Ability: acf/create-custom-post-type

- Description
- Input
  - Required Properties
  - Optional Properties
- Output
- Annotations
- Permissions
- Example
- Errors
- Notes

Creates a new custom post type in ACF.

Description

Creates a new custom post type using ACF's post type registration system. The post type is registered with WordPress and optionally enabled for AI access.

Input

```
{
  "post_type": "product",
  "label": "Product",
  "plural_label": "Products"
}
```

Required Properties

`post_type` *(string)* The post type key/slug. Must be lowercase alphanumeric with underscores or hyphens. Maximum 20 characters.

`label` *(string)* The singular label for the post type.

`plural_label` *(string)* The plural label for the post type.

Optional Properties

`description` *(string)* Description of the post type.

`public` *(boolean)* Whether the post type is publicly accessible. Default: `true`

`hierarchical` *(boolean)* Whether the post type supports parent/child relationships (like Pages). Default: `false`

`supports` *(array)* Features the post type supports. Can be an array of strings or an object with boolean values.

Available supports: – `title`: Post title – `editor`: Content editor – `author`: Author selection – `thumbnail`: Featured image – `excerpt`: Post excerpt – `comments`: Comments – `trackbacks`: Trackbacks – `revisions`: Revision history – `page-attributes`: Page attributes (menu order, parent) – `custom-fields`: Custom fields metabox – `post-formats`: Post formats

`show_in_rest` *(boolean)* Whether to expose in the REST API. Required for block editor and AI abilities. Default: `true`

`rest_base` *(string)* Custom REST API base path. Defaults to post type key.

`has_archive` *(boolean)* Whether the post type has an archive page. Default: `false`

`has_archive_slug` *(string)* Custom slug for the archive page.

`menu_icon` *(string|object)* Admin menu icon. Can be a dashicon class (e.g., `dashicons-cart`), image URL, or object with `type` and `value`.

`menu_position` *(integer)* Position in the admin menu. Values 5-100.

`taxonomies` *(array)* Array of taxonomy slugs to associate with this post type.

`allow_ai_access` *(boolean)* Whether to allow AI access to this post type. Default: `true`

`ai_description` *(string)* Description to help AI understand the purpose of this post type.

Output


```
{
  "success": true,
  "post_type": {
    "post_type": "product",
    "label": "Product",
    "plural_label": "Products",
    "public": true,
    "show_in_rest": true,
    "allow_ai_access": true
  },
  "message": "Custom post type 'product' created successfully."
}
```


Annotations

Annotation	Value
`readonly`	false
`destructive`	false
`idempotent`	false

Permissions

Requires the ACF capability (default: `manage_options`).

Example

Create a Team Member post type:

```
{
  "post_type": "team_member",
  "label": "Team Member",
  "plural_label": "Team Members",
  "description": "Staff and team member profiles",
  "public": true,
  "supports": ["title", "editor", "thumbnail"],
  "has_archive": true,
  "menu_icon": "dashicons-groups",
  "menu_position": 25,
  "allow_ai_access": true,
  "ai_description": "Team member profiles with bio, role, and contact information"
}
```

Response:


```
{
  "success": true,
  "post_type": {
    "post_type": "team_member",
    "label": "Team Member",
    "plural_label": "Team Members",
    "description": "Staff and team member profiles",
    "public": true,
    "hierarchical": false,
    "supports": ["title", "editor", "thumbnail"],
    "show_in_rest": true,
    "has_archive": true,
    "menu_icon": "dashicons-groups",
    "menu_position": 25,
    "allow_ai_access": true,
    "ai_description": "Team member profiles with bio, role, and contact information"
  },
  "message": "Custom post type 'team_member' created successfully."
}
```


Errors

Error	Cause
Invalid post type key	Key contains invalid characters or exceeds 20 characters
Post type already exists	A post type with this key is already registered
Missing required field	`post_type`, `label`, or `plural_label` not provided

Notes

- Post type keys must be unique across WordPress
- Reserved post type keys (`post`, `page`, `attachment`, etc.) cannot be used
- The post type is immediately registered and available in the admin
- Dynamic abilities (query, create, view, update, delete) are registered for post types with `show_in_rest` and `allow_ai_access` enabled

---

□□□ #1
□ □□□ □□□ 2026-08-29 23:44:03 UTC
□ □□□ □□□ 2026-08-29 23:44:03 UTC