

Event types

This page explains the `eventType` property, and specifications of event types available in the Google Calendar API.

Google Calendar allows users to create generic events, as well as events that are designed for specific use cases and with custom properties.

The event type can be discovered in the following places in the API:

- All events are returned with an `eventType` (`/workspace/calendar/api/v3/reference/events#eventType`).
- `eventType` (`/workspace/calendar/api/v3/reference/events/insert#eventType`) needs to be set when creating or updating an event resource. If unset, 'default' type will be used.
- `eventTypes` (`/workspace/calendar/api/v3/reference/events/list#eventTypes`) can be specified in an `Events:list` (`/workspace/calendar/api/v3/reference/events/list`) call to list events of specific types. If no type is specified, all event types will be returned.
- `eventTypes` (`/workspace/calendar/api/v3/reference/events/watch#eventTypes`) can be specified in an `Events:watch` (`/workspace/calendar/api/v3/reference/events/watch`) call to subscribe to updates on events of specific types. If no type is specified, the request will result in subscribing to all event types.

Event type cannot be changed after an event is created. Different event types have different restrictions, and API operations that work for some event types might not work for the others.

Default event

Events with `default` event type are created and used as one of the main resources of Google Calendar API. They support a wide range of properties (`/workspace/calendar/api/v3/reference/events#properties`) which can be used to further customize the event.

See Create events (`/workspace/calendar/api/guides/create-events`) to start working with Google Calendar events.

Birthday

Birthdays are special all-day events with an annual recurrence.

Users can manually create birthday events on Google Calendar. In addition, the birthday information syncs with Google Calendar when users add a person and include their birthday and other significant dates in [Google Contacts](https://contacts.google.com) (<https://contacts.google.com>). Users' own birthday is also synced to Google Calendar from their [Google Account profile](https://myaccount.google.com/profile) (<https://myaccount.google.com/profile>).

Google Calendar API supports [`get`](/workspace/calendar/api/v3/reference/events/get) (</workspace/calendar/api/v3/reference/events/get>), [`instances`](/workspace/calendar/api/v3/reference/events/instances) (</workspace/calendar/api/v3/reference/events/instances>), and [`list`](/workspace/calendar/api/v3/reference/events/list) (</workspace/calendar/api/v3/reference/events/list>) methods for reading birthday events. [`eventTypes`](/workspace/calendar/api/v3/reference/events/list#eventTypes) (</workspace/calendar/api/v3/reference/events/list#eventTypes>) can be set to `'birthday'` to list only birthday events. If no type is specified, birthdays will be listed alongside all other event types.

In the returned Event objects, inspect the [`birthdayProperties`](/workspace/calendar/api/v3/reference/events#birthdayProperties) (</workspace/calendar/api/v3/reference/events#birthdayProperties>) field for more details about this special event. `birthdayProperties` has the following fields:

- [`type`](/workspace/calendar/api/v3/reference/events#birthdayProperties.type) (</workspace/calendar/api/v3/reference/events#birthdayProperties.type>): Type of this special event, whether it's a birthday, an anniversary, or another significant date.
- [`customTypeName`](/workspace/calendar/api/v3/reference/events#birthdayProperties.customTypeName) (</workspace/calendar/api/v3/reference/events#birthdayProperties.customTypeName>): User-specified label for this special event. This is populated if `type` is set to `'custom'`.
- [`contact`](/workspace/calendar/api/v3/reference/events#birthdayProperties.contact) (</workspace/calendar/api/v3/reference/events#birthdayProperties.contact>): Resource name of the contact this special event is linked to, if any. This has the format `'people/c12345'` and can be used to fetch contact details from the [People API](/people) (</people>).

The API allows creating birthday events using the [`insert`](/workspace/calendar/api/v3/reference/events/insert) (</workspace/calendar/api/v3/reference/events/insert>) method with the following specifications:

- [`eventType`](/workspace/calendar/api/v3/reference/events#eventType) (</workspace/calendar/api/v3/reference/events#eventType>) is set to `'birthday'`.
- [`start`](/workspace/calendar/api/v3/reference/events#start) (</workspace/calendar/api/v3/reference/events#start>) and [`end`](/workspace/calendar/api/v3/reference/events#end) (</workspace/calendar/api/v3/reference/events#end>) fields need to define an all-day event which spans exactly one day.

- visibility (/workspace/calendar/api/v3/reference/events#visibility) field value must be 'private'.
- transparency (/workspace/calendar/api/v3/reference/events#transparency) field value must be 'transparent'.
- Need to have an annual recurrence, which means that the recurrence (/workspace/calendar/api/v3/reference/events#recurrence) field must be 'RRULE:FREQ=YEARLY'. Birthday events falling on 29th of February must have the following recurrence rule: 'RRULE:FREQ=YEARLY;BYMONTH=2;BYMONTHDAY=-1'.
- Can have a colorId (/workspace/calendar/api/v3/reference/events#colorId), summary (/workspace/calendar/api/v3/reference/events#summary) and reminders (/workspace/calendar/api/v3/reference/events#reminders).
- Can have birthdayProperties (/workspace/calendar/api/v3/reference/events#birthdayProperties). If specified, type (/workspace/calendar/api/v3/reference/events#birthdayProperties.type) must be 'birthday', and both customTypeName (/workspace/calendar/api/v3/reference/events#birthdayProperties.customTypeName) and contact (/workspace/calendar/api/v3/reference/events#birthdayProperties.contact) must be empty.
- Cannot have any other event properties.

The API allows updating the colorId (/workspace/calendar/api/v3/reference/events/update#colorId), summary (/workspace/calendar/api/v3/reference/events/update#summary) and reminders (/workspace/calendar/api/v3/reference/events/update#reminders) of birthday events using the update (/workspace/calendar/api/v3/reference/events/update) and patch (/workspace/calendar/api/v3/reference/events/patch) methods. start (/workspace/calendar/api/v3/reference/events/update#start) and end (/workspace/calendar/api/v3/reference/events/update#end) fields can also be updated to change the event date. In this case, the new values need to define an all-day event which spans exactly one day. Timing details of a birthday event cannot be updated if the event is linked to a contact (/workspace/calendar/api/v3/reference/events#birthdayProperties.contact), or its type (/workspace/calendar/api/v3/reference/events#birthdayProperties.type) is 'self'.

Google Calendar API doesn't allow creating birthday events with custom birthdayProperties (/workspace/calendar/api/v3/reference/events#birthdayProperties), or updating these properties. Significant dates can be edited with the People API (/people), and the changes are synced with Google Calendar. Similarly, users can edit their own birthday on their Google Account profile

(<https://myaccount.google.com/profile>), and it syncs with Google Calendar.

Requests that try to create or update a birthday in an unsupported way will fail. In this case, inspect the error message to identify the issue.

The API supports [import](/workspace/calendar/api/v3/reference/events/import) (/workspace/calendar/api/v3/reference/events/import) operation for birthday events; however, the event will be imported as a default event. In other words, the [eventType](/workspace/calendar/api/v3/reference/events#eventType) (/workspace/calendar/api/v3/reference/events#eventType) will be 'default'.

The API supports the [watch](/workspace/calendar/api/v3/reference/events/watch) (/workspace/calendar/api/v3/reference/events/watch) method to subscribe to changes on birthday events on Google Calendar. [eventTypes](/workspace/calendar/api/v3/reference/events/watch#eventTypes) (/workspace/calendar/api/v3/reference/events/watch#eventTypes) can be set to 'birthday' to subscribe to updates on birthday events. If no type is specified, all event types, including birthdays, will be subscribed to.

Birthday events can be deleted using the [delete](/workspace/calendar/api/v3/reference/events/delete) (/workspace/calendar/api/v3/reference/events/delete) method of the Google Calendar API. Deleting a birthday event from Google Calendar doesn't affect data on [Google Contacts](https://contacts.google.com) (<https://contacts.google.com>) or [Google Account profile](https://myaccount.google.com/profile) (<https://myaccount.google.com/profile>).

Changing the organizer of a birthday event using [move](/workspace/calendar/api/v3/reference/events/move) (/workspace/calendar/api/v3/reference/events/move) or [update](/workspace/calendar/api/v3/reference/events/update) (/workspace/calendar/api/v3/reference/events/update) methods is not supported.

Events from Gmail

Events [automatically generated from Gmail](https://support.google.com/calendar/answer/6084018) (<https://support.google.com/calendar/answer/6084018>) have the 'fromGmail' event type.

Google Calendar API doesn't allow creating this event type using the [insert](/workspace/calendar/api/v3/reference/events/insert) (/workspace/calendar/api/v3/reference/events/insert) method.

The API allows updating the [colorId](/workspace/calendar/api/v3/reference/events/update#colorId) (/workspace/calendar/api/v3/reference/events/update#colorId), [reminders](/workspace/calendar/api/v3/reference/events/update#reminders) (/workspace/calendar/api/v3/reference/events/update#reminders), [visibility](/workspace/calendar/api/v3/reference/events/update#visibility) (/workspace/calendar/api/v3/reference/events/update#visibility), [transparency](/workspace/calendar/api/v3/reference/events/update#transparency) (/workspace/calendar/api/v3/reference/events/update#transparency), [status](/workspace/calendar/api/v3/reference/events/update#status) (/workspace/calendar/api/v3/reference/events/update#status), [attendees](/workspace/calendar/api/v3/reference/events/update#attendees) (/workspace/calendar/api/v3/reference/events/update#attendees), [private](/workspace/calendar/api/v3/reference/events/update#private) (/workspace/calendar/api/v3/reference/events/update#extendedProperties.private) and [shared](/workspace/calendar/api/v3/reference/events/update#shared)

(/workspace/calendar/api/v3/reference/events/update#extendedProperties.shared) extended properties using the [update](#) (/workspace/calendar/api/v3/reference/events/update) and [patch](#) (/workspace/calendar/api/v3/reference/events/patch) methods.

The API supports [get](#) (/workspace/calendar/api/v3/reference/events/get) and [list](#) (/workspace/calendar/api/v3/reference/events/list) methods for reading events from Gmail. [eventTypes](#) (/workspace/calendar/api/v3/reference/events/list#eventTypes) can be set to 'fromGmail' to list only events generated from Gmail. If no type is specified, events from Gmail will be listed alongside all other event types.

The API supports the [watch](#) (/workspace/calendar/api/v3/reference/events/watch) method to subscribe to changes on events from Gmail on Google Calendar. If no type is specified, all event types, including 'fromGmail', will be subscribed to.

Events from Gmail can be deleted using the [delete](#) (/workspace/calendar/api/v3/reference/events/delete) method of the Google Calendar API.

Changing the organizer of an event from Gmail using [move](#) (/workspace/calendar/api/v3/reference/events/move) or [update](#) (/workspace/calendar/api/v3/reference/events/update) methods is not supported.

Focus time, out of office and working location

The Google Calendar API can be used to create and manage events that show the status of Google Calendar users.

These features are only available on primary calendars, and to some Google Calendar users. See [Manage focus time, out of office, and working location events](#) (/workspace/calendar/api/guides/calendar-status) to learn more.

Explore event types in Google Apps Script

[Google Apps Script](#) (/apps-script/overview) is a JavaScript-based cloud scripting language that lets you build business applications that integrate with Google Workspace. Scripts are developed in a browser-based code editor, and they are stored and run on Google's servers. See also [Google Apps Script quickstart](#) (/workspace/calendar/api/quickstart/apps-script) to start using Apps Script to send requests to the Google Calendar API.

The following instructions describe how to read and manage events using the [Google Calendar API](#) (/apps-script/advanced/calendar) as an advanced service in Google Apps Script. For a complete list of Google Calendar API resources and methods, see the [reference documentation](#) (/workspace/calendar/api/guides/overview).

Create and set up the script

1. Create a script by going to script.google.com/create (https://script.google.com/create).
2. On the left pane next to **Services**, click Add a service **+** .
3. Select **Google Calendar API** and click **Add**.
4. After enabled, the API appears on the left pane. Available methods and classes in the API can be listed using the *Calendar* keyword in the editor.

(Optional) Update the Google Cloud project

Each Google Apps Script project has an associated Google Cloud project. Your script can use the default project that Google Apps Script automatically creates. If you want to use a custom Google Cloud project, see [Switch to a different standard Cloud project](#) (/apps-script/guides/cloud-platform-projects#switch_to_a_different_standard). After setting the Google Cloud project, select **Editor** **< >** on the left side to navigate back to the code editor.

Add code to the script

The following code sample shows how to list, read, and create events with different `eventType` values.

1. Paste the following into the code editor.

```
const CALENDAR_ID = 'CALENDAR_ID' || 'primary';

/** Lists default events. */
function listDefaultEvents() {
  listEvents('default');
}

/** Lists birthday events. */
```

```

function listBirthdays() {
    listEvents('birthday');
}

/** Lists events from Gmail. */
function listEventsFromGmail() {
    listEvents('fromGmail');
}

/**
 * Lists events with the given event type. If no type is specified, lists :
 * See https://developers.google.com/workspace/calendar/api/v3/reference/e
 */
function listEvents(eventType = undefined) {
    // Query parameters for the list request.
    const optionalArgs = {
        eventTypes: eventType ? [eventType] : undefined,
        singleEvents: true,
        timeMax: '2024-07-30T00:00:00+01:00',
        timeMin: '2024-07-29T00:00:00+01:00',
    };
    try {
        var response = Calendar.Events.list(CALENDAR_ID, optionalArgs);
        response.items.forEach(event => console.log(event));
    } catch (exception) {
        console.log(exception.message);
    }
}

/**
 * Reads the event with the given eventId.
 * See https://developers.google.com/workspace/calendar/api/v3/reference/e
 */
function readEvent() {
    try {
        var response = Calendar.Events.get(CALENDAR_ID, 'EVENT_ID ✎');
        console.log(response);
    } catch (exception) {
        console.log(exception.message);
    }
}

/** Creates a default event. */
function createDefaultEvent() {
    const event = {

```

```

        start: { dateTime: '2024-07-30T10:30:00+01:00' },
        end: { dateTime: '2024-07-30T12:30:00+01:00' },
        description: 'Created from Apps Script.',
        eventType: 'default',
        summary: 'Sample event',
    }
    createEvent(event);
}

/** Creates a birthday event. */
function createBirthday() {
    const event = {
        start: { date: '2024-01-29' },
        end: { date: '2024-01-30' },
        eventType: 'birthday',
        recurrence: ["RRULE:FREQ=YEARLY"],
        summary: "My friend's birthday",
        transparency: "transparent",
        visibility: "private",
    }
    createEvent(event);
}

/**
 * Creates a Calendar event.
 * See https://developers.google.com/workspace/calendar/api/v3/reference/e
 */
function createEvent(event) {

    try {
        var response = Calendar.Events.insert(event, CALENDAR_ID);
        console.log(response);
    } catch (exception) {
        console.log(exception.message);
    }
}

```

Replace the following:

- **CALENDAR_ID**: Email address of the calendar to retrieve and create events on. This constant is initially set to 'primary', which is a keyword to access the primary calendar of the signed-in user. Changing this value lets you read events on the calendars of other users you have access to.

- *EVENT_ID*: Id of the event. You can call the [Events:list](#) (/workspace/calendar/api/v3/reference/events/list) to retrieve event ids.

Run the code sample

1. Above the code editor, select the function to run from the drop-down menu, and click **Run**.
2. In the first execution, it prompts you to authorize access. Review and allow Apps Script to access your calendar.
3. You can inspect the results of the script execution in the **Execution Log** that appears at the bottom of the window.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-07-30 UTC.