

# Postman高级使用

接口自动化测试课程



黑马程序员  
[www.itheima.com](http://www.itheima.com)

传智教育旗下  
高端IT教育品牌

# 学习目标

Learning Objectives

1. 掌握Postman断言/关联/参数化/测试报告使用
2. 能够使用Postman进行接口手工测试



# 目录

Contents

- ◆ Postman断言
- ◆ Postman自动关联
- ◆ Postman参数化
- ◆ Postman测试报告

# Postman断言

01

- 响应状态码断言
- 包含指定字符串断言
- JSON数据断言

## 学习目标

1. 能够使用Postman对响应结果自动断言

Postman断言：让Postman工具代替人工**自动判断**预期结果和实际结果是否一致

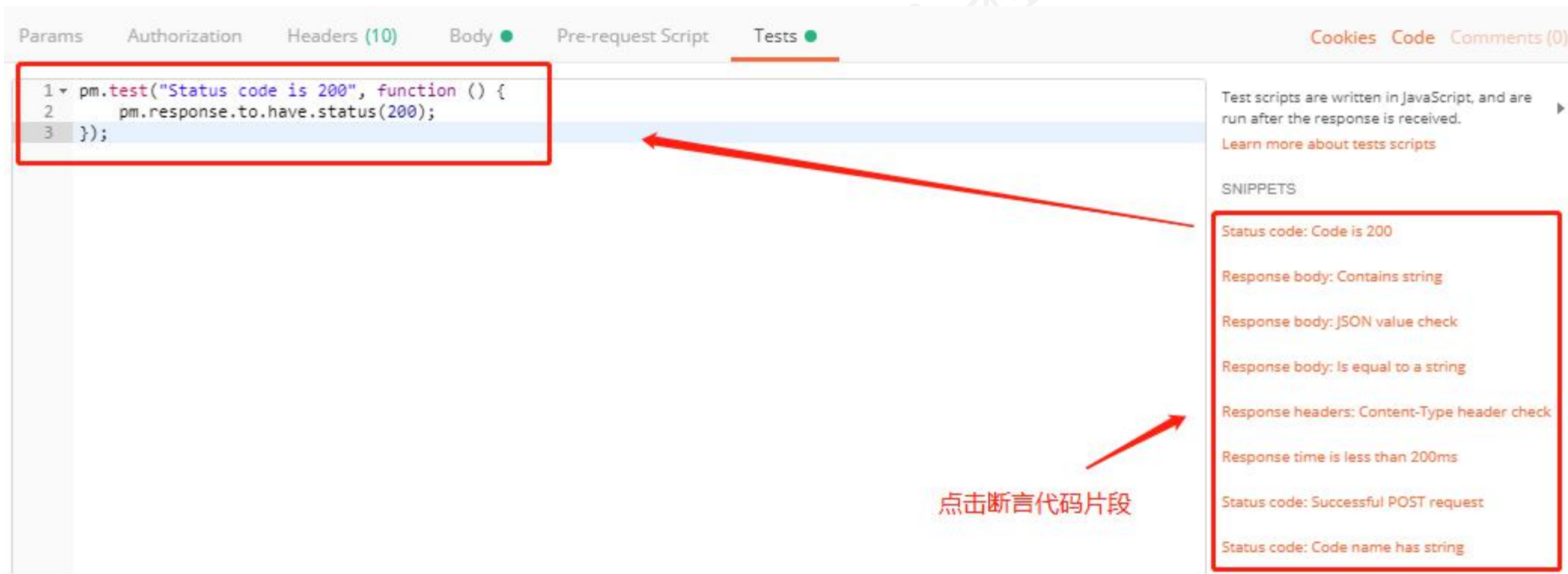
## 用法：

- ‘Tests’后置脚本标签页编写JavaScript断言代码
- 断言结果（PASS/FAIL）在‘Test Results’标签页中展示

## Postman断言

### 选择断言代码片段

- 在Tests输入框的右侧配置了一些常用的断言代码片段，点击后自动填写到Tests输入框中，然后再进行修改



## 响应状态码断言

- 模板名称: Status code: Code is 200
- 模板内容:

```
1 // 判断响应状态码是否等于200
2 pm.test("Status code is 200", function () {
3     pm.response.to.have.status(200);
4 });
5
6 pm.test() 函数:
7 参数1: 字符串-测试断言名称
8 参数2: 回调函数-具体断言语句
9
10 pm.response.to.have.status(code:Number): 判断是否包含指定的状态码
```

## 包含指定字符串断言

- 模板名称: Response body: Contains string
- 模板内容:

```
1 // 判断响应结果是否包含指定字符串
2 pm.test("Body matches string", function () {
3     pm.expect(pm.response.text()).to.include("string_you_want_to_search");
4 });
5
6 //通过一系列调用链判断是否符合预期
7 pm.expect(): 接收实际结果
8 .to : 连接符, 用于连接断言与判断
9 .include() : 用于指定断言方式和预期结果
10 pm.response.text() : 获取响应结果的文本格式数据
```



## JSON数据断言

- 模板名称: Response body: JSON value check
- 模板内容:

```
1 // 校验响应的JSON数据
2 pm.test("Your test name", function () {
3     var jsonData = pm.response.json();
4     pm.expect(jsonData.value).to.eql(100);
5 });
6
7 pm.response.json() 获取响应结果的json数据
8 .eql() : 用于指定断言方式和预期结果
```



## 针对ihrm系统登录成功接口用例完成断言设置

```
1. 响应状态码断言
2. JSON数据断言
{
  "success": true,
  "code": 10000,
  "message": "操作成功!"
}
```



思考

1. 断言的作用是什么?
2. 常见断言有哪些方式
  - ★ 响应状态码断言
  - ★ 包含字符串断言
  - ★ JSON数据断言



# 目录

Contents

- ◆ Postman断言
- ◆ Postman自动关联
- ◆ Postman参数化
- ◆ Postman测试报告

## 02

# Postman自动关联

- 解决方案

### 学习目标

1. 能够使用Postman进行自动关联数据处理



思考

1. ihrm系统：用户登录后才能访问对应的资源

所有接口都需要依赖登录接口返回的token值

2. 业务场景：

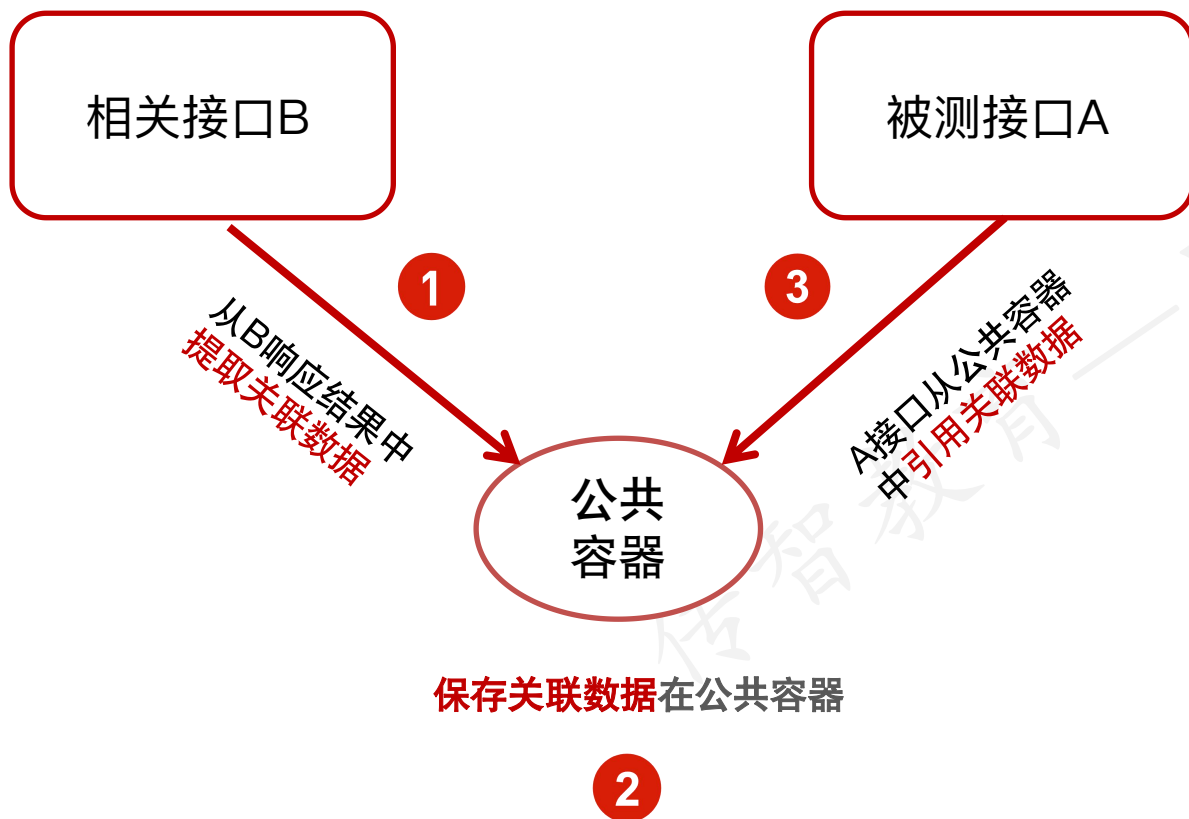
新增员工—> 查询员工—> 修改员工—> 删除员工

后三个接口依赖新增员工接口返回的员工id

手动关联

自动关联

## 自动关联解决思路



### 核心代码:

```
// 1.提取响应数据
```

```
var jsonData = pm.response.json();
```

```
// 2.设置环境变量
```

```
pm.environment.set("var_name", value);
```

```
// 3.引用环境变量
```

请求参数中引用: `{{var_name}}`

代码中引用:

```
var value = pm.environment.get("var_name");
```

## 案例

### ihrm系统**查询员工接口**关联实现

需求：

1. 查询员工接口： /api/sys/user/:target
2. 登录接口： /api/sys/login
3. 查询员工接口请求头Authorization字段—>登录接口返回的data字段

分析：

- ① 登录接口响应体中提取data
  - ② 设置环境变量保存data
  - ③ 查询员工接口引用data
- 在登录接口**后置脚本**内实现





思考

Postman**自动**关联：解决什么问题？

1提取关联数据

- `pm.response.json()`

2保存关联数据

- `pm.environment.set()`

3引用关联数据

- `{{变量名}}`
- `pm.environment.get()`



# 目录

Contents

- ◆ Postman断言
- ◆ Postman自动关联
- ◆ Postman参数化
- ◆ Postman测试报告

# 03

## Postman参数化

- 测试数据文件准备
- 参数化实现

### 学习目标

1. 能够使用Postman进行接口参数化处理

应用场景：

测试脚本中**仅测试数据不一样**，使用参数化提高脚本复用。

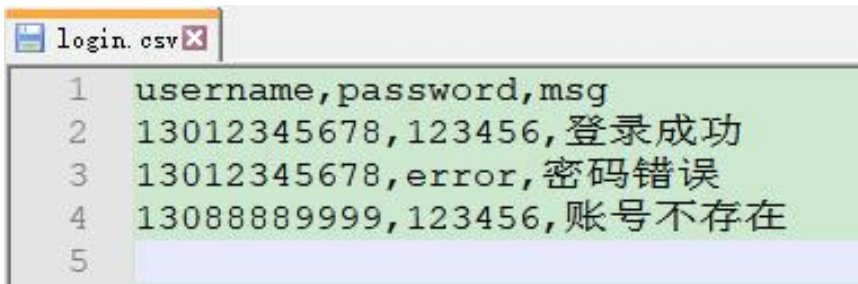
实现步骤：

1. 测试数据保存在**数据文件单独维护**
2. 引用数据文件**实现脚本迭代调用**




## 定义数据文件

- 常用的数据文件格式有：CSV、JSON



```
login.csv
1 username,password,msg
2 13012345678,123456,登录成功
3 13012345678,error,密码错误
4 13088889999,123456,账号不存在
5
```



```
login.json
1 [
2   {"username":"13012345678", "password":"123456", "msg":"登录成功"},
3   {"username":"13012345678", "password":"error", "msg":"密码错误"},
4   {"username":"13088889999", "password":"123456", "msg":"账号不存在"}
5 ]
6
```

## 引用数据文件数据

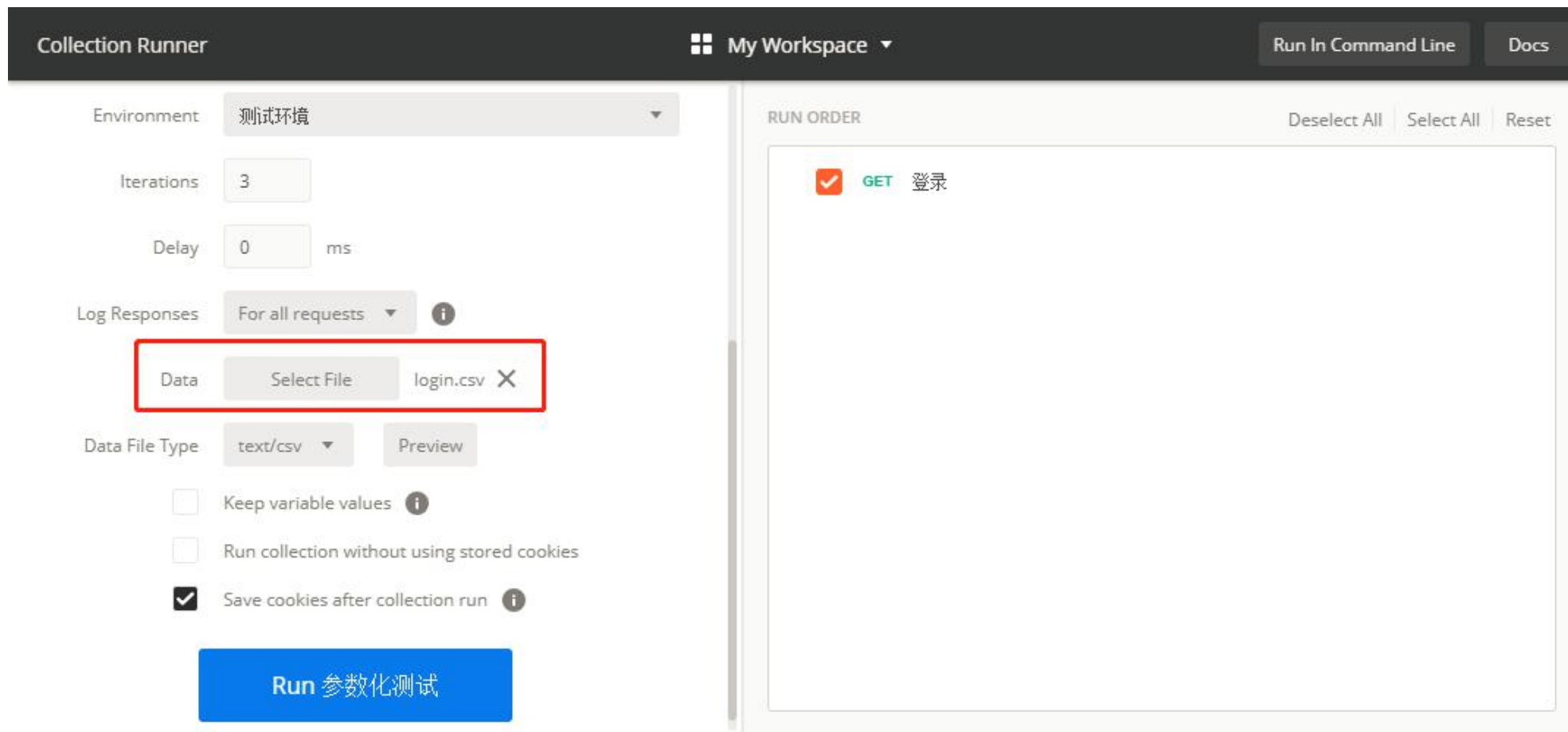
### 1. 请求参数中获取

{{}} 引用相关对象的key  
eg: {{username}}

### 2. 代码中获取

Postman内置data对象引用key  
eg. data.username

## 选择数据文件：运行测试集时选择数据文件



## 案例

## 批量查询手机号归属地

**需求：**批量查询手机号所属运营商信息，校验运营商数据是否正确

- 接口地址：`http://cx.shouji.360.cn/phonearea.php?number=13012345678`
- 部分测试数据：

手机号: 13012345678	运营商: 联通
手机号: 13800001111	运营商: 移动
手机号: 18966778899	运营商: 电信

**分析：**

- ① 如何定义数据文件？
- ② 如何引用测试数据？
- ③ 如何关联数据文件？



## 案例

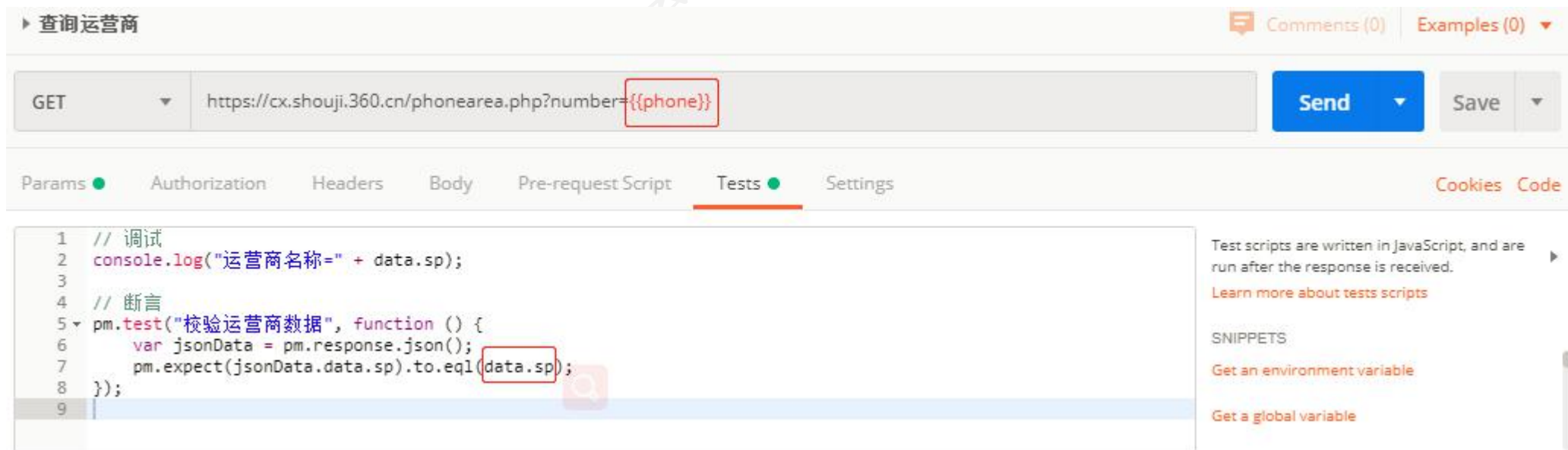
## 批量查询手机号归属地

### CSV格式

```
phone,sp  
13012345678,联通  
13800001111,移动  
18966778899,电信
```

### JSON格式

```
[  
  {"phone":"13041092162", "sp":"联通"},  
  {"phone":"13820003536", "sp":"移动"},  
  {"phone":"18911236688", "sp":"电信"}  
]
```





## ihrm系统登录接口参数化实现

**需求：**针对登录接口，将请求数据和断言数据组织成JSON文件，并使用PM批量运行

**分析：**

- ①：JSON文件格式如何组织？
- ②：如何使用Postman批量运行？



思考

1. 参数化解决的问题是什么?
2. 如何引用数据文件批量执行测试用例?



# 目录

Contents

- ◆ Postman断言
- ◆ Postman自动关联
- ◆ Postman参数化
- ◆ Postman测试报告

## 04

# Postman测试报告

- Newman使用
- 测试报告分析

### 学习目标

1. 掌握使用Newman生成测试报告

## 为什么需要测试报告?

测试**统计分析** + 测试**结果确认**

**Newman生成测试报告**

# Newman Report

Collection	demo	
Time	Thu Jul 18 2019 10:45:41 GMT+0800 (中国标准时间)	
Exported with	Newman v4.5.1	

	Total	Failed
Iterations	1	0
Requests	2	0
Prerequisite Scripts	0	0
Test Scripts	1	0
Assertions	0	0

Total run duration	672ms
Total data received	453.34KB (approx)
Average response time	203ms

Total Failures	0
----------------	---

## Requests

### 查询天气

Method	GET
URL	<a href="http://www.weather.com.cn/data/sk/101010100.html">http://www.weather.com.cn/data/sk/101010100.html</a>
Mean time per request	94ms
Mean size per request	227B
Total passed tests	0
Total failed tests	0
Status code	200

Newman: 基于nodejs, 在命令行执行Postman脚本的插件

Newman  
环境搭建

## 1. 安装nodejs

- 校验: `node -v`

## 2. 安装newman插件

- 安装newman
  - `npm install -g newman`
  - 校验newman `-v`
- 安装newman-reporter-html
  - `npm install -g newman-reporter-html`

## Newman生成测试报告命令

```
newman run 测试集文件 -e 环境变量文件 -d 测试数据文件 -r html --reporter-html-export report.html
```

### 命令说明:

- run xxx.json: 执行测试集文件
- -e source: 环境变量文件
- -d source: 测试数据文件
- -r html: 生成测试报告类型
- --reporter-html-export source: 存放报告的路径





练习

针对ihrm系统的登录接口进行参数化，并批量运行产出测试报告

## 分析测试报告：

1. 总共运行了多少条测试用例？
2. 成功了多少条case？失败了多少条case？
3. 查看每一天case的请求数据是否正确引用了数据文件
4. 查看断言响应是否正确
5. 分析失败的case是什么原因造成的？



思考

1. 请描述生成测试报告的步骤?

1) `newman run 用例集名称.json -e 环境文件名称  
.json -d 数据文件名称.json -r html --reporter-html-  
export report.html`

2. 分析测试报告：查看用例的具体数据，分析失败原因



## 总结

1. 掌握Postman断言
2. 掌握Postman关联
3. 掌握Postman参数化
4. 掌握Postman测试报告



传智教育旗下高端IT教育品牌