About Me

Name: Jonathan Brown

Job: Full Stack Web Developer

Experience: Almost 3 years of experience in WordPress development

Getting started

With the REST API and Angular JS

Folder Structure

How to setup the application structure

The application will consist of two folders.

1. The actual angular folder
2. The WordPress folder

WordPress Setup

The WordPress setup will be a fresh install of WP upgraded to 4.9

- No themes are necessary
- Plugins required will be acf pro, acf to rest, and CPT manager.
- You don’t really need CPT manager if you want to register them with code, dealers choice.

Angular Setup

Creating the folder structure

- Index.html <- root file
- App <- all the code
- Assets <- sass/css, images, js, vendor stuff

App

- Shared
  - Sidebar
  - Global Services
- Components
  - Articles
  - Home
  - Movies
  - Shows

Common files in folder

Most folders will have the same structure in them (view, controller, service, possible directives).

Ex.

- Components
  - Home
    - homeController.js
    - homeService.js
    - homeView.html

Creating your main app file

Declaring angular

Angular has to be defined to a variable. When you tell a variable that its an angular module, you have to give it a name, and tell the module what dependencies it will have access to. The dependencies are an array of string names that correlate to a particular dependency.

Creating the index file

Creating the index.html file

This file will contain your app definition as well as load your views

Simply declare ng-app in the html block, then define ng-view in the main content block. Include any angular js files you need in the footer. Think of the index in the scope of this conversation as both the header and footer. The only thing that will be swapped out is the main content. Try to make sure you download all the angularjs files you need and don't hardcode directly to the cdn links.

Initializing angular in your markup

You will need to add ng-app=”yourappname” to your html tag. Yourappname is the name you give to your angular module when you declare angular.module();

Route(s) 66

Routes are basically just definitions that tell your app how to handle certain urls. Handling usually involves what templateUrl(view) and what controller to use. Friendly warning, when writing urls in angular, make sure that you use #! <- hashbang. If you try to do relative urls, it’s not going to work.

Controllers - Dat power

What are controllers?

Controllers are the heavy lifters between the model and the view.

Since Angular is a MV* framework and this presentation is about the REST API, we can think of controllers as the getters, setters, and senders of data.

Controllers

- Handle requesting data from the REST API
- Modifying the data it receives from the API
- Passing that data to the view

What’s $scope

$scope is a bag. That bag is directly related to the controller that the scope variable is in. If you define a scope variable in one controller, another will not have that variable.

Dependency injection

Dependency injection is one of the main building blocks of angularjs.

Angular has a bunch of different services that they've built out that you can use. But you can't use them by default. You have to tell your app that you are loading in that dependency, and then tell your controller that they have access to it. The good news, is that you do it by default with the $scope variable, so it's nothing new.

What’s a service?

Services are basically just collections of code that perform a specific task. If you have a bunch of code in a controller that is reusable, take it out and drop it into a service so that it is more DRY. Then inject it into the controller like you would other Angular services.

It’s always nice to have a View

What are views?

Views represent the “view” of the application. By that I mean that they are essentially the markup that users see

- Views can be as big or small as you need them to be.
- Views are powered by their respective controllers
- Views are the markup that the users see and interact with

Accessing scope variables in a view

So long as the view is tied to a controller (it should be), you can access that controller's $scope variable. You access that $scope variable by calling just the scope attribute name. If you have $scope.message in your controller, in the view you would just use {{message}}, scope is implied.

Looping over an object in $scope

You can use ng-repeat on an element to duplicate that element and insert more data into it, much like a foreach loop in php. Let's say you contact your api and get a collection of posts back, and you store that in $scope.posts. You can use ng-repeat="post in posts" to loop over each post in the posts array.

Ng-if and more.

Angular has a whole host of other useful directives such as ng-if, ng-repeat, ng-app, ng-init.

Filters

Angular filters make life easy

Angular comes with a few helpful filters out of the box some of the common ones include

- Currency
- Date
- limitTo
- Uppercase
- lowercase

To call a filter you use {{post.publish_date | date}}. This will use the prebuilt angular date filter. You can pass in formatting options if you want/need to.

Getting Your REST (API)

Contacting the WordPress REST API

So long as you have a new install of WordPress you can access the rest api out of the box

You can see some routes from the rest api by going to http://yourdomain.com/wp-json. That will give you a list of api endpoints currently available by your WordPress installation.

Contacting the WordPress REST API - All about those types

Getting post types is super simple.

That route is http://yourdomain.com/wp-json/wp/v2/types.

To get types you simply add /types to the end of the rest api url.

If we run that now we will get a list of all post types currently registered by your WordPress install.

Get all posts of a specific type

Getting all of the posts of a post type is simple.

You simply add the /:post_type to the rest url where :post_type is the type you're trying to get. Once you ping that url you will get all posts that are published that are in that post type.

Contacting the WordPress REST API - Get a post

Getting a post is as simple as getting posts.

You simply add the /:post_type/:post_id to the rest url (:post_id is the id of the post you want) and WordPress will feed you all of the information available for the post. If you have the acf to rest plugin you will get all of the acf fields you created with it.

ACF to Rest

ACF to rest makes things incredibly easy to work with in the REST API. By default WordPress doesn't give you meta data. With acf to rest, all of your meta data you're getting is stored in an acf key in the post you request.

Outputting your $scope data from a post or posts

Multiple posts

You will need to loop over your posts and output the field or post data attribute you want by calling it from the post itself. For instance ng-repeat=”post in posts” {{post.id}}

Single post in scope

To access data in a single post scope, just call post.id or post.acf.field_name

Let’s look at some code

Questions?

Helpful Links

1. https://angularjs.org/
2. https://scotch.io/courses/getting-started-with-angularjs-1x
3. https://code.tutsplus.com/tutorials/creating-single-page-applications-with-wordpress-and-angularjs--cms-25095
4. https://github.com/xxjonfenxx/wcjax-2018-angular-wordpress