



# Event Management API

1.0.0

OAS3

</static/swagger.yaml>

API documentation for the Event Management system.

## Servers

 ▾[Authorize](#)

## Users

[GET](#)

/users Get users with advanced search



### Parameters

[Try it out](#)**Name****Description****query**  
**string**  
*(query)*

Search string to match in username or email (case-insensitive).

**page**  
**integer**  
*(query)*

Page number for pagination (default: 1).

**size**  
**integer**  
*(query)*

Number of results per page (default: 20, max: 100).

### Responses

| Code | Description      | Links    |
|------|------------------|----------|
| 200  | A list of users. | No links |

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "username": "alice",  
    "email": "alice@example.com",  
    "bio": "Some short bio"  
  }  
]
```

**POST**

/users Create/Register a new user



## Parameters

[Try it out](#)

No parameters

**Request body** required

application/json



[Example Value](#) | [Schema](#)

```
{  
  "username": "john_doe",  
  "email": "john@example.com",  
  "password": "secret123",  
  "bio": "Short bio text"  
}
```

## Responses

| Code | Description               | Links    |
|------|---------------------------|----------|
| 201  | User created successfully | No links |
| 400  | Missing required fields   | No links |

**PUT**

/users Update the authenticated user's account

**Parameters****Try it out**

No parameters

**Request body** required**application/json****Example Value** | Schema

```
{  
  "email": "updated@example.com",  
  "password": "newSecret123",  
  "bio": "Updated bio text"  
}
```

**Responses****Code****Description****Links**

200

User updated successfully

*No links*

Media type

**application/json**

Controls Accept header.

**Example Value** | Schema

```
{  
  "message": "User updated successfully."  
}
```

401

Unauthorized access

*No links*

403

Unauthorized access. You can only update your own account.

*No links***DELETE**

/users Delete the authenticated user's account

**Parameters****Try it out**

No parameters

## Responses

| Code | Description               | Links    |
|------|---------------------------|----------|
| 200  | User deleted successfully | No links |
| 401  | Invalid or missing token  | No links |

POST

/users/login User login, returns JWT



## Parameters

Try it out

No parameters

## Request body required

application/json



## Example Value | Schema

```
{  
  "username": "john_doe",  
  "password": "secret123"  
}
```

## Responses

| Code | Description                   | Links    |
|------|-------------------------------|----------|
| 200  | Login successful, returns JWT | No links |

Media type

application/json



Controls Accept header.

## Example Value | Schema

```
{  
  "message": "string",  
  "token": "string"  
}
```

| Code  | Description                                     | Links             |
|---|---|-------------------|
| 400   | Username/password missing                       | No links          |
| 401   | Invalid credentials                             | No links          |
| <b>GET</b>  | /users/{username} Get a single user by username | ^                 |
| <b>Parameters</b>   |   | <b>Try it out</b> |
| Name  | Description                                     |                   |
| <b>username</b> * required<br>string<br>(path)  | The username of the user to fetch.              |                   |
|   | username  |                   |
| <b>Responses</b>  |   |                   |
| Code  | Description                                     | Links             |
| 200   | The user data                                   | No links          |
| Media type  |   |                   |
| application/json  |   | ▼                 |
| Controls Accept header.   |   |                   |
| <b>Example Value</b>  |   | Schema            |
| {<br>"username": "alice",<br>"email": "alice@example.com",<br>"bio": "Short bio"<br>} |   |                   |
| 404   | User not found                                  | No links          |

## Events

^

**POST**

/events Create a new event

**Parameters****Try it out**

No parameters

**Request body** required

application/json

[Example Value](#) | Schema

```
{  
  "group_id": 1,  
  "event_name": "Team Meeting",  
  "event_date": "2025-01-10",  
  "event_location": "Conference Room A",  
  "event_description": "Quarterly team meeting."  
}
```

**Responses****Code****Description****Links**

201

Event created successfully

*No links*

400

Missing required fields

*No links***PUT**

/events/{event\_id} Update an event

**Parameters****Try it out****Name****Description****event\_id** \* required**integer***(path)*

event\_id

**Request body** required

application/json

[Example Value](#) | Schema

```
{  
    "event_name": "Updated Team Meeting",  
    "event_date": "2025-01-15",  
    "event_location": "Updated Conference Room B",  
    "event_description": "Updated meeting description."  
}
```

## Responses

| Code | Description                | Links    |
|------|----------------------------|----------|
| 200  | Event updated successfully | No links |
| 400  | Missing required fields    | No links |

**DELETE** /events/{event\_id} Delete an event ↗ 

## Parameters

Try it out

| Name                                     | Description |
|--|-------------|
| event_id * required<br>integer<br>(path) | event_id    |

## Responses

| Code | Description                | Links    |
|------|----------------------------|----------|
| 200  | Event deleted successfully | No links |
| 404  | Event not found            | No links |

**GET** /events/user/groups Retrieve events for groups the user belongs to ↗ 

## Parameters

Try it out

No parameters

## Responses

| Code | Description                          | Links           |
|------|--------------------------------------|-----------------|
| 200  | List of events for the user's groups | <i>No links</i> |

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "event_id": 1,  
    "event_name": "Team Meeting"  
  }  
]
```

**GET**

/events/sorted Retrieve past and upcoming events



## Parameters

[Try it out](#)

No parameters

## Responses

| Code | Description                        | Links           |
|------|------------------------------------|-----------------|
| 200  | Past and upcoming events separated | <i>No links</i> |

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "past_events": [  
    {  
      "event_id": 1,  
      "event_name": "Team Meeting"  
    }  
  ]  
}
```

| Code | Description  | Links |
|------|--|-------|
|      | <pre>         "event_name": "Past Event"     }, ], "future_events": [     {         "event_id": 2,         "event_name": "Future Event"     } ] } </pre> |       |

**POST** /events/tags Add multiple tags to an event ^ 

---

**Parameters** **Try it out**

No parameters

---

**Request body** required application/json 

---

[Example Value](#) | [Schema](#)

```
{
  "event_id": 1,
  "tags": [
    "urgent",
    "team"
  ]
}
```

---

**Responses**

| Code | Description             | Links           |
|------|-------------------------|-----------------|
| 201  | Tags added successfully | <i>No links</i> |
| 400  | Missing required fields | <i>No links</i> |

**GET** /events/{event\_id}/tags Retrieve all tags for a specific event ^

---

**Parameters** **Try it out**

---

| Name  | Description |
|---|-------------|
| <b>event_id</b> * required<br>integer<br>(path) | event_id    |

## Responses

| Code | Description                | Links    |
|------|----------------------------|----------|
| 200  | List of tags for the event | No links |

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[  
  {  
    "tag_name": "urgent"  
  }  
]
```

PUT /events/{event\_id}/tags Update tags for a specific event ^ 

Parameters Try it out

| Name  | Description |
|---|-------------|
| <b>event_id</b> * required<br>integer<br>(path) | event_id    |

Request body required

application/json

Example Value | Schema

```
{  
  "tags": [  
    "updated",  
    "critical"  
  ]  
}
```

```
    }]
```

## Responses

| Code | Description               | Links    |
|------|---------------------------|----------|
| 200  | Tags updated successfully | No links |
| 400  | Tags are required         | No links |

GET

/events/tags/events/{tag\_name} Get all events with a specific tag



## Parameters

Try it out

| Name                                    | Description |
|---|-------------|
| tag_name * required<br>string<br>(path) | tag_name    |

## Responses

| Code | Description                 | Links    |
|------|-----------------------------|----------|
| 200  | List of events with the tag | No links |

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "event_id": 1,  
    "event_name": "Team Meeting"  
  }  
]
```

**GET**

/events/tags/popular Retrieve popular tags based on usage

**Parameters****Try it out**

No parameters

**Responses**

| Code | Description          | Links           |
|------|----------------------|-----------------|
| 200  | List of popular tags | <i>No links</i> |

Media type

application/json

Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "tag_name": "urgent",  
    "usage_count": 42  
  }  
]
```

## Membership

**POST**

/membership/ Add the authenticated user to a group as a Member

**Parameters****Try it out**

No parameters

**Request body** required

application/json

[Example Value](#) | [Schema](#)

```
{  
  "group_id": 1  
}
```

## Responses

| Code | Description                                     | Links    |
|------|---|----------|
| 201  | User added to the group successfully            | No links |
| 400  | Group ID is invalid or user is already a member | No links |
| 404  | Group not found                                 | No links |

GET

/membership/ List all members of a group with their roles



## Parameters

Try it out

| Name | Description |
|------|-------------|
|------|-------------|

**group\_id** \* required  
integer  
(query) The ID of the group to fetch members for

group\_id

## Responses

| Code | Description                          | Links    |
|------|--------------------------------------|----------|
| 200  | List of all members with their roles | No links |
| 400  | Group ID is required                 | No links |

Media type

application/json



Controls Accept header.

Example Value | Schema

```
[  
  {  
    "username": "alice",  
    "user_role": "Member"  
  }  
]
```

No links

| Code   | Description  | Links             |
|--|--|-------------------|
| 404  | Group not found  | No links          |
| <b>GET</b>   | /membership/user/groups List all groups a user is a member of with their roles | ^                 |
| <b>Parameters</b>  |  | <b>Try it out</b> |
| Name   | Description  |                   |
| <b>username</b> * required   | The username of the user to fetch groups for                                   |                   |
| string<br>(query)  |  | username          |
| <b>Responses</b>   |  |                   |
| Code   | Description  | Links             |
| 200  | List of all groups with their roles the user is a member of                    | No links          |
| Media type   |  |                   |
| application/json   |  | ▼                 |
| Controls Accept header.  |  |                   |
| <a href="#">Example Value</a>   <a href="#">Schema</a>                           |  |                   |
| <pre>[   {     "group_name": "Developers",     "user_role": "Member"   } ]</pre> |  |                   |
| 400  | Username is required   | No links          |
| 404  |  | No links          |

|                   |   |   |
|-------------------|---|---|
| <b>GET</b>        | /membership/admins List all admins with their group names | ^ |
| <b>Parameters</b> |   |   |

Try it out

No parameters

## Responses

| Code | Description                                 | Links           |
|------|---|-----------------|
| 200  | A list of all admins with their group names | <i>No links</i> |

Media type

application/json

Controls Accept header.

Example Value | Schema

```
[  
  {  
    "group_name": "Developers",  
    "username": "alice",  
    "user_role": "Admin"  
  }  
]
```

PUT

/membership/role Update a user's role in their group



Allows a user to update the role of a group member. Admins can update any user's role, while non-admin users can only update their own role. Ensures that there is always at least one admin in the group.

## Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  
  "group_id": 1,  
  "user_id": 42,  
  "role": "Member"  
}
```

## Responses

| Code | Description   | Links    |
|------|---|----------|
| 200  | User role updated successfully.   | No links |
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>{ "message": "User role updated successfully." }</pre> |          |
| 400  | Invalid input, such as missing or invalid parameters.   | No links |
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p>Example Value   Schema</p> <pre>{ "error": "Invalid role. Only 'Member', 'Guest', and 'Admin' are allowed." }</pre>   |          |
| 403  | Unauthorized action or insufficient permissions.  | No links |
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p>Example Value   Schema</p> <pre>{ "error": "You are not a member of this group." }</pre>                              |          |
| 404  | Target user is not a member of the group.   | No links |
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p>Example Value   Schema</p>  |          |

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

```
{  
  "error": "Target user is not a member of the group."  
}
```

**DELETE**

/membership/remove Remove a user from a group



Allows an admin or the user themselves to remove a user from a group. Ensures there is always at least one admin in the group.

## Parameters

**Try it out**

No parameters

## Request body required

application/json



Example Value | Schema

```
{  
  "group_id": 1,  
  "user_id": 42  
}
```

## Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

200

User removed from the group successfully.

*No links*

Media type

application/json



Controls Accept header.

Example Value | Schema

```
{  
  "message": "User removed from the group successfully."  
}
```

400

Invalid input, such as missing parameters or trying to remove the last admin.

*No links*

| Code | Description   | Links           |
|------|---|-----------------|
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{  "error": "Cannot remove the last admin from the group."}</pre> |                 |
| 403  | Unauthorized action or insufficient permissions.  | <i>No links</i> |
| 404  | User not a member of the group.   | <i>No links</i> |

## Groups



GET

/groups Get all groups



### Parameters

[Try it out](#)

No parameters

### Responses

| Code | Description       | Links    |
|------|-------------------|----------|
| 200  | A list of groups. | No links |

Media type

application/json

▼

Controls Accept header.

Example Value | Schema

```
[  
  {  
    "group_id": 1,  
    "group_name": "Developers",  
    "group_description": "Group for developers"  
  }  
]
```

POST /groups Create a new group ^ 

**Parameters** Try it out

No parameters

**Request body** required application/json

Example Value | Schema

```
{  
  "group_name": "Developers",  
  "group_description": "Group for developers"  
}
```

**Responses**

| Code | Description                | Links    |
|------|----------------------------|----------|
| 201  | Group created successfully | No links |
| 400  | Missing required fields    | No links |

**GET**

/groups/{group\_id} Get a single group by ID

**Parameters****Try it out**

| Name | Description |
|------|-------------|
|------|-------------|

|                            |  |
|----------------------------|--|
| <b>group_id</b> * required |  |
|----------------------------|--|

|         |  |
|---------|--|
| integer |  |
|---------|--|

|        |  |
|--------|--|
| (path) |  |
|--------|--|

|          |  |
|----------|--|
| group_id |  |
|----------|--|

**Responses**

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

|     |  |  |
|-----|--|--|
| 200 |  |  |
|-----|--|--|

|                |  |          |
|----------------|--|----------|
| The group data |  | No links |
|----------------|--|----------|

|            |  |
|------------|--|
| Media type |  |
|------------|--|

|                  |   |
|------------------|---|
| application/json | ▼ |
|------------------|---|

|                         |  |
|-------------------------|--|
| Controls Accept header. |  |
|-------------------------|--|

|                               |                        |
|-------------------------------|------------------------|
| <a href="#">Example Value</a> | <a href="#">Schema</a> |
|-------------------------------|------------------------|

|  |
|--|
| {<br>"group_name": "Developers",<br>"group_description": "Group for developers"<br>} |
|--|

|     |  |
|-----|--|
| 404 |  |
|-----|--|

|                 |          |
|-----------------|----------|
| Group not found | No links |
|-----------------|----------|

**PUT**

/groups/{group\_id} Update an existing group by ID

**Parameters****Try it out**

| Name | Description |
|------|-------------|
|------|-------------|

|                            |  |
|----------------------------|--|
| <b>group_id</b> * required |  |
|----------------------------|--|

|         |  |
|---------|--|
| integer |  |
|---------|--|

|        |  |
|--------|--|
| (path) |  |
|--------|--|

|          |  |
|----------|--|
| group_id |  |
|----------|--|

**Request body** required

application/json



Example Value | Schema

```
{  
    "group_name": "Updated Developers",  
    "group_description": "Updated group for developers"  
}
```

## Responses

| Code | Description                | Links           |
|------|----------------------------|-----------------|
| 200  | Group updated successfully | <i>No links</i> |
| 400  | Missing required fields    | <i>No links</i> |
| 403  | Unauthorized access        | <i>No links</i> |
| 404  | Group not found            | <i>No links</i> |

**DELETE**

/groups/{group\_id} Delete a group by ID



**Try it out**

## Parameters

| Name | Description |
|------|-------------|
|------|-------------|

**group\_id** \* required

integer

(path)

group\_id

## Responses

| Code | Description                | Links           |
|------|----------------------------|-----------------|
| 200  | Group deleted successfully | <i>No links</i> |

| Code | Description         | Links    |
|------|---------------------|----------|
| 403  | Unauthorized access | No links |
| 404  | Group not found     | No links |

## Messages



GET

/messages List all messages ordered from newest to oldest



### Parameters

Try it out

| Name | Description |
|------|-------------|
|------|-------------|

**group\_id** \* required

integer

(query)

Group ID to filter messages.

1

page

integer

(query)

Page number for pagination (default: 1).

1

size

integer

(query)

Number of results per page (default: 20, max: 100).

20

### Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

200

All messages ordered from newest to oldest

No links

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

| Code | Description   | Links    |
|------|---|----------|
|      | <pre>[<br/>  {<br/>    "message_id": 1,<br/>    "group_id": 1,<br/>    "user_id": 101,<br/>    "user_message": "This is a message",<br/>    "message_time": "2024-01-15T12:34:56"<br/>  }<br/>]</pre> |          |
| 400  | Group ID is required  | No links |
| 401  | Unauthorized access or missing token  | No links |
| 403  | Unauthorized access. You are not a member of this group.  | No links |

**POST** /messages Create a new message ^ 

**Parameters** **Try it out**

No parameters

**Request body** required application/json 

**Example Value** | **Schema**

```
{  
  "user_message": "This is a new message",  
  "group_id": 1  
}
```

**Responses**

| Code | Description                  | Links    |
|------|------------------------------|----------|
| 201  | Message created successfully | No links |

Media type application/json 

Controls Accept header.

| Code | Description   | Links    |
|------|---|----------|
|      | Example Value   Schema  |          |
|      | <pre>{   "message": "Message created successfully",   "message_id": 123 }</pre> |          |
| 400  | Message text or group ID is required  | No links |
| 401  | Unauthorized access or missing token  | No links |
| 403  | Unauthorized access. You are not a member of this group.                        | No links |

**GET /messages/filter** Filter messages by date range with pagination [^](#) [🔒](#)

**Parameters** [Try it out](#)

| Name  | Description  |
|---|--|
| <b>group_id</b> * required<br>integer<br>(query)          | Group ID to filter messages.<br><input type="text" value="1"/>                         |
| <b>start_date</b> * required<br>string(\$date)<br>(query) | Start date for filtering (YYYY-MM-DD).<br><input type="text" value="2024-01-01"/>      |
| <b>end_date</b> * required<br>string(\$date)<br>(query)   | End date for filtering (YYYY-MM-DD).<br><input type="text" value="2024-02-01"/>        |
| <b>page</b><br>integer<br>(query)                         | Page number for pagination (default: 1).<br><input type="text" value="1"/>             |
| <b>size</b><br>integer<br>(query)                         | Number of results per page (default: 20, max: 100).<br><input type="text" value="20"/> |

## Responses

| Code | Description  | Links    |
|------|--|----------|
| 200  | Filtered messages  | No links |
|      | <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>▼</p> <p>Controls Accept header.</p>                       |          |
|      | <p>Example Value   Schema</p> <pre>[<br/>  {<br/>    "message_id": 1,<br/>    "user_message": "Hello, world!",<br/>    "message_time": "2024-01-15T12:34:56"<br/>  }<br/>]</pre> |          |
| 400  | Group ID, start date, and end date are required  | No links |
| 401  | Unauthorized access or missing token   | No links |
| 403  | Unauthorized access. You are not a member of this group.   | No links |

**DELETE**

/messages/{message\_id} Delete a message by its ID



## Parameters

**Try it out**

| Name  | Description   |
|---|---|
| <b>message_id</b> * required<br>integer<br>(path) | The ID of the message to delete.<br><br><input type="text" value="message_id"/> |

## Responses

| Code | Description   | Links    |
|------|---|----------|
| 200  | Message deleted successfully  | No links |
|      | <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <div style="margin-left: 10px;">▼</div> <p>Controls Accept header.</p> |          |
|      | <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{ "message": "Message deleted successfully" }</pre>  |          |
| 401  | Unauthorized access or missing token  | No links |
| 403  | Unauthorized access. Only the creator or a group admin can delete this message.   | No links |
| 404  | Message not found   | No links |

PUT /messages/{message\_id} Update a message by its ID ↗ 

**Parameters** [Try it out](#)

| Name  | Description   |
|---|---|
| <b>message_id</b> * required<br>integer<br>(path) | The ID of the message to update.<br><input type="text" value="message_id"/> |

**Request body** required application/json ▼

[Example Value](#) | [Schema](#)

```
{ "user_message": "Updated message content" }
```

## Responses

| Code | Description   | Links    |
|------|---|----------|
| 200  | Message updated successfully  | No links |
|      | <p>Media type</p> <div><p>application/json</p><p>▼</p></div> <p>Controls Accept header.</p> |          |
|      | <p>Example Value   Schema</p> <pre>{ "message": "Message updated successfully" }</pre>      |          |
| 400  | Invalid request data  | No links |
| 401  | Unauthorized access or missing token  | No links |
| 403  | Unauthorized access. Only the creator can update this message.                              | No links |
| 404  | Message not found   | No links |

## Tags

^

POST

/tags Create a new tag



### Parameters

Try it out

No parameters

### Request body required

application/json

▼

Example Value | Schema

```
{  
  "tag_name": "Event Management"  
}
```

## Responses

| Code | Description              | Links    |
|------|--------------------------|----------|
| 201  | Tag created successfully | No links |
| 400  | Tag name is required     | No links |

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "message": "Tag created successfully",  
  "tag_id": 1  
}
```

400

Tag name is required

No links

GET

/tags List all tags with optional search



## Parameters

[Try it out](#)

| Name                       | Description  |
|----------------------------|--|
| query<br>string<br>(query) | Search string to filter tags by name.<br><br>Event |

## Responses

| Code | Description     | Links    |
|------|-----------------|----------|
| 200  | A list of tags. | No links |

| Code | Description   | Links |
|------|---|-------|
|      | <p>Media type</p> <div style="border: 2px solid #80C080; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>[<br/>  {<br/>    "tag_id": 1,<br/>    "tag_name": "Event Management"<br/>  }<br/>]</pre> |       |

PUT

/tags/{tag\_id} Update a tag by its ID



## Parameters

[Try it out](#)

| Name | Description |
|------|-------------|
|------|-------------|

**tag\_id** \* required

integer  
(path)

tag\_id

**Request body** required

application/json

Example Value | Schema

```
{  
  "tag_name": "Updated Tag Name"  
}
```

## Responses

| Code | Description              | Links    |
|------|--------------------------|----------|
| 200  | Tag updated successfully | No links |
| 400  | Tag name is required     | No links |

| Code   | Description                           | Links   |             |   |                              |                   |                          |          |     |               |          |
|--|---------------------------------------|---|-------------|---|------------------------------|-------------------|--------------------------|----------|-----|---------------|----------|
| 404  | Tag not found                         | No links  |             |   |                              |                   |                          |          |     |               |          |
| <b>DELETE</b>  | /tags/{tag_id} Delete a tag by its ID | ^  |             |   |                              |                   |                          |          |     |               |          |
| <b>Parameters</b>  |                                       |   |             |   |                              |                   |                          |          |     |               |          |
| <table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td><b>tag_id</b> * required<br/>integer<br/>(path)</td><td>The ID of the tag to delete.</td></tr> </tbody> </table>  |                                       | Name  | Description | <b>tag_id</b> * required<br>integer<br>(path) | The ID of the tag to delete. | <b>Try it out</b> |                          |          |     |               |          |
| Name   | Description                           |   |             |   |                              |                   |                          |          |     |               |          |
| <b>tag_id</b> * required<br>integer<br>(path)  | The ID of the tag to delete.          |   |             |   |                              |                   |                          |          |     |               |          |
| <b>Responses</b>   |                                       |   |             |   |                              |                   |                          |          |     |               |          |
| <table border="1"> <thead> <tr> <th>Code</th><th>Description</th><th>Links</th></tr> </thead> <tbody> <tr> <td>200</td><td>Tag deleted successfully</td><td>No links</td></tr> <tr> <td>404</td><td>Tag not found</td><td>No links</td></tr> </tbody> </table> |                                       |   | Code        | Description                                   | Links                        | 200               | Tag deleted successfully | No links | 404 | Tag not found | No links |
| Code   | Description                           | Links   |             |   |                              |                   |                          |          |     |               |          |
| 200  | Tag deleted successfully              | No links  |             |   |                              |                   |                          |          |     |               |          |
| 404  | Tag not found                         | No links  |             |   |                              |                   |                          |          |     |               |          |

## Attendance

|  |  |   |
|--|--|---|
| <b>POST</b>  | /events/{event_id}/attendance Add attendance to an event | ^  |
| Marks the current user as attended or not attended for an event. Requires JWT. |  |   |
| <b>Parameters</b>  |  | <b>Try it out</b>   |
|  |  |   |

| Name  | Description  |
|---|--|
| <b>event_id</b> <small>* required</small>         | The ID of the event to add attendance for.<br><br><input type="text" value="event_id"/>  |
| Request body <small>required</small>              |  |
| <a href="#">application/json</a> <small>▼</small> |  |
| Example Value   Schema                            |  |
| <pre>{   "status": "attended" }</pre>             |  |
| Responses   |  |
| Code  | Description  |
| 201   | <p>Attendance added successfully</p> <p>Media type<br/> <input style="border: 2px solid green; padding: 2px; width: 150px; height: 20px; border-radius: 5px;" type="text" value="application/json"/> <small>▼</small></p> <p>Controls Accept header.</p> <p>Example Value   Schema</p> <pre>{   "message": "Attendance added successfully" }</pre>       |
| 400   | <p>Attendance status is required or duplicate attendance record exists</p> <p>Media type<br/> <input style="border: 2px solid green; padding: 2px; width: 150px; height: 20px; border-radius: 5px;" type="text" value="application/json"/> <small>▼</small></p> <p>Example Value   Schema</p> <pre>{   "error": "Attendance status is required." }</pre> |

| Code   | Description   | Links             |
|--|---|-------------------|
| 404  | Event not found   | No links          |
| <b>GET</b>   | /events/{event_id}/attendance List attendance for an event      | ↗                 |
| Retrieves a list of users who are attending or not attending for a specific event.   |   |                   |
| <b>Parameters</b>  |   | <b>Try it out</b> |
| Name   | Description   |                   |
| <b>event_id</b> * required<br>integer<br>(path)  | The ID of the event to retrieve attendance for.<br><br>event_id |                   |
| <b>Responses</b>   |   |                   |
| <b>Code</b>  | <b>Description</b>  | Links             |
| 200  | List of users and their attendance status for the event         | No links          |
| <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>[{"user_id": 101, "username": "alice", "event_status": "attended"}]</pre> |   |                   |

|            |  |   |
|------------|--|---|
| <b>PUT</b> | /events/{event_id}/attendance Update attendance for an event | ↗ |
|            |  |   |

Updates the attendance status of the current user. Requires JWT.

## Parameters

[Try it out](#)

| Name | Description |
|------|-------------|
|------|-------------|

### event\_id \* required

integer  
(path)

event\_id

## Request body required

application/json



[Example Value](#) | [Schema](#)

```
{  
  "status": "not_attended"  
}
```

## Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

200

Attendance updated successfully

No links

Media type

application/json



Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "message": "Attendance updated successfully"  
}
```

400

New attendance status is required

No links

Media type

application/json



| Code | Description  | Links |
|------|--|-------|
|      | <p>Example Value   Schema</p> <pre>{<br/>  "error": "New attendance status is required."<br/>}</pre> |       |

|     |                             |          |
|-----|-----------------------------|----------|
| 404 | Attendance record not found | No links |
|-----|-----------------------------|----------|

|               |  |   |
|---------------|--|---|
| <b>DELETE</b> | /events/{event_id}/attendance Delete attendance for an event |   |
|---------------|--|---|

Deletes the current user's attendance record for an event. Requires JWT.

## Parameters

[Try it out](#)

| Name  | Description  |
|---|--|
| <b>event_id</b> * required<br>integer<br>(path) | The ID of the event to delete attendance for.<br><br><input type="text" value="event_id"/> |

## Responses

| Code | Description                     | Links    |
|------|---------------------------------|----------|
| 200  | Attendance deleted successfully | No links |

Media type

**application/json** 

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "message": "Attendance deleted successfully"  
}
```

|     |  |          |
|-----|--|----------|
| 404 | Attendance record not found or already deleted | No links |
|-----|--|----------|

| Code   | Description  | Links             |             |  |  |   |                 |
|--|--|-------------------|-------------|--|--|---|-----------------|
|  | <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div>   |                   |             |  |  |   |                 |
|  | <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>{   "error": "Attendance record not found or already deleted." }</pre>  |                   |             |  |  |   |                 |
| <b>GET</b>   | /users/{user_id}/attendance List a user's attended or not attended events  | <a href="#">^</a> |             |  |  |   |                 |
| Retrieves a list of all events that a specific user is attending or not attending.   |  |                   |             |  |  |   |                 |
| <b>Parameters</b> <hr/> <table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td><b>user_id</b> * required<br/>integer<br/>(path)</td><td>The ID of the user to retrieve attendance for.<br/><br/><input type="text" value="user_id"/></td></tr> </tbody> </table> |  | Name              | Description | <b>user_id</b> * required<br>integer<br>(path) | The ID of the user to retrieve attendance for.<br><br><input type="text" value="user_id"/> |   |                 |
| Name   | Description  |                   |             |  |  |   |                 |
| <b>user_id</b> * required<br>integer<br>(path)   | The ID of the user to retrieve attendance for.<br><br><input type="text" value="user_id"/>   |                   |             |  |  |   |                 |
| <b>Responses</b> <hr/> <table border="1"> <thead> <tr> <th>Code</th><th>Description</th><th>Links</th></tr> </thead> <tbody> <tr> <td>200</td><td>List of events the user is attending or not attending</td><td><i>No links</i></td></tr> </tbody> </table>  |  | Code              | Description | Links  | 200  | List of events the user is attending or not attending | <i>No links</i> |
| Code   | Description  | Links             |             |  |  |   |                 |
| 200  | List of events the user is attending or not attending  | <i>No links</i>   |             |  |  |   |                 |
|  | <p>Media type</p> <div style="border: 1px solid #007bff; padding: 2px; display: inline-block;">application/json</div>  |                   |             |  |  |   |                 |
|  | <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>[   {     "event_id": 10,     "event_name": "Sample Event",     "event_status": "attended",     "event_date": "2025-02-20"   } ]</pre> |                   |             |  |  |   |                 |

| Code   | Description   | Links             |
|--|---|-------------------|
|  | <pre>}</pre>  |                   |
| <b>GET</b>   | /events/attendance/top-users List top users by event attendance | ^                 |
| Retrieves users who have attended the most events.   |   |                   |
| <b>Parameters</b>  |   | <b>Try it out</b> |
| No parameters  |   |                   |
| <b>Responses</b>   |   |                   |
| Code   | Description   | Links             |
| 200  | List of top users and their attendance count                    | <i>No links</i>   |
| <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <span style="font-size: small;">▼</span> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>[   {     "user_id": 101,     "username": "alice",     "total_attendance": 5   } ]</pre> |   |                   |

## Feedback

|   |                            |   |
|---|----------------------------|---|
| <b>POST</b>                                       | /feedback Add new feedback | ^  |
| Adds feedback for a specific event. Requires JWT. |                            |   |
| <b>Parameters</b>                                 |                            | <b>Try it out</b>   |
|   |                            |   |

No parameters

**Request body** required

application/json

[Example Value](#) | [Schema](#)

```
{  
  "event_id": 1,  
  "rating": 5,  
  "feedback": "Great event!"  
}
```

## Responses

| Code | Description                      | Links           |
|------|----------------------------------|-----------------|
| 201  | Feedback added successfully      | <i>No links</i> |
| 400  | event_id and rating are required | <i>No links</i> |

GET

/feedback/events/{event\_id} Get feedback for an event



Retrieves all feedback associated with a specific event.

## Parameters

[Try it out](#)

| Name | Description |
|------|-------------|
|------|-------------|

**event\_id** \* required

integer  
(path)

Event ID

event\_id

## Responses

| Code | Description   | Links    |
|------|---|----------|
| 200  | <p>List of feedback for the event</p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>[<br/>  {<br/>    "feedback_id": 10,<br/>    "user_id": 101,<br/>    "username": "alice",<br/>    "rating": 5,<br/>    "feedback": "Great event!",<br/>    "created_at": "2025-01-04T12:00:00"<br/>  }<br/>]</pre> | No links |

**GET** /feedback/users/{user\_id} Get feedback by user ^

Retrieves all feedback created by a specific user.

#### Parameters

[Try it out](#)

| Name                      | Description                          |
|---------------------------|--------------------------------------|
| <b>user_id</b> * required | User ID                              |
| integer<br>(path)         | <input type="text" value="user_id"/> |

#### Responses

| Code | Description  | Links    |
|------|--|----------|
| 200  | <p>List of feedback by the user</p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div> | No links |

| Code  | Description  | Links                                 |
|---|--|---------------------------------------|
|   | <p>Controls Accept header.</p> <p><a href="#">Example Value</a>   <a href="#">Schema</a></p> <pre>[<br/>  {<br/>    "feedback_id": 15,<br/>    "event_id": 2,<br/>    "event_name": "Sample Event",<br/>    "rating": 4,<br/>    "feedback": "It was good.",<br/>    "created_at": "2025-01-04T14:00:00"<br/>  }<br/>]</pre> |                                       |
| <a href="#">GET</a>                             | <a href="#">/feedback/stars/{event_id}</a> Get average rating and total feedback count for an event  | <a href="#">^</a>                     |
|   | <p>Retrieves the average rating and total feedback count for a specific event.</p>   |                                       |
|   | <p><b>Parameters</b></p>   | <a href="#">Try it out</a>            |
| Name  | Description  |                                       |
| <b>event_id</b> * required<br>integer<br>(path) | Event ID   | <input type="text" value="event_id"/> |
|   |  |                                       |
|   | <b>Responses</b>   |                                       |

| Code | Description                       | Links    |
|------|-----------------------------------|----------|
| 200  | Average rating and total feedback | No links |

Media type

application/json

▼

Controls Accept header.

Example Value | Schema

```
{  
    "average_rating": 4.5,  
    "total_feedback": 12  
}
```

PUT

/feedback/{feedback\_id} Update feedback



Updates an existing feedback. Only the owner of the feedback can do this. Requires JWT.

## Parameters

Try it out

| Name                          | Description |
|-------------------------------|-------------|
| <b>feedback_id</b> * required | Feedback ID |

integer  
(path)

## Request body required

application/json

Example Value | Schema

```
{  
    "rating": 4,  
    "feedback": "It was good, but could be better."  
}
```

## Responses

| Code | Description                           | Links    |
|------|---------------------------------------|----------|
| 200  | Feedback updated successfully         | No links |
| 400  | rating is required                    | No links |
| 403  | You can only update your own feedback | No links |
| 404  | Feedback not found                    | No links |

**DELETE**

/feedback/{feedback\_id} Delete feedback



Deletes a feedback. Only the owner of the feedback can do this. Requires JWT.

#### Parameters

[Try it out](#)

| Name | Description |
|------|-------------|
|------|-------------|

**feedback\_id** \* required

integer  
(path)

Feedback ID

feedback\_id

#### Responses

| Code | Description                           | Links    |
|------|---------------------------------------|----------|
| 200  | Feedback deleted successfully         | No links |
| 403  | You can only delete your own feedback | No links |
| 404  | Feedback not found                    | No links |