User Groups REST API routes

- Schema
- List User Groups
  - Arguments
  - Definition
  - Example of use
  - JSON Response
- Create a User Group
  - Arguments
  - Definition
  - Example of use
  - JSON Response
- Retrieve a specific User Group
  - Arguments
  - Definition
  - Example of use
  - JSON Response
- Update a specific User Group
  - Arguments
  - Definition
  - Example of use
  - JSON Response
- Delete a specific User group
  - Arguments
  - Definition
  - Example of use
  - JSON Response
- List the User Groups of the logged in member
  - Arguments

- Definition
- Example of use
- JSON Response

The Groups component allow your users to organize themselves into specific public, private or hidden sections of your community site with separate activity streams and member listings.

> The User Groups component is an optional one. This means the `buddypress/v2/groups/*` endpoints will only be available if the component is active on the community site.

Schema

The schema defines all the fields that exist for a Groups component single item.

Property	Description
`id`	A unique numeric ID for the Groups single item. JSON data type: *integer*. Read only. Context: `embed`, `view`, `edit`.
`creator_id`	The ID of the user who created the Groups single item. JSON data type: *integer*. Default: the ID of the logged in member. Context: `embed`, `view`, `edit`.
`name`	The name of the Groups single item. JSON data type: *string*. Required: `true`. Context: `embed`, `view`, `edit`.
`slug`	The URL-friendly slug for the Groups single item. JSON data type: *string*. Context: `embed`, `view`, `edit`.
`link`	The permalink to the Groups single item on the site. JSON data type: *string*, format: *URI*. Read only. Context: `embed`, `view`, `edit`.
`description`	The description of the Groups single item. JSON data type: *object* (properties: `raw`, `rendered`). Required: `true`. Context: `embed`, `view`, `edit`.

Property	Description
`status`	The status of the Groups single item (e.g.: `private`). JSON data type: *string*. One of: `public`, `private`, `hidden`. Context: `embed`, `view`, `edit`.
`enable_forum`	Whether the Group has a forum enabled or not[^1]. JSON data type: *boolean*. Context: `embed`, `view`, `edit`.
`parent_id`	ID of the parent Groups single item[^2]. JSON data type: *integer*. Context: `embed`, `view`, `edit`.
`date_created`	The date the Groups single item was created, in the site's timezone. JSON data type: *string* | *null*, format: *date-time*. Read only. Context: `embed`, `view`, `edit`.
`date_created_gmt`	The date the Groups single item was created, as GMT. JSON data type: *string* | *null*, format: *date-time*. Read only. Context: `view`, `edit`.
`created_since`	Time elapsed since the Groups single item was created, in the site's timezone. JSON data type: *string*. Default: `''`. Read only. Context: `embed`, `view`, `edit`.
`types`	The BP Group type(s) assigned to the Groups single item. See this documentation page for more information about Group Types. JSON data type: *array*. Read only. Context: `embed`, `view`, `edit`.
`admins`	The list of Groups single item administrators. JSON data type: *array*. Read only. Context: `edit`.
`mods`	The list of Groups single item moderators. JSON data type: *array*. Read only. Context: `edit`.
`total_member_count`	Count of all Group members. JSON data type: *integer*. Read only. Context: `embed`, `view`, `edit`.

Property	Description
`last_activity`	The date the Groups single item was last active, in the site's timezone. JSON data type: *string* | *null*, format: *date-time*. Read only. Context: `embed`, `view`, `edit`.
`last_activity_gmt`	The date the Groups single item was last active, as GMT. JSON data type: *string* | *null*, format: *date-time*. Read only. Context: `view`, `edit`.
`last_activity_diff`	Time elapsed since the Groups single item was last active. JSON data type: *string*. Default: `''`. Read only. Context: `embed`, `view`, `edit`.
`avatar_urls`[^3]	Avatar URLs for the Groups single item (Full & Thumb sizes). JSON data type: *object* (properties: `full`, and `thumb`). Read only. Context: `embed`, `view`, `edit`.

List User Groups

Arguments

Name	Description
`context`	Scope under which the request is made; determines fields present in response. JSON data type: *string*. Default: `view`. One of: `view`, `embed`, `edit`.
`page`	Current page of the collection. JSON data type: *integer*. Default: `1`.
`per_page`	Maximum number of groups single items to be returned in result set. JSON data type: *integer*. Default: `10`.
`search`	Limit results to those matching a string. JSON data type: *string*.
`exclude`	Ensure result set excludes Groups with specific IDs JSON data type: *array*. Default: `[]`.

Name	Description
`include`	Ensure result set includes Groups with specific IDs. JSON data type: *array*. Default: `[]`.
`type`	Shorthand for certain orderby/order combinations JSON data type: *string*. Default: `active`. One of: `active`, `newest`, `alphabetical`, `random`, `popular`.
`order`	Order sort attribute ascending or descending. JSON data type: *string*. Default: `desc`. One of: `desc`, `asc`.
`orderby`	Order Groups by which attribute. JSON data type: *string*. Default: `date_created`. One of: `date_created`, `last_activity`, `total_member_count`, `name`, `random`.
`user_id`	Limit result set to Groups single items that this user (ID) is a member of. JSON data type: *integer*. Default: `0`.
`status`	Group statuses to limit results to. JSON data type: *array*. Default: `[]`. One or more of: `public`, `private`, `hidden`.
`parent_id`	Get Groups single items that are children of the specified Group(s) IDs[^2]. JSON data type: *array*. Default: `[]`.
`meta`	Get Groups based on their meta data information. JSON data type: *array*. Default: `[]`.
`group_type`	Limit results set to a certain Group type. See this documentation page for more information about Group Types. JSON data type: *string*. Default: `''`. One of: the registered Group types on the site.
`enable_forum`	Whether the group has a forum enabled or not. JSON data type: *boolean*. Default: `false`.
`populate_extras`	Whether to fetch extra BP data about the returned groups. JSON data type: *boolean*. Default: `false`.

Name	Description
`show_hidden`	Whether results should include hidden Groups. JSON data type: *boolean*. Default: `false`.

Definition

`GET /buddypress/v2/groups`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups?context=view&populate_extras=true', {
	method: 'GET',
	headers: requestHeaders,
} ).then( ( response ) => {
	return response.json();
} ).then( ( data ) => {
	console.table( data );
} );
```

JSON Response

- An array of objects representing the matching groups single items on success.
- An object containg the error code, data and message on failure.

Create a User Group

Logged in users can create Groups single items, unless the Site's administrator has disabled their capacity from BuddyPress settings.

Arguments

Name	Description
`creator_id`	The ID of the user who created the Group. JSON data type: *integer*. Default: the ID of the logged in member.
`name`	The name of the Group. JSON data type: *string*. Required.
`description`	The description of the Group. JSON data type: *string*. Required.
`slug`	The URL-friendly slug for the Group. JSON data type: *string*.
`status`	The status of the Group. JSON data type: *string*. One of: `public`, `private`, `hidden`. Default: `public`.
`types`	The BP Group type(s) assigned to the Groups single item. See this documentation page for more information about Group Types. To assign more than one type, use a comma separated list of types. JSON data type: *string*. One of: the registered group types.
`parent_id`	ID of the parent Group[^2]. JSON data type: *integer*.
`enable_forum`	Whether the Group has a forum enabled or not. JSON data type: *boolean*.

Definition

`POST /buddypress/v2/groups`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups', {
	method: 'POST',
	headers: requestHeaders,
	body: JSON.stringify(
```

```
□□{
□□□name: 'Bapuus',
□□□description: 'bapuu is the BuddyPress wapuu',
□□}
□),
} ).then( ( response ) => {
□return response.json();
} ).then( ( data ) => {
□ console.log( data );
} );
```

JSON Response

- An object representing the created Groups single item on success.
- An object containg the error code, data and message on failure.

Retrieve a specific User Group

Arguments

Name	Description
`id`	A unique numeric ID for the Groups single item. **Required**. JSON data type: *integer*.
`context`	Scope under which the request is made; determines fields present in response. JSON data type: *string*. Default: `view`. One of: `view, embed, edit`.
`populate_extras`	Whether to fetch extra BP data about the returned groups. JSON data type: *boolean*. Default: `false`.

Definition

`GET /buddypress/v2/groups/<id>`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups/4', {
	method: 'GET',
	headers: requestHeaders,
} ).then( ( response ) => {
	return response.json();
} ).then( ( data ) => {
	console.log( data );
} );
```

JSON Response

- An object representing the Groups single item on success.
- An object containg the error code, data and message on failure.

Update a specific User Group

Arguments

Name	Description
`id`	A unique numeric ID for the Groups single item. **Required**. JSON data type: *integer*.
`creator_id`	The ID of the user who created the Group. JSON data type: *integer*. Default: the ID of the logged in member.
`name`	The name of the Group. JSON data type: *string*. Required.
`description`	The description of the Group. JSON data type: *string*. Required.
`slug`	The URL-friendly slug for the Group. JSON data type: *string*.

Name	Description
`status`	The status of the Group. JSON data type: *string*. One of: `public`, `private`, `hidden`. Default: `public`.
`parent_id`	ID of the parent Group[^2]. JSON data type: *integer*.
`enable_forum`	Whether the Group has a forum enabled or not. JSON data type: *boolean*.
`types`	The BP Group type(s) assigned to the Groups single item. See this documentation page for more information about Group Types. To assign more than one type, use a comma separated list of types. JSON data type: *string*. One of: the registered group types.
`append_types`	Append one or more BP Group type(s) to a group. To append more than one BP Group type, use a comma separated list of BP Group types. JSON data type: *string*. One of: the registered group types.
`remove_types`	Remove one or more BP Group type(s) from a group. To remove more than one BP Group type, use a comma separated list of BP Group types. JSON data type: *string*. One of: the registered group types.

Definition

`PUT /buddypress/v2/groups/<id>`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups/4', {
	method: 'PUT',
	headers: requestHeaders,
	body: JSON.stringify(
		{
```

```
    creator_id: 2,
    status: 'private',
  }
),
} ).then( ( response ) => {
return response.json();
} ).then( ( data ) => {
 console.log( data );
} );
```

JSON Response

- An object representing the updated Groups single item on success.
- An object containg the error code, data and message on failure.

Delete a specific User group

Arguments

Name	Description
`id`	A unique numeric ID for the Groups single item. **Required**. JSON data type: *integer*.

Definition

`DELETE /buddypress/v2/groups/<id>`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups/4', {
  method: 'DELETE',
```

```
headers: requestHeaders,
} ).then( ( response ) => {
return response.json();
} ).then( ( data ) => {
 console.log( data );
} );
```

JSON Response

- An object informing about the `deleted` status and the `previous` Groups single item on success.
- An object containg the error code, data and message on failure.

List the User Groups of the logged in member

Arguments

Name	Description
`context`	Scope under which the request is made; determines fields present in response. JSON data type: *string*. Default: `view`. One of: `view, embed, edit`.
`max`	The maximum amount of groups the user is member of to return. Defaults to all groups. JSON data type: *integer*. Default: `0`.

Definition

`GET /buddypress/v2/groups/me`

Example of use

> The `requestHeaders` object needs to be set according to the WordPress REST API nonce. Read more about the REST API authentification.

```
fetch( '/wp-json/buddypress/v2/groups/me', {
	method: 'GET',
	headers: requestHeaders,
} ).then( ( response ) => {
	return response.json();
} ).then( ( data ) => {
	console.log( data );
} );
```

JSON Response

- An array of objects representing the matching groups single items on success.
- An object containg the error code, data and message on failure.

[^1]: the `enable_forum` is used by the bbPress plugin to inform the corresponding group has a forum associated to it. [^2]: the `parent_id` field is not used by BuddyPress internally to provide a groups hierarchy feature leaving this part to BuddyPress Add-ons. See changeset 11095. [^3]: This property is only available if the WordPress discussion settings allow avatars and the Site Administrator allowed group administrators to upload profile photos for groups.

□□□ #2
□ □□□ □□□ 2026-08-29 22:50:51 UTC
□ □□□ □□□ 2026-08-29 22:50:51 UTC