参考网址 ： <https://www.cnblogs.com/jasonwang2y60/p/7302449.html>

# [vue+mockjs 模拟数据，实现前后端分离开发](https://www.cnblogs.com/jasonwang2y60/p/7302449.html)

在项目中尝试了mockjs，mock数据，实现前后端分离开发。

关于mockjs，官网描述的是

1.前后端分离

2.不需要修改既有代码，就可以拦截 Ajax 请求，返回模拟的响应数据。

3.数据类型丰富

4.通过随机数据，模拟各种场景。

等等优点。

总结起来就是在后端接口没有开发完成之前，前端可以用已有的接口文档，在真实的请求上拦截ajax，并根据mockjs的mock数据的规则，模拟真实接口返回的数据，并将随机的模拟数据返回参与相应的数据交互处理，这样真正实现了前后台的分离开发。

与以往的自己模拟的假数据不同，mockjs可以带给我们的是：在后台接口未开发完成之前模拟数据，并返回，完成前台的交互；在后台数据完成之后，你所做的只是去掉mockjs：停止拦截真实的ajax，仅此而已。

下面一步步的来实现vue-cli创建项目并添加一条新闻类的数据模拟接口：

1.安装vue-cli全局脚手架

|  |  |
| --- | --- |
| 1 | npm install --global vue-cli |

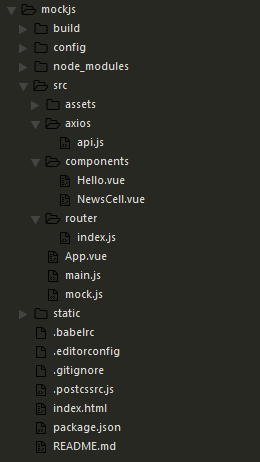
2.创建vue项目

|  |  |
| --- | --- |
| 1 | vue init webpack mockjs<br>cd mockjs<br>npm install axios --save |

3.安装mockjs

|  |  |
| --- | --- |
| 1 | npm install mockjs --save-dev |

4.项目目录



axios/api    用来封装axios

Hello.vue     页面首页

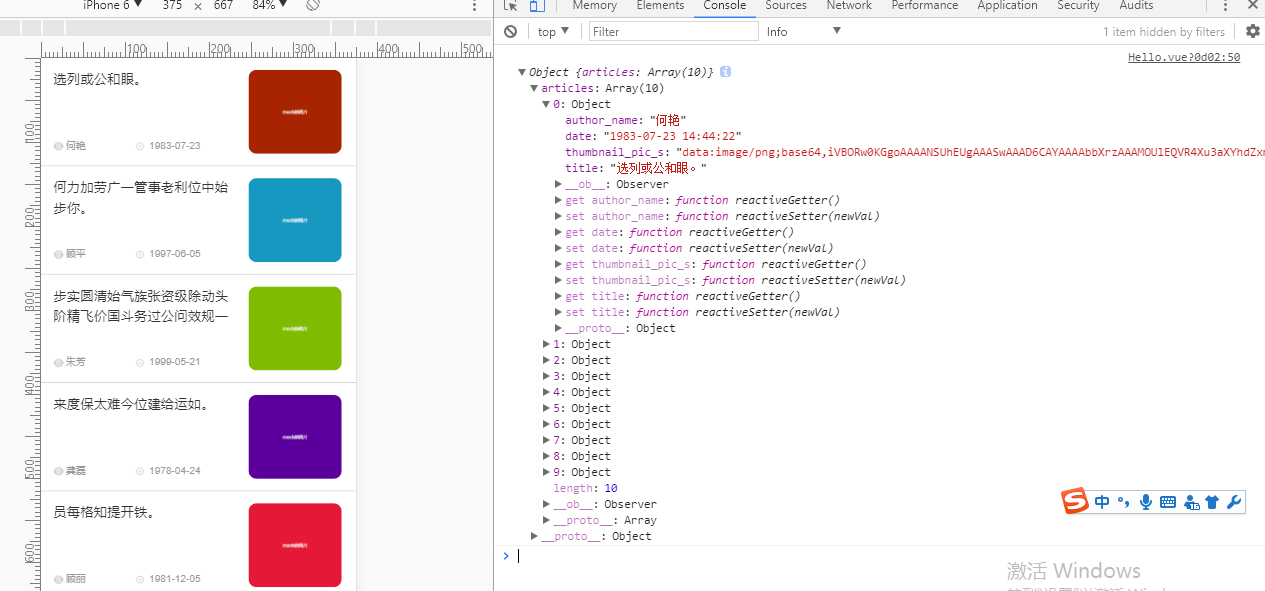
NeswCell.vue   新闻组件

router/index.js   路由

main.js      入口js

mock.js     mockjs文件

在来看下完成后的效果



5.在入口js（main.js）里引入mockjs

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | // The Vue build version to load with the `import` command  // (runtime-only or standalone) has been set in webpack.base.conf with an alias.  import Vue from 'vue'  import App from './App'  import router from './router'    Vue.config.productionTip = false    // 引入mockjs  require('./mock.js')    /\* eslint-disable no-new \*/  new Vue({      el: '#app',      router,      template: '<App/>',      components: {          App      }  })    Vue.filter('getYMD', function(input) {      return input.split(' ')[0];  }) |

这里我添加了额一个常用的时间整理过滤器 getYMD

6. 添加一个mock规则（mock.js）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | // 引入mockjs  const Mock = require('mockjs');  // 获取 mock.Random 对象  const Random = Mock.Random;  // mock一组数据  const produceNewsData = function() {      let articles = [];      for (let i = 0; i < 100; i++) {          let newArticleObject = {              title: Random.csentence(5, 30), //  Random.csentence( min, max )              thumbnail\_pic\_s: Random.dataImage('300x250', 'mock的图片'), // Random.dataImage( size, text ) 生成一段随机的 Base64 图片编码              author\_name: Random.cname(), // Random.cname() 随机生成一个常见的中文姓名              date: Random.date() + ' ' + Random.time() // Random.date()指示生成的日期字符串的格式,默认为yyyy-MM-dd；Random.time() 返回一个随机的时间字符串          }          articles.push(newArticleObject)      }        return {          articles: articles      }  }    // Mock.mock( url, post/get , 返回的数据)；  Mock.mock('/news/index', 'post', produceNewsData); |

7.在Hello.vue 中请求文档接口，并接收mock数据

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76 | <template>    <div class="index">      <div v-for="(item, key) in newsListShow">        <news-cell        :newsDate="item"        :key="key"        ></news-cell>      </div>    </div>  </template>    <script>  import api from './../axios/api.js'  import NewsCell from './NewsCell.vue'    export default {    name: 'index',    data () {      return {        newsListShow: [],      }    },    components: {      NewsCell    },    created() {      this.setNewsApi();    },    methods:{      setNewsApi: function() {        api.JH\_news('/news/index', 'type=top&key=123456')        .then(res => {          console.log(res);          this.newsListShow = res.articles;        });      },    }  }  </script>    <!-- Add "scoped" attribute to limit CSS to this component only -->  <style scoped>  .topNav{    width: 100%;    background: #ED4040;    position: fixed;    top:0rem;    left: 0;    z-index: 10;  }  .simpleNav{    width: 100%;    line-height: 1rem;    overflow: hidden;    overflow-x: auto;    text-align: center;    font-size: 0;    font-family: '微软雅黑';    white-space: nowrap;  }  .simpleNav::-webkit-scrollbar{height:0px}  .simpleNavBar{    display: inline-block;    width: 1.2rem;    color:#fff;    font-size:0.3rem;  }  .navActive{    color: #000;    border-bottom: 0.05rem solid #000;  }  .placeholder{    width:100%;    height: 1rem;  }  </style> |

 注意：api.JH\_news是我封装的axios函数

axios/api.js如下

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | import axios from 'axios'  import vue from 'vue'    axios.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded'    // 请求拦截器  axios.interceptors.request.use(function(config) {      return config;    }, function(error) {      return Promise.reject(error);    })    // 响应拦截器  axios.interceptors.response.use(function(response) {    return response;  }, function(error) {    return Promise.reject(error);  })    // 封装axios的post请求  export function fetch(url, params) {    return new Promise((resolve, reject) => {      axios.post(url, params)        .then(response => {          resolve(response.data);        })        .catch((error) => {          reject(error);        })    })  }    export default {    JH\_news(url, params) {      return fetch(url, params);    }  } |

8.在NewsCell.vue展示数据

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122 | <template>    <section class="financial-list">      <section class="collect" @click="jumpPage">        <aside>          <h2>{{newsDate.title}}</h2>          <section class="Cleft clearfix">            <img class="fl" src="./../assets/icon/eyes.png" style="width:0.24rem;height:0.2rem;">            <span class="fl">{{newsDate.author\_name}}</span>          </section>          <section class="Cright">            <img src="./../assets/icon/clock.png" style="width:0.2rem;height:0.2rem;">            <span>{{newsDate.date | getYMD}}</span>          </section>          <div style="clear: both"></div>        </aside>        <aside>          <img :src="newsDate.thumbnail\_pic\_s" style="border-radius: 0.2rem;">        </aside>        <div style="clear: both"></div>      </section>    </section>  </template>    <script>  export default {    name: 'NewsCell',    props: {      newsDate: Object    },    data () {      return {      }    },    computed: {    },    methods: {      jumpPage: function () {        window.location.href = this.newsDate.url      }    }  }  </script>    <style scoped>  .financial-list {    width: 100%;    height: 100%;    background-color: white;    padding: 0.28rem 0;    border-bottom: 1px solid #ccc;  }    .financial-list .collect {    width: 92%;    margin: 0 auto;  }    .financial-list .collect aside:nth-of-type(1) {    width: 63%;    float: left;  }    .financial-list .collect aside:nth-of-type(2) {    width: 32%;    height: 2rem;    float: left;    margin-left: 0.3rem;  }    .financial-list .collect h2 {    width: 100%;    height: 0.96rem;    font-size: 0.32rem;    color: #333333;    line-height: 0.48rem;    text-overflow: ellipsis;    -o-text-overflow: ellipsis;    overflow: hidden;  }    .financial-list .collect aside:nth-of-type(2) img {    width: 100%;    height: 100%;  }    .financial-list .collect aside .Cleft {    width: 45%;    float: left;    margin-top: 0.66rem;  }    .financial-list .collect aside .Cleft span{    display: block;    width: 1.4rem;    margin-left: 0.05rem;    white-space: nowrap;    text-overflow: ellipsis;    -o-text-overflow: ellipsis;    overflow: hidden;  }    .financial-list .collect aside .Cright {    width: 55%;    float: right;    margin-top: 0.66rem;  }  .financial-list .collect aside .Cright span{    display: inline-block;    margin: 0.04rem 0 0 0.05rem;  }  .financial-list .collect aside span {    font-size: 0.2rem;    color: #999999;  }    .financial-list .collect aside .Cleft img,  .financial-list .collect aside .Cright img {    width: 0.18rem;    height: 0.24rem;    margin-top: 0.09rem;  }  </style> |

　　完成

9.所有代码可以查看我的github：  https://github.com/Jasonwang911/vue\_mockjs