WordPress to Gatsby

Jack Pritchard

Jack Pritchard

Freelance Fullstack Website Developer

Shameless Plug

Taking on projects for Autumn 2019

I focus on front-end applications and have experience with -

- React
- NextJS
- Gatsby
- WordPress
- Timber

WordPress to Gatsby

In this presentation I hope to show you the power of Gatsby, before creating a bridge between dependable development tools, processes and frameworks like WordPress.

Before We Get Started!

The Final Product

Gatsby

"Gatsby is a free and open source framework based on React that helps developers build blazing fast websites and apps"

Modern web tech without the headache

Enjoy the power of the latest web technologies – React.js , Webpack , modern JavaScript and CSS and more — all set up and waiting for you to start building.

Bring your own data

Gatsby’s rich data plugin ecosystem lets you build sites with the data you want — from one or many sources: Pull data from headless CMSs, SaaS services, APIs, databases, your file system, and more directly into your pages using GraphQL .

Scale to the entire internet

Forget complicated deploys with databases and servers and their expensive, time-consuming setup costs, maintenance, and scaling fears. Gatsby.js builds your site as “static” files which can be deployed easily on dozens of services.

Future-proof your website

Do not build a website with last decade's tech. The future of the web is mobile, JavaScript and APIs—the JAMstack. Every website is a web app and every web app is a website.

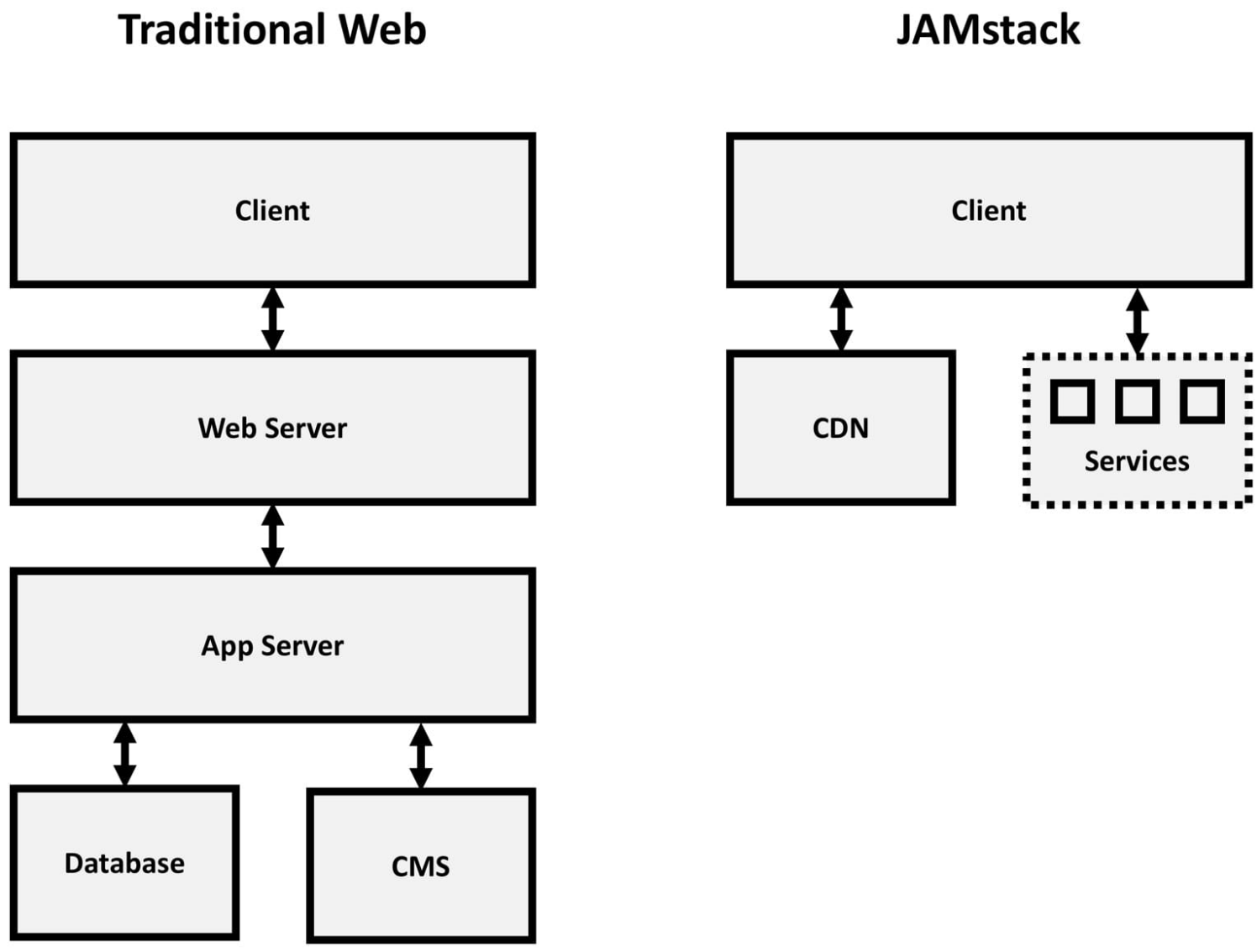
Speed past the competition

Instead of waiting to generate pages when requested, pre-build pages and lift them into a global cloud of servers — ready to be delivered instantly to your users.

My Favourite Benefits of Gatsby

- Hot reloading
- Enables an offline first approach
- No need for heavy queries
- No need for server queries
- Console warnings as you develop, to encourage best practices
- Incredibly quick products

Traditional server vs Gatsby (JAM)



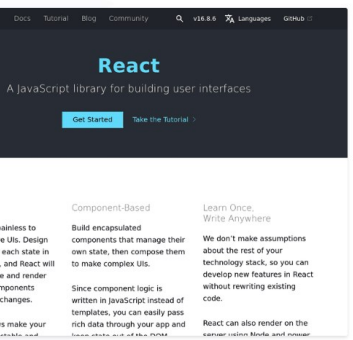
Gatsby Showcase

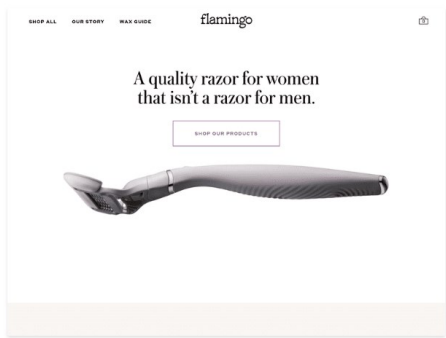
v: "What Can Gatsby Plugins Do?" - Learn about different types of plugins and how to build one!

by Docs Tutorials Plugins Features Blog Showcase Contributing Search

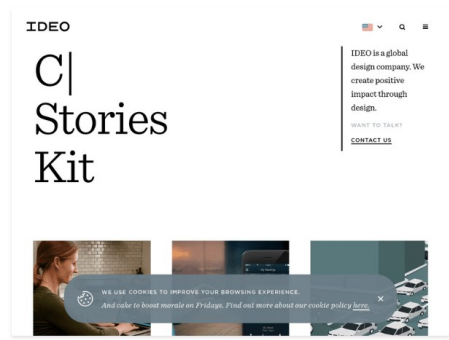
ed Sites View all →

Want to get featured? Submit your

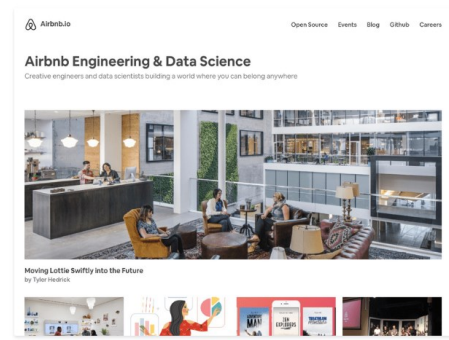




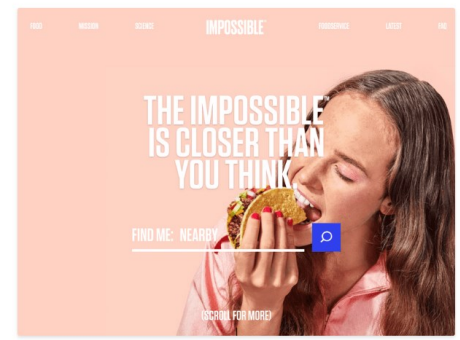
Flamingo



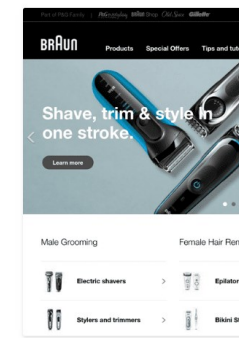
IDEO



Airbnb Engineering & Data Science



Impossible Foods



Braun

Agency, Technology, Featured, Consulting, User Experience

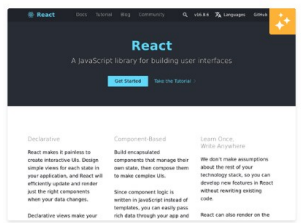
Blog, Gallery, Featured

Food, Featured

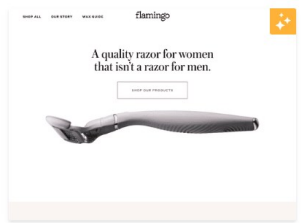
eCommerce, Featured

eCommerce, Beauty, Featured

Sites (562)

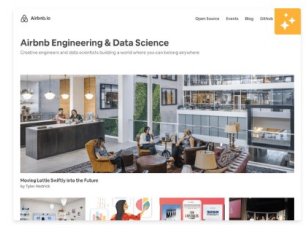


ReactJS
Web Development, Featured

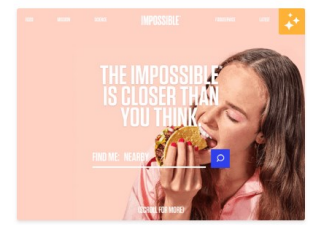


Flamingo
eCommerce, Beauty, Featured

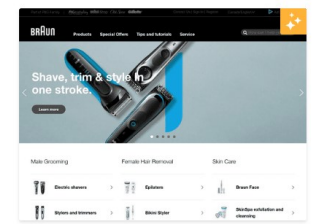
IDEO
Agency, Technology, Featured, Consulting, User Experience



Airbnb Engineering & Data Science
Blog, Gallery, Featured



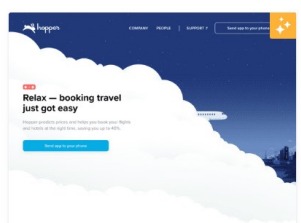
Impossible Foods
Food, Featured



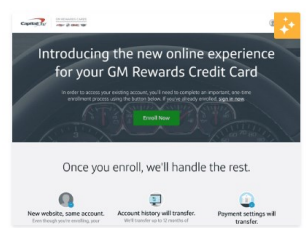
Braun
eCommerce, Featured

NYC Pride 2019 | WorldPride NYC | Stonewall50
Education, Marketing, Nonprofit, Featured

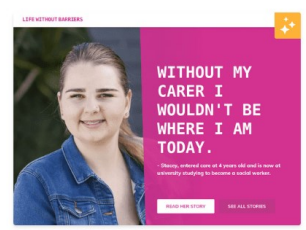
The State of European Tech



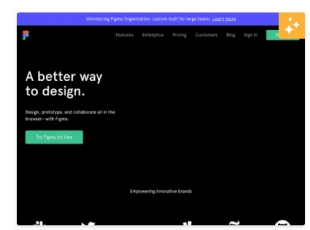
Hopper



GM Capital One



Life Without Barriers | Foster Care



Figma

Gatsby Plugins

ch Gatsby Library

gin-react-helmet 526k
cument head data with react-helmet.
op-in server rendering support for

ge 393.4k
g React image component with optional
the blur-up effect.

gin-manifest 339.6k
in which adds a manifest.webmanifest
es progressive web apps

gin-offline 321.3k
in which sets up a site to be able to run

gin-page-creator 621.5k
in that automatically creates pages
components in specified directories

rce-filesystem 544.1k
in which parses files within a directory
arsing by other plugins

gin-sharp 466.6k
the Sharp image manipulation library
lugins

sformer-sharp 379.3k
sformer plugin for images using Sharp

sformer-remark 271.3k
sformer plugin for Markdown using the
ary and ecosystem

gin-sitemap 189.9k
in that automatically creates a sitemap

Official Plugin View plugin on GitHub See starters using this

gatsby-plugin-offline

Adds drop-in support for making a Gatsby site work offline and more resistant to bad network connections. It creates a service worker for the site and loads the service worker into the client.

If you're using this plugin with `gatsby-plugin-manifest` (recommended) this plugin should be listed *after* that plugin so the manifest file can be included in the service worker.

Install

```
npm install --save gatsby-plugin-offline
```

How to use

JS

```
// In your gatsby-config.js
plugins: [`gatsby-plugin-offline`]
```

Overriding options

When adding this plugin to your `gatsby-config.js`, you can pass in options to override the default Workbox config.

The default config is as follows. Warning: you can break the offline support by changing these options, so tread carefully.

JS

```
const options = {
  importWorkboxFrom: `local`,
  globDirectory: rootDir,
  globPatterns,
  modifyUrlPrefix: {
    // If `pathPrefix` is configured by user, we should replace
    // the default prefix with `pathPrefix`.
    "/": `${pathPrefix}/`,
  },
  cacheId: `gatsby-plugin-offline`,
  // Don't cache-bust JS or CSS files, and anything in the static directory,
```

Sold Yet?

(Gatsby I'll happily take any sponsorship at this point)

Let's Build a Site 🔨

Our Backend 🍑


WORDPRESS

WordPress

I assume many of you in the room know what WordPress is.

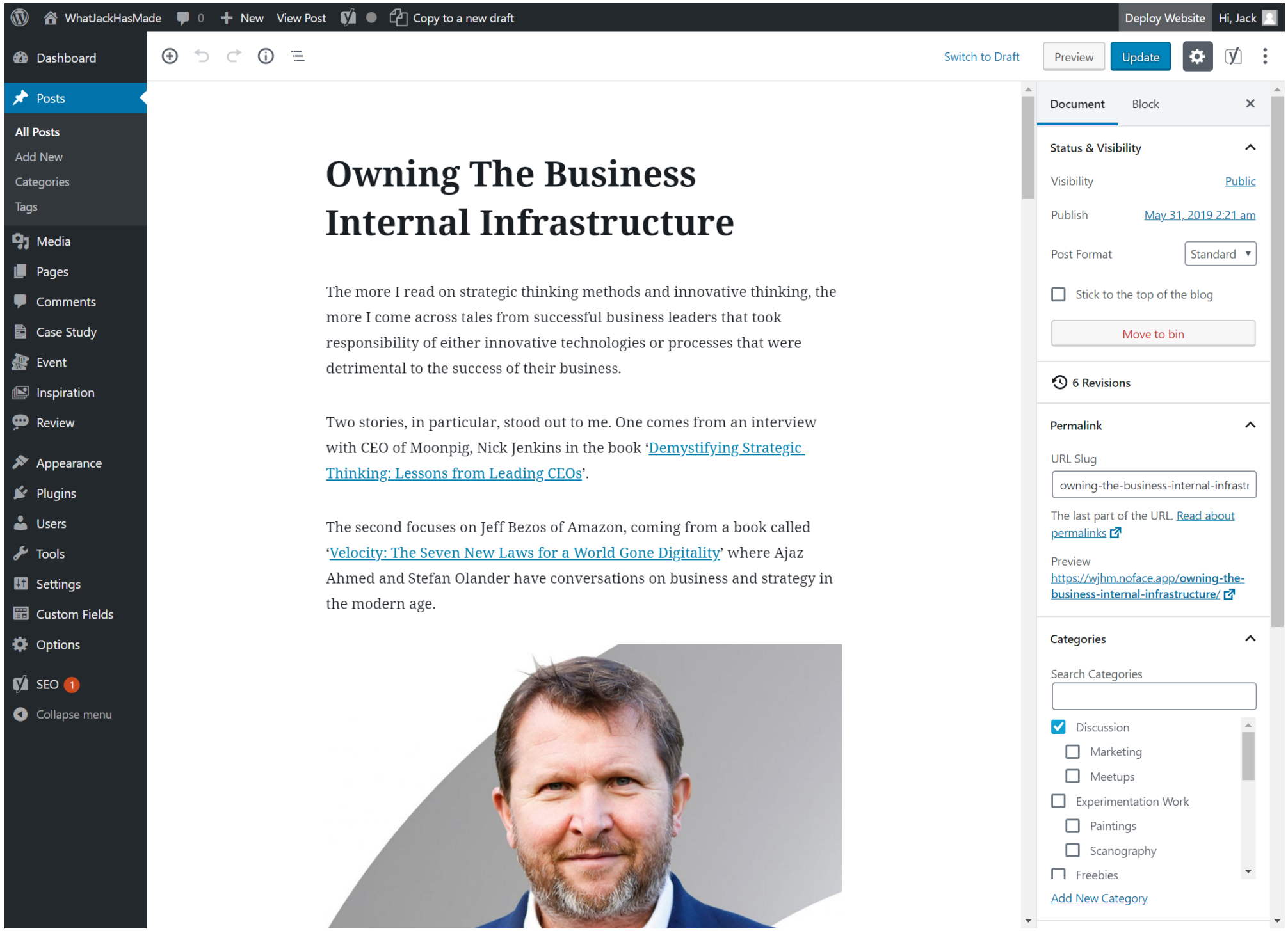
However, for those who need a recap, WordPress is an open source CMS (Content Management System).

Gutenberg

For anyone using WordPress, I assume you know what Gutenberg is; For better or for worse.

Gutenberg is now the default content editor for WordPress.

WordPress - Gutenberg



Headless WordPress

Headless WordPress

WordPress setups often involve a theme which provides front-end and back-end files to render webpages.

Instead, we are going to be creating custom API endpoints in our WordPress theme and importing that data to Gatsby to render pages.

Essentially using WordPress as only a backend.

Headless WordPress Endpoint

What we have in all WordPress setups (as of release 4.7.0+) - https://wjhm.noface.app/wp-json/wp/v2/posts/

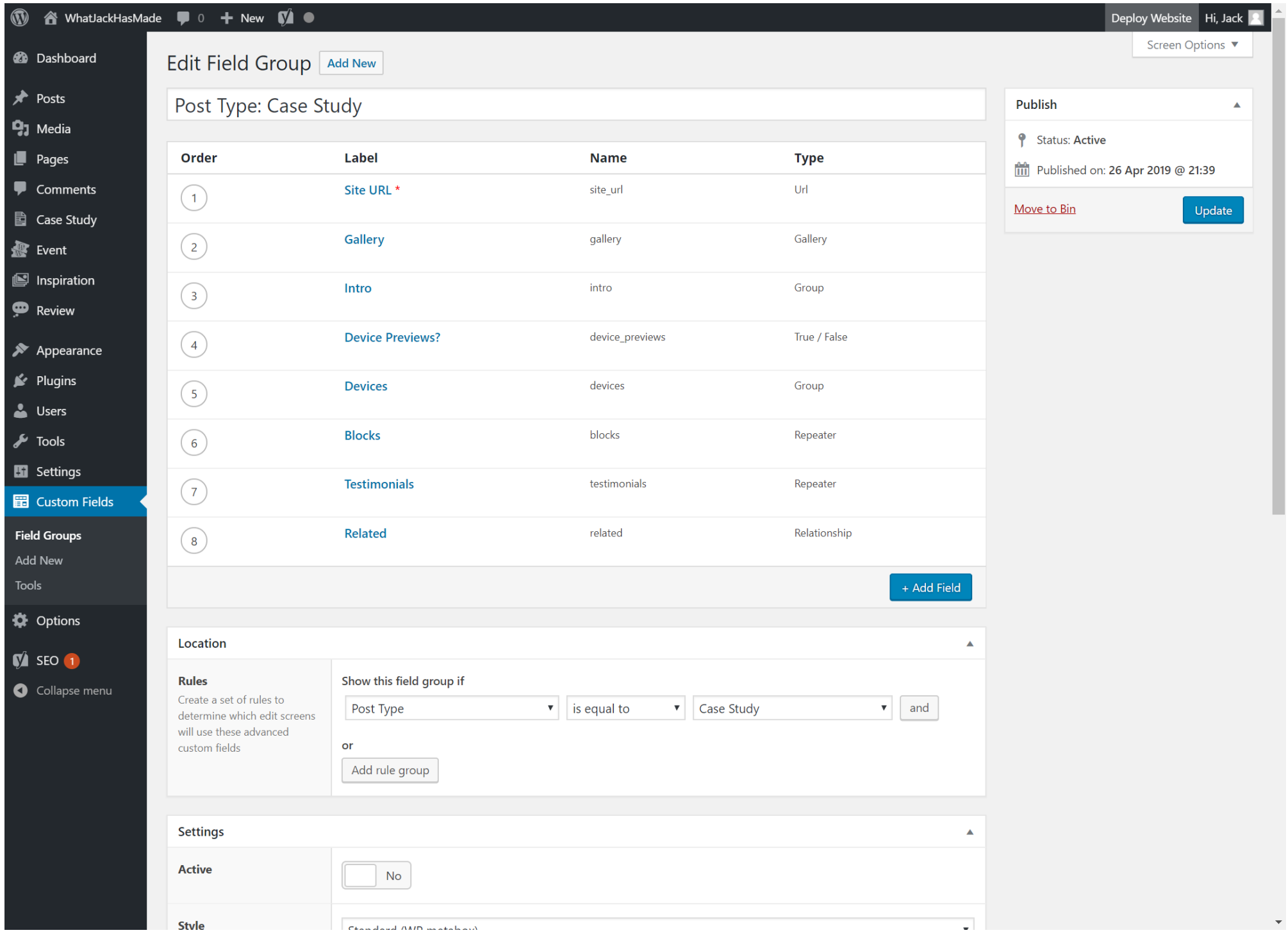

RANGER CUSTOM
FIELDS

Advanced Custom Fields (ACF)

ACF is a must have for any WordPress developer looking to improve their themes.

ACF allows developers to add custom fields (who'd have thought?) to posts and pages on a WordPress site.

Advanced Custom Fields (ACF)



That's All of Our Backend Tech Stack 👍

Details to Follow

I know we've covered a few different technologies, but this talk is a step by step process.

So I will be going in to some more detail shortly.

Setting a Task

I had set myself a task, to rebuild my personal website https://whatjackhasmade.co.uk 🔌

Designs

I'm a big fan of atomic design, for anyone unfamiliar, atomic design is a concept of looking at pages and slicing them into smaller and smaller components.

Homepage

AWARD WINNING WEBSITE DEVELOPER

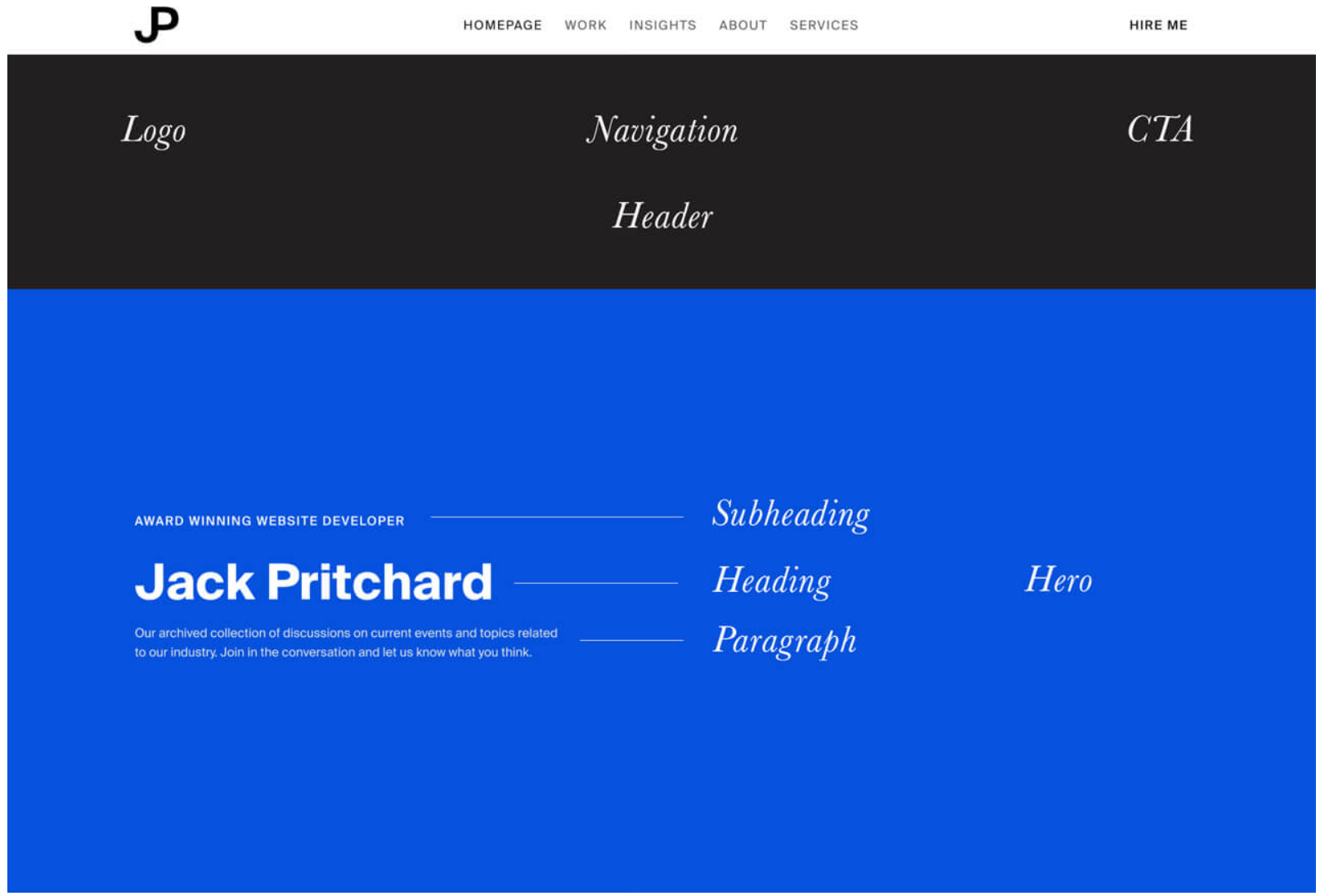
Jack Pritchard

Our archived collection of discussions on current events and topics related to our industry. Join in the conversation and let us know what you think.

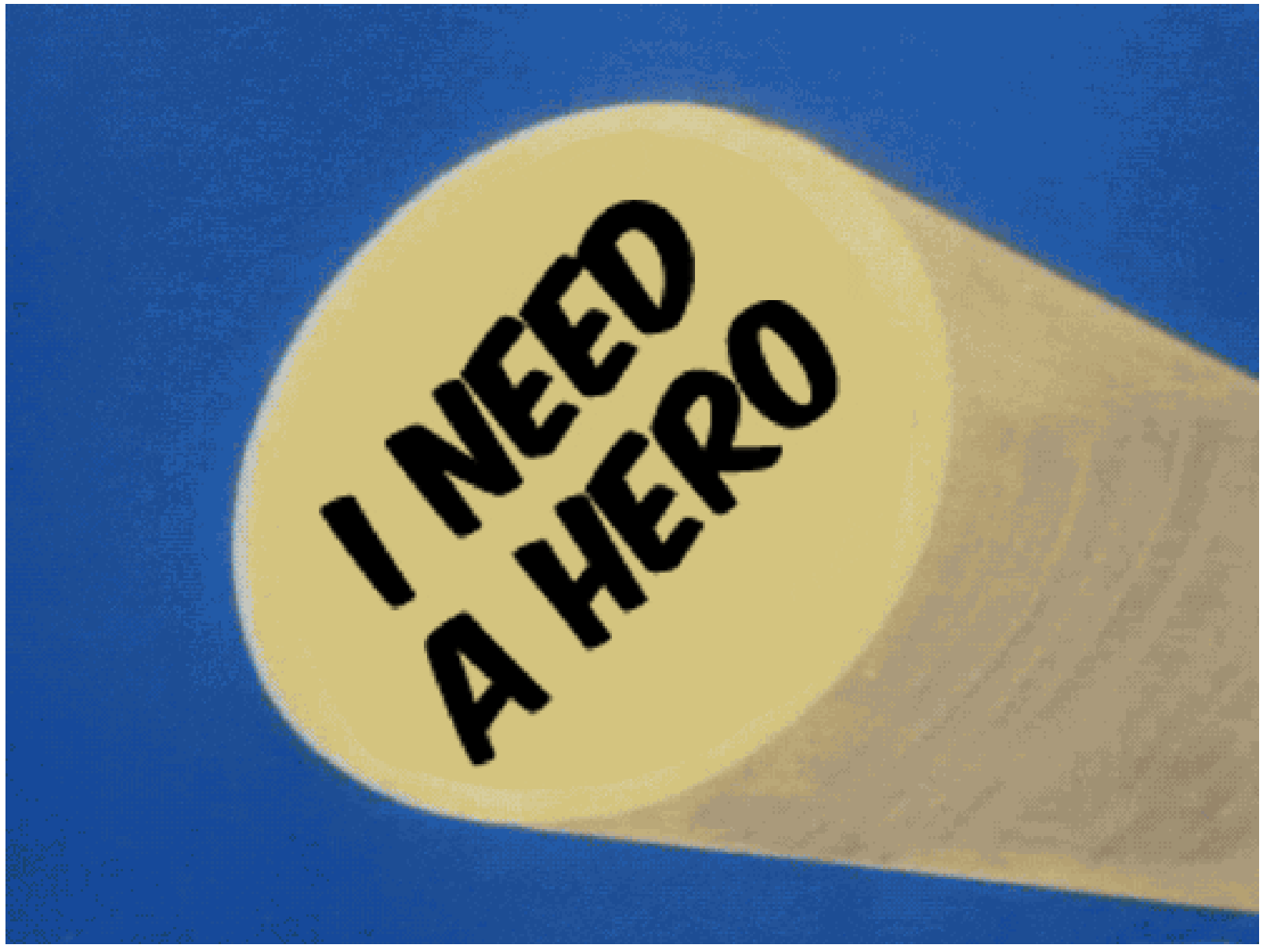
I'VE HELPED MY CLIENTS ACHIEVE THEIR GOALS AND IMPROVE THEIR BUSINESSES

What Goes In To Building Exceptional Products?

Homepage - Disected


HOMEPAGE
WORK
INSIGHTS
ABOUT
SERVICES
HIRE ME
Logo
Navigation
CTA
Header
AWARD WINNING WEBSITE DEVELOPER
Jack Pritchard
Our archived collection of discussions on current events and topics related to our industry. Join in the conversation and let us know what you think.
Subheading
Heading
Paragraph
Hero

We'll Only Be Focussing on The Hero


I NEED
A HE

AWARD WINNING WEBSITE DEVELOPER

Jack Pritchard

Our archived collection of discussions on current events and topics related to our industry. Join in the conversation and let us know what you think.

Subheading

Heading

Paragraph

Hero

Development Time

So we've got our designs, now time to start Developing

WordPress

All we need to know is that we are running version 5 or above.

My WordPress environment is hosted at https://wjhm.noface.app on a shared hosting platform with PHP available.

WordPress Plugins - Minimum Requirements

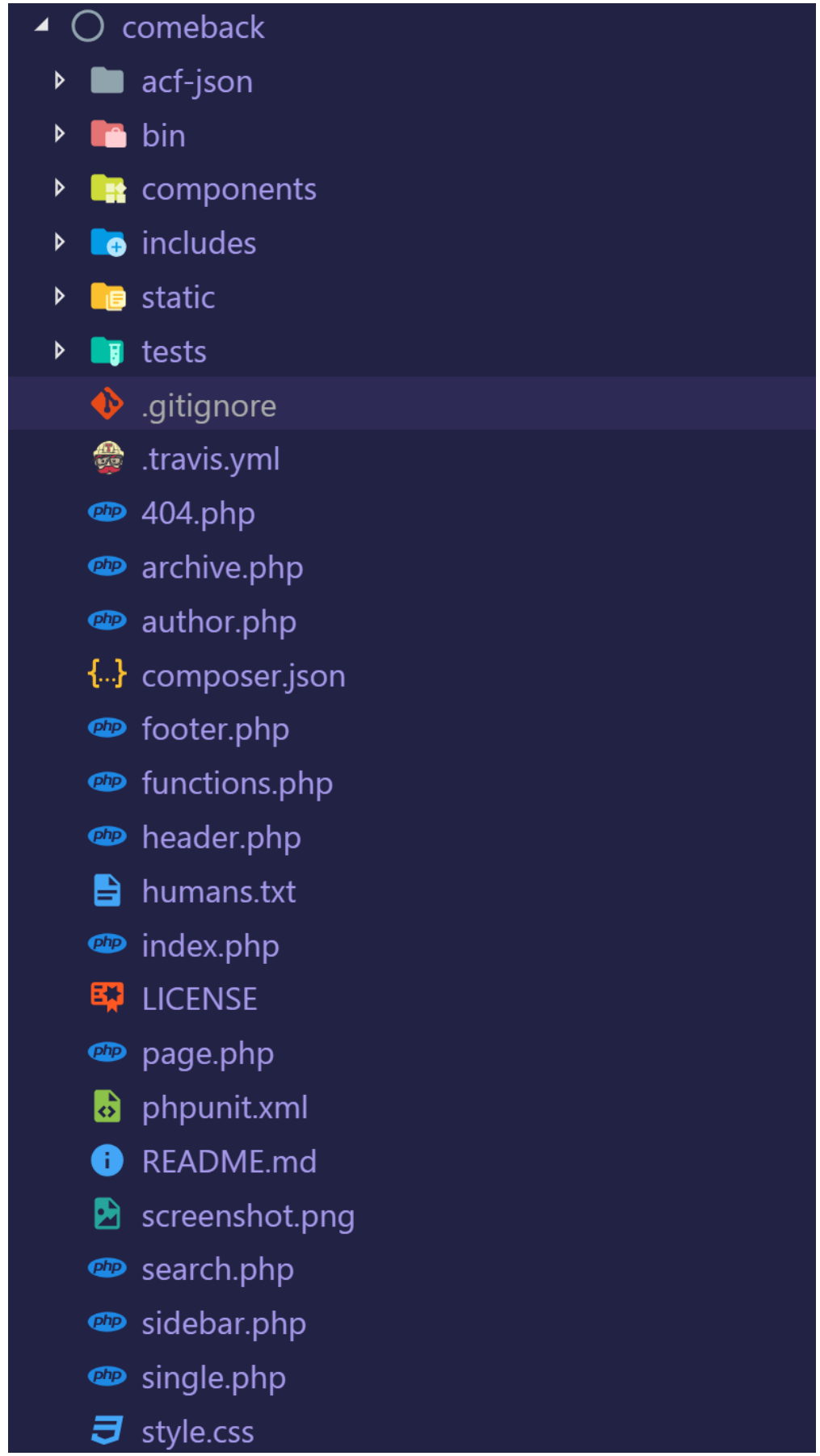
The only plugin we need to follow this process is ACF 5.8.0

WordPress Theme

To kickstart the WordPress theme, I've downloaded and installed the Timber Starter Theme.

Feel free to download my theme repository as a starting point and alter as you need to - https://github.com/whatjackhasmade/WJHM-Wordpress-Theme

WordPress Theme Structure

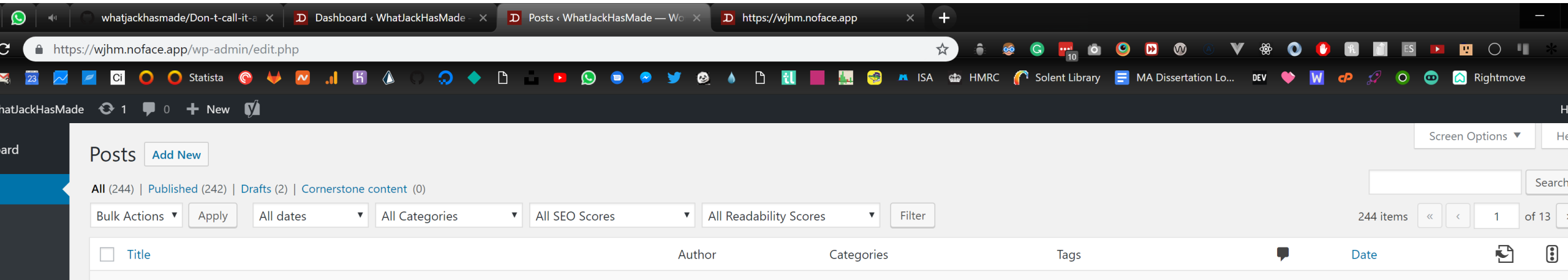

comeback
acf-json
bin
components
includes
static
tests
.gitignore
.travis.yml
404.php
archive.php
author.php
composer.json
footer.php
functions.php
header.php
humans.txt
index.php
LICENSE
page.php
phpunit.xml
README.md
screenshot.png
search.php
sidebar.php
single.php
style.css

Removing Front-End Theme Files

To keep focus on a Headless WordPress build, I have for this project scraped all twig file templates and their components.

WordPress Posts and Pages - WP Admin

With a fresh theme and environment, I then pulled in my local machines database full of posts and pages from my previous site.



whatjackhasmade/Don-t-call-it-a | Dashboard ‹ WhatJackHasMade | Posts ‹ WhatJackHasMade — Wo | https://wjhm.noface.app

https://wjhm.noface.app/wp-admin/edit.php

Statista ISA HMRC Solent Library MA Dissertation Lo... Rightmove

hatJackHasMade 1 0 New

Screen Options

Posts

Add New

All (244) | Published (242) | Drafts (2) | Cornerstone content (0)

Search

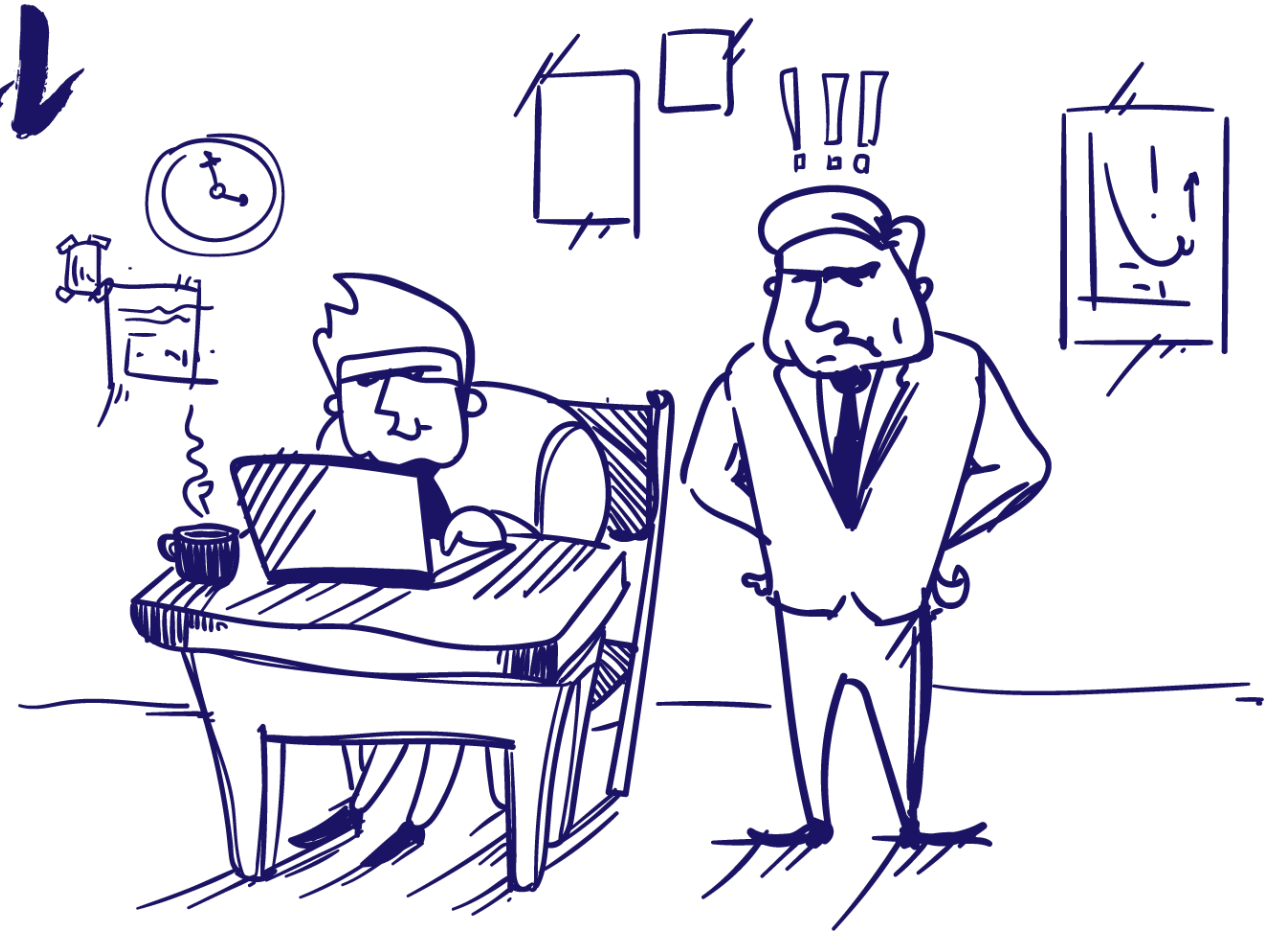
Bulk Actions | Apply | All dates | All Categories | All SEO Scores | All Readability Scores | Filter

244 items « ‹ 1 of 13

Title	Author	Categories	Tags		Date		
Performance Matters: Adaptive Images — Draft	Jack	Performance, Website Design	—	—	Last Modified 2018/11/21	0	
Return of the Jack	Jack	Discussion	—	—	Published 2018/09/22	0	
What is Gutenberg?	Jack	Discussion, Website Design	cms, content, editor, gutenberg, management, pineapple, system, tinymce, wordpress	—	Published 2017/09/14	0	
M E A N	Jack	Experimentation Work	—	—	Published 2017/09/12	1	
ShareUp 5 – Learning and Discussions	Jack	Discussion, Meetups	etch, Etch UK, event, events, jack, Jack Davies, local, meetup, moov2, Organise, shareup, shareup soton, soton, southampton	—	Published 2017/06/18		
How to enable WebP in WordPress	Jack	Website Design	browser support, compression, content, Custom, editor, file, file type, google chrome, image, image compression, mce, network, network speed, plugin, support, webp, wordpress	—	Published 2017/06/17		
WebP Image Format	Jack	Performance, Website Design	file, files, format, image, image extension, image file, image format, webp	—	Published 2017/06/13		
Compress Presentations	Jack	Discussion	categories, compress, document, documents, download, media, pdf, pdfs, performance, powerpoint, presentation, slide, slides, speed	—	Published 2017/06/11		
3 Common Pitfalls of Growth Hacking	Jack	Discussion, Marketing	action, growth, growth hacking, hacking, Marketing, newsletter, promotion	—	Published 2017/06/09		
Winchester Creatives June 2017	Jack	Discussion, Meetups	creatives, event, events, meetup, meetups, networking, presentation,	—	Published 2017/06/06		

This is what we're left with

WHERE THE HELL IS OUR SITE?



Headless WordPress Endpoint

Besides an angry project manager that expected something to look at, we have our API endpoints - https://wjhm.noface.app/wp-json/wp/v2/posts/

How far can you get with the existing API?

You can get pretty far in what you want to achieve with the existing API.

However, there are a few issues I have with the endpoints...

Issues with the existing API

Bloated

The endpoints were built to include as many use cases as possible, as a result, it is bloated with meta information I don't need

Issues with the existing API

Limitations

You are limited to 100 results per page (I have over 250+ blog posts I want to query without passing in several endpoints)

Issues with the existing API

Missing Data

I would love to get access to some ACF or Yoast SEO information from the endpoints but the current endpoints are currently setup without plugins in mind.

Issues with the existing API

Great for setting HTML - Not for generating

Creating Our Endpoints

Fortunately, you can hook into WordPress and create your own endpoints.

I've got a whole blog post about it on my company site - https://noface.co.uk/rest-endpoint-wordpress-menus

Creating our WordPress Posts Endpoint

```php
<?php
/* Register function to run at rest_api_init hook */
add_action('rest_api_init', function () {
    /* Setup siteurl/wp-json/posts/v2/all */
    register_rest_route('posts/v2', '/all', array(
        'methods' => 'GET',
        'callback' => 'rest_posts',
        'args' => array(
            'slug' => array(
                'validate_callback' => function ($param, $request, $key) {
                    return is_string($param);
                },
            ),
```

Before we generate our API data, we need to register the endpoint

Creating our Endpoint Content

```php
<?php
function rest_posts($data)
{
    $args = array(
        'posts_per_page' => -1,
        'post_status' => 'publish',
        'post_type' => 'post',
    );

$loop = new WP_Query($args);

if ($loop) {
    $insightItems = array();
    while ($loop->have_posts()): $loop->the_post();
        $the_content = convert_content(get_the_content());
        $the_content = get_the_content();
        array_push(
            $insightItems, array(
                'content' => $the_content,
                'date' => get_the_time('c'),
                'excerpt' => get_post_meta(get_the_ID(), '_yoast_wpseo_metadesc', true),
                'id' => get_the_ID(),
                'imageLargest' => get_the_post_thumbnail_url(get_the_ID(), 'largest'),
```

We're going to create the callback function we just mentioned for our rest endpoint

Custom Endpoints ❤

Old Endpoint - https://wjhm.noface.app/wp-json/wp/v2/posts/

Our Endpoint - https://wjhm.noface.app/wp-json/posts/v2/all

Now we have our endpoints, let's create some content

Revisiting Our Atomic Designs

Now before we go ahead and develop a building block, let's first pick the block we want to build and break it down.

One of my favourite components for a website is as mentioned, the hero component, so let's stick with that.

Atoms That Create The Hero

In the hero component we have a few parts of data that will be editable. Including the background colour, subheading, heading, and paragraph/intro.

Mapping the anatomy of the Hero

To simplify the block even further, as I am the only content editor, I am going to make the hero component require two fields.

- Background Colour (ACF - Colour Picker)
- Content (ACF - WYSIWYG)

Register the block in functions.php

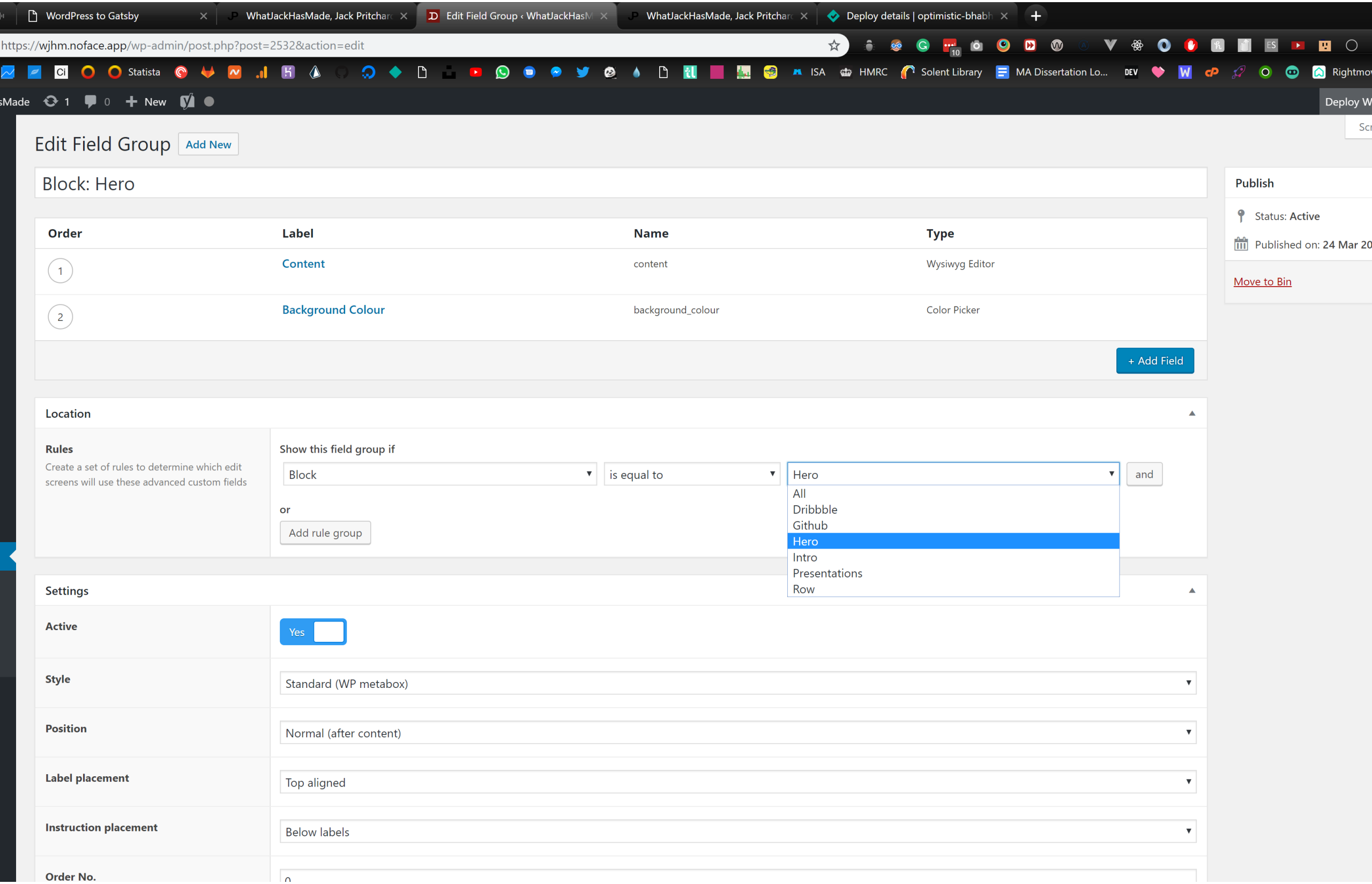
The current documentation can be found at - https://www.advancedcustomfields.com/resources/acf_register_block/

With Timber documentation on 'acf_register_block' found at - https://timber.github.io/docs/guides/gutenberg/

Register the block in functions.php

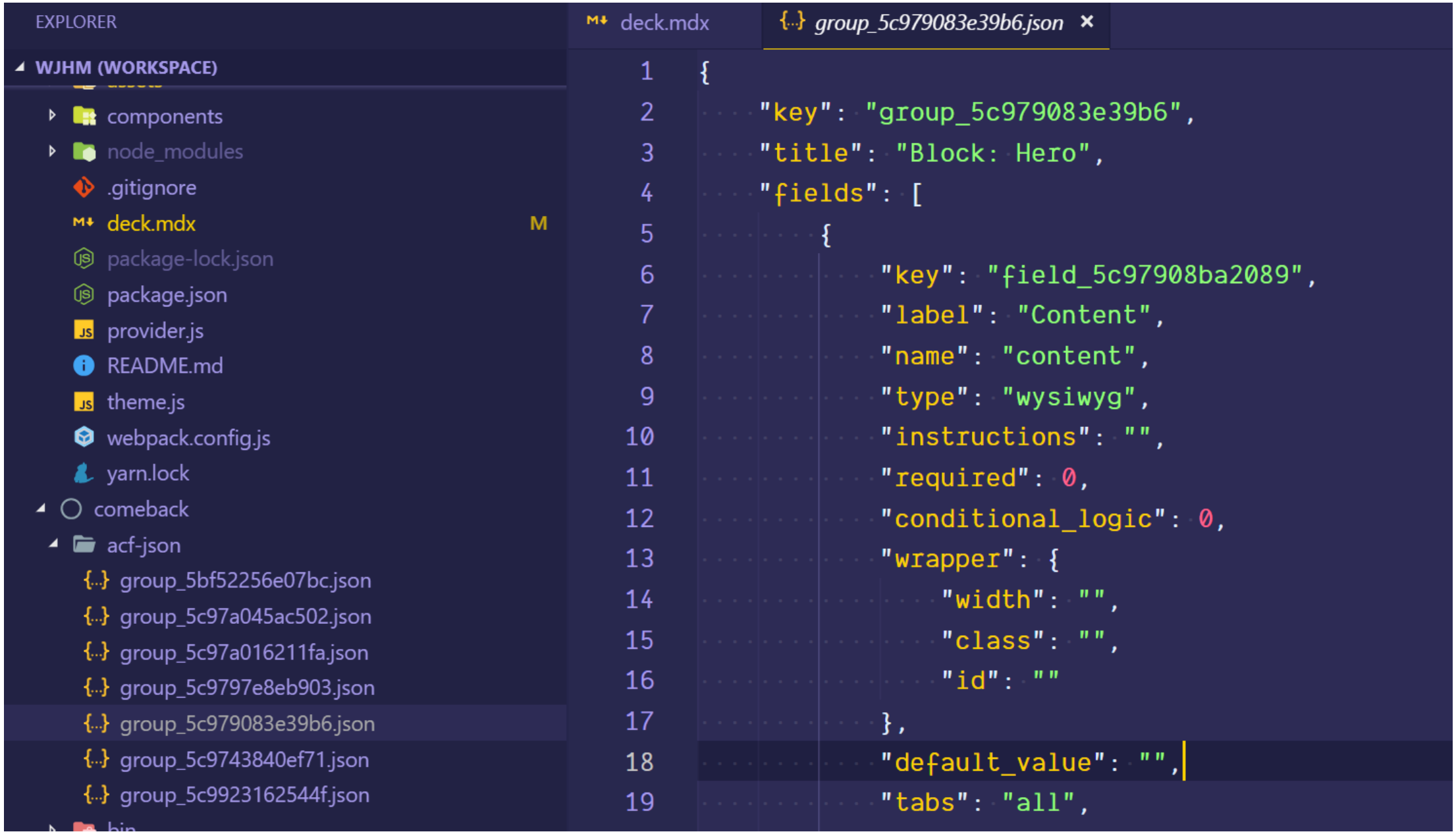
I'll briefly go through my register blocks function

Register the block in ACF



Protip: acf-json folder

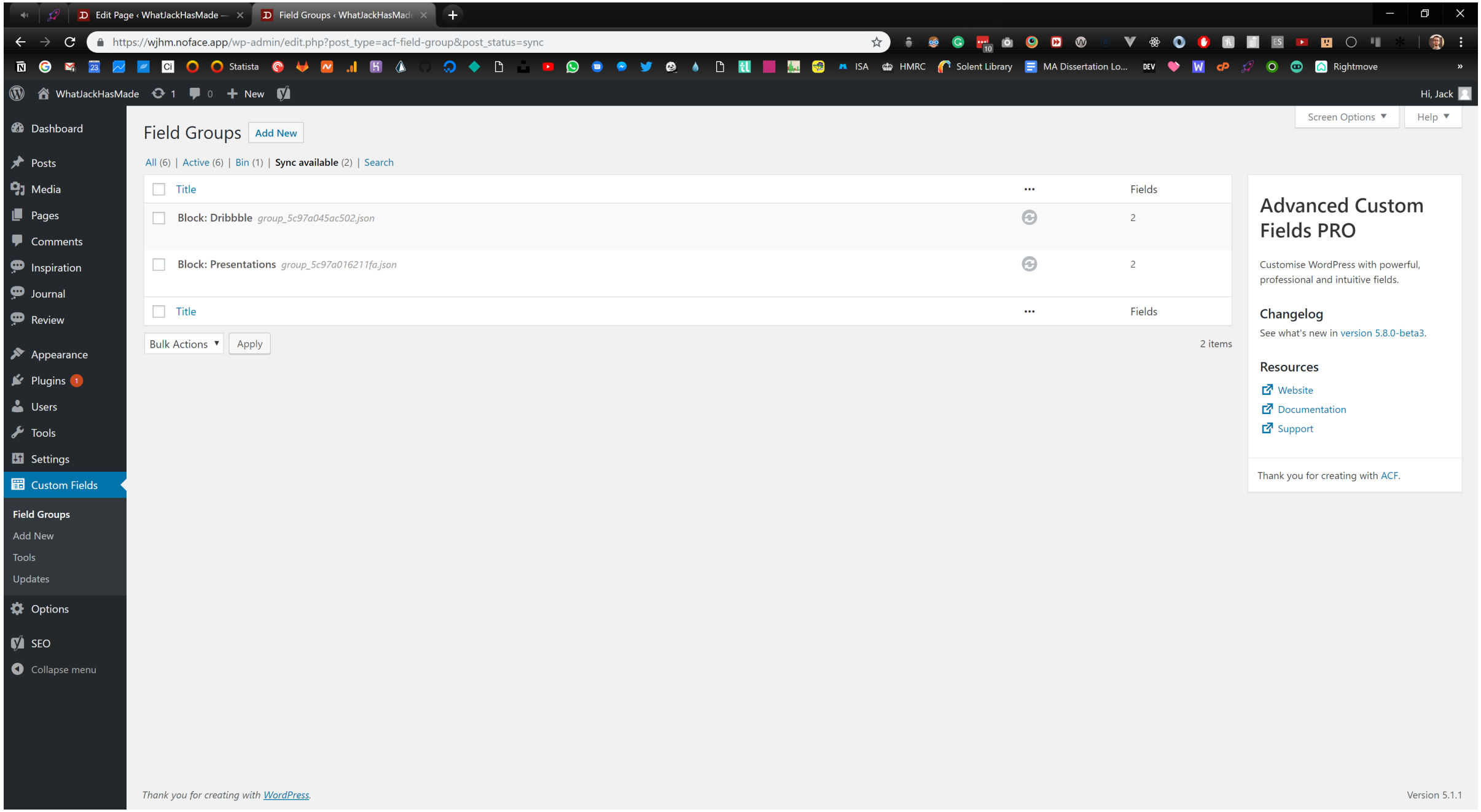
Quick protip, if you are an ACF Pro plugin user, creating a folder named 'acf-json' in your theme directory will save any configurations of your custom field groups to the theme.


EXPLORER
WJHM (WORKSPACE)
components
node_modules
.gitignore
deck.mdx
M
package-lock.json
package.json
provider.js
README.md
theme.js
webpack.config.js
yarn.lock
comeback
acf-json
group_5bf52256e07bc.json
group_5c97a045ac502.json
group_5c97a016211fa.json
group_5c9797e8eb903.json
group_5c979083e39b6.json
group_5c9743840ef71.json
group_5c9923162544f.json
deck.mdx
group_5c979083e39b6.json
{
"key": "group_5c979083e39b6",
"title": "Block: Hero",
"fields": [
{
"key": "field_5c97908ba2089",
"label": "Content",
"name": "content",
"type": "wysiwyg",
"instructions": "",
"required": 0,
"conditional_logic": 0,
"wrapper": {
"width": "",
"class": "",
"id": ""
},
"default_value": "",
"tabs": "all",

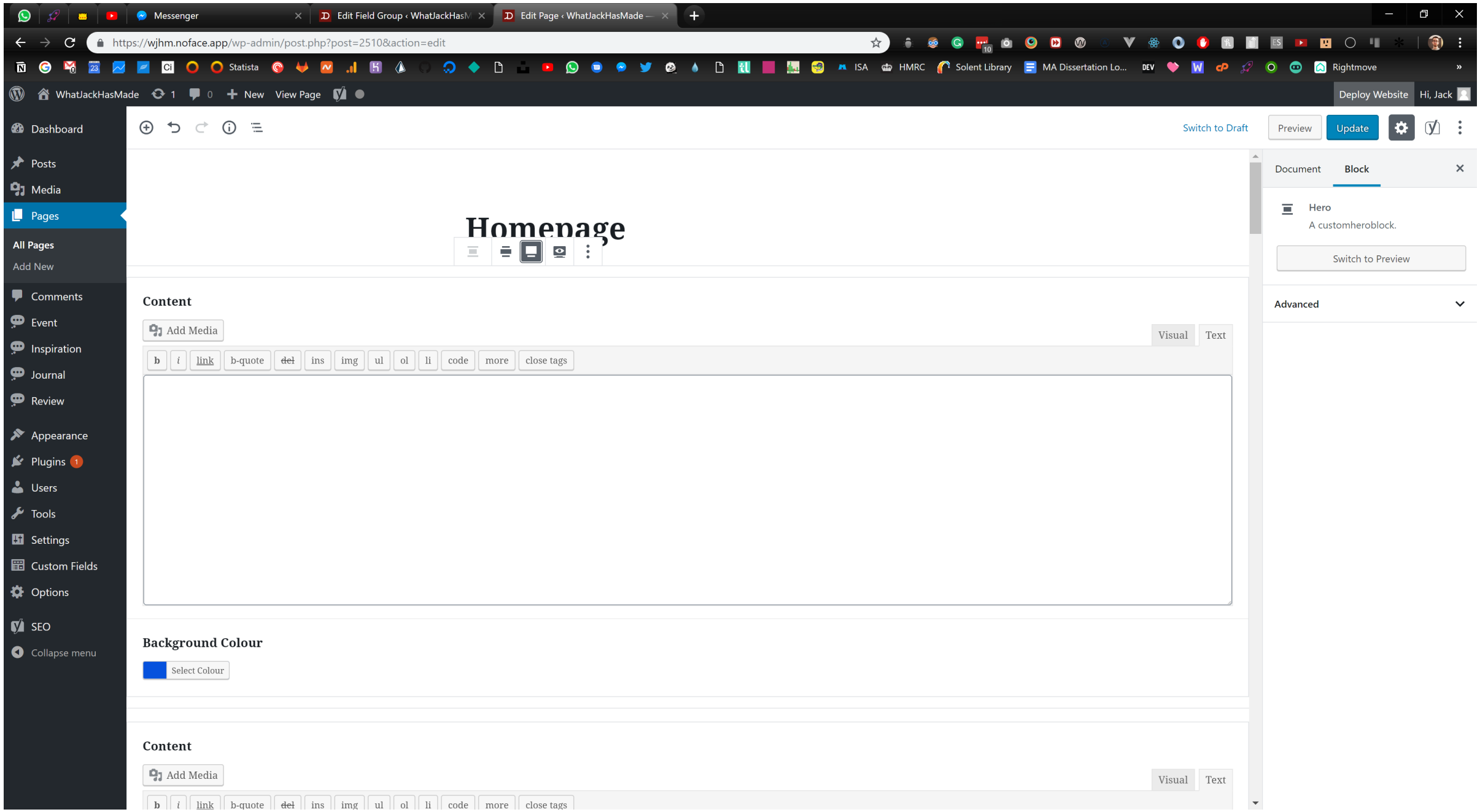
Push this to your online WordPress environment

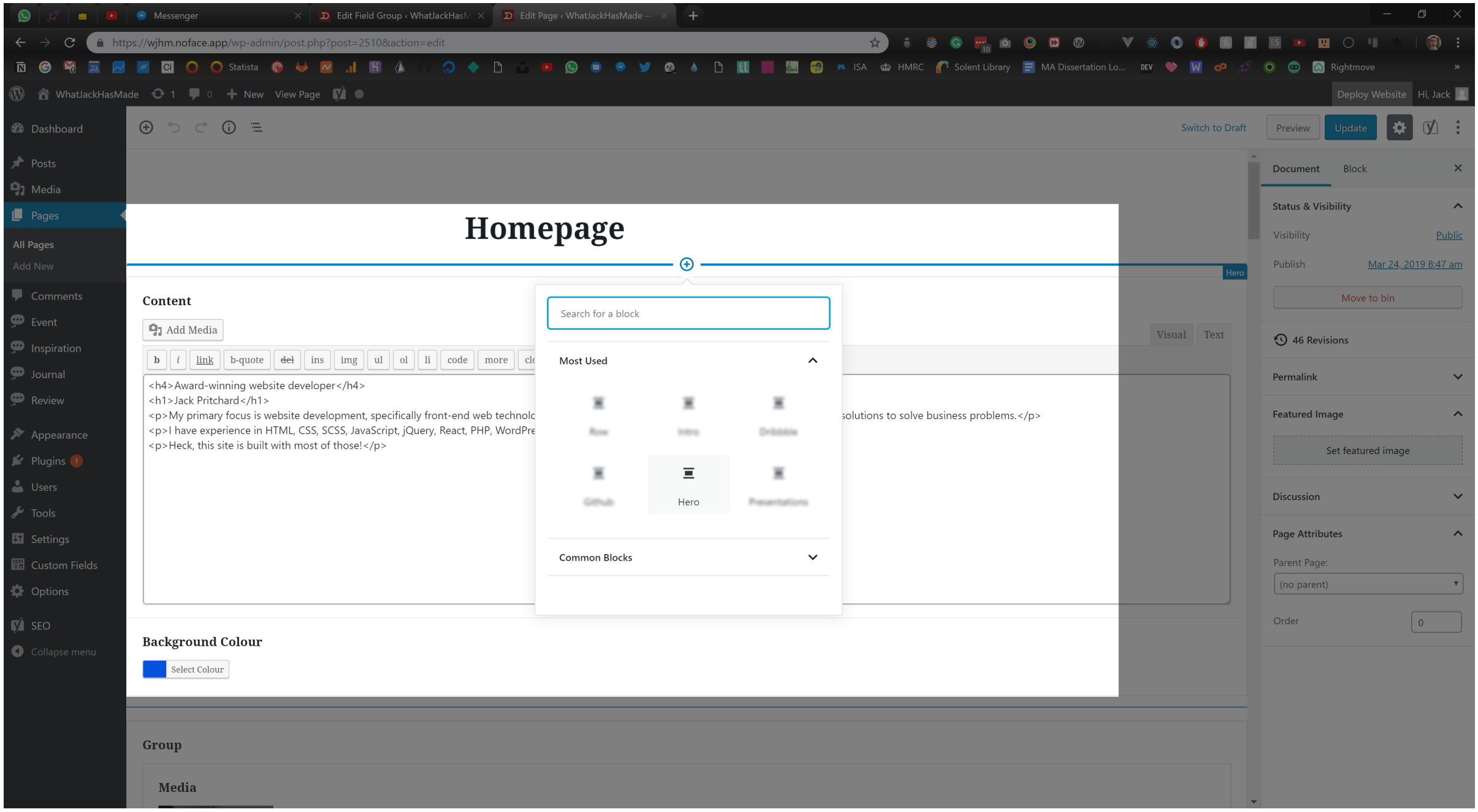
With continous integration, my new block will be registered when I deploy.

Then I simply sync the ACF changes on the live site to grab the fields it expects.


Edit Page ‹ WhatJackHasMade —
Field Groups ‹ WhatJackHasMade
https://wjhm.noface.app/wp-admin/edit.php?post_type=acf-field-group&post_status=sync
Statista
ISA
HMRC
Solent Library
MA Dissertation Lo...
Rightmove
WhatJackHasMade
1
0
New
Hi, Jack
Dashboard
Posts
Media
Pages
Comments
Inspiration
Journal
Review
Appearance
Plugins
Users
Tools
Settings
Custom Fields
Field Groups
Add New
Tools
Updates
Options
SEO
Collapse menu
Field Groups
Add New
Screen Options
Help
All (6) | Active (6) | Bin (1) | Sync available (2) | Search
Title
Fields
Block: Dribbble group_5c97a045ac502.json
2
Block: Presentations group_5c97a016211fa.json
2
Title
Fields
Bulk Actions
Apply
2 items
Advanced Custom Fields PRO
Customise WordPress with powerful, professional and intuitive fields.
Changelog
See what's new in version 5.8.0-beta3.
Resources
Website
Documentation
Support
Thank you for creating with ACF.
Thank you for creating with WordPress.
Version 5.1.1

The Hero Component in Gutenberg




Messenger
Edit Field Group ‹ WhatJackHasM
Edit Page ‹ WhatJackHasMade —
https://wjhm.noface.app/wp-admin/post.php?post=2510&action=edit
Statista
ISA
HMRC
Solent Library
MA Dissertation Lo...
Rightmove
WhatJackHasMade
1
0
New
View Page
Deploy Website
Hi, Jack
Dashboard
Posts
Media
Pages
All Pages
Add New
Comments
Event
Inspiration
Journal
Review
Appearance
Plugins
Users
Tools
Settings
Custom Fields
Options
SEO
Collapse menu
Switch to Draft
Preview
Update
Homepage
Content
Add Media
b
i
link
b-quote
del
ins
img
ul
ol
li
code
more
<h4>Award-winning website developer</h4>
<h1>Jack Pritchard</h1>
My primary focus is website development, specifically front-end web technolo
solutions to solve business problems.
I have experience in HTML, CSS, SCSS, JavaScript, jQuery, React, PHP, WordPre
Heck, this site is built with most of those!
Search for a block
Most Used
Row
Intro
Dribbble
Github
Hero
Presentations
Common Blocks
Hero
Visual
Text
Background Colour
Select Colour
Group
Media
Document
Block
Status & Visibility
Visibility
Public
Publish
Mar 24, 2019 8:47 am
Move to bin
46 Revisions
Permalink
Featured Image
Set featured image
Discussion
Page Attributes
Parent Page:
(no parent)
Order
0

Magic 🧝‍♀️

It's like magic

This is something to get excited about for any and all of your WordPress projects.

Even if you aren't interested in React, Gatsby or the front-end portion of this presentation.

So far so good

We've registered our custom block and it functions nicely in our backend.

The only issue is that the block doesn't solve the problem I have when it comes to pages.

So far so good?

The blocks are great at generating HTML in our endpoints, but I don't want that, I want to generate HTML and CSS in our React application.

I want the endpoints to act as a list of ingredients, not cook the meal.

Post JSON vs. Block JSON


```
"content":
        "<!-- wp:acf/hero {'id':'block_5c9791d20f725','data':{'field_5c97908ba2089'
        '<h4>Award-winning website developer</h4><h1>Jack Pritchard</h1><p>My prima
        focus is website development, specifically front-end web technologies and
        implementing the latest frameworks to create bespoke solutions to solve
        business problems.</p><p>I have experience in HTML, CSS, SCSS, JavaScript,
        jQuery, React, PHP, WordPress, Gatsby.</p><p>Heck, this site is built with
        most of those!</p>','field_5c979098a208a':'#0652DD','field_5c9bc20dc99d0':
        '2626'},'name':'acf/hero','align':'full','mode':'edit'} /-->"
```

Post JSON vs. Block JSON

Converting the content to JSON objects

I've created the following PHP functional files -

- get-acf-images.php
- get-acf-titles.php
- convert-the-content.php

Converting the Content to JSON objects

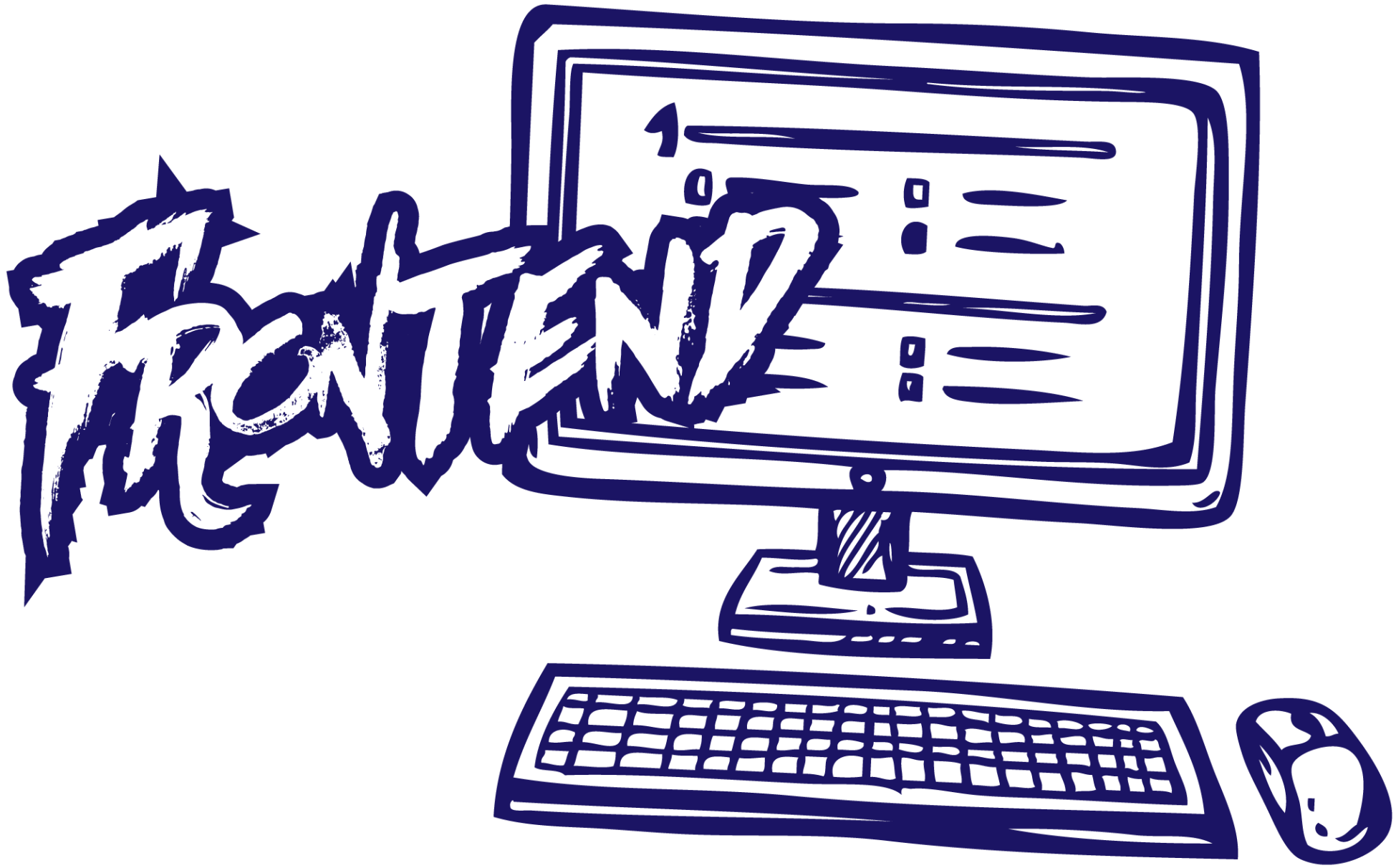
To convert the blocks into usable JSON, I rely on some custom functions and new WP functions

Our New Content

Backend Complete 😅

- We've got a WordPress setup ✔
- We've got a WordPress theme installed ✔
- We've got content on our site to use ✔
- We've got custom endpoints for us to use with Gatsby ✔

Time to Take 2 minutes for Questions or Heckling


FRONTEND

Frontend

Right let's get into the front-end of the website.

Gatsby - Recap

Just to recap, the front-end will be built with Gatsby which is completely dependant on React.

Gatsby's Blog Starter

Boot up the starter

Use the Gatsby CLI to create a new site, specifying the blog starter.

```
gatsby new my-blog-starter https://github.com/gatsbyjs/gatsby-starter-blog
```

Start developing

Navigate into your new site’s directory and start it up.

```
cd my-blog-starter/
gatsby develop
```

Open the source code and start editing

Your site is now running at `http://localhost:8000`

Note: You'll also see a second link:
`http://localhost:8000/___graphql`.

This is a tool you can use to experiment with querying your data. Learn more about using this tool in the Gatsby tutorial.

Gatsby's blog starter - Directory

```
.
├── node_modules
├── src
├── .gitignore
├── .prettierrc
├── gatsby-browser.js
├── gatsby-config.js
├── gatsby-node.js
├── gatsby-ssr.js
├── LICENSE
├── package-lock.json
├── package.json
└── README.md
```

Gatsby - Creating Pages

To create pages in Gatsby, you can either create a React component within the 'pages' directory and Gatsby will generate the component as a HTML file.

Similar to how you could create mypage.php and it would become accessible on your server.

Gatsby - Creating Pages

Alternatively we can generate pages from API sources (like WordPress endpoints 👀).

Gatsby Plugins

To gather the data, one of Gatsby's open source plugins can be plucked from their directory and installed via NPM and the gatsby-config.js file in our site root directory.

There are existing plugins for WordPress data, but most require a specific theme or setup which deviates from our setup.

So we'll create our own 😄

Creating our Gatsby WordPress plugin

```
.
├── plugins
│   └── gatsby-source-wordpress
│       ├── gatsby-node.js
│       ├── package.json
│       └── yarn.lock
├── node_modules
├── src
├── .gitignore
├── .prettierrc
├── gatsby-browser.js
├── gatsby-config.js
├── gatsby-node.js
├── gatsby-ssr.js
├── LICENSE
├── package-lock.json
├── package.json
└── README.md
```

Creating our Gatsby WordPress plugin

Loading the Plugin

```
module.exports = {
        plugins: [
                {
                        resolve: "gatsby-source-wordpress"
                }
        ]
};
```

gatsby-config.js

Firing up our Gatsby site

To run the site, you'll need to have terminal access to the Gatsby root folder and run `yarn run dev` or `npm run dev`.


npm
npm ERR! Failed at the whatjackhasmade@2.0.0 develop script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
npm ERR! A complete log of this run can be found in:
npm ERR! C:\Users\Jack\AppData\Roaming\npm-cache_logs\2019-03-24T14_22_00_997Z-debug.log
error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
Jack@DESKTOP-U0JRGI1 MINGW64 ~/Desktop/Don-t-call-it-a-comeback (master)
$ yarn run dev
yarn run v1.7.0
warning ..\..\package.json: No license field
$ npm run develop
> whatjackhasmade@2.0.0 develop C:\Users\Jack\Desktop\Don-t-call-it-a-comeback
> gatsby develop
success open and validate gatsby-configs — 0.009 s
success load plugins — 0.395 s
success onPreInit — 0.996 s
success initialize cache — 0.007 s
success copy gatsby files — 0.160 s
success onPreBootstrap — 0.014 s
success source and transform nodes — 1.841 s
success building schema — 0.712 s
success createPages — 0.003 s
success createPagesStatefully — 0.087 s
success onPreExtractQueries — 0.001 s
success update schema — 0.016 s
success extract queries from components — 0.237 s
success run graphql queries — 0.027 s — 2/2 94.97 queries/second
success write out page data — 0.014 s
success write out redirect data — 0.001 s
success onPostBootstrap — 0.014 s
info bootstrap finished - 9.682 s
DONE Compiled successfully in 4674ms
2:22:41 PM
You can now view whatjackhasmade in the browser.
http://localhost:8000/
View GraphiQL, an in-browser IDE, to explore your site's data and schema
http://localhost:8000/___graphql
Note that the development build is not optimized.
To create a production build, use npm run build
i 「wdm」:
i 「wdm」: Compiled successfully.

Access to data via GraphQL

For most developers, querying data in WordPress is often done in the templating language itself via WordPress functions like the_title(); or get_field();

Access to data via GraphQL

However, with Gatsby we have access to the data in a language called GraphQL.

GraphQL

To query the data Gatsby has access to, we can either pop open their friendly URL which for me is http://localhost:8000/___graphql

GraphQL

I prefer to use a tool called GraphQL Playground as it has a dark mode which helps with my vision!

GraphQL Playground

Application Workspace Edit Window

RECENT

Last opened 23.03.2019

New Workspace

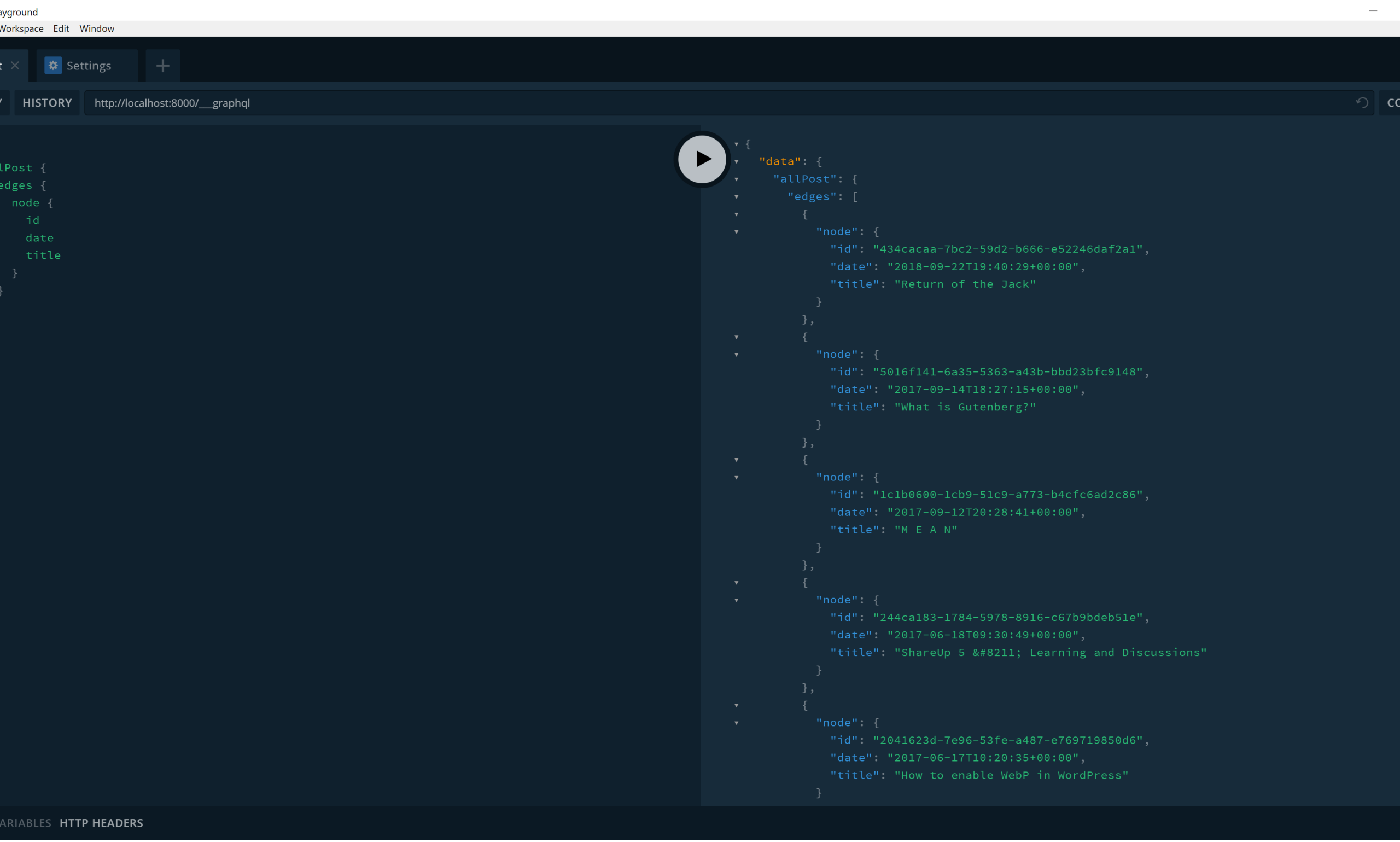
Either load a local repository with a .graphqlconfig file, or just open a HTTP endpoint

LOCAL URL ENDPOINT

http://localhost:8000/___graphql

OPEN

My First Gatsby GraphQL Query



Querying Our Nodes

```
{
        allPage {
                edges {
                        node {
                                id
                                slug
                                title
                        }
                }
        }
}
```

Pages With Queries

We now have 'nodes' that are accessible as the node type of 'Page'.

We'll be using these nodes to auto generate pages based on the JSON imported from our custom endpoints.

Generating Pages With Our Gatsby Plugin

```
exports.createPages = ({ graphql, actions }) => {
        const { createPage } = actions;
        return new Promise((resolve, reject) => {
                graphql(`
                        {
allPage {
        edges {
                node {
                        content {
                                id
                                align
                                data {
                                        background_colour
                                        content
                                }
                                mode
                                name
                        }
                        id
                        imageXS
                        imageSM
                        imageMD
                        imageLG
                        imageXL
                        imageFull
                        slug
```

Accessing the data via props

```
import React, { Component } from "react";
import Base from "./Base";
import ACFParser from "../particles/ACFParser";

export default class PageTemplate extends Component {
        render() {
const { content } = this.props.pageContext;

return (
        <Base>
                <ACFParser content={content} />
        </Base>
);
```

The page template first imports React and some components I've created to assist with generating the page

Sanitizing and rendering components with the data

The data itself is still simply that, data.

To process the data we need to write some functions which check what type of block is available and then render the correct React component based on that.

ACFParser block

The ACF Parser component first imports React and the hero component

ACFParser block

If the parser detects the name `acf/hero` which we have in our endpoint then it will import that component with all of the layout, styles and logic related to it.

Our Hero Component

To kick things off, we import React and Styled-Components

AWARD-WINNING WEBSITE DEVELOPER

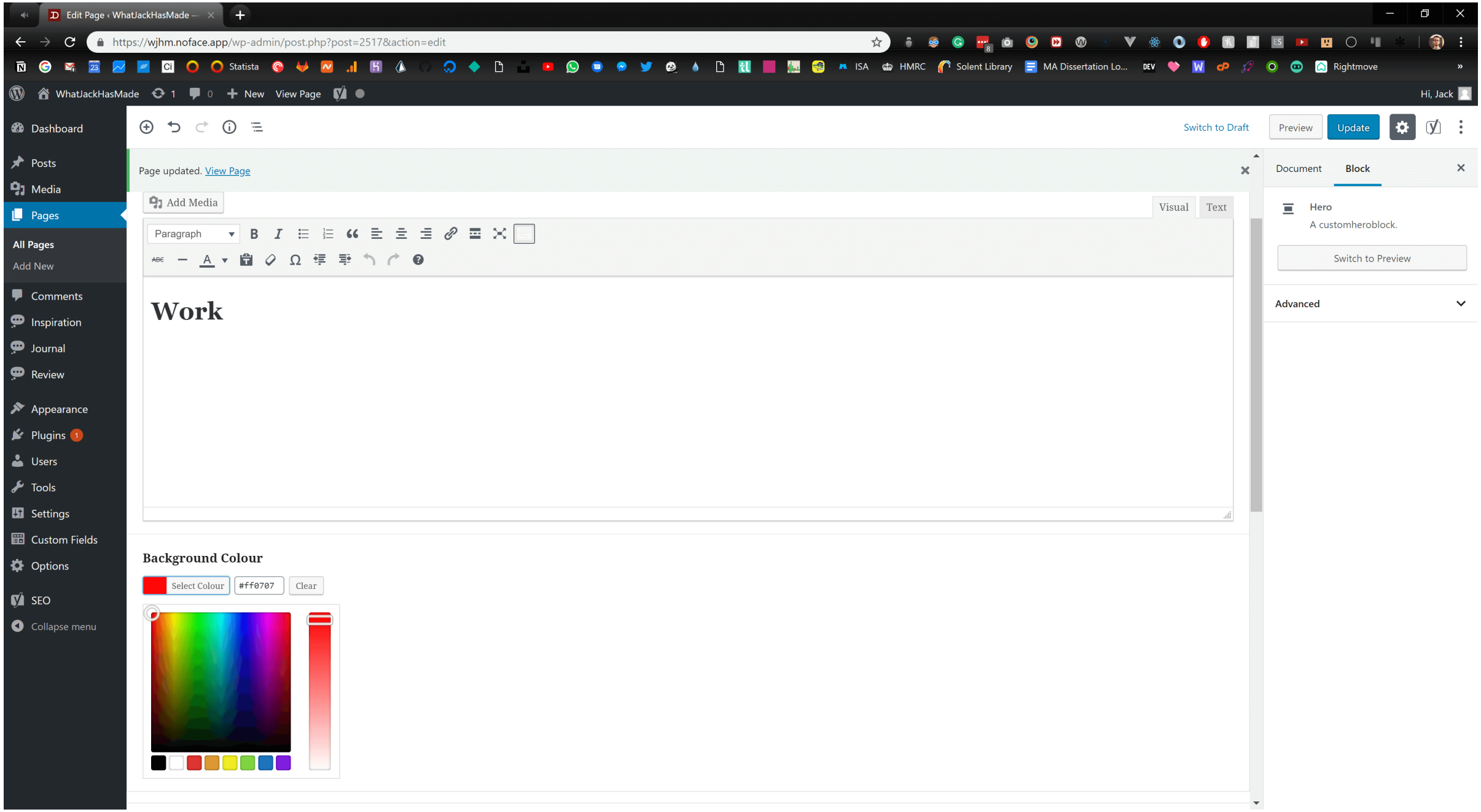
Jack Pritchard

Your website is the digital brochure and a potential storefront of your business. I create bespoke Fullstack websites from the front-end designs to the outstanding backend capabilities which become fully customisable for any ability of the user.

Beautiful websites are what I do best, coding in range of languages including HTML, CSS, SCSS, JavaScript, jQuery, PHP and working with on popular frameworks like React, WordPress and Gatsby I can create a space for you to grow your business and share your success.

Building our pages

All that is left now is to start creating content in WordPress to be digested by Gatsby.


Edit Page ‹ WhatJackHasMade
https://wjhm.noface.app/wp-admin/post.php?post=2517&action=edit
Statista
ISA
HMRC
Solent Library
MA Dissertation Lo...
Rightmove
WhatJackHasMade
1
0
New
View Page
Hi, Jack
Dashboard
Posts
Media
Pages
All Pages
Add New
Comments
Inspiration
Journal
Review
Appearance
Plugins
1
Users
Tools
Settings
Custom Fields
Options
SEO
Collapse menu
Switch to Draft
Preview
Update
Page updated. View Page
Add Media
Visual
Text
Paragraph
Work
Document
Block
Hero
A customheroblock.
Switch to Preview
Advanced
Background Colour
Select Colour
#ff0707
Clear

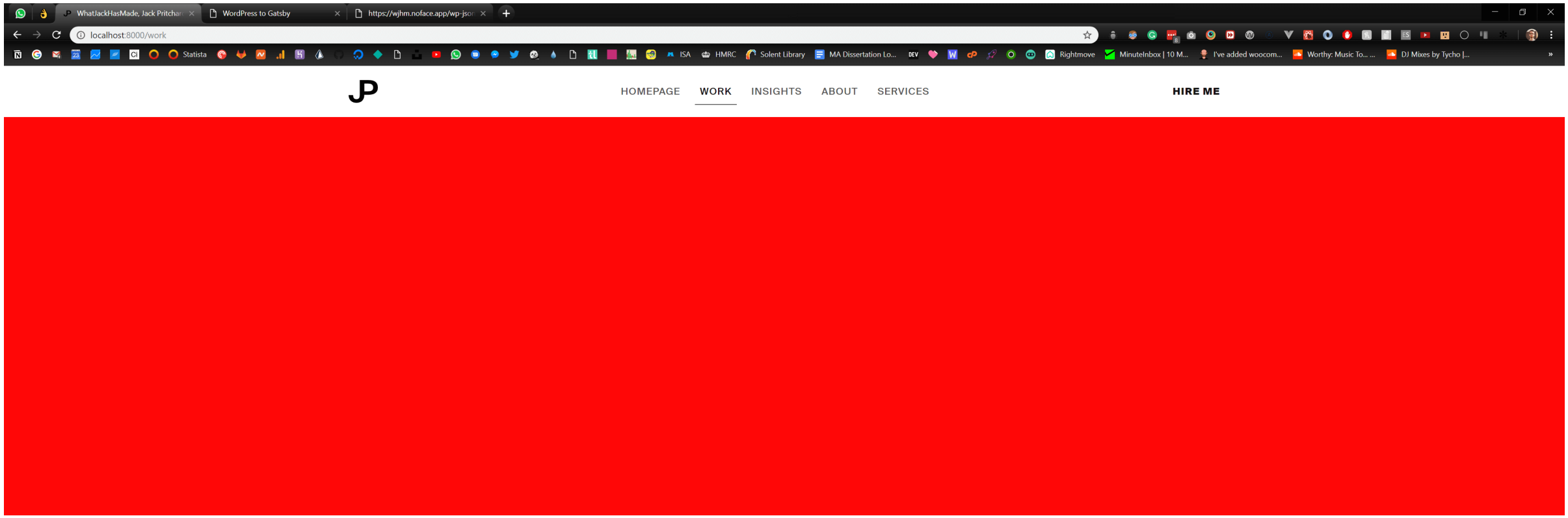
Things to be aware of

During development, Gatsby only has access to data imported on the time of `yarn run dev` or `npm run dev`.

If you create new pages or add content to WordPress, restart your local server.

Styled Components

The hero component is a great example of how it can be used effectively.



Always available for a chat

JACK PRITCHARD

07393 357520

jack@noface.co.uk

Creating Future Components

1. Register block - Functions
2. Register block - Twig (Optional)
3. Register block - ACF
4. Find and Replace - Functions
5. React Component

Hooking Netlify Up

So now we have a static site being generated. It's time to get this bad boy live for the world to see.

Hooking Netlify Up

https://app.netlify.com/ are a great hosting solution as again it's FREE and they continously deploy on repository changes, warn you of insecure code references (HTTP vs. HTTPS).

Hooking Netlify Up

Hello World

Generate on Content Changes

https://github.com/crgeary/wp-jamstack-deployments

JAMstack Deployments (Settings)

General

Webhook URL

https://api.netlify.com/build_hooks/5c9a23b3264687ae

Your Webhook URL. See Netlify docs.

Webhook Method

POST ▾

Set either GET or POST for the webhook request. Defaults to POST.

Post Types

- [x] Posts `post`
- [x] Pages `page`
- [ ] Media `attachment`
- [ ] Revisions `revision`
- [ ] Navigation Menu Items `nav_menu_item`
- [ ] Custom CSS `custom_css`

That's all folks

Thanks for letting me ramble, there may be a few points I had to trim out for the sanity of your ears, so if you have questions you want to ask, then please do and I'll try my best to answer!

Contact Information

- jack@noface.co.uk
- whatjackhasmade on all socials
- whatjackhasmade.co.uk
- noface.co.uk
- NoFace