### mockjs作用

生成随机数据，拦截ajax请求

### mockjs

#### 使用mockjs

项目安装mock

npm install mockjs

项目中新建mock文件

//引入mock模块

**import** Mock **from** 'mockjs'

将mock文件在main.js中导入

**import** Vue **from** 'vue'

**import** App FROM './App.vue'

**import** './mock/index.js'

Vue.config.productionTip = **false**

**new** Vue({

render:h => h(App),

}).$mount('#app')

#### mock语法

##### 生成字符串

* 生成指定次数字符串

**import** Mock **from** 'mockjs'

**const** data = Mock.mock({

"string|4":"哈哈"

})

* 生成指定范围长度字符串

**const** data = Mock.mock({

"string|1-8":"哈哈"

})

##### 生成文本

* 生成一个随机字符串

**const** data = Mock.mock({

"string":"@cword"

})

* 生成指定长度和范围

**const** data = Mock.mock({

string:"@cword(1)"

str :"@cword(10,15)"

})

##### 生成标题和句子

* 生成标题和句子

**const** data = Mock.mock({

title:"@ctitle(8)"

sentence:"@csentence"

})

* 生成指定长度的标题和句子

**const** data = Mock.mock({

title:"@ctitle(8)"

sentence:"@csentence(50)"

})

* 生成指定范围的

**const** data = Mock.mock({

title:"@ctitle(5,8)"

sentence:"@csentence(50,100)"

})

##### 生成段落

* 随机生成段落

**const** data = Mock.mock({

content:"@cparagraph()"

})

##### 生成数字

* 生成指定数字

**const** data = Mock.mock({

"number|80":1

})

* 生成范围数字

**const** data = Mock.mock({

"number|1-99":1

})

##### 生成自增id

* 随机生成标识

**const** data = Mock.mock({

id:"@increment"

})

##### 生成姓名-地址-身份证

* 随机生成姓名-地址-身份证

**const** data = Mock.mock({

name:"@cname()"

idCard:"@id()"

address:"@city(true)"

})

##### 随机生成图片

会显示图片太大

访问这里解决：[(3条消息) mock 点击图片显示Too big of an image!\_战斗中的老段的博客-CSDN博客](https://blog.csdn.net/qq_48203828/article/details/126254689)

* 生成图片：@image（“300\*200”，‘#ff0000','#fff','gif','坤坤'）
* 参数1：图片大小

[

'300\*250','250\*250','240\*400','336\*280'

'180\*150','720\*300','468\*60','234\*60'

'388\*31','250\*250','240\*400','120\*40'

'125\*125','250\*250','240\*400','336\*280'

]

* 参数2：图片背景色
* 参数3：图片前景色
* 参数4：图片格式
* 参数5：图片文字

##### 生成时间

* @Date
* 生成指定格式时间：@date(yyyy-MM-dd hh:mm:ss)

指定数组返回的参数

* 指定长度：‘date|5’
* 指定范围:'data|5-10'

**const** data = Mock.mock({

'list|50-99':[

{

name:'@cname'

address:'@city(true)'

id:'@increment()'

}

]

})

##### 生成随机数据

Random前面都要加一个Mock

1、返回一个随机的布尔值。

Random.boolean()  
 Random.boolean( min, max, current )  
2、返回一个随机的整数  
 Random.integer()  
 Random.integer( min )  
 Random.integer( min, max )  
3、返回一个随机字符  
 Random.character()  
 Random.character( ‘lower/upper/number/symbol’ )  
 Random.character( pool )  
4、返回一个随机字符串  
 Random.string()  
 Random.string( length )  
 Random.string( pool, length )  
 Random.string( min, max )  
 Random.string( pool, min, max )  
5、返回一个整数数组  
 Random.range( stop )  
 Random.range( start, stop )  
 Random.range( start, stop, step )  
6、返回一个随机的日期字符串  
 Random.date()  
 Random.date(format)  
7、返回一个随机的时间字符串  
 Random.time()  
 Random.time( format )  
8、返回当前的日期和时间字符串。  
 Ranndom.now( unit, format )  
 Ranndom.now()  
 Ranndom.now( format )  
 Ranndom.now( unit )  
9、随机生成一个常见的英文姓名。  
 Random.name()  
 Random.name( true )【true；可选，指示是否生成中间名】  
10、随机生成一个常见的中文姓名。  
 Random.cname()

#### mock拦截请求

##### 定义get请求

Mock.mock('api/get/news','get',()=>{

**return**{

status:200,

message:"获取数据成功"

}

})

##### 定义post请求

Mock.mock('api/post/news','post',()=>{

**return**{

status:200,

message:"获取数据成功"

}

})

#### 实现新闻管理案例

##### 获取数据

接口地址：：/api/get/news

接口参数：

pageindex：页码

pagesize:每页的条数

请求类型：get

返回的数据：

{

status:200,

message:"获取新闻列表成功",

list:[

{

"id":1,

"title":"解忧杂货店",

"content":"《解忧杂货店》是日本作家东野圭吾写作的长篇小说。2011年《小说野性时代》连载，于2012年3月由角川书店发行单行本",

"img\_url":"http://t15.baidu.com/it/u=2090705107,20534764&fm=224&app=112&f=JPEG?w=500&h=500&s=61D0718656561FFFE504A51703000067",

"add\_time":"1984-04-03 11:43:37"}

],

total:50

}

}

##### 添加新闻

接口地址：：/api/add/news

接口参数：

title：'标题'

content：内容

请求类型：post

返回的数据：

{

status:200,

message:"获取新闻列表成功",

list:[

{

"id":1,

"title":"解忧杂货店",

"content":"《解忧杂货店》是日本作家东野圭吾写作的长篇小说。2011年《小说野性时代》连载，于2012年3月由角川书店发行单行本",

"img\_url":"http://t15.baidu.com/it/u=2090705107,20534764&fm=224&app=112&f=JPEG?w=500&h=500&s=61D0718656561FFFE504A51703000067",

"add\_time":"1984-04-03 11:43:37"}

],

total:50

}

}

##### 删除新闻

接口地址：：/api/delete/news

接口参数：

id；新闻id

请求类型：post

返回的数据：

{

status:200,

message:"获取新闻列表成功",

list:[

{

"id":1,

"title":"解忧杂货店",

"content":"《解忧杂货店》是日本作家东野圭吾写作的长篇小说。2011年《小说野性时代》连载，于2012年3月由角川书店发行单行本",

"img\_url":"http://t15.baidu.com/it/u=2090705107,20534764&fm=224&app=112&f=JPEG?w=500&h=500&s=61D0718656561FFFE504A51703000067",

"add\_time":"1984-04-03 11:43:37"}

],

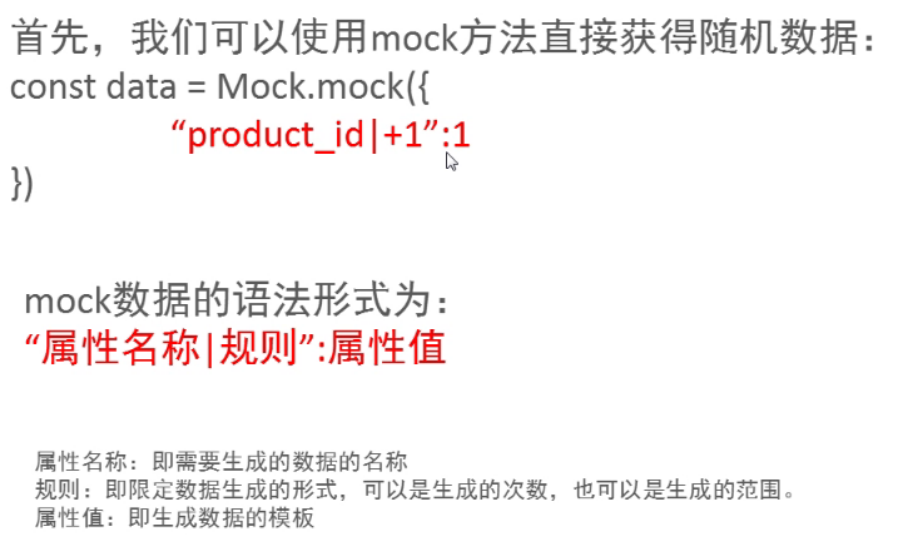
total:50

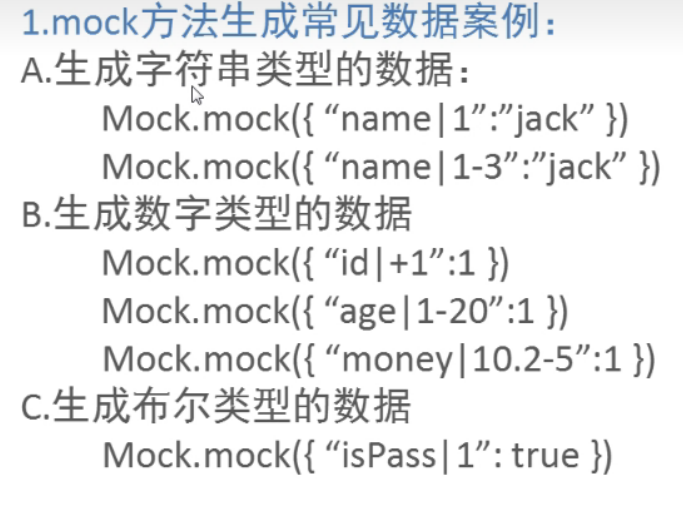
}

}

### mock数据的常用形式







“money|10.2-5”:1 整数部分一定是10，小数部分 2~5位数

“money|1-20.2-5”:1 整数部分取1~20的随机范围，小数部分 2~5位数

“name|1-3”:”jack” jack随机重复1到3次

“name|10”:”jack” 把jack重复10遍

“isPass|1”: true 随机生成true或false |后面的如果是0，就全是false

#### 数组

"list|5": [ // 固定5个元素 1-10 随机1~10个元素

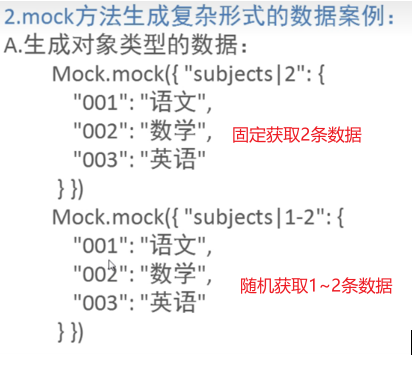
      {

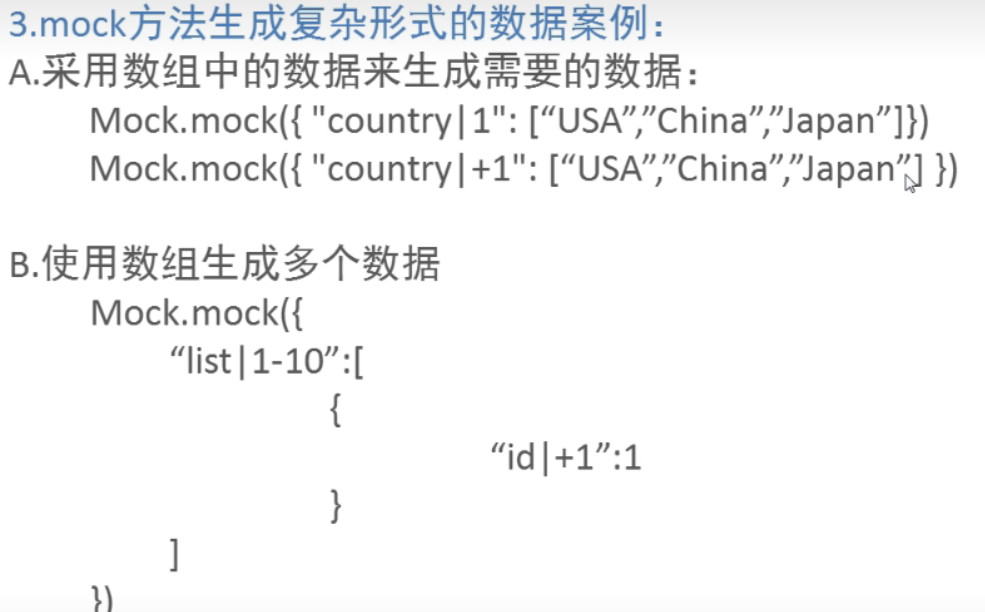
        "number|1-20.2-5": 1,

      },

    ],

#### 对象





#### Random



#### 自定义随机数据

// 自定义随机函数

  Mock.Random.extend({

    status: function () {

      const arr = ["已收货", "未收货", "已付款", "未付款", "已评价"];

      return this.pick(arr);

    },

  });

  const data = Mock.mock({

    "list|5": [

      {

        phone: /\d{11}/,

        statue: "@status()", // 使用随机函数

      },

    ],

  });

#### 正则

##### 生成随机11位数字模拟手机号码

"phone": /\d{11}/,