Introduction to Nuxt.js

About me

- Hilary Cheng (hilarycheng@gmail.com)
- Full Stack Developer
- Develop Machine Related Application (IoT) with Server and Frontend Management Console, Embedded Android
- Interested in Elixir / Flutter
- Software Architect in Hungwai Innovation Ltd.

Problem of Nowaday Javascript Framework

- Single Page Application (SPA)
- Embedded lots of Javascript inside HTML
- Data download from Server by API
- Rendering perform on Client Side
- Lots of Search Engine cannot crawl your Web Site Content

What is Nuxt.js

- Universal Vue Framework like vue-cli
- Render HTML
  - At the Server Side by Refresh/First Visit
  - At Client Side when events occurred
- Web site can even works without Javascript
- Got Community Plugins to customize your needs

Use Case

- Content / News Website
- Migration from Old Website (Wordpress / Drupal) with SPA Style
- Needs to maintain existing search engine result

Let’s do it

Create Nuxt Project

```
vue init nuxt-community/starter-template nuxtdemo

cd nuxtdemo

npm install

npm run dev
```

Directory Structure

- layouts
    - Common Page Styles/Layout
- pages
    - Your Application
- plugins
    - like Vueitfy / Local Storage / Axios
- static
    - jpg/png/favicon.ico
- middleware
    - authentication

```
npm install @nuxtjs/axios @nuxtjs/bulma --save
```

```
vim nuxt.config.js
```

Append the following text

```
  modules: [
     '@nuxtjs/axios',
     '@nuxtjs/bulma',
     '@nuxtjs/font-awesome'
  ]
```

Create the First index.vue

Use the JSON API from https://jsonplaceholder.typicode.com/

Edit pages/index.vue

Add Async Data

```
async asyncData({ app }) {
    const posts = await app.$axios.$get('https://jsonplaceholder.typicode.com/posts');
    console.log('Loading Posts');
    return { posts };
},
```

Modify index.vue template

```
<section class="section">
    <div class="container" >
     <div class="rows">
       <h1 class="title">Posts</h1>
         <div class="columns is-multiline">
            <div v-for="post in posts" :key="post.id" class="column is-one-quarter">
             <div class="card">
               <div class="card-header">
                 <div class="card-header-title">
                          {{ post.title }}
                 </div>
               </div>
               <div class="card-content">
                 {{ post.body }}
               </div>
             </div>
            </div>
           </div>
         </div>
     </div>
</section>
```

Add Events onClick to Card

```
<div class="card" v-on:click="postClick(post.id, $event)">

methods: {
    postClick: function(postid, event) {
      event.stopPropagation();
      this.$router.push('/post/' + postid);
    }
}
```

Create Route /post/:id

```
mkdir -p pages/post
touch pages/_id.vue

<template>
   <div>posts</div>
</template>
<script>
export default {
       async asyncData({ app, params }) {
         const comments = await app.$axios.$get('https://jsonplaceholder.typicode.com/posts/' +
params.id + '/comments');

         console.log('Loading Comments ' + params.id);

         return { comments };
       },
}
</script>
```

Add Template to post/_id.vue

```
<section class="section">
      <div class="container" >
        <div class="rows">
          <h1 class="title">Comments</h1>
          <div v-for="comment in comments" :key="comment.id" class="row is-one-quarter">
            <div class="columns">
             <div class="column">
               <div class="card">
                 <div class="card-header">
                   <div class="card-header-title">
                     {{ comment.name }} {{ comment.email }}
                   </div>
                 </div>
                 <div class="card-content">
                   {{ comment.body }}
                 </div>
               </div>
             </div>
           </div>
         </div>
        </div>
      </div>
</section>
```

Add Back Button to return to previous page

```
<button class="primary" v-on:click="goBack">Back</button>

methods: {
       goBack: function(event) {
         event.stopPropagation()
         this.$router.go(-1);
       }
}
```

Reminder

- Be careful of using variable at browser in asyncData
- using “process.client” to determine Client Side / Server Side
- Default router is not flexible
- Use @nuxtjs/router
- Use @nuxtjs/proxy for Local Development

Thanks