Build Your Own Web App With The WordPress REST API

WP + Javascript = Amazing

What Is A SPA?

- A single page web app is a website that is built inside of a single HTML document.
- A javascript library or framework is used to provide functionality.
- Content is typically loaded via an API into the page.
- Users may navigate to additional URLs on the site, the document itself does not change. The framework simply updates the DOM to reflect the changing view and content.

We’re Using Vue.js

- Easy to learn even if you don’t have experience with other libraries.
- Good documentation.
- Strong community involvement and lots of auxiliary projects.
- Small file size.
- Speedy.

Development Environment

- Install Node on your computer if you don’t have it already. https://nodejs.org/en/download/
- Install the Vue CLI
- VS Code is very nice for working with Vue and other frameworks. https://code.visualstudio.com/
- The starter project for this talks is available at https://github.com/billrobbins/WPVue

Vue Is Similar To WordPress

```
<main>
    <?php while ( have_posts() ) : the_post(); ?>
        <?php the_pos_thumbnail(); ?>
        <?php the_title( '<h1>', '</h1>' ); ?>
        <?php the_content(); ?>
    <?php endwhile; ?>
</main>
```

```
<main class="main-content" v-if="post && post != ''">
   <img :src="post.featured_image_url" />
   <h1 v-html="post.title.rendered"></h1>
   <div v-html="post.content.rendered"></div>
</main>
```

Conditionals

These allow us to display items based on variables.

```
<?php if ( $some_variable !='' ) { ?>
    <main>
        <p><?php echo $some_variable; ?></p>
    </main>
<?php } ?>
```

```
  <main v-if="some.value">
    <p>{{some.value}}</p>
  </main>
```

Loops

Create a group of items from a data set.

```
<article v-for="post of posts":key="post.id">

  <router-link :to="{ name: 'Post', params: { id: post.slug }}">
    <img :src="post.featured_image_url" />
  </router-link>

  <h2>
    <router-link :to="{ name: 'Post', params: { id: post.slug }}">
      {{post.title.rendered}}
    </router-link>
  </h2>
  <div v-html="post.excerpt.rendered"></div>

</article>
```

Vue.js From the Ground Up

- The index.html file loads first.
- Inside it, a special div loads up the app.
- The root of the app, is a file called App.vue. All of the components inside the app begin in this file.
- Each component can have a template as well as functionality. The template is placed at the top of the component file and the functionality in the script section below that.
- The components are loaded into the App.vue file via a special tag called the Router View.

Routing

- The browser URL tells the router which component to load.
- The component inserts its content in the router view tag inside of App.vue.
- Similar to WordPress’ template hierarchy.

```
routes: [
  {
    path: '/blog/page/:id',
    name: 'Blog',
    component: Archive
  },
  {
    path: '/post/:id',
    name: 'Post',
    component: Post
  },
  {
    path: '/:id',
    name: 'Page',
    component: Page
  }
]
```

How Do We *Get* The Content?

- Use a library called Axios to help us make HTTP calls

```
import axios from 'axios';

export const HTTP = axios.create({
  baseURL: `http://yoursite.com/wp-json/`
})
```

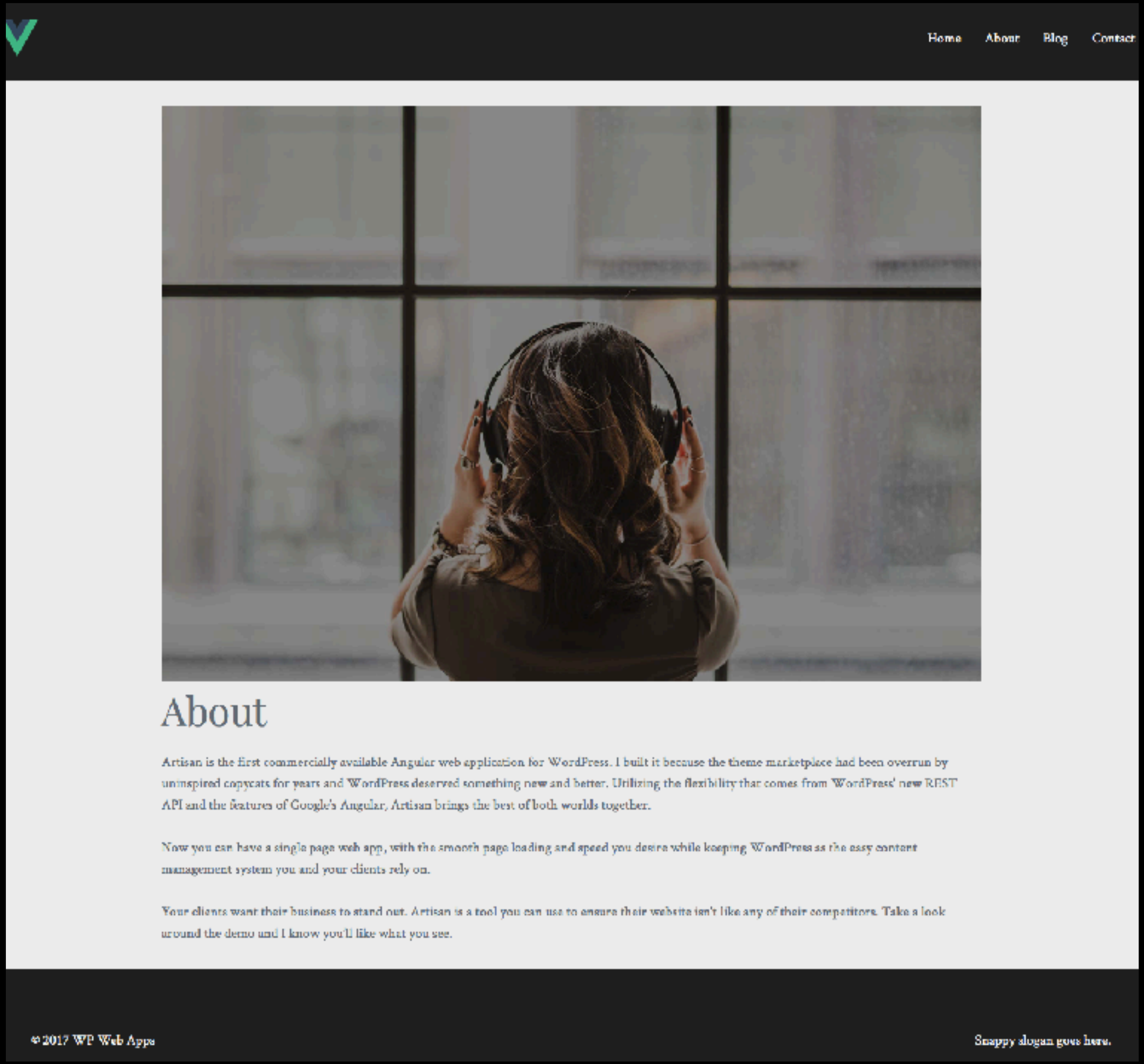
- This environment is used as a base so we can set the URL to our source in one place.

- Each time after that, we can use HTTP as a constant to reference all of this.

```
fetchData() {
  HTTP.get('wp/v2/pages?slug='+this.$route.params.id)

    .then((resp) => {
      this.page = resp.data[0]
    })
}
```

- This works with the router to pull the content that corresponds to the URL in the browser.
- All of the data returned gets assigned to the page variable. For us to use in our template.

```
<template>
  <main
   class="main-content"
   v-if="page && page != ''"
   :key="page.id">

   <img :src="page.featured_image_url" />
   <h1>{{page.title.rendered}}</h1>
   <div v-html="page.content.rendered"></div>

  </main>
</template>
```



About

Artisan is the first commercially available Angular web application for WordPress. I built it because the theme marketplace had been overrun by uninspired copycats for years and WordPress deserved something new and better. Utilizing the flexibility that comes from WordPress' new REST API and the features of Google's Angular, Artisan brings the best of both worlds together.

Now you can have a single page web app, with the smooth page loading and speed you desire while keeping WordPress as the easy content management system you and your clients rely on.

Your clients want their business to stand out. Artisan is a tool you can use to ensure their website isn't like any of their competitors. Take a look around the demo and I know you'll like what you see.

© 2017 WP Web Apps

Snappy slogan goes here.

- We check to see if the API is loaded first.
- We then pull the content from the response and assign it to the spots in our template.
- The browser replaces the content in the Router View with the new content.

Go Make Something Amazing

- Starter project on Github for you at https://github.com/billrobbins/WPVue
- Includes Page, Post and Archive components, routing and navigation menus.
- Has simple fade transition between views.
- What to add?
  - Comments
  - Dynamic titles and descriptions.

Bill Robbins

WPWebApps.com

@billrobbins