acf/schema/post_field_objects/post_type={post_type}

- Description
- Changelog
- Parameters
- Return
- Example

Filter field objects for a specific post type.

Description

This filter is a post type specific version of `acf/schema/post_field_objects`. It allows you to modify field objects only for a specific post type without affecting other post types.

The dynamic portion of the hook name, `{post_type}`, refers to the post type slug.

Changelog

- Added in version 6.8.0

Parameters

```
apply_filters( 'acf/schema/post_field_objects/post_type={post_type}', $field_objects, $post );
```

- `$field_objects` *(array|null)* Field objects array from previous filters
- `$post` *(WP_Post)* The post object

Return

(array|null) Modified field objects array.

Example

This example adds product-specific data for WooCommerce products.

```
/**
 * Add WooCommerce product data to JSON-LD.
 *
 * @param array|null $field_objects The field objects array.
 * @param WP_Post    $post          The post object.
 * @return array Modified field objects.
 */
function my_acf_schema_product_fields( $field_objects, $post ) {
    if ( null === $field_objects ) {
        $field_objects = get_field_objects( $post->ID, false );
    }

    if ( ! is_array( $field_objects ) ) {
        $field_objects = array();
    }

    // Get WooCommerce product
    $product = wc_get_product( $post->ID );
    if ( ! $product ) {
        return $field_objects;
    }

    // Add price as Offer
    $field_objects['wc_price'] = array(
        'value'           => $product->get_price(),
        'schema_property' => 'Offer.price',
        'type'            => 'number',
    );
```

```
    $field_objects['wc_currency'] = array(
        'value'           => get_woocommerce_currency(),
        'schema_property' => 'Offer.priceCurrency',
        'type'            => 'text',
    );

    $field_objects['wc_availability'] = array(
        'value'           => $product->is_in_stock() ? 'https://schema.org/InStock' :
'https://schema.org/OutOfStock',
        'schema_property' => 'Offer.availability',
        'type'            => 'url',
    );

    return $field_objects;
}
add_filter( 'acf/schema/post_field_objects/post_type=product', 'my_acf_schema_product_fields',
10, 2 );
```

This example adds recipe-specific calculated fields.

```
/**
 * Add calculated fields for recipes.
 *
 * @param array|null $field_objects The field objects array.
 * @param WP_Post    $post          The post object.
 * @return array Modified field objects.
 */
function my_acf_schema_recipe_fields( $field_objects, $post ) {
    if ( null === $field_objects ) {
        $field_objects = get_field_objects( $post->ID, false );
    }

    if ( ! is_array( $field_objects ) ) {
        $field_objects = array();
    }

    // Add recipe category from taxonomy
    $categories = get_the_terms( $post->ID, 'recipe_category' );
    if ( $categories && ! is_wp_error( $categories ) ) {
        $field_objects['recipe_category'] = array(
            'value'           => $categories[0]->name,
```

```
            'schema_property' => 'recipeCategory',
            'type'            => 'text',
        );
    }

    return $field_objects;
}
add_filter( 'acf/schema/post_field_objects/post_type=recipe', 'my_acf_schema_recipe_fields',
10, 2 );
```

□□□ #2
□ □□□ □□□ 2026-08-29 15:44:03 UTC
□ □□□ □□□ 2026-08-29 15:44:03 UTC