1 基于 Spring Cloud Alibaba 的外卖订餐系统

1、需求分析

2、设计数据库

3、技术选型

前端：Vue + Element UI + Mint UI + MUI

买家端（移动端）： Vue + Mint UI + MUI

卖家端（PC 端）： Vue + Element UI

后端：

低配版 Spring Cloud Alibaba + MyBatis Plus + MySQL

高配版 Spring Cloud Alibaba + Rocket MQ + MyBatis Plus + MySQL + Redis + Elastic Search

2 前端开发

前端原型（不需要对接后台，使用假数据完成的页面）

> 买家端

1、创建 Vue 工程

2、添加 axios 插件

3、安装 Mint UI

```
cnpm install mint-ui -S
```

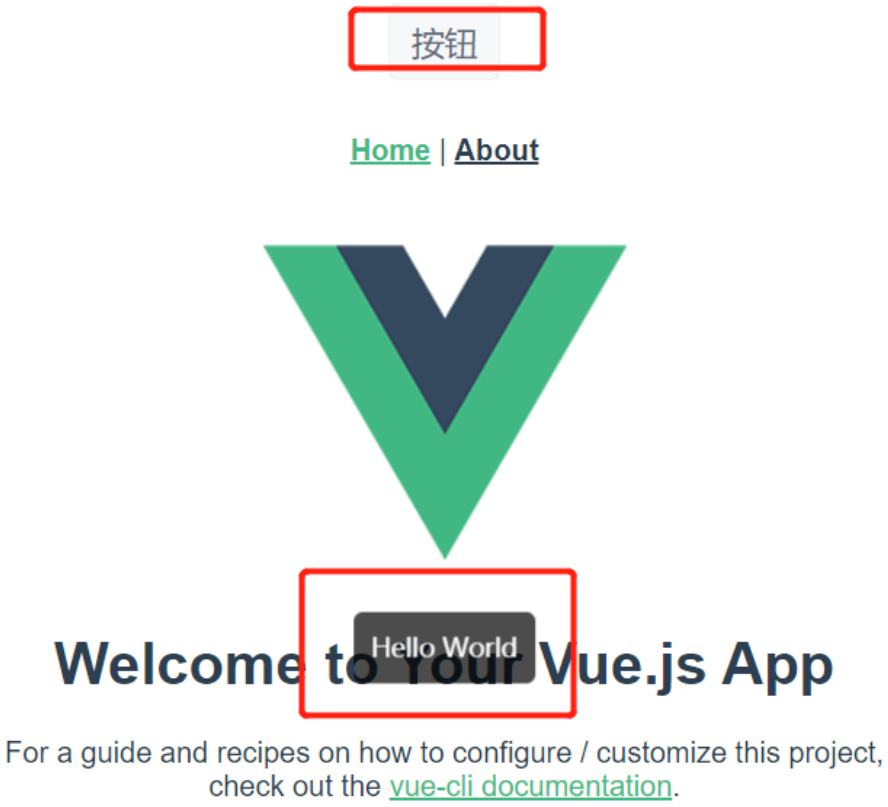
4、main.js 中引入 Mint UI 组件

```
import Vue from 'vue'
import './plugins/axios'
import App from './App.vue'
import router from './router'
import store from './store'
import Mint from 'mint-ui'
import 'mint-ui/lib/style.css'

Vue.config.productionTip = false

Vue.use(Mint)

new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

5、测试，App.vue 使用 Mint UI 组件

```
<template>
  <div id="app">
    <mt-button @click.native="test">按钮</mt-button>
    <div id="nav">
      <router-link to="/">Home</router-link> |
```

```
      <router-link to="/about">About</router-link>
    </div>
    <router-view/>
  </div>
</template>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}

#nav {
  padding: 30px;
}

#nav a {
  font-weight: bold;
  color: #2c3e50;
}

#nav a.router-link-exact-active {
  color: #42b983;
}
</style>

<script>
  export default {
    methods: {
      test:function () {
        this.$toast('Hello World');
      }
    }
  }
</script>
```



安装成功

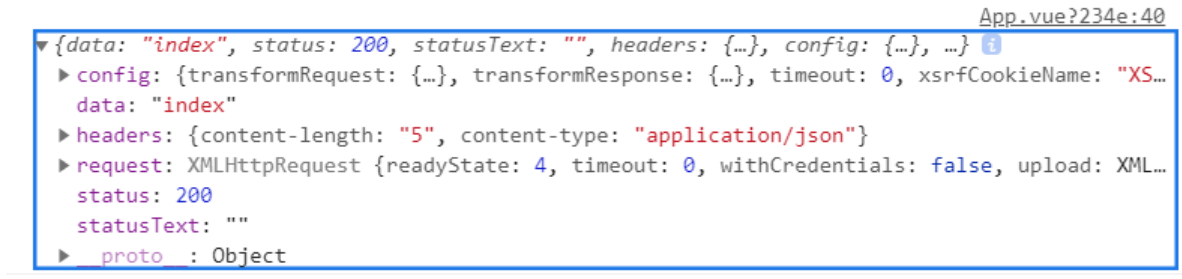
6、测试 axios

```
test:function () {
  axios.get('http://localhost:8181/index').then(function
(resp) {
      console.log(resp)
  })
}
```

```
@GetMapping("/index")
public String index(){
    return "index";
}
```

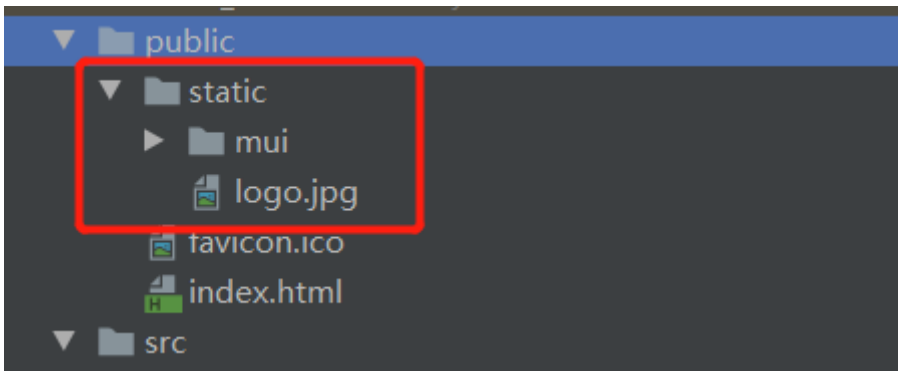
```
package com.southwind.configuration;

import
org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.CorsRegi
stry;
```

```
import
org.springframework.web.servlet.config.annotation.WebMvcCo
nfigurer;

@Configuration
public class CrosConfiguration implements WebMvcConfigurer
{

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
                .allowedOrigins("*")
                .allowedMethods("GET", "HEAD", "POST",
"PUT", "DELETE", "OPTIONS")
                .allowCredentials(true)
                .maxAge(3600)
                .allowedHeaders("*");
    }
}
```



测试成功

7、引入 mui，导入静态文件，注意位置 public/static/



8、main.js 引入相关依赖

```
import Vue from 'vue'
```

```
import './plugins/axios'
import App from './App.vue'
import router from './router'
import store from './store'
import Mint from 'mint-ui'
import 'mint-ui/lib/style.css'
import '../public/static/mui/css/mui.min.css'

Vue.config.productionTip = false

Vue.use(Mint)

new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

9、测试，App.vue 使用 mui 组件

```
<template>
  <div id="app">
    <mt-button @click.native="test">按钮</mt-button>
    <button type="button" class="mui-btn mui-btn-danger">
红色</button>
    <div id="nav">
      <router-link to="/">Home</router-link> |
      <router-link to="/about">About</router-link>
    </div>
    <router-view/>
  </div>
</template>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}
```

```
#nav {
  padding: 30px;
}

#nav a {
  font-weight: bold;
  color: #2c3e50;
}

#nav a.router-link-exact-active {
  color: #42b983;
}
</style>

<script>
  export default {
    methods: {
      test:function () {

 axios.get('http://localhost:8181/index').then(function
(resp) {
             console.log(resp)
        })
      }
    }
  }
</script>
```

添加成功，takeout_buyer 基于移动端的买家前端工程 Vue + MinUI + mui + axios 环境搭建成功。

接下来完成前端页面原型的开发，**使用假数据把用户交互界面做出来。**