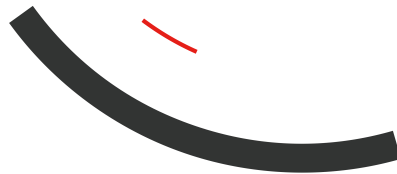




黑马程序员™  
www.itheima.com

传智播客旗下  
高端IT教育品牌

# Mock.js



问题：

前端工程师准备调后端接口时，如果发现后端接口还没开发完成，该怎么办？

干等？划水？

**NO!** 出来混，迟早要还的，我们必须立即行动起来！

**Mock:** 一种模拟后端接口的解决方案，能让我们提前调用模拟接口，完成前端开发

官网地址：<http://mockjs.com/>



## 前后端分离

让前端攻城师独立于后端进行开发。



## 开发无侵入

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。



## 数据类型丰富

支持生成随机的文本、数字、布尔值、日期、邮箱、链接、图片、颜色等。



## 增加单元测试的真实性

通过随机数据，模拟各种场景。



## 用法简单

符合直觉的接口。



## 方便扩展

支持支持扩展更多数据类型，支持自定义函数和正则。

# Mock.js 初体验

安装:

```
npm install mockjs
```

导入:

```
var Mock = require('mockjs')
```

使用:

```
var data = Mock.mock({....})
```

Mock.js 的语法规则包括两部分：

## 1. 数据模板定义规范（Data Template Definition, DTD）

数据模板中的每个属性由 3 部分构成：属性名(name)、生成规则(rule)、属性值(value)：

```
'name|rule': value
```

## 2. 数据占位符定义规范（Data Placeholder Definition, DPD）

占位符 只是在属性值字符串中占个位置，并不出现在最终的属性值中

@占位符

@占位符 (参数 [, 参数])

## 属性值是字符串 String

1. 'name|min-max': string

通过重复 string 生成一个字符串，重复次数大于等于 min，小于等于 max。

2. 'name|count': string

通过重复 string 生成一个字符串，重复次数等于 count。

## 属性值是数字 Number

1. 'name|min-max': number

生成一个大于等于 min、小于等于 max 的整数，属性值 number 只是用来确定类型。

# ■ 数据占位符定义规范

@id() : 得到随机的id

@cname(): 随机生成中文名字

@date('yyyy-MM-dd'): 随机生成日期

@paragraph(): 描述

@email(): 邮箱地址

案例：生成一个随机用户对象，结构如下所示：

```
{  
  "id": "230000201903033537",  
  "username": "卢平",  
  "date": "1978-09-16",  
  "description": "Kcb jkkvdypcst xef .",  
  "email": "n.qoimyx@ewoea.mh",  
  "age": 29  
}
```



## 1. Mock.js有什么作用？

一种模拟后端接口的解决方案，能让我们提前调用模拟接口，完成前端开发

## 2. Mock.js 的语法规则包括哪两部分？

数据模板和数据占位符

## 3. Mock.js基本代码格式

```
Mock.mock({  
  id: "@id()", //得到随机的id, 对象  
  username: "@cname()", //随机生成中文名字  
  date: "@date()", //随机生成日期  
  description: "@paragraph()", //描述  
  email: "@email()", //email  
  'age|18-38': 0  
})
```

## 问题：

我们如何在Vue项目中，使用Mock.js来模拟接口，并完成调用呢？

## 步骤：

1. 定义接口路由，在接口中并返回mock模拟的数据
2. 在vue.config.js中配置devServer，在before属性中引入接口路由函数
3. 使用axios调用该接口，获取数据

在Vue中使用Mock.js来模拟接口，需要哪几步？

## 1. 定义接口路由，在接口中并返回mock模拟的数据

```
module.exports = function (app) {  
  app.use('/api/userinfo', (req, res) => {  
    res.json(obj)  
  })  
}
```

## 2. 在vue.config.js中配置devServer，在before属性中引入接口路由函数

```
module.exports = {  
  devServer: {  
    before: require('./mock/index.js')//引入mock/index.js  
  }  
}
```

## 3. 使用axios调用该接口，获取数据

如何控制Mock接口的开关？

1. 新建.env.development，定义环境变量

```
MOCK=true
```

2. 定义接口路由前，判断当前MOCK环境变量是否为true

```
module.exports = function (app) {  
  if (process.env.MOCK == 'true') {  
    app.use('/api/userinfo', (req, res) => {  
      res.json(obj)  
    })  
  }  
}
```



传智播客旗下高端IT教育品牌