this resource.

Example

Start watching for changes to a collection of events on a given calendar:

```
POST https://www.googleapis.com/calendar/v3/calendars/my_calendar@gmail.com/even
Authorization: Bearer auth_token_for_current_user
Content-Type: application/json

{
  "id": "01234567-89ab-cdef-0123456789ab", // Your channel ID.
  "type": "web_hook",
  "address": "https://mydomain.com/notifications", // Your receiving URL.
  ...
  "token": "target=myApp-myCalendarChannelDest", // (Optional) Your channel toke
  "expiration": 1426325213000 // (Optional) Your requested channel expiration ti
}
```

Required properties

With each `watch` request, you must provide these fields:

- An `id` property string that uniquely identifies this new notification channel within your project. We recommend using a universally unique identifier (UUID (http://en.wikipedia.org/wiki/UUID)) or any similar unique string. Maximum length: 64 characters.

  The ID value you set is echoed back in the `X-Goog-Channel-Id` HTTP header of every notification message that you receive for this channel.

- A `type` property string set to the value `web_hook`.
- An `address` property string set to the URL that listens and responds to notifications for this notification channel. This is your webhook callback URL, and it must use HTTPS.

  Note that the Google Calendar API is able to send notifications to this HTTPS address only if there's a valid SSL certificate installed on your web server. Invalid certificates include:

  - Self-signed certificates.
  - Certificates signed by an untrusted source.
  - Certificates that have been revoked.
  - Certificates that have a subject that doesn't match the target hostname.

Optional properties

You can also specify these optional fields with your `watch` request:

- A `token` property that specifies an arbitrary string value to use as a channel token. You can use notification channel tokens for various purposes. For example, you can use the token to verify that each incoming message is for a channel that your application created —to ensure that the notification is not being spoofed—or to route the message to the right destination within your application based on the purpose of this channel. Maximum length: 256 characters.

  The token is included in the `X-Goog-Channel-Token` HTTP header in every notification message that your application receives for this channel.

  If you use notification channel tokens, we recommend that you:

  - Use an extensible encoding format, such as URL query parameters. Example:

    forwardTo=hr&createdBy=mobile

  - Don't include sensitive data such as OAuth tokens.

★ **Note:** If you must send highly-sensitive data, make sure it's encrypted before adding it to the token.

- An `expiration` property string set to a Unix timestamp (http://en.wikipedia.org/wiki/Unix_time) (in milliseconds) of the date and time when you want the Google Calendar API to stop sending messages for this notification channel.

  If a channel has an expiration time, it's included as the value of the `X-Goog-Channel-Expiration` HTTP header (in human-readable format) in every notification message that your application receives for this channel.

For more details on the request, refer to the `watch` method for the Acl (/workspace/calendar/v3/reference/acl/watch), CalendarList (/workspace/calendar/v3/reference/calendarList/watch), Events (/workspace/calendar/v3/reference/events/watch), and Settings (/workspace/calendar/v3/reference/settings/watch) resources in the API Reference.

Watch response

If the `watch` request successfully creates a notification channel, it returns an HTTP `200 OK` status code.

The message body of the watch response provides information about the notification channel you just created, as shown in the example below.

```
{
  "kind": "api#channel",
  "id": "01234567-89ab-cdef-0123456789ab"", // ID you specified for this channel
  "resourceId": "o3hgv1538sdjfh", // ID of the watched resource.
  "resourceUri": "https://www.googleapis.com/calendar/v3/calendars/my_calendar@g
  "token": "target=myApp-myCalendarChannelDest", // Present only if one was prov
  "expiration": 1426325213000, // Actual expiration time as Unix timestamp (in m
}
```

In addition to the properties you sent as part of your request, the returned information also includes the `resourceId` and `resourceUri` to identify the resource being watched on this notification channel.

Note: The `resourceId` property is a stable, version-independent identifier for the resource. The `resourceUri` property is the canonical URI of the watched resource in the context of the current API version, so it's version-specific.

You can pass the returned information to other notification channel operations, such as when you want to stop receiving notifications (#stopping).

For more details on the response, refer to the `watch` method for the Acl (/workspace/calendar/v3/reference/acl/watch), CalendarList (/workspace/calendar/v3/reference/calendarList/watch), Events (/workspace/calendar/v3/reference/events/watch), and Settings (/workspace/calendar/v3/reference/settings/watch) resources in the API Reference.

Sync message

After creating a notification channel to watch a resource, the Google Calendar API sends a `sync` message to indicate that notifications are starting. The `X-Goog-Resource-State` HTTP header value for these messages is `sync`. Due to network timing issues, it's possible to receive the `sync` message even before you receive the `watch` method response.

It's safe to ignore the `sync` notification, but you can also use it. For example, if you decide you don't want to keep the channel, you can use the `X-Goog-Channel-ID` and `X-Goog-Resource-ID` values in a call to stop receiving notifications (#stopping). You can also use the `sync` notification to do some initialization to prepare for later events.

The format of `sync` messages the Google Calendar API sends to your receiving URL is shown below.

```
POST https://mydomain.com/notifications // Your receiving URL.
X-Goog-Channel-ID: channel-ID-value
X-Goog-Channel-Token: channel-token-value
X-Goog-Channel-Expiration: expiration-date-and-time // In human-readable format.
X-Goog-Resource-ID: identifier-for-the-watched-resource
X-Goog-Resource-URI: version-specific-URI-of-the-watched-resource
X-Goog-Resource-State: sync
```

```
X-Goog-Message-Number: 1
```

Sync messages always have an `X-Goog-Message-Number` HTTP header value of `1`. Each subsequent notification for this channel has a message number that's larger than the previous one, though the message numbers will not be sequential.

Renew notification channels

A notification channel can have an expiration time, with a value determined either by your request or by any Google Calendar API internal limits or defaults (the more restrictive value is used). The channel's expiration time, if it has one, is included as a Unix timestamp (http://en.wikipedia.org/wiki/Unix_time) (in milliseconds) in the information returned by the `watch` method. In addition, the expiration date and time is included (in human-readable format) in every notification message your application receives for this channel in the `X-Goog-Channel-Expiration` HTTP header.

Currently, there's no automatic way to renew a notification channel. When a channel is close to its expiration, you must replace it with a new one by calling the `watch` method. As always, you must use a unique value for the `id` property of the new channel. Note that there's likely to be an "overlap" period of time when the two notification channels for the same resource are active.

Receive notifications

Whenever a watched resource changes, your application receives a notification message describing the change. The Google Calendar API sends these messages as HTTPS `POST` requests to the URL you specified as the `address` property (#address_prop) for this notification channel.

Note: Notification delivery HTTPS requests specify a user agent of `APIs-Google` and respect robots.txt directives, as described in APIs Google User Agent (/search/docs/crawling-indexing/apis-user-agent).

Interpret the notification message format

All notification messages include a set of HTTP headers that have `X-Goog-` prefixes. Some types of notifications can also include a message body.

Headers

Notification messages posted by the Google Calendar API to your receiving URL include the following HTTP headers:

Header	Description
Always present	
`X-Goog-Channel-ID`	UUID or other unique string you provided to identify this notification channel.
`X-Goog-Message-Number`	Integer that identifies this message for this notification channel. Value is always `1` for `sync` messages. Message numbers increase for each subsequent message on the channel, but they're not sequential.
`X-Goog-Resource-ID`	An opaque value identifying the watched resource. This ID is stable across API versions.
`X-Goog-Resource-State`	The new resource state that triggered the notification. Possible values: `sync`, `exists`, or `not_exists`.
`X-Goog-Resource-URI`	An API-version-specific identifier for the watched resource.
Sometimes present	
`X-Goog-Channel-Expiration`	Date and time of notification channel expiration, expressed in human-readable format. Only present if defined.
`X-Goog-Channel-Token`	Notification channel token that was set by your application, and that you can use to verify the notification source. Only present if defined.

Notification messages posted by the Google Calendar API to your receiving URL do not include a message body. These messages do not contain specific information about updated resources, you will need to make another API call to see the full change details.

Examples

Change notification message for modified collection of events:

```
POST https://mydomain.com/notifications // Your receiving URL.
Content-Type: application/json; utf-8
Content-Length: 0
```

```
X-Goog-Channel-ID: 4ba78bf0-6a47-11e2-bcfd-0800200c9a66
X-Goog-Channel-Token: 398348u3tu83ut8uu38
X-Goog-Channel-Expiration: Tue, 19 Nov 2013 01:13:52 GMT
X-Goog-Resource-ID:  ret08u3rv24htgh289g
X-Goog-Resource-URI: https://www.googleapis.com/calendar/v3/calendars/my_calenda
X-Goog-Resource-State:  exists
X-Goog-Message-Number: 10
```

Respond to notifications

To indicate success, you can return any of the following status codes: **`200`**, **`201`**, **`202`**, **`204`**, or **`102`**.

If your service uses Google's API client library (/admin-sdk/directory/v1/libraries) and returns **`500`**,**`502`**, **`503`**, or **`504`**, the Google Calendar API retries with exponential backoff (https://www.google.com/search?q=define%3Aexponential+backoff&oq=define%3Aexponential+backoff). Every other return status code is considered to be a message failure.

Understand Google Calendar API notification events

This section provides details on the notification messages you can receive when using push notifications with the Google Calendar API.

X-Goog-Resource-State	Applies to	Delivered when
`sync`	ACLs, Calendar lists, Events, Settings.	A new channel was successfully created. You can expect to start receiving notifications for it.
`exists`	ACLs, Calendar lists, Events, Settings.	There was a change to a resource. Possible changes include the creation of a new resource, or the modification or deletion of an existing resource.

Stop notifications

The **`expiration`** property controls when the notifications stop automatically. You can choose to stop receiving notifications for a particular channel before it expires by calling the **`stop`** method at the following URI:

```
https://www.googleapis.com/calendar/v3/channels/stop
```

This method requires that you provide at least the channel's `id` and the `resourceId` properties, as shown in the example below. Note that if the Google Calendar API has several types of resources that have `watch` methods, there's only one `stop` method.

Only users with the right permission can stop a channel. In particular:

- If the channel was created by a regular user account, only the same user from the same client (as identified by the OAuth 2.0 client IDs from the auth tokens) who created the channel can stop the channel.
- If the channel was created by a service account, any user from the same client can stop the channel.

The following code sample shows how to stop receiving notifications:

```
POST https://www.googleapis.com/calendar/v3/channels/stop

Authorization: Bearer CURRENT_USER_AUTH_TOKEN
Content-Type: application/json

{
  "id": "4ba78bf0-6a47-11e2-bcfd-0800200c9a66",
  "resourceId": "ret08u3rv24htgh289g"
}
```

Special considerations

When working with push notifications, keep the following in mind:

Events and ACLs are per-calendar If you want to get notified about all event or ACL changes for calendars A and B, you need to separately subscribe to the events/ACL collections for A and for B.

Settings and calendar lists are per-user Settings and calendar lists only have one collection per user, so you can subscribe just once.

Also, you won’t be notified when you gain access to a new collection (for example, a new calendar) although you *will* be notified if that calendar is added to the calendar list (assuming you are subscribed to the calendar list collection).

Notifications are not 100% reliable. Expect a small percentage of messages to get dropped under normal working conditions. Make sure to handle these missing messages gracefully, so that the application still syncs even if no push messages are received.