

Google Calendar Events API Documentation

Introduction

Welcome to the documentation for the Google Calendar Events API. This API allows you to manage events in your Google Calendar. This document provides information on how to use the API, including available endpoints, required inputs, possible outputs, error handling, setup, and running instructions.

API Endpoints

1. Get Event by ID

- **Endpoint URL:** `GET /api/events/{eventId}`
- **Description:** Retrieve a specific event from the user's Google Calendar by its ID.
- **Required Inputs:**
 - `eventId` (String): The unique identifier of the event.
- **Possible Outputs:** The event details or an error message.
- **Error Handling:** Possible error responses include `200 OK` for a successful request or `400 Bad Request` for invalid inputs.

2. Create Event

- **Endpoint URL:** `POST /api/events`
- **Description:** Create a new event in the user's Google Calendar.
- **Required Inputs:**
 - Request body in the format of a `GoogleCalendarCreate` object with the following properties:
 - `Summary` (String, Max Length: 200): A summary of the event.
 - `Description` (String, Max Length: 500): A detailed description of the event.
 - `Location` (String, Max Length: 100): The location where the event takes place.
 - `Start` (Date, Required): The start date and time of the event.
 - `End` (Date, Required): The end date and time of the event.
 - `TimeZone` (String, Max Length: 100): The time zone in which the event occurs.
- **Possible Outputs:** The created event or an error message.
- **Error Handling:** Possible error responses include `201 Created` for a successful creation or `400 Bad Request` for invalid data or if events are created in the past or on Fridays/Saturdays.

3. Get Events

- **Endpoint URL:** `GET /api/events`
- **Description:** Retrieve a list of events from the user's Google Calendar.
- **Required Inputs:**
 - `fromDate` (Date): Start date for filtering events.
 - `toDate` (Date): End date for filtering events.
 - `pageToken` (String): Token for paginating results.
 - `resultsCount` (Integer): Number of results to retrieve.
 - `searchQuery` (String): A search query to filter events.
- **Possible Outputs:** A list of events matching the filter criteria and a `NextPageToken` for pagination.

- **Error Handling:** Possible error responses include **200 OK** for a successful request or **400 Bad Request** for invalid inputs.

4. Delete Event by ID

- **Endpoint URL:** **DELETE** `/api/events/{eventId}`
- **Description:** Delete a specific event from the user's Google Calendar by its ID.
- **Required Inputs:**
 - **eventId** (String): The unique identifier of the event.
- **Possible Outputs:** No content or an error message.
- **Error Handling:** Possible error responses include **204 No Content** for a successful deletion or **400 Bad Request** for errors.

Setup and Running Instructions

Prerequisites

- .NET Core runtime and development tools
- Access to Google Calendar API with credentials

Download and Install

- Clone the project repository from <https://github.com/Kyrillos-Gaber/GoogleCalenderEvents.git>
- Build the project using **dotnet build**.

Configuration

- Update the app settings with Google Calendar API credentials and other configurations.
- **Authentication:** The service uses OAuth 2.0 for Google Calendar API access. It stores credentials in a file.

Run the Application

- Start the application using **dotnet run**.
- The API will be available at **<https://localhost:7220>**.

Testing Instructions

- Use tools like Postman to test the API endpoints.
- Provide required input parameters in the request.
- Check the responses and error handling.

Conclusion

Thank you for using the Google Calendar Events API. If you have any questions or need further assistance, please don't hesitate to contact us.

Please ensure you configure the necessary Google Calendar API credentials for the service to function correctly.