## 搭建过程：

参考：<https://segmentfault.com/a/1190000008049815>

**问题1:**

搭建过程中会出现，安装不了chromerdriver的问题，手动安装

：

npm install chromedriver --chromedriver\_cdnurl= <http://npm.taobao.org/mirrors/chromedriver>

参考：<https://segmentfault.com/a/1190000013289136?utm_source=channel-hottest>

**问题2：**

使用cnpm 安装依赖包的过程中，出现下面错误：

TypeError: this is not a typed array.

at Function.from (native)

at Object.hashesMatch (index.js:29:18

at Context.<anonymous> (test/test.js:25:22)

node.js的版本使用最新的版本，重装前一定要卸载干净上一个版本的。

参考：<https://stackoverflow.com/questions/36899888/getting-typeerror-this-is-not-a-typed-array-using-buffer-from-in-mocha>

## 学习思路：

1. 先看vue.js基础

参考：<https://www.runoob.com/vue2/vue-install.html>

1. vuex

一个全局管理的变量和对象的框架。

参考：<https://segmentfault.com/a/1190000015782272>

1. ElementUI

一些可以直接使用UI组件

<https://element.eleme.cn/#/zh-CN>

1. vue组件的生命周期（重要）

<https://segmentfault.com/a/1190000008570622>

1. Router相关疑惑

<http://www.cnblogs.com/zhansu/p/6252570.html>

# 知识点：

1.一个 vue 对象绑定一个组件

2.v-if条件语句或者v-model双向绑定

3.v-for

4.computed类似method，但是有缓存

5.v-bind:class="{ active: isActive }"

6. v-on:click= 事件修饰符， 按键修饰符

7.v-model 双向绑定表单，vue会选择合适的方式去同步数据

8.组件

组件具有基础组件监听，绑定的能力，也可以自定义方法，发送事件等能力

<div id="app">

<div id="counter-event-example">

<p>{{ total }}</p>

<button-counter v-on:increment="incrementTotal"></button-counter>

<button-counter v-on:increment="incrementTotal"></button-counter>

</div>

</div>

<script>

Vue.component('button-counter', {

template: '<button v-on:click="incrementHandler">{{ counter }}</button>',

data: function () {

return {

counter: 0

}

},

methods: {

incrementHandler: function () {

this.counter += 1

this.$emit('increment')

}

},

})

new Vue({

el: '#counter-event-example',

data: {

total: 0

},

methods: {

incrementTotal: function () {

this.total += 1

}

}

})

</script>

9.自定义组件

<div id="app">

<p>页面载入时，input 元素自动获取焦点：</p>

<input v-focus>

</div>

<script>

// 注册一个全局自定义指令 v-focus

Vue.directive('focus', {

// 当绑定元素插入到 DOM 中。

inserted: function (el) {

// 聚焦元素

el.focus()

}

})

// 创建根实例

<div id="app">

<p>页面载入时，input 元素自动获取焦点：</p>

<input v-focus>

</div>

10.路由

创建路由，router-link:to ,router -view(路由到的组件渲染的位置)

11.动画可以使用css,

<transition name = "fade">

<p v-show = "show" v-bind:style = "styleobj">动画实例</p>

</transition>

12.混入（继承）

同java类似，方法重写也一样

var Component = Vue.extend({

mixins: [myMixin]

})

13.ajax，链式编程

// 基于全局Vue对象使用http

Vue.http.get('/someUrl', [options]).then(successCallback, errorCallback);

Vue.http.post('/someUrl', [body], [options]).then(successCallback, errorCallback);

// 在一个Vue实例内使用$http

this.$http.get('/someUrl', [options]).then(successCallback, errorCallback);

this.$http.post('/someUrl', [body], [options]).then(successCallback, errorCallback);

14.监听及全局set方法

vm.$watch('counter', function(nval, oval) {

alert('计数器值的变化 :' + oval + ' 变为 ' + nval + '!');

});

Vue.set( target, key, value )