

# API

04/13/2026 1:10 pm PDT

**PRO** ⓘ API requires a Business subscription. [Learn more.](#)

## Getting started

### Rate Limits

Each API key is limited to **5 requests per second** across all endpoints combined. Requests over the limit return HTTP `429` with a `Retry-After` header. All responses also include `X-RateLimit-Limit` and `X-RateLimit-Remaining` headers so you can pace yourself.

### Credentials

The Heylo team will issue you with an API key for your group or organization (if on the [Organization](#) plan). Replace `HEYLO_API_KEY` below with your API key. If you need to rotate or delete an API key, please [contact the Heylo team](#).

### Endpoints

All endpoints accept `POST` requests with a JSON body. The specific operation is selected via the `type` field in the request body.

- `https://api.heylo.com/apiv1-fetch`
- `https://api.heylo.com/apiv1-remove`
- `https://api.heylo.com/apiv1-update`

### Example Usage

Fetching events for a group:

```
curl -X POST \
-H "Authorization: Bearer HEYLO_API_KEY" \
-H "Content-Type: application/json" \
<https://api.heylo.com/apiv1-fetch> \
-d '{
  "communityId": "HEYLO_GROUP_ID",
  "type": "communityEvents"
}'
```

Removing a member from a group:

```
curl -X POST \\  
-H "Authorization: Bearer HEYLO_API_KEY" \\  
-H "Content-Type: application/json" \\  
<https://api.heylo.com/apiv1-remove> \\  
-d '{  
  "communityId": "HEYLO_GROUP_ID",  
  "type": "member",  
  "userId": "abc..."  
'
```

## Operations

### Community Events

#### Fetch

Returns events for a group, ordered by start time. Supports pagination via `lastEventId`.

**Endpoint:** `POST <https://api.heylo.com/apiv1-fetch >`

#### Parameters

- `type` : "communityEvents"
- `communityId` : the Heylo group ID
- `batchSize` (optional): max events to return – default 50, max 100
- `endTimestamp` (optional): unix timestamp in ms – only return events starting before this
- `lastEventId` (optional): pass the last event ID from the previous response to fetch the next page
- `startTimestamp` (optional): unix timestamp in ms – only return events starting after this

**Response:** `{ "events": [...] }`

### Event Attendance

#### Fetch

Returns attendance records for a single event.

**Endpoint:** `POST <https://api.heylo.com/apiv1-fetch >`

#### Parameters

- `type` : "eventAttendance"
- `communityId` : the Heylo group ID that owns the event
- `eventId` : the event ID

**Response:** `{ "attendance": [...] }`

# Members

## Remove

Removes a member from a group. Also cancels any pending invites previously created for the same [userId, communityId] pair.

**Endpoint:** `POST <https://api.heylo.com/apiv1-remove >`

### Parameters

- `type` : "member"
- `communityId` : the Heylo group ID from which to remove the member
- `userId` : the stable ID of the user in your system

# Roles

## Community Roles

### Fetch

Returns all roles defined for a group.

**Endpoint:** `POST <https://api.heylo.com/apiv1-fetch >`

### Parameters

- `type` : "communityRoles"
- `communityId` : the Heylo group ID

**Response:** `{ "roles": [...] }`

## Role Members

### Fetch

Returns the user IDs of members assigned to a specific role.

**Endpoint:** `POST <https://api.heylo.com/apiv1-fetch >`

### Parameters

- `type` : "roleMembers"
- `communityId` : the Heylo group ID
- `roleId` : the role ID

**Response:** `{ "memberUserIds": [...] }`

### Update

Add and/or remove members from a role in a single request. User IDs that aren't active or hidden

members of the community are silently dropped from the update.

**Endpoint:** POST <https://api.heylo.com/apiv1-update >

### Parameters

- `type` : "roleMembers"
- `communityId` : the Heylo group ID
- `roleId` : the role ID
- `addedUserIds` (optional): array of user IDs to add to the role – max 100 per request
- `removedUserIds` (optional): array of user IDs to remove from the role – max 100 per request

At least one of `addedUserIds` or `removedUserIds` must be non-empty.

**Response:** { "addedUserIds": [...], "removedUserIds": [...] } – the user IDs actually applied after filtering

---