



Axios toolkit for WordPress API: Working with custom post types

The last two articles on using Axios with the WordPress REST API has worked with standard items in the WordPress world like post and pages.

WordPress also gives you the option of creating your own content types with [Custom Post Types](#) (CPTs)

We create custom post types in PHP, either in a theme's `functions.php` or in a plugin.

When working with the REST API, the important configuration is `show_in_rest`. Specifying this attribute with a value of `true` will make the custom post type available in the WordPress API.

The CPT definition for an essay CPT looks like this:

```
<?php
function rivendellweb_custom_essay_type() {
    register_post_type( 'essa'essays'// WordPress CPT Options Start
    array(
        'labels' => array(
            'name' => __( 'Essays' ),
            'singular_name' => __( 'Essay' )
        ),
        'has_archive' => true,
        'public' => true,
        'rewrite' => array('slug' => 'essay'),
        'show_in_rest' => true,
        'supports' => array('title',
            'editor',
            'excerpt',
            'author',
            'thumbnail',
```

```

        'comments',
        'revisions',
        'custom-fields',
        'permalinks',
        'featured_image')
    )
};
}
add_action( 'init', 'rivendellweb_custom_essay_type' );

```

We can then treat essays like we do posts. We can retrieve pages of essays

```

async function getEssays(numberOfEssays = 10, pageNumber = 1) {
  await axios
    .get('/wp-json/wp/v2/essays/', {
      params: {
        per_page: numberOfEssays,
        page: pageNumber,
      },
      headers: {
        Authorization: `Bearer ${access_token}`,
      },
    })
    .then((res) => {
      console.log(res.data);
    })
    .catch((err) => {
      console.log(err);
    });
}

```

Just like we did with posts, we can create a full CRUD set of functions around these custom post types.