

Create Comment

Endpoint

URL: `/:projectId/comments`

Method: `POST`

Authentication Required: Yes

Description

Creates a new comment or reply on a specific entity.

- Comments can include text, a GIF, mentions, attachments, and metadata.
- Only authenticated users (or master clients) may create comments.
- Automatically handles reputation and notification logic.

Request

Headers

Header	Type	Required	Description
<code>Authorization</code>	<code>string</code>	Yes	Bearer token for authenticated users.

Body Parameters

Field	Type	Required	Description
<code>entityId</code>	<code>string</code>	Yes	ID of the entity to attach the comment to.
<code>content</code>	<code>string</code>	No*	Text content of the comment. Required unless <code>gif</code> is provided.
<code>gif</code>	IGif	No*	GIF object. Required unless <code>content</code> is provided.
<code>mentions</code>	<code>IMention[]</code>	No	List of mentioned users.
<code>parentId</code>	<code>string</code>	No	ID of the parent comment (for replies).
<code>referencedCommentId</code>	<code>string</code>	No	ID of a referenced (quoted) comment.
<code>foreignId</code>	<code>string</code>	No	Optional external reference ID.
<code>attachments</code>	<code>Record<string, any>[]</code>	No	List of attached files or resources.
<code>metadata</code>	<code>Record<string, any></code>	No	Custom metadata for this comment.
<code>userId</code>	<code>string</code>	No	Allowed only for master clients. Posts the comment on behalf of a user.

Example Request

```
{
  "entityId": "a1b2c3d4",
  "content": "Great article!",
  "mentions": [{ "id": "user-123", "username": "alex" }],
  "gif": null,
  "parentId": "comment-456",
  "attachments": [],
  "metadata": { "source": "web" }
}
```

Response

Success Response (201 Created)

Returns the newly created comment object:

```
{
  "id": "comment-789",
  "entityId": "a1b2c3d4",
  "userId": "user-456",
  "content": "Great article!",
  "mentions": [{ "id": "user-123", "username": "alex" }],
  "gif": null,
  "attachments": [],
  "metadata": { "source": "web" },
  "createdAt": "2025-05-11T10:00:00.000Z",
  "updatedAt": "2025-05-11T10:00:00.000Z"
}
```

Error Responses

Missing Entity ID (400 Bad Request)

```
{
  "error": "Missing entity ID",
  "code": "comment/missing-entity-id"
}
```

Missing Content (400 Bad Request)

```
{
  "error": "Missing required comment content",
  "code": "comment/missing-content"
}
```

Missing User ID (400 Bad Request)

```
{
```

```
"error": "Missing user ID",  
"code": "comment/missing-user-id"  
}
```

Entity Not Found (404 Not Found)

```
{  
  "error": "Entity not found",  
  "code": "comment/entity-not-found"  
}
```

Server Error (500 Internal Server Error)

```
{  
  "error": "Internal server error",  
  "code": "comment/server-error",  
  "details": "[error message]"  
}
```

Notes

- Either `content` or `gif` must be present.
- If `userId` is provided, the request must come from a master client.
- Reputation is awarded to the user who posts the comment.
- Notifications are triggered automatically:
 - To the entity author
 - To the parent comment author (for replies)
 - To each mentioned user (excluding duplicates)
- API usage for comments is tracked via Redis.

Gif Structure

```
interface IGif {
```

```
    altText: string;
    gifPreviewUrl: string;
    gifUrl: string;
    id: string;
    url: string;
    aspectRatio: string;
}
```

Mention Structure

```
interface IMention {
    id: string; // User ID
    username: string; // Username at time of mention
}
```

Last updated on May 11, 2025