

Metadata

- meta
- meta
- meta
- meta data

meta

meta WordPress

WordPress

WordPress meta

add_post_meta()

`add_post_meta()` (int post_id, string meta_key, string meta_value, bool unique)

meta_key mycrazymetakeyname wporg_featured_menu meta_key

meta_value

unique

unique meta_key

update_post_meta()

`update_post_meta()` (int post_id, string meta_key, string meta_value, string prev_value)

`add_post_meta()` (int post_id, string meta_key, string meta_value, string prev_value)

delete_post_meta()

`delete_post_meta()` (int post_id, string meta_key, string meta_value)

stripslashes()

`stripslashes()` (string json)

JSON {"key": "value with \"escaped quotes\""}

```
$escaped_json = '{"key": "value with \"escaped quotes\""}';
update_post_meta( $id, 'escaped_json', $escaped_json );
$broken = get_post_meta( $id, 'escaped_json', true );
/*
$broken, after stripslashes(), ends up unparsable:
{"key": "value with "escaped quotes""}
*/
```

##

`wp_slash()` WP 3.6 `stripslashes()`

```
$escaped_json = '{"key":"value with \"escaped quotes\""}';
update_post_meta( $id, 'double_escaped_json', wp_slash( $escaped_json ) );
$fixed = get_post_meta( $id, 'double_escaped_json', true );
/*
$fixed, after stripslashes(), ends up as desired:
{"key":"value with \"escaped quotes\""}
*/
```

`meta_key` `the_meta()` WordPress `“ ”`

`add_meta_box()`

`meta_key` `“color”` `meta_value` `“red”`

```
add_post_meta( 68, '_color', 'red', true );
```

`meta_value` `meta_key`

meta

HTML HTML

`add_meta_box()` `add_meta_boxes()`

`post` `wporg_cpt`

```
function wporg_add_custom_box() {
    $screens = [ 'post', 'wporg_cpt' ];
    foreach ( $screens as $screen ) {
        add_meta_box(
            'wporg_box_id',          // Unique ID
            'Custom Meta Box Title', // Box title
            'wporg_custom_box_html', // Content callback, must be of type callable
            $screen                  // Post type
        );
    }
}

add_action( 'add_meta_boxes', 'wporg_add_custom_box' );
```

wporg_custom_box_html() HTML

HTML

```
function wporg_custom_box_html( $post ) {  
    ?>  
    <label for="wporg_field">Description for this field</label>  
    <select name="wporg_field" id="wporg_field" class="postbox">  
        <option value="">Select something...</option>  
        <option value="something">Something</option>  
        <option value="else">Else</option>  
    </select>  
    <?php  
}
```

POST HTML " " "

[get_post_meta\(\)](#) postmeta

```
function wporg_custom_box_html( $post ) {  
    $value = get_post_meta( $post->ID, '_wporg_meta_key', true );  
    ?>  
    <label for="wporg_field">Description for this field</label>  
    <select name="wporg_field" id="wporg_field" class="postbox">  
        <option value="">Select something...</option>  
        <option value="something" <?php selected( $value, 'something' ); ?>>Something</option>  
        <option value="else" <?php selected( $value, 'else' ); ?>>Else</option>  
    </select>  
    <?php  
}
```

[selected\(\)](#)

WP

save_post save_post

postmeta WordPress

wporg_field _wporg_meta_key

```
function wporg_save_postdata( $post_id ) {
    if ( array_key_exists( 'wporg_field', $_POST ) ) {
        update_post_meta(
            $post_id,
            '_wporg_meta_key',
            $_POST['wporg_field']
        );
    }
}

add_action( 'save_post', 'wporg_save_postdata' );
```

WP

`add_meta_boxes` callback `add_meta_boxes` `div`

WP

`remove_meta_box()` `add_meta_box()`

`add_meta_box()` wp-includes/edit-form-advanced.php .

WP

OOO OOP OOOOOOOOOOOOOOOOOOOOOOOOOOOOO

OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

```
abstract class WPOrg_Meta_Box {

    /**
     * Set up and add the meta box.
     */
    public static function add() {
        $screens = [ 'post', 'wporg_cpt' ];
        foreach ( $screens as $screen ) {
            add_meta_box(
                'wporg_box_id',      // Unique ID
                'Custom Meta Box Title', // Box title
                [ self::class, 'html' ], // Content callback, must be of type callable
                $screen              // Post type
            );
        }
    }

    /**
     * Save the meta box selections.
     */
    /** @param int $post_id The post ID.
     */
    public static function save( int $post_id ) {
        if ( array_key_exists( 'wporg_field', $_POST ) ) {
            update_post_meta(
                $post_id,
                '_wporg_meta_key',
                $_POST['wporg_field']
            );
        }
    }

    /**
     * Display the meta box HTML to the user.
     */
}
```



```

/*
 * @param WP_Post $post Post object.
 */
public static function html( $post ) {
    $value = get_post_meta( $post->ID, '_wporg_meta_key', true );
    ?>
    <label for="wporg_field">Description for this field</label>
    <select name="wporg_field" id="wporg_field" class="postbox">
        <option value="">Select something...</option>
        <option value="something" <?php selected( $value, 'something' ); ?>>Something</option>
        <option value="else" <?php selected( $value, 'else' ); ?>>Else</option>
    </select>
    <?php
    }
}

add_action( 'add_meta_boxes', [ 'WPOrg_Meta_Box', 'add' ] );
add_action( 'save_post', [ 'WPOrg_Meta_Box', 'save' ] );

```

Ajax

form `$_POST`

AJAX

JavaScript

change AJAX

```

/*jslint browser: true, plusplus: true */
(function ($, window, document) {
    'use strict';
    // execute when the DOM is ready
    $(document).ready(function () {
        // js 'change' event triggered on the wporg_field form field
        $('#wporg_field').on('change', function () {
            // our code
        });
    });
});

```

```
}(jQuery, window, document));
```

```
    
```

```
    
```

```
POST wporg_field 
```

```
/*jslint browser: true, plusplus: true */
(function ($, window, document) {
    'use strict';
    // execute when the DOM is ready
    $(document).ready(function () {
        // js 'change' event triggered on the wporg_field form field
        $('#wporg_field').on('change', function () {
            // jQuery post method, a shorthand for $.ajax with POST
            $.post(wporg_meta_box_obj.url, // or ajaxurl
                {
                    action: 'wporg_ajax_change', // POST data, action
                    wporg_field_value: $('#wporg_field').val(), // POST data, wporg_field_value
                    post_ID: jQuery('#post_ID').val() // The ID of the post currently being edited
                }, function (data) {
                    // handle response data
                    if (data === 'success') {
                        // perform our success logic
                    } else if (data === 'failure') {
                        // perform our failure logic
                    } else {
                        // do nothing
                    }
                }
            );
        });
    });
})(jQuery, window, document);
```

wporg_meta_box_obj JavaScript WordPress AJAX URL

WordPress AJAX URL; ajaxurl

AJAX post wporg_cpt

/plugin-name/admin/meta-boxes/js/admin.js plugin-name

/plugin-name/plugin.php

```
function wporg_meta_box_scripts()
{
    // get current admin screen, or null
    $screen = get_current_screen();
    // verify admin screen object
    if (is_object($screen)) {
        // enqueue only for specific post types
        if (in_array($screen->post_type, ['post', 'wporg_cpt'])) {
            // enqueue script
            wp_enqueue_script('wporg_meta_box_script', plugin_dir_url(__FILE__) . 'admin/meta-boxes/js/admin.js',
                ['jquery']);
            // localize script, create a custom js object
            wp_localize_script(
                'wporg_meta_box_script',
                'wporg_meta_box_obj',
                [
                    'url' => admin_url('admin-ajax.php'),
                ]
            );
        }
    }
}

add_action('admin_enqueue_scripts', 'wporg_meta_box_scripts');
```

```
// The piece after `wp_ajax_` matches the action argument being sent in the POST request.
add_action( 'wp_ajax_wporg_ajax_change', 'my_ajax_handler' );
```

```
/**
 * Handles my AJAX request.
 */

function my_ajax_handler() {
    // Handle the ajax request here
    if ( array_key_exists( 'wporg_field_value', $_POST ) ) {
        $post_id = (int) $_POST['post_ID'];
        if ( current_user_can( 'edit_post', $post_id ) ) {
            update_post_meta(
                $post_id,
                '_wporg_meta_key',
                $_POST['wporg_field_value']
            );
        }
    }

    wp_die(); // All ajax handlers die when finished
}
```

□ AJAX □ AJAX □ ~~AJAX~~ □ ~~index~~ □

- WordPress [] [] [] []
- [] [] WordPress [] [] [] [] [] [] [] []
- WordPress [] [] [] [] [] [] [] [] [] []

meta data

- `the_meta()` -
- `get_post_custom()` `get_post_meta()` -
- `get_post_custom_values()` -