

link: null  
title: 珠峰架构师成长计划  
description: null  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=19 sentences=31, words=244

## 1. 什么是Mock #

- 什么是Mock?? Mock其实就是真实数据存在之前，即调试期间的代替品

## 2. 如何Mock数据? #

- 最low的方式将 Mock 数据写在代码里、json文件里；
- 利用 Charles、Fiddler等代理工具，将 URL 映射到本地文件；
- 本地起 Mock Server，即mockjs。有点麻烦每次修改了后还要重启服务

## 3. Mock语法 #

- [mockjs \(http://mockjs.com/examples.html\)](http://mockjs.com/examples.html)

占位符 含义 @ip 随机输出一个IP @id 随机输出长度18的字符，不接受参数 array/1-10 随机输出1-10长度的数组，也可以直接是固定长度 object/2 输出一个两个key值的对象 @image() 返回一个占位图url，支持size, background, foreground, format, text @cname 生成一个中文名 @datetime 生成一个随机的时间

```
let Mock = require('mockjs');
let result = Mock.mock({
  "code": 0,
  "message": "请求成功",
  "data|20": [{
    "id": "@id",
    "ip": "@ip",
    "name": "@cname",
    "userId": "@id",
    "stars|2": ['★'],
    "colors|2": {red: 'red', yellow: 'yellow', blue: 'blue'},
    "avatar": "@image()",
    "createAt": "@datetime"
  }]
})
console.log(result);
```

## 4. easy-mock #

- [easy-mock \(https://easy-mock.com/\)](https://easy-mock.com/)
- Easy Mock就是一个在线创建mock的服务平台，帮你省去你 配置、安装、起服务、维护、多人协作Mock数据不互通等一系列繁琐的操作

## 5. 基本用法 #

```
{
  "data|5": [{
    "id|1-1000": 1,
    "title": "@csentence",
    "url": "@url",
    "image": "@image(300x200)",
    "createAt": "@datetime"
  }],
  "code": 0
}
```

## 5.高阶用法 #

对象 描述 MockMock 对象 \_req.url 获得请求 url 地址 \_req.method 获取请求方法 \_req.params 获取 url 参数对象 \_req.querystring 获取查询参数字符串(url中?后面的部分)，不包含? \_req.query 将查询参数字符串进行解析并以对象的形式返回，如果没有查询参数字符串则返回一个空对象 \_req.body 当 post 请求以 x-www-form-urlencoded 方式提交时，我们可以拿到请求的参数对象 ... \_req.cookies、ip、host等等  
[docs \(https://easy-mock.com/docs\)](https://easy-mock.com/docs)

```
GET /mock/5cfa0f3b9a819c502224e47f/news?%3Fcode=0&name=zhufeng HTTP/1.1
Accept: application/json, *
```

```
{
  success: true,
  data: {
    name: function({
      _req
    }) {
      return _req.query.name;
    },
    code: function({
      _req
    }) {
      return _req.query.code ? 0 : 1;
    },
    data: function({
      _req,
      Mock
    }) {
      return {
        token: Mock.mock("@guid()"),
        userId: Mock.mock("@id(5)"),
        cname: Mock.mock("@cname()"),
        name: Mock.mock("@name()"),
        avatar: Mock.mock("@image(200x100, #FF6600)")
      }
    }
  }
}
```

## 6.在线调试 #

- [在线调试 \(https://easy-mock.com/mock/5a0aad39eace86040209063d/pjhApi\\_1510649145466/api/common/login#!method=post\)](https://easy-mock.com/mock/5a0aad39eace86040209063d/pjhApi_1510649145466/api/common/login#!method=post)