# .NET Core Calendar API

A company is launching a new service that provides scheduling appointments and meetings of an individual, similar to popular services such as Gmail Calendar, Outlook, etc. A 'calendar' needs to be created, and as part of this challenge, you are required to come up with a service to maintain this calendar.

As step 1, create a service that supports REST APIs for creating, deleting, and updating events in a calendar. An event will have details such as event name, scheduled time, scheduled location, members, etc.. A few more APIs required would be to fetch the event details, the events in a particular location, sort the events as per the time, finding all events for a particular organizer, etc. A detailed explanation about the APIs and data is given below.

Each event object is a JSON object with the following keys -

1. name - Name of the event. [STRING]
2. time - Scheduled time for the event in UTC (GMT + 0). [EPOCH INTEGER]
3. location - Location of the event. [STRING]
4. members - String of member names separated by a comma. [ARRAY OF STRINGS]
5. eventOrganizer - Name of the organizer of the event. [STRING]
6. id - Unique ID of the event as generated by the system. [INTEGER]

Example

{

  "name": "Agenda discussion",

  "time": 1573843210,

  "location": “Miami”,

  "members": “Any,Jay”

  “eventOrganizer”: “Sam”,

  "id": "1"

}

## APIs

The following APIs need to be implemented:

1. Adding a new event - POST request should be created to add a new event. The API endpoint would be /calendar. The request body contains the details of the event. HTTP response should be 201.
2. Deleting any event by id -  DELETE request to endpoint /calendar/{id} should delete the event. If the item does not exist return not found.
3. Editing the event - PUT request to endpoint /calendar/{id}. The request body would contain the id of the event and the information that needs to be edited. If the item does not exist return not found.
4. Getting all events - GET request to endpoint /calendar should return all the events in the system. The HTTP response code should be 200. If no event exists, return the empty array.
5. Getting all events of the organizer - GET request to endpoint /calendar/query?eventOrganizer={eventOrganizer} should return the entire list of events organized by this organizer. The HTTP response code should be 200. For empty response return empty array.
6. Getting event by id - GET request to endpoint /calendar/query?id={id} should return the details of the event with this unique id. The HTTP response code should be 200.
7. Getting all events by location - GET request to endpoint /calendar/query?location={location} should return the entire list of events happening at that location. The HTTP response code should be 200.
8. Getting event by name - GET request to endpoint /calendar/query?name={name}should return the details of the event with this name. The HTTP response code should be 200.
9. Sort the event as per the time - GET request to endpoint /calendar/sort should return the events sorted in descending order of time.

## Deliverables

1. The source code published in your GitHub account.
2. A working Docker image published in your account at hub.docker.com with the instructions to run it.

You can send the final links to [hr@keesingtechnologies.com](mailto:hr@keesingtechnologies.com).