

WordPress 入門

- [WordPressのquote order](#)
- [WordPressでACF](#)
- [ACFのOptions PageのPost Type](#)
- [ACF の](#)
- [acf_form\(\) WordPress の wp_insert_post\(\) の](#)
- [WordPress](#)
- [WordPress](#)
- [WP](#)
 - [Feature Image](#)

WordPress quote order

WooCommerce

WooCommerceWordPress quote order

- 1. Gravity FormsFormidable Forms quote order
- 2. WP Email ManagerFlamingo quote order
- 3. WP CRMZero BS WordPress CRM quote order
- 4. WP InvoiceSliced Invoices quote order
- 5. Custom Post TypeAdvanced Custom Fields quote order
- 6. WP QueryWP REST API quote order
- 7. WordPressPage Template quote order quote order

WordPress quote orderWooCommerce

WordPress 安装 ACF 插件教程

WordPress 安装 Advanced Custom Fields (ACF) 插件教程

第 1 步：安装 ACF 插件

1. 安装 ACF 插件
 - 在 WordPress 后台点击“插件”>“安装新插件”
2. 搜索并安装 ACF 插件
 - 在 WordPress 后台搜索“Advanced Custom Fields”>“Add New”
 - 点击“安装”按钮
 - 安装完成后，点击“激活”按钮
 - 安装 ACF 插件时，可以选择安装以下插件：
 - customer_name (姓名)
 - customer_email (电子邮件)
 - order_item (订单项目)
 - quantity (数量)
 - total_price (总价)
 - order_status (订单状态)
 - 安装 ACF 插件时，可以选择安装以下插件：

第 2 步：创建自定义后类型

1. 创建自定义后类型
 - 在 WordPress 后台点击“插件”>“安装新插件”

```
function create_order_post_type() {
    register_post_type('order',
        array(
            'labels' => array(
                'name' => __('Orders'),
                'singular_name' => __('Order')
            ),
            'public' => true,
            'has_archive' => true,
            'supports' => array('title', 'editor'),
            'menu_position' => 5,
            'show_in_menu' => true,
```

```

    )
    );
}

add_action('init', 'create_order_post_type');
```

2. ACF 创建

- ACF 创建“”

3. 创建

1. 创建

- order-form.php 创建

2. 创建

- acf_form() ACF 创建
- ```

<?php
/* Template Name: Order Form */
get_header();
?>

<h2></h2>

<?php
acf_form(array(
 'post_id' => 'new_post',
 'post_title' => true,
 'post_content' => true,
 'new_post' => array(
 'post_type' => 'order',
 'post_status' => 'publish'
),
 'field_groups' => array('your-acf-field-group-id'), // ID
 'submit_value' => '提交'
));
?>

<?php get_footer(); ?>
```

### 3. 创建

- ACF[acf/save\_post] ACF 创建

# 第 4 章

1. 

• “”

ACF
2. 

AC order\_status

# 第 5 章

1. 

get\_field() ACF
2. 

“” ACF Admin Columns 0 ACF

ACF

# ACF Options Page Post Ty

ACF Options Page Post Type

## 1. Options Page

- Options Page
- Options Page
- Options Page

- 
- Options Page

- 
- Options Page

## 2. Post Type

- 
- 
- ACF WooCommerce

- 
- Post Type
- API

- Options Page



# 如何调用ACF 自定义字段

如何调用 ACF 自定义字段

## 1. 如何调用 ACF 自定义字段

- `post_id` `acf_form()` `post_id` `'new_post'` 调用 ACF 自定义字段 ID
- `new_post` `new_post` `post_type` `post_status` `'order'` 调用 ACF 自定义字段 “ ”

```
acf_form(array(
 'post_id' => 'new_post',
 'new_post' => array(
 'post_type' => 'order', // 调用 ACF 自定义字段
 'post_status' => 'publish' // 调用 ACF 自定义字段
),
 'submit_value' => '提交'
));
```

## 2. 如何调用 ACF 自定义字段

- `AC order` 调用 ACF 自定义字段
- `field_XXXXXXX` 调用 ACF 自定义字段

## 3. 如何调用 ACF 自定义字段

- `WordPress` 调用 ACF 自定义字段
- 调用 ACF 自定义字段

## 4. 如何调用 ACF 自定义字段

- `ACF` 调用 ACF 自定义字段
- `functions.php` 调用 ACF 自定义字段 WordPress 调用 ACF 自定义字段

## 5. 如何调用 ACF 自定义字段



- WordPress
- **PHP**

PHP



A

acf\_form()

acf/save\_post

# WordPress 使用 `acf_form()` 与 `wp_insert_post()` 插入新文章

本文将介绍如何在 WordPress 中使用 ACF 插件来插入新文章。

## 使用 `acf_form()` 插入新文章

步骤

1. 安装 ACF 插件并激活。
2. 在 WordPress 后台创建一个新的 ACF 字段组。
3. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。

代码

1. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。
2. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。

## 使用 `wp_insert_post()` 插入新文章

步骤

1. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。
2. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。
3. 使用 `wp_insert_post()` 或 `update_field()` 插入新文章。

代码

1. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。
2. 在 WordPress 后台创建一个新的文章，并选择刚刚创建的 ACF 字段组。

总结

- 使用 `acf_form()` 可以方便地在 WordPress 后台插入新文章。
- 使用 `wp_insert_post()` 可以在 WordPress 后台插入新文章。

wp\_insert\_post() acf\_form()

# WordPress

WordPress



WordPress ACF WordPress

## 1.

functions.php

```
add_action('admin_menu', 'currency_exchange_settings_page');
function currency_exchange_settings_page() {
 add_menu_page(
 'Currency Exchange', // Title
 'Currency Exchange', // Menu Label
 'manage_options', // Capability
 'currency-exchange-settings', // Slug
 'render_currency_exchange_page', // Callback
 'dashicons-chart-line', // Icon
 100 // Position
);
}
```

## 2.

```
function render_currency_exchange_page() {
 // Render content
 if (isset($_POST['submit'])) {
 check_admin_referer('currency_exchange_save', 'currency_exchange_nonce'); // nonce
 $currency_rates = isset($_POST['currency_rates']) ? $_POST['currency_rates'] : [];
 update_option('currency_exchange_rates', $currency_rates); // Save rates
 }
}
```

```

 echo '<div class="updated"><p>[]</p></div>';
 }

 // []

 $currency_rates = get_option('currency_exchange_rates', []);

 ?>
 <div class="wrap">
 <h1>[]</h1>
 <form method="post">
 <?php wp_nonce_field('currency_exchange_save', 'currency_exchange_nonce'); ?>
 <table class="form-table">
 <thead>
 <tr>
 <th>[]</th>
 <th>[]</th>
 <th>[]</th>
 </tr>
 </thead>
 <tbody id="currency-rates-table">
 <?php if (!empty($currency_rates)) : ?>
 <?php foreach ($currency_rates as $index => $rate) : ?>
 <tr>
 <td><input type="text" name="currency_rates[<?php echo $index; ?>][code]"
value="<?php echo esc_attr($rate['code']); ?>" /></td>
 <td><input type="number" step="0.0001" name="currency_rates[<?php echo $index;
?>][rate]" value="<?php echo esc_attr($rate['rate']); ?>" /></td>
 <td><button type="button" class="remove-rate button">[]</button></td>
 </tr>
 <?php endforeach; ?>
 <?php endif; ?>
 </tbody>
 </table>
 <p><button type="button" class="button" id="add-rate">[]</button></p>
 <?php submit_button('[]'); ?>
 </form>
 </div>

 <script>

```

```
document.getElementById('add-rate').addEventListener('click', function() {
 const table = document.getElementById('currency-rates-table');
 const rowCount = table.rows.length;
 const row = table.insertRow(rowCount);
 row.innerHTML = `
 <td><input type="text" name="currency_rates[${rowCount}][code]" /></td>
 <td><input type="number" step="0.0001" name="currency_rates[${rowCount}][rate]" /></td>
 <td><button type="button" class="remove-rate button">❌</button></td>
 `;
 row.querySelector('.remove-rate').addEventListener('click', function() {
 row.remove();
 });
});

document.querySelectorAll('.remove-rate').forEach(button => {
 button.addEventListener('click', function() {
 this.closest('tr').remove();
 });
});
</script>
<?php
}
```

3. 

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|

```
[get_option()]
```

```
<?php
//
$currency_rates = get_option('currency_exchange_rates', []);
if (!empty($currency_rates)) {
 foreach ($currency_rates as $rate) {
 $code = esc_html($rate['code']);
 $value = esc_html($rate['rate']);
 echo "<p>{$code} : {$value}</p>";
 }
}
```

# 4.

```
function convert_currency($amount, $from_currency, $to_currency) {
 $currency_rates = get_option('currency_exchange_rates', []);
 $from_rate = 1;
 $to_rate = 1;

 foreach ($currency_rates as $rate) {
 if ($rate['code'] === $from_currency) {
 $from_rate = $rate['rate'];
 }
 if ($rate['code'] === $to_currency) {
 $to_rate = $rate['rate'];
 }
 }

 return ($amount / $from_rate) * $to_rate;
}

// 100
$converted_price = convert_currency(100, 'USD', 'EUR');
echo ' : ' . $converted_price;
```

# WordPress



[illegible]

# Feature Image

```
" class="rounded">
```