

# 数据Mock与前后端联调

从入门到工作2020版

# 版权声明

- 本内容版权归属杭州饥人谷教育科技有限公司所有。
- 任何媒体、网站、个人未经本公司授权不得转载、链接、转帖，或以其他方式复制、发布、发表。
- 已获得饥人谷授权的媒体、网站、个人在使用时必须注明“资料来源：饥人谷”
- 如咨询课程，微信联系xiedaimala02 或 xiedaimala03 ， 或者点击链接直接咨询 <https://dwz.cn/pPsjntxY>

# 数据Mock

# Mock方法

- 使用node.js 手写 server
  - ✓ 见案例
- Mock.js & Mock平台
  - ✓ 淘宝 Rap2 <http://rap2.taobao.org/>

# 实现一个Server

# 20行代码实现server mock

```
const http = require('http')
const fs = require('fs')
const url = require('url')

http.createServer((req, res) => {
  let pathObj = url.parse(req.url, true)
  switch (pathObj.pathname) {
    case '/getWeather':
      if(pathObj.query.city === 'beijing')
        res.end(JSON.stringify({city: 'beijing', weather: 'sunny'}))
      else
        res.end(JSON.stringify({city: pathObj.query.city, weather: 'unknown'}))
      break
    default:
      try {
        let pathname = pathObj.pathname === '/' ? '/index.html' : pathObj.pathname
        res.end( fs.readFileSync(__dirname + pathname) )
      } catch(e) {
        res.writeHead(404, 'Not Found')
        res.end('<h1>404 Not Found~</h1>')
      }
  }
}).listen(8080)
```

# 高级用法

- 从实现简易 Server 到实现node后端  
框架 Express.js
  - ✓ <https://github.com/jirengu/node-server>

# Mock.js



# Mock.js

- 用法演示

- ✓ <http://mockjs.com/examples.html>

- 常见用法

- ✓ @ctitle(3, 10)、@cparagraph、@cword、@cname、@integer(10, 100)、@float(20, 30, 2, 3)、@color、@date、@time、@now、@id、@url、@email、@image('200x100')

- ✓ 范例

```
{
  'statusCode|1': [1, 3, 2, 8],
  'msg': '@cword(4, 10)',
  'data|4': [
    {
      id: '@id',
      title: '@ctitle',
      author: '@cname',
      createdAt: '@datetime'
    }
  ]
}
```

# Rap2

- 地址
  - ✓ <http://rap2.taobao.org/>

# 范例

```
{
  "string|1-10": "★",
  "number|1-100": 1,
  "boolean|1-2": true,
  "regexp": "[a-z][A-Z][0-9]/",
  "function": "() => Math.random()",
  "array|1-10": [
    {
      "foo|+1": 5,
      "bar|1-10": "★"
    }
  ],
  "items": [
    1,
    true,
    "hello",
    "/\\w{10}/"
  ],
  "object": {
    "foo|+1": 2,
    "bar|1-10": "★"
  },
  "placeholder": "@title"
}
```

```
{
  "city": "function(){\n  return '$city$'\n}",
  "weather|1": [
    "晴",
    "阴",
    "小雨",
    "大雨",
    "雷阵雨",
    "小雪",
    "大雪"
  ],
  "date": "@now",
  "description": "@cparagraph",
  "temperature": "@integer(10, 15)°C ~ @integer(15, 20)°C"
}
```

```
{
  "status|1": [
    0,
    1,
    2
  ],
  "msg": "function(){ \n  if(this.status === 0)\n    return '登录成功'\n  else if(this.status === 1) \n    return '用户不存在'\n  else \n    return '用户名或者密码错误'\n}"
}
```

# 接口规范

# 接口约定

- 当前接口的路径是什么？ 如 /auth/register
- 当前接口提交数据的类型是什么？ 如
  - ✓ GET 获取数据
  - ✓ POST 提交或者创建
  - ✓ PATCH 修改数据，部分修改
  - ✓ DELETE 删除数据
  - ✓ PUT 修改数据，整体替换原有数据
- 参数类型/格式， 比如
  - ✓ fromdata, 或者 application/x-www-form-urlencoded
- 参数字段， 及限制条件
- 返回成功的数据格式
- 返回失败的数据格式

# 范例

- 查看范例

✓ <https://www.yuque.com/docs/share/08fd7cfb-6716-4409-9b15-fd9e9d491f34>

- 扫码手机看



# 接口测试

# curl

// GET请求

```
curl "http://rap2api.taobao.org/app/mock/244238/getWeather?city=beijing"
```

// -d 提交的参数, 默认是POST请求

```
curl -d "username=aaaa&password=bbb" "http://rap2api.taobao.org/app/mock/244238/login"
```

// -i 展示响应头

```
curl -d "username=hunger1&password=123456" "http://blog-server.hunger-valley.com/auth/login" -i
```

// -H 设置请求头

```
curl -H "Content-Type:application/json" -X POST -d '{"user": "admin", "passwd":"12345678"}'  
http://127.0.0.1:8000/login
```

// -X 设置请求类型

```
curl -d "username=aaaa&password=bbb" -X POST  
"http://rap2api.taobao.org/app/mock/244238/login"
```

// -b 请求带上cookie

```
curl "http://blog-server.hunger-valley.com/auth" -b  
"connect.sid=s%3AmeDbrn03UtTM8fqChaPQ20wmWlnKeHiu.e3uMtu7j1zQ1iNeaajCmxkYYG  
Q%2FyHV1ZsozMvZYWC6s"
```