Comments

- Schema
- List Comments
- Create a Comment
  - Arguments
- Retrieve a Comment
- Update a Comment
  - Arguments
- Delete a Comment
  - Arguments

Schema

The schema defines all the fields that exist within a comment record. Any response from these endpoints can be expected to contain the fields below unless the `_filter` query parameter is used or the schema field only appears in a specific context.

`id`	Unique identifier for the comment. JSON data type: integer Read only Context: `view`, `edit`, `embed`
`author`	The ID of the user object, if author was a user. JSON data type: integer Context: `view`, `edit`, `embed`
`author_email`	Email address for the comment author. JSON data type: string, Format: email Context: `edit`
`author_ip`	IP address for the comment author. JSON data type: string, Format: ip Context: `edit`
`author_name`	Display name for the comment author. JSON data type: string Context: `view`, `edit`, `embed`

`author_url`	URL for the comment author. JSON data type: string, Format: uri Context: `view`, `edit`, `embed`
`author_user_agent`	User agent for the comment author. JSON data type: string Context: `edit`
`content`	The content for the comment. JSON data type: object Context: `view`, `edit`, `embed`
`date`	The date the comment was published, in the site's timezone. JSON data type: string, Format: datetime (details) Context: `view`, `edit`, `embed`
`date_gmt`	The date the comment was published, as GMT. JSON data type: string, Format: datetime (details) Context: `view`, `edit`
`link`	URL to the comment. JSON data type: string, Format: uri Read only Context: `view`, `edit`, `embed`
`parent`	The ID for the parent of the comment. JSON data type: integer Context: `view`, `edit`, `embed`
`post`	The ID of the associated post object. JSON data type: integer Context: `view`, `edit`
`status`	State of the comment. JSON data type: string Context: `view`, `edit`
`type`	Type of the comment. JSON data type: string Read only Context: `view`, `edit`, `embed`
`author_avatar_urls`	Avatar URLs for the comment author. JSON data type: object Read only Context: `view`, `edit`, `embed`
`meta`	Meta fields. JSON data type: object Context: `view`, `edit`

```
</div>
```

List Comments

Query this endpoint to retrieve a collection of comments. The response you receive can be controlled and filtered using the URL query parameters below.

```
<h3>Definition</h3>

<code>GET /wp/v2/comments</code>

<h3>Example Request</h3>

<code>$ curl https://example.com/wp-json/wp/v2/comments</code>
</div>
<div class="secondary">
<h3>Arguments</h3>
<table class="arguments">
<tr>
<td>
<code>context</code><br />
</td>
<td>
<p>Scope under which the request is made; determines fields present in response.</p>
<p class="default">
Default: <code>view</code>
</p>
<p>One of: <code>view</code>, <code>embed</code>, <code>edit</code></p>
</td>
</tr>
<tr>
<td>
<code>page</code><br />
</td>
<td>
<p>Current page of the collection.</p>
<p class="default">
Default: <code>1</code>
</p>
</td>
```

□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>per_page</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Maximum number of items to be returned in result set.
□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□Default: <code>10</code>
□□□□□
□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>search</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit results to those matching a string.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>after</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit response to comments published after a given ISO8601 compliant date.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>author</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit result set to comments assigned to specific user IDs. Requires authorization
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>author_exclude</code>
□□□□□□□□□</td>

□□□<td>
□□□□□□□□□□ Ensure result set excludes comments assigned to specific user IDs. Requires authorization.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>author_email</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit result set to that from a specific author email. Requires authorization.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>before</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit response to comments published before a given ISO8601 compliant date.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>exclude</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Ensure result set excludes specific IDs.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>include</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit result set to specific IDs.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>offset</code>

□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Offset the result set by a specific number of items.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>order</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Order sort attribute ascending or descending.
□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□Default: <code>desc</code>
□□□□□
□□□□□□□□□□□□□□□□ One of: <code>asc</code>, <code>desc</code>
□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>orderby</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Sort collection by comment attribute.
□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□Default: <code>date_gmt</code>
□□□□□
□□□□□□□□□□□□□□□□ One of: <code>date</code>, <code>date_gmt</code>, <code>id</code>, <code>incl
<code>post</code>, <code>parent</code>, <code>type</code>
□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>parent</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□ Limit result set to comments of specific parent IDs.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>

□□□□□□□□□□□<code>parent_exclude</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□□ Ensure result set excludes specific parent IDs.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□□<code>post</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□□ Limit result set to comments assigned to specific post IDs.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□□<code>status</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□□ Limit result set to comments assigned a specific status. Requires authorization.</
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□Default: <code>approve</code>
□□□□□□
□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□□<code>type</code>
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□□ Limit result set to comments assigned a specific type. Requires authorization.
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□Default: <code>comment</code>
□□□□□□
□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□□<code>password</code>
□□□□□□□□□</td>

```
□□□<td>
□□□□□□□□□□<p>The password for the post if it is password protected.</p>
□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□</table>

</div>
```

Create a Comment

Arguments

`author`	The ID of the user object, if author was a user.
`author_email`	Email address for the comment author.
`author_ip`	IP address for the comment author.
`author_name`	Display name for the comment author.
`author_url`	URL for the comment author.
`author_user_agent`	User agent for the comment author.
`content`	The content for the comment.
`date`	The date the comment was published, in the site's timezone.
`date_gmt`	The date the comment was published, as GMT.
`parent`	The ID for the parent of the comment.
`post`	The ID of the associated post object.
`status`	State of the comment.
`meta`	Meta fields.

```
</div>
<div class="secondary">
□<h3>Definition</h3>

□<code>POST /wp/v2/comments</code>
</div>
```

Retrieve a Comment

```
□<h3>Definition & Example Request</h3>

□<code>GET /wp/v2/comments/&lt;id&gt;</code>

□<p>Query this endpoint to retrieve a specific comment record.</p>

□<code>$ curl https://example.com/wp-json/wp/v2/comments/&lt;id&gt;</code>
</div>
<div class="secondary">
□□<h3>Arguments</h3>
<table class="arguments">
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>id</code><br />
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□<p>Unique identifier for the comment.</p>
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>context</code><br />
□□□□□□□□□</td>
□□□<td>
□□□□□□□□□□<p>Scope under which the request is made; determines fields present in response.</p>
□□□□□□□□□□□□□□□□□□□□□□□□<p class="default">
□□□□□□□Default: <code>view</code>
□□□□□</p>
□□□□□□□□□□□□□□□□□<p>One of: <code>view</code>, <code>embed</code>, <code>edit</code></p>
□□□□□□□□□</td>
□□</tr>
□□□□<tr>
□□□<td>
□□□□□□□□□□<code>password</code><br />
□□□□□□□□□</td>
```

```
□□□<td>
□□□□□□□□□□<p>The password for the parent post of the comment (if the post is password protected
□□□□□□□□□□□□□□□□□□□□□□□□□□□□</td>
□□</tr>
□□</table>

</div>
```

Update a Comment

Arguments

`id`	Unique identifier for the comment.
`author`	The ID of the user object, if author was a user.
`author_email`	Email address for the comment author.
`author_ip`	IP address for the comment author.
`author_name`	Display name for the comment author.
`author_url`	URL for the comment author.
`author_user_agent`	User agent for the comment author.
`content`	The content for the comment.
`date`	The date the comment was published, in the site's timezone.
`date_gmt`	The date the comment was published, as GMT.
`parent`	The ID for the parent of the comment.
`post`	The ID of the associated post object.
`status`	State of the comment.
`meta`	Meta fields.

```
</div>
<div class="secondary">
□<h3>Definition</h3>

□<code>POST /wp/v2/comments/&lt;id&gt;</code>
```

```
□<h3>Example Request</h3>

□<code></code>
</div>
```

Delete a Comment

Arguments

`id`	Unique identifier for the comment.
`force`	Whether to bypass Trash and force deletion.
`password`	The password for the parent post of the comment (if the post is password protected).

```
</div>
<div class="secondary">
□<h3>Definition</h3>

□<code>DELETE /wp/v2/comments/&lt;id&gt;</code>

□<h3>Example Request</h3>

□<code>$ curl -X DELETE https://example.com/wp-json/wp/v2/comments/&lt;id&gt;</code>
</div>
```

□□□ #2
□ □□□ □□□ 2026-08-29 23:39:08 UTC
□ □□□ □□□ 2026-08-29 23:39:08 UTC