注册自定义文章类型

目录

- 命名最佳实践
- URL
  - A Custom Slug for a Custom Post Type

WordPress 默认包含五种文章类型: `post`、`page`、`attachment`、`revision` 和 `menu`。

在开发插件时，您可能需要创建特定的内容类型：例如，电子商务网站的产品、在线学习网站的作业，或影评网站的影片。

通过自定义文章类型，您可以注册自己的文章类型。一旦自定义文章类型注册成功，它将获得一个新的顶级管理界面，用于管理和创建该类型的文章。

要注册新的文章类型，请使用 `register_post_type()` 函数。

[alert]
我们建议将自定义文章类型放在插件中，而不是主题中。这样可以确保即使用户更换主题，其内容仍然可以移植。 [/alert]

以下是一个最小示例，注册了一个名为 “产品 ”的新文章类型，在数据库中的标识符为 `wporg_product`。

```
function wporg_custom_post_type() {
  register_post_type('wporg_product',
    array(
      'labels'      => array(
        'name'          => __('Products', 'textdomain'),
        'singular_name' => __('Product', 'textdomain'),
      ),
      'public'      => true,
      'has_archive' => true,
    )
  );
}
add_action('init', 'wporg_custom_post_type');
```

请访问 `register_post_type()` 的参考页面查看参数说明。

[warning]您必须在 `admin_init` 钩子之前和 `after_setup_theme` 钩子之后调用 `register_post_type()`。推荐使用 `init` 动作钩子。[/warning]

命名最佳实践

请务必为您的文章类型函数和标识符添加与您的插件、主题或网站对应的简短前缀。

[warning]请确保您的自定义文章类型标识符不超过 **20** 个字符，因为数据库中 `post_type` 列目前是该长度的 VARCHAR 字段。[/warning]

[warning]为确保向前兼容，请勿使用 `wp_` 作为您的标识符——它已被 WordPress 核心使用。[/warning]

[warning]如果您的标识符过于通用（例如: `product`），可能会与选择使用相同标识符的其他插件或主题发生冲突。[/warning]

[info]若不禁用其中一个冲突的文章类型，则无法解决重复的文章类型标识符问题。[/info]

URL

自定义文章类型在网站 URL 结构中拥有其专属的固定链接段。

类型为 `wporg_product` 的文章默认使用以下 URL 结构: `https://example.com/wporg_product/%product_name%`。

其中 `wporg_product` 是自定义文章类型的固定链接段，`%product_name%` 则是特定产品的固定链接段。

最终的固定链接将呈现为: `https://example.com/wporg_product/wporg-is-awesome`。

您可以在自定义文章类型的编辑界面查看固定链接，操作方式与默认文章类型完全一致。

A Custom Slug for a Custom Post Type

To set a custom slug for the slug of your custom post type all you need to do is add a key => value pair to the `rewrite` key in the `register_post_type()` arguments array.

Example:

```
function wporg_custom_post_type() {
  register_post_type('wporg_product',
```

```
    array(
      'labels'      => array(
        'name'          => __( 'Products', 'textdomain' ),
        'singular_name' => __( 'Product', 'textdomain' ),
      ),
      'public'      => true,
      'has_archive' => true,
      'rewrite'     => array( 'slug' => 'products' ), // my custom slug
    )
  );
}
add_action('init', 'wporg_custom_post_type');
```

The above will result in the following URL structure: `https://example.com/products/%product_name%`

[warning]Using a generic slug like `products` can potentially conflict with other plugins or themes, so try to use one that is more specific to your content.[/warning]

[info]Unlike the custom post type identifiers, the duplicate slug problem can be solved easily by changing the slug for one of the conflicting post types.[/info]

If the plugin author included an `apply_filters()` call on the arguments, this can be done programmatically by overriding the arguments submitted via the `register_post_type()` function.

□□□ #2
□ □□□ □□□ 2026-03-26 17:08:39 UTC
□ □□□ □□□ 2026-03-26 17:08:39 UTC