WordPress as a headless CMS

Prasad Ajinkya

Working with WordPress

First impressions with WordPress

- EduPristine as a marketing and outreach platform

Meeting the WordPress community

- Mumbai WordPress meetup
- 13Llama Studio as a web design studio focusing on WordPress based websites
- WordCamp 2015

Moving on from being a WordPress developer to a myriad things in tech

- WordPress at Homeville
- WordPress as a headless CMS

What is a headless CMS?

A content management system where the front end is effectively separated from the admin/backend.

- It could be the backend for a custom JS based front end or

- the API for a mobile application

- the content service for multiple web and mobile apps
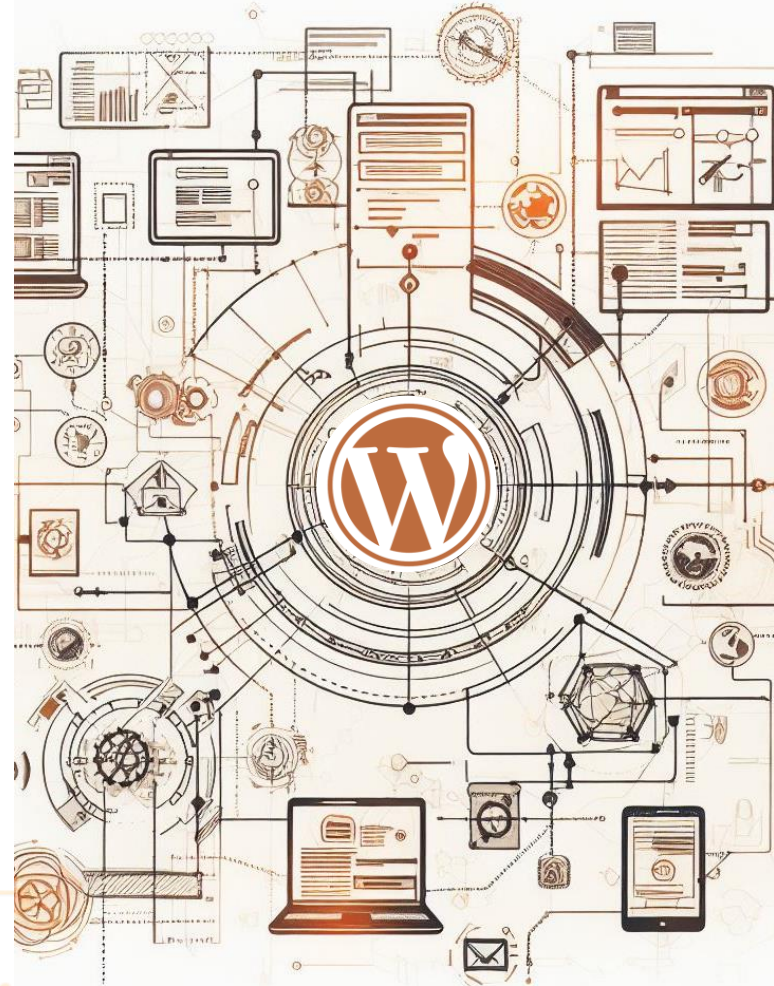
WordPress and its evolution

- WordPress v3.0 - 2010
  - Ability to register Custom Post Types (CPTs)
  - Ability to register Custom Taxonomy
- Initial work REST API for WordPress - 2013
  - [Make note](#) by Ryan McCue
  - REST API plugin released - 2014
- WordPress v4.7 – 2016
  - REST API

When should you consider a headless CMS

- Rapid application prototyping and MVP development

- Re-using a set of content across multiple applications and use cases e.g blog articles, job listings, faqs, etc.

- Multiple business edits in content needing developer intervention

- Re-use of CMS across multiple use cases

Building blocks for headless

- Registering a new Custom Post Type
- Adding REST API capabilities

```php
// Register Custom Post Type
function custom_post_type() {

    $labels = array(
        'name'                  => _x( 'Post Types', 'Post Type General Name', 'text_domain' ),
        'singular_name'         => _x( 'Post Type', 'Post Type Singular Name', 'text_domain' ),
        'menu_name'             => __( 'Post Types', 'text_domain' ),
        'name_admin_bar'        => __( 'Post Type', 'text_domain' ),
        'archives'              => __( 'Item Archives', 'text_domain' ),
        'attributes'            => __( 'Item Attributes', 'text_domain' ),
        'parent_item_colon'     => __( 'Parent Item:', 'text_domain' ),
        'all_items'             => __( 'All Items', 'text_domain' ),
        'add_new_item'          => __( 'Add New Item', 'text_domain' ),
        'add_new'               => __( 'Add New', 'text_domain' ),
        'new_item'              => __( 'New Item', 'text_domain' ),
        'edit_item'             => __( 'Edit Item', 'text_domain' ),
        'update_item'           => __( 'Update Item', 'text_domain' ),
        'view_item'             => __( 'View Item', 'text_domain' ),
        'view_items'            => __( 'View Items', 'text_domain' ),
        'search_items'          => __( 'Search Item', 'text_domain' ),
        'not_found'             => __( 'Not found', 'text_domain' ),
        'not_found_in_trash'    => __( 'Not found in Trash', 'text_domain' ),
        'featured_image'        => __( 'Featured Image', 'text_domain' ),
        'set_featured_image'    => __( 'Set featured image', 'text_domain' ),
        'remove_featured_image' => __( 'Remove featured image', 'text_domain' ),
        'use_featured_image'    => __( 'Use as featured image', 'text_domain' ),
        'insert_into_item'      => __( 'Insert into item', 'text_domain' ),
        'uploaded_to_this_item' => __( 'Uploaded to this item', 'text_domain' ),
        'items_list'            => __( 'Items list', 'text_domain' ),
        'items_list_navigation' => __( 'Items list navigation', 'text_domain' ),
        'filter_items_list'     => __( 'Filter items list', 'text_domain' ),
    );
    $args = array(
        'label'                 => __( 'Post Type', 'text_domain' ),
        'description'           => __( 'Post Type Description', 'text_domain' ),
        'labels'                => $labels,
        'supports'              => array( 'title', 'editor' ),
        'taxonomies'            => array( 'category', 'post_tag' ),
        'hierarchical'          => false,
        'public'                => true,
        'show_ui'               => true,
        'show_in_menu'          => true,
        'menu_position'         => 5,
        'show_in_admin_bar'     => true,
        'show_in_nav_menus'     => true,
        'can_export'            => true,
        'has_archive'           => true,
        'exclude_from_search'   => false,
        'publicly_queryable'    => true,
        'capability_type'       => 'page',
        'show_in_rest'          => true,
    );
    register_post_type( 'post_type', $args );

}
add_action( 'init', 'custom_post_type', 0 );
```

Things to consider if you go the headless route

- Focus on performance and server response times

- Secure the API and wp-admin

- GraphQL + Your JS framework (Next.js, Gatsby, etc)

- Maximum number of items per response needs to be taken into consideration

- REST API Handbook (link given in references)

Pros and Cons

- Pros
  - Flexibility of content and content architectures
  - Solid user friendly CMS
  - Scalability
  - Custom UX
  - Front end de-coupled from WordPress backend means one more layer of security
- Cons
  - Multiple codebases (frontend, WordPress)
  - No preview
  - Front end is to be built from scratch – accessibility, site optimization needs to be thought through

References

- [REST API Handbook](#)

- [WordPress REST API Tutorial](#) – Jetpack blog on REST API, highlights hygiene factors when using a REST API

- [GenerateWP](#) – Easy code generator for all things WP

- [WPGraphQL](#) – Plugin to extend the WP API into a GraphQL schema

- [Pods](#) – Plugin to customize CPTs, create new ones and extend existing ones on REST

- [Frontity](#) – Opinionated React framework for headless WordPress sites

- Image credits – [DALL-E 3 Powered by Bing](#)

Headless CMS covered in WordCamps

WordCamp	Year	Speaker	Topic and Link
Mumbai	2015	Amit Sharma	WordPress as The Backbone of a Mobile App
Sydney	2016	Bronson Quick	Using WordPress As A Headless CMS
Europe	2017	Rian Rietveld	Accessibility In The Age Of The Headless CMS
CapeTown	2017	Petya Raykovska	The WordPress REST API – a guide for non developers
NYC	2017	Wes Turner	"Headless" WordPress and What It Really Means
Hamilton	2018	Thiago Loureiro	Headless WordPress + React
Singapore	2019	Ian Labao	Face Your Fear: The Headless WordPress!
Milano	2019	Cristiano Zanca	WordPress loves GatsbyJS and GraphQl
Brighton	2019	Craig West	Decoupled/Headless WP and WP Components for non-WP sites
Europe	2020	Luis Herranz	Headless WordPress – current status and remaining challenges
Nicaragua	2021	Billi Ramirez	JAMStack usando WordPress como Headless CMS
Colombo	2021	Hussain Thajutheen	Headless WordPress Authentication with React
Kathmandu	2022	Sandip Basnet	Harnessing the Power of GraphQL in WordPress
Malaysia	2023	Liton Arefin	What is Headless WordPress and why should you use it?

Thank you!

@kidakaka