For this blog I am going to talk about how to create a simple blog page with vue client, After install node.js, we can create vue client in webpack-simple by typing the following command line in the console:

$ vue init wekpack-simple my-first-blog

In this way we can initialize a new vue project by creating a project folder with name 'my-first-blog', then we can go to the folder by using 'cd my-first-blog' command line. Inside the folder, we can type command line:

$ npm install

$ npm run dev

Now, we have created our template for our project! There are a lot of files that are created. In main.js, this is the file where the whole project starts, and inside of this document we've included a file named App.vue. In App file which was created we can do all what we want. Such as include more components which we want to import to the main file.

To build a new component, we might want to firstly create a new vue file and then register the file to the App.vue file, so we can import and use it. So, how do we do that?

Let's take a look of a simple structure of a vue file. It consists of parts: template, script, and style. In the template, this is where you want your content goes. In script, you have export area, this is where you want to make your data and methods, as well as to register components.

After we've created a new file in a folder called components, how could we import it to App.vue? Within the script, we can write 'import addBlog from './components/addBlog.vue', and then we can register a new vue addBlog to our app.vue by placing

components: {

'add-blog’: addBlog

}

to our export default part. Finally, we will put some content in addBlog.vue. Template in blogs holds the stuff of which we want to show in the app template, late we can call this template in side app template by putting tag of what we initialized as <addBlog></addBlog>

For example, we wish the users fill some information in this page, so they should write some kind of inputs in a form tag. Let’s say we have <input type=”text” placeholder=” add title”>, but this is not interactive with the file. How could we get around of this?

In vue.js there is a thing called v-model which is similar to ng-model in angular (two-way binding), but v-model is a single way binding. The way to write is like the following:

1. <input v-model= “sth”/>
2. <input v-bind: value= “sth” v-on:input=”sth = $event.target.value”/>

Or 3. <input: value=” sth @input=” sth=$event.target.value”/> (abbreviated)

To understand the second line of code, we should know that element input has an event called oninput. This is included in HTML5. Each time when the text area receives some changes it will automatically toggle oninput, and updated sth with a new value.

So, when we add v-model property to <input/>, the default will take value as an element property, and regard input tag as a trigger to pass value. Now, sth is bind to v-model, and whenever the users produce any kind of input, ’sth’ get updated. We can name ‘sth’ as blog.title, blog.content for our case and place them in data(){} of addblog file.

On the other hand, we can also add modifiers to v-model such as lazy,

v-model.lazy: this will delay the input for what the property received.