

# WordPress 101

- [WordPress 101 quote order](#)
- [WordPress 101 ACF](#)
- [101 ACF 101 Options Page 101 Post Type](#)
- [101 ACF 101 ACF](#)
- [101 acf\\_form\(\) 101 WordPress 101 wp\\_insert\\_post\(\) 101](#)
- [101 WordPress](#)
- [WordPress 101](#)
- [WP 101](#)
  - [101 Feature Image](#)
- [Contact Form 7 101 Mailchimp 101 API 101 Email 101 by Meta AI](#)
- [101 wordpress 101 wp\\_postmeta 101](#)

# WordPress quote order

Woocommerce

WooCommerce WordPress quote order

- Gravity Forms Formidable Forms quote order
- WP Email Manager Flamingo quote order
- WP CRM Zero BS WordPress CRM quote order
- WP Invoice Sliced Invoices quote order
- Custom Post Type Advanced Custom Fields quote order
- WP Query WP REST API quote order
- WordPress Page Template quote order quote order

WordPress quote order WooCommerce

# WordPress ACF

WordPress Advanced Custom Fields (ACF)

## 1

1. ACF
  - WordPress “Advanced Custom Fields”
2.
  - WordPress ACF “Custom Fields” > “Add New”
  - “” Post Type “”
  - - customer\_name ( )
    - customer\_email ( )
    - order\_item ( )
    - quantity ( )
    - total\_price ( )
    - order\_status ( )
  -

## 2 “”

1.
  - Custom Post functions.php

```
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

- o ACF “ ”

## 3

### 1.

- [order-form.php ]

### 2.

- acf\_form() ACF

```

<?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' => ' '
));
?>

<?php get_footer(); ?>

```

### 3.

- o ACF [acf/save\_post " ACF



# ACF Options Page Post Type

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

- [Barcode]



- **Options Page** [Barcode]
- [Barcode] **Post type** [Barcode]

# ACF

ACF

## 1.

- `post_id` `acf_form()` `post_id` `'new_post'` ACF ID
- `new_post` `new_post` `post_type` `post_status` `'order'` ACF " "

```
acf_form(array(  
    'post_id' => 'new_post',  
    'new_post' => array(  
        'post_type' => 'order', //  
        'post_status' => 'publish' //  
    ),  
    'submit_value' => ''  
));
```

## 2.

- `AC` `order`
- `field_XXXXXXX`

## 3.

- `WordPress`
- 

## 4.

- ACF
- `functions.php` ACF WordPress

## 5.



# acf\_form() WordPress

## wp\_insert\_post()

.....

[acf\_form()]

...

1. ACF .....
2. .... ACF .....
3. .... HTML .....

...

1. .... HTML .....ACF .....
2. ....

[wp\_insert\_post()]

...

1. .... **HTML** .... HTML .....
2. ....
3. [wp\_insert\_post()] [update\_field()] .....

...

1. ....
2. ....

....

- [acf\_form()] .....ACF .....
- [wp\_insert\_post()] .....

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 Settings', // Title
        'Currency Exchange Settings', // Menu Title
        'manage_options', // Capabilities
        'currency-exchange-settings', // Slug
        'render_currency_exchange_page', // Callback
        'dashicons-chart-line', // Icon
        100 // Priority
    );
}
```

## 2.

```
function render_currency_exchange_page() {
    //
    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); //
    }
}
```

```

    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>

```



## 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 



WP

# Feature Image

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

# Contact Form 7 & Mailchimp [ API & Email ] by Me

“ wordpress contact form 7 email API Mailchimp ”

Contact Form 7 & Mailchimp API email

## 1 Mailchimp API

1. Mailchimp
2. "API"
3. "API"
4. "API"
5. API

## 2 Contact Form 7 & Mailchimp

1. WordPress
2. " > "
3. "Contact Form 7 Mailchimp Extension"
4. "
- 5.

## 3 Contact Form 7 & Mailchimp

1. " > "
2. Mailchimp
3. "Mailchimp"
4. Mailchimp API
5. Mailchimp
6. "

## 4

- 1.
- 2.
3. Mailchimp

■■■■■■■■■■■■■■■■■■■■ Contact Form 7 ■■■■ email ■■■ Mailchimp ■■■■

# wordpress wp\_postmeta

wp\_postmeta WordPress CPT ACF WooCommerce  
orphaned data

## “ ”

### 1 Orphaned meta

- post\_id wp\_posts
- meta

```
SELECT COUNT(*)  
FROM wp_postmeta pm  
LEFT JOIN wp_posts p ON pm.post_id = p.ID  
WHERE p.ID IS NULL;
```

### 2 /

- WordPress

```
_edit_lock  
_edit_last  
_wp_old_slug  
_wp_trash_meta_time
```

### 3

- meta key

```
_yoast_wpseo_*  
_elementor_*  
_wc_last_active
```

### 4

- meta\_value = " meta\_value IS NULL



```
DELETE FROM wp_postmeta
WHERE meta_key LIKE '_yoast_wpseo_%';
```



SQL

- **Advanced Database Cleaner**  
postmeta commentmeta options
- **WP Optimize**

“ ”



SQL

```
OPTIMIZE TABLE wp_postmeta;
```

SQL



- SQL WP-Cron
- meta ACF Pro Custom Database Table wc\_product\_meta\_lookup
- ACF meta

PHP /wp-content/mu-plugins/cleanup-postmeta.php wp-cli

```
if (defined('WP_CLI') && WP_CLI) {
    /**
     * WP-CLI wp cleanup postmeta
     */
    WP_CLI::add_command('cleanup postmeta', function() {
```

```

global $wpdb;
WP_CLI::log('[] [] wp_postmeta...');

$orphans = $wpdb->query("
    DELETE pm FROM {$wpdb->postmeta} pm
    LEFT JOIN {$wpdb->posts} p ON pm.post_id = p.ID
    WHERE p.ID IS NULL
");
WP_CLI::success("[] meta[]{$orphans} []");

$empty_values = $wpdb->query("
    DELETE FROM {$wpdb->postmeta}
    WHERE meta_value = '' OR meta_value IS NULL
");
WP_CLI::success("[] meta[]{$empty_values} []");

$system_keys = [
    '_edit_lock',
    '_edit_last',
    '_wp_old_slug',
    '_wp_trash_meta_time',
    '_wp_trash_meta_status'
];
$keys_in = implode("','", $system_keys);
$system_deleted = $wpdb->query("
    DELETE FROM {$wpdb->postmeta}
    WHERE meta_key IN ('{$keys_in}')
");
WP_CLI::success("[]{}{$system_deleted} []");

$wpdb->query("OPTIMIZE TABLE {$wpdb->postmeta}");
WP_CLI::success("[] wp_postmeta []");
});
}

```

[] []

wp cleanup postmeta

[] []

