

Table of Contents

接口测试课程	1.1
Postman实现接口测试	1.2
Postman介绍和安装	1.2.1
Postman基本用法	1.2.2
Postman高级用法	1.2.3
Postman测试报告	1.2.4
项目实战	1.2.5

接口测试课程

课程目标

- 能够根据接口API文档编写接口测试用例
- 能够使用Postman工具进行接口测试，并能够对大量接口用例进行管理、对接口响应结果进行断言、处理多接口的依赖及生成测试报告
- 能够使用Python+Requests封装的接口测试框架，实现接口对象封装、测试用例编写、测试数据管理及生成测试报告

课程大纲

章节	知识点	
第1章 接口测试基础	1.接口及接口测试概念 2.HTTP协议 3.接口规范	4.接口测试流程 5.项目环境说明
第2章 Postman实现接口测试	1.Postman介绍和安装 2.Postman基本用法 3.Postman高级用法	4.Postman测试报告 5.项目实战
第3章 数据库操作	1. 数据库介绍 2. 数据库基本操作	3. 数据库事务操作
第4章 代码实现接口测试	1. Requests库 2. 集成UnitTest	3. 接口测试框架开发 4. 项目实战
第5章 持续集成	1. 持续集成介绍 2. Git与Git代码托管平台 3. Jenkins	4. 持续集成之Postman 5. 持续集成之代码
第6章 接口测试扩展	1. 接口Mock测试	2. 接口测试总结

Postman实现接口测试

目标

1. 掌握如何安装Postman
2. 掌握Postman的基本用法
3. 掌握全局变量与环境变量
4. 掌握Postman断言和关联
5. 掌握如何读取外部文件实现参数化
6. 掌握如何使用Newman生成HTML测试报告

Postman介绍和安装

目标

1. 了解什么是Postman
2. 掌握如何安装Postman

1. Postman介绍

Postman是一款非常流行的接口调试工具，它使用简单，而且功能也很强大。不仅测试人员会使用，开发人员也会经常使用。

官方网站：<https://www.getpostman.com/>

主要特点

1. 简单易用的图形用户界面
2. 可以保存接口请求的历史记录
3. 使用测试集Collections可以更有效的管理组织接口
4. 可以在团队之间同步接口数据

2. Postman安装

2.1 安装方式

方式一

- Postman最早是以Chrome浏览器插件的形式存在的，可以到chrome商店搜索下载安装。由于2018年初Chrome停止对Chrome应用程序的支持，所以不再推荐使用该方式。

方式二

- Postman提供了独立的安装包，不再依赖于Chrome浏览器了。同时支持MAC、Windows和Linux，推荐使用这种方式安装。
- 下载地址：<https://www.getpostman.com/apps>

2.2 安装步骤

Windows

- 双击运行安装程序，即可完成安装。

MacOS

- 下载的应用程序，你可以拖动文件到“应用程序”文件夹。

3. Postman入门示例

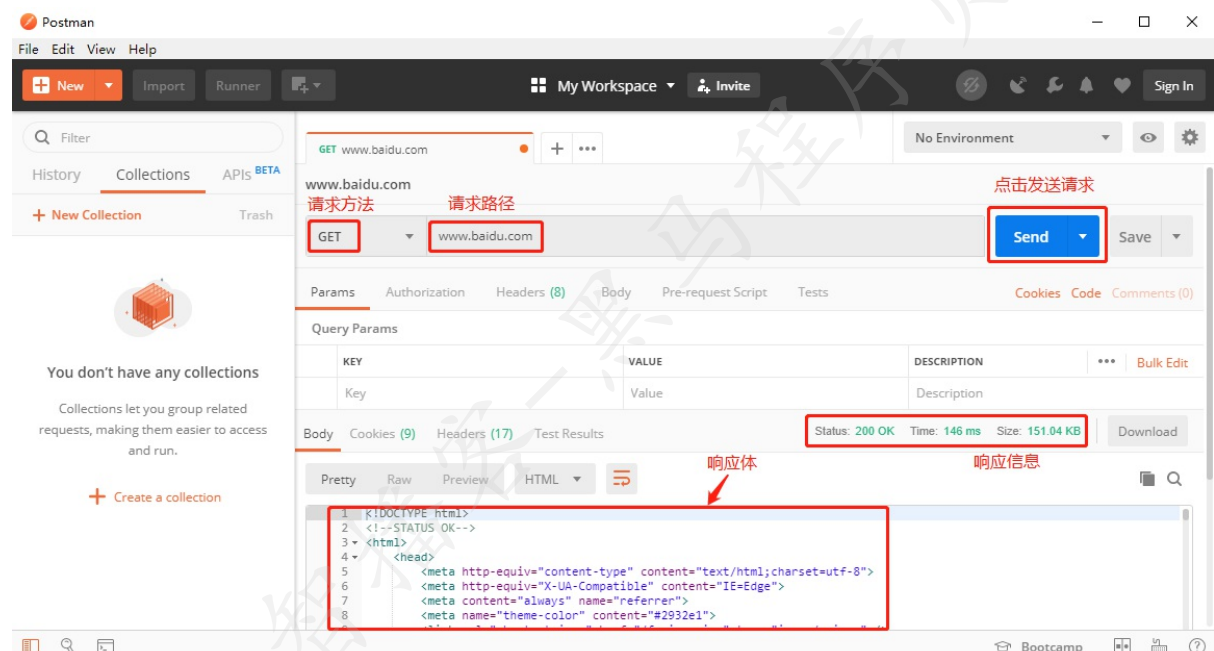
需求

使用Postman访问百度 `http://www.baidu.com`，并查看响应结果

实现步骤

1. 设置请求方式
2. 输入接口地址
3. 点击发送请求
4. 查看响应结果

示例截图



Postman基本用法

目标

1. 掌握使用Postman发送URL参数
2. 掌握使用Postman发送表单数据
3. 掌握使用Postman发送JSON数据

1. 案例一

1.1 需求

1. 访问TPshop搜索商品的接口，通过查询字符串的方式传递搜索的关键字 iPhone，并查看响应数据
2. 请求路径格式为：`http://tpshop-test.itheima.net/Home/Goods/search.html?q=iPhone`

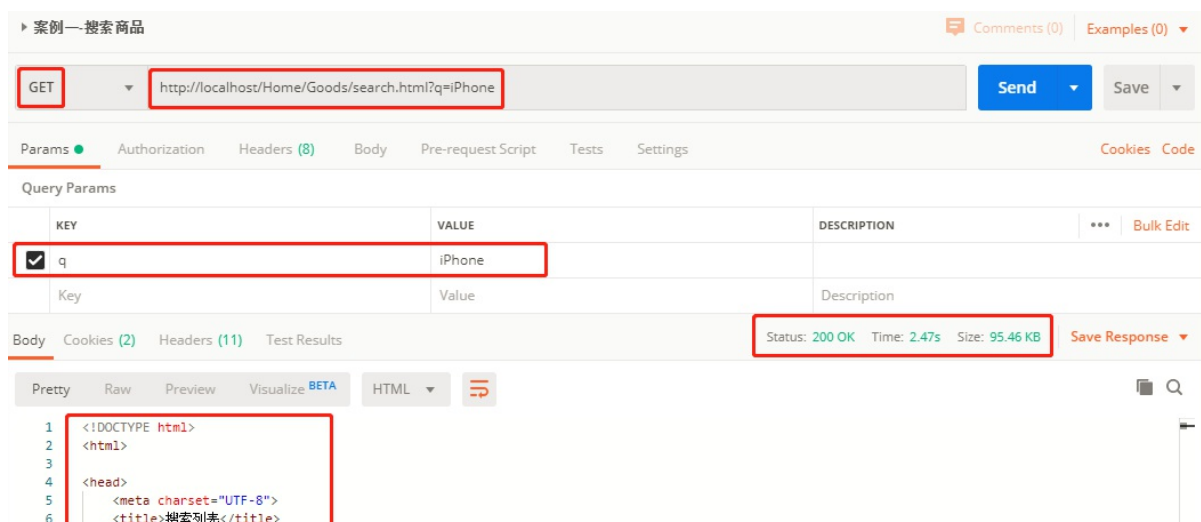
1.2 实现分析

- 请求方式：GET
- 请求路径：`http://tpshop-test.itheima.net/Home/Goods/search.html`
- 传参方式：查询字符串（q=iPhone）

1.3 实现步骤

1. 设置请求方式为GET
2. 输入请求路径
3. 设置请求数据（查询字符串）
4. 点击发送请求
5. 查看响应数据

1.4 示例截图



2. 案例二

2.1 需求

1. 请求TPshop项目的登录接口，请求数据（username: 13088888888, password: 123456, verify_code: 1234）
2. 登录接口URL: http://tpshop-test.itheima.net/index.php?m=Home&c=User&a=do_login

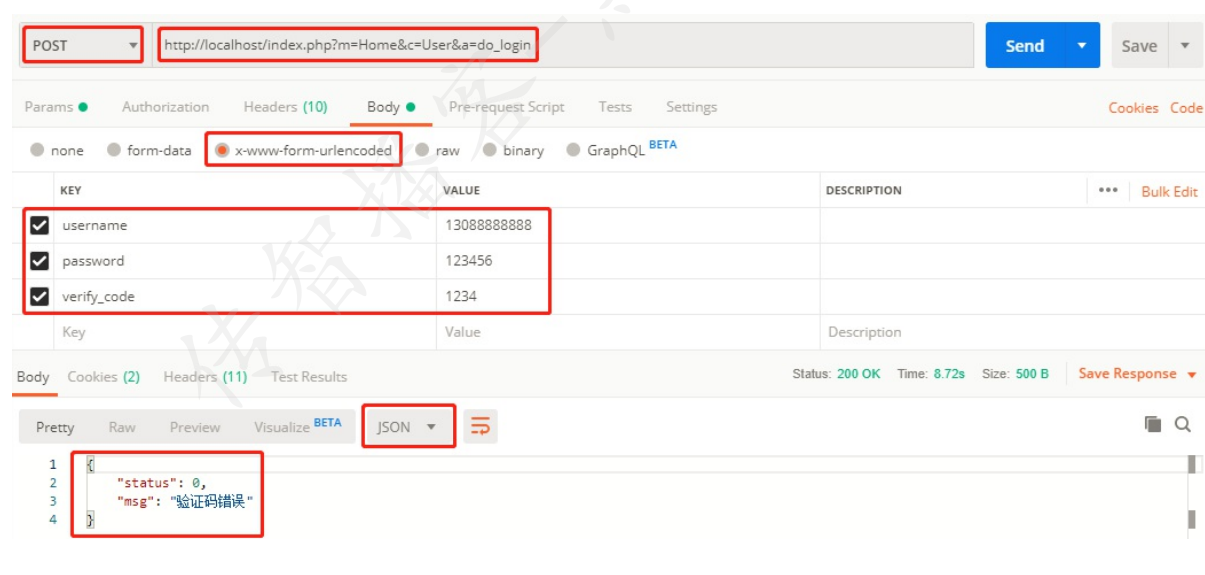
2.2 实现分析

- 请求方式: POST
- 请求路径: `http://tpshop-test.itheima.net/index.php?m=Home&c=User&a=do_login`
- 请求数据: 请求体-表单提交（username: 13088888888, password: 123456, verify_code: 1234）

2.3 实现步骤

1. 设置请求方式为POST
2. 输入请求路径
3. 设置请求数据（请求体-表单提交）
4. 点击发送请求
5. 查看响应数据

2.4 示例截图



3. 案例三

3.1 需求

1. 请求IHRM项目的登录接口，请求数据（{"mobile": "13800000002", "password": "123456"}）
2. 登录接口URL: <http://ihrm-test.itheima.net/api/sys/login>

3.2 实现分析

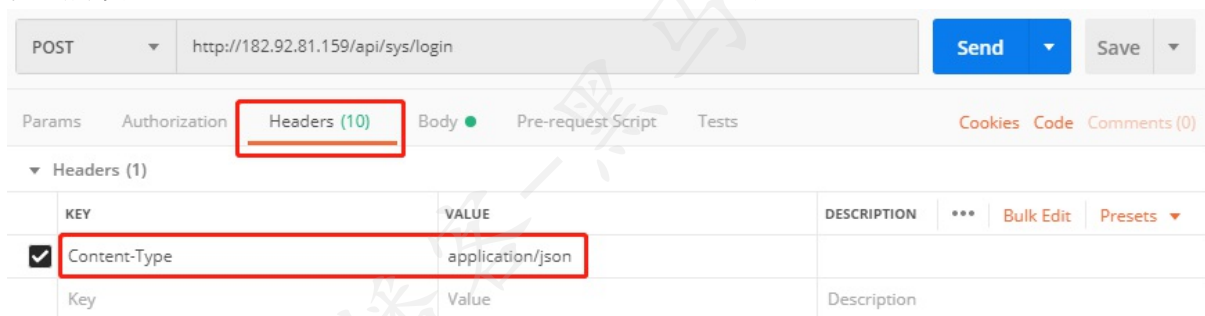
- 请求方式: POST
- 请求路径: `http://ihrm-test.itheima.net/api/sys/login`
- 请求数据:
 - 请求头- (Content-Type: application/json)
 - 请求体-JSON数据 (`{"mobile": "13800000002", "password": "123456"}`)

3.3 实现步骤

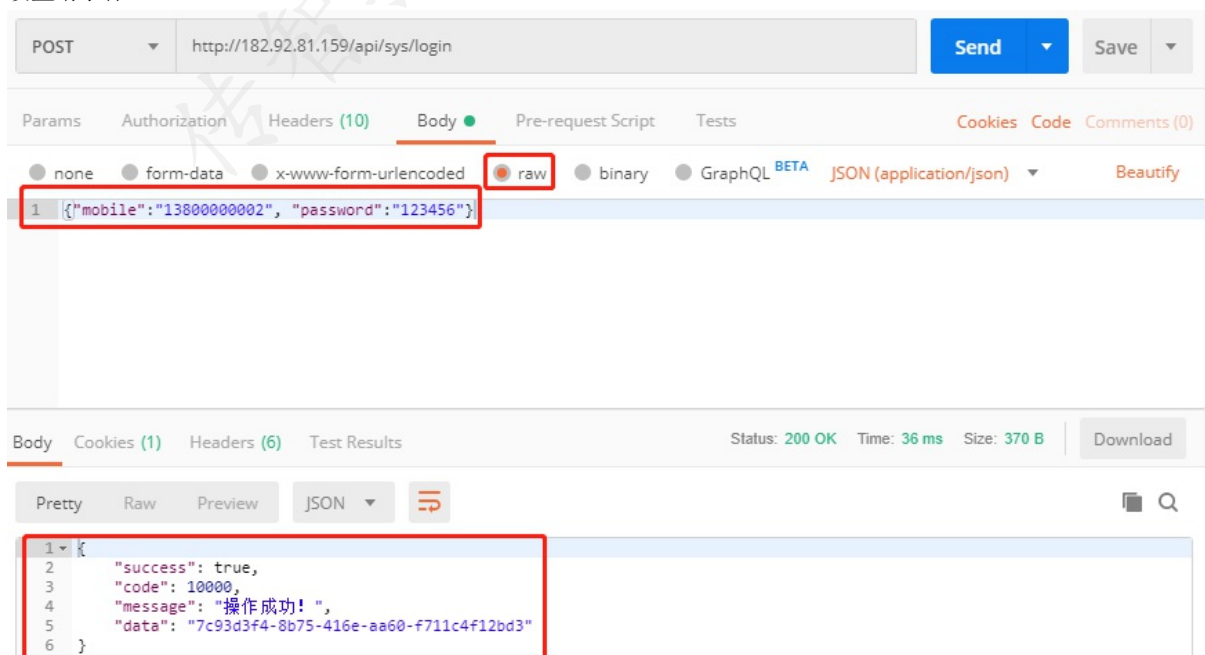
1. 设置请求方式为POST
2. 输入请求路径
3. 设置请求头
4. 设置请求体 (请求体-raw)
5. 点击发送请求
6. 查看响应数据

3.4 示例截图

设置请求头:



设置请求体:



佐智播客-黑马程序员

Postman高级用法

目标

1. 掌握如何创建用例的目录结构
2. 掌握Postman断言
3. 掌握全局变量与环境变量
4. 掌握Postman关联
5. 掌握如何读取外部文件实现参数化

1. 管理用例

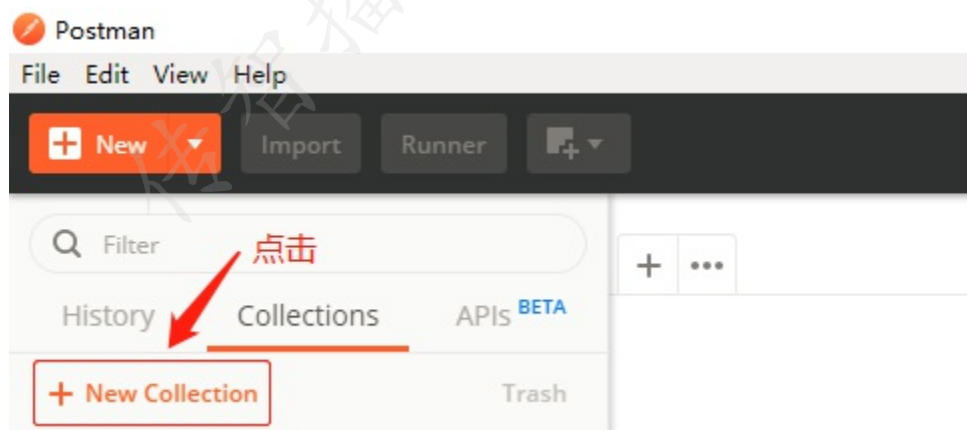
思考：针对一个大型的项目进行接口测试，会编写很多测试用例。怎么更好的管理维护测试用例呢？

1.1 实现步骤

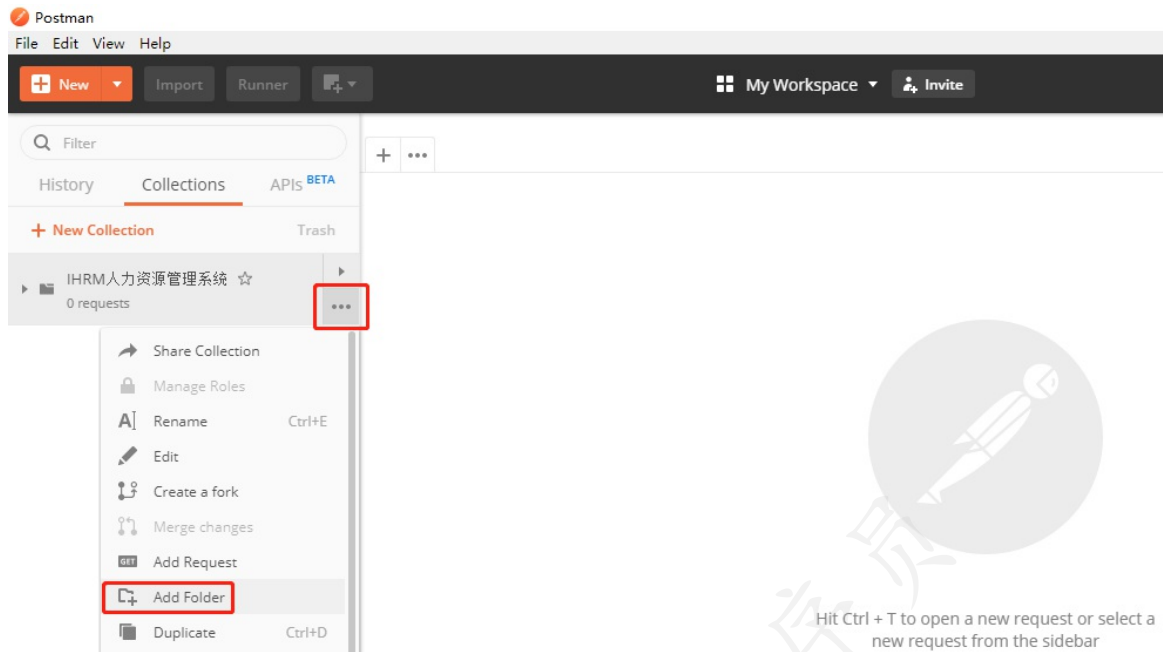
1. 创建测试集-New Collection
 - Collection可以理解成一个项目（系统），包含多个接口请求
2. 创建目录-Add Folder
 - Folder可以理解成一个项目模块
3. 添加请求-Add Request

1.2 实现步骤截图

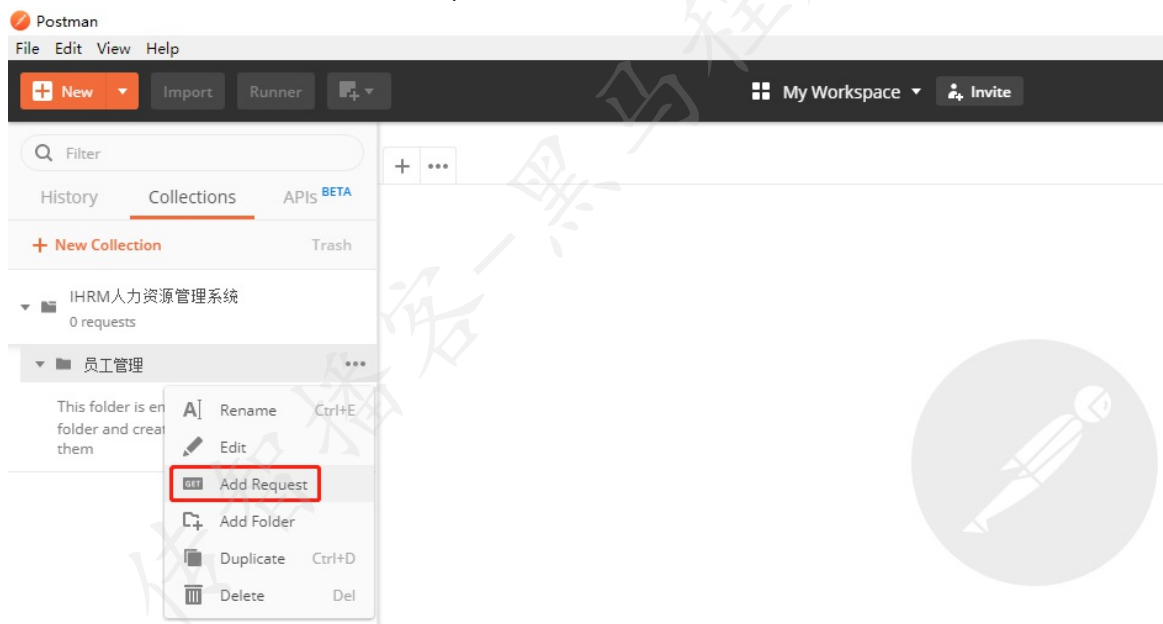
1. 创建测试集- 点击'New Collection'，在弹出框中输入测试集的名称和其他信息，再点击'Create'按钮



2. 创建目录- 在测试集上点击右键选择'Add Folder'，在弹出框中输入目录的名称和其他信息，再点击'Create'按钮



3. 添加请求- 在目录上点击右键选择‘Add Request’，在弹出框中输入请求的名称和其他信息，再点击保存按钮



2. Postman断言

断言：让程序判断预期结果和实际结果是否一致。

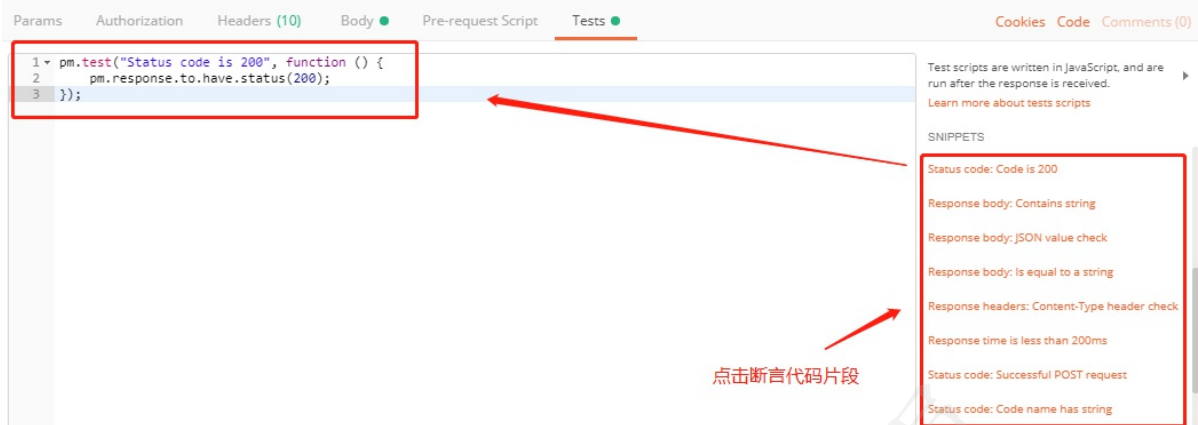
2.1 特点

- Postman的断言是使用JavaScript语言编写的，写在‘Tests’标签页里
- Tests中的脚本在发送请求之后执行，会把断言的结果（PASS/FAIL）最终在‘Test Results’标签页中展示

2.2 常用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



断言代码片段介绍

1. Status code: Code is 200

```
// 判断响应状态码是否等于200
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});
```

2. Response body: Contains string

```
// 判断响应体中是否包含指定的字符串
pm.test("Body matches string", function () {
  pm.expect(pm.response.text()).to.include("string_you_want_to_search");
});
```

3. Response body: Is equal to a string

```
// 判断响应体数据是否等于指定的字符串
pm.test("Body is correct", function () {
  pm.response.to.have.body("response_body_string");
});
```

4. Response body: JSON value check

```
// 校验响应的JSON数据
pm.test("Your test name", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.value).to.eql(100);
});
```

5. Response headers: Content-Type header check

```
// 判断响应头中是否包含指定的头标签
pm.test("Content-Type is present", function () {
  pm.response.to.have.header("Content-Type");
});
```

测试结果



3. 全局变量与环境变量

3.1 概念

- 全局变量：全局变量是全局唯一的，不可重复定义的变量
- 环境变量：
 - 一个变量只能属于某个环境，在某一个环境中变量不可重复定义
 - 在环境与环境之间可以定义重复的变量
 - 一个环境可以包含多个环境变量
 - 常见环境分类：开发环境、测试环境、生产环境

3.2 设置变量

- 全局变量
 1. 手动设置
 2. 代码设置： `pm.globals.set("var_name", value);`
- 环境变量
 1. 手动设置
 2. 代码设置： `pm.environment.set("var_name", value);`

3.3 获取变量值

- 全局变量
 1. 请求参数中获取： `{{var_name}}`
 2. 代码中获取： `var value = pm.globals.get("var_name");`
- 环境变量
 1. 请求参数中获取： `{{var_name}}`
 2. 代码中获取： `var value = pm.environment.get("var_name");`

4. 请求前置脚本

请求前置脚本就是在“Pre-request Script”标签中所编写的脚本，是在请求发送之前要执行的代码片段。

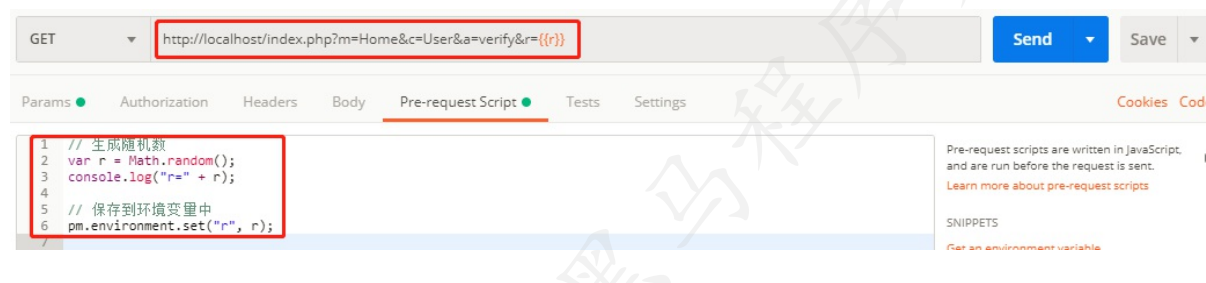
应用场景：

- 请求参数中包含一个随机数或者时间戳
- 请求之前需要对请求参数做进一步处理（如：对敏感参数进行加密）

案例

- 需求：请求TPshop项目中获取图片验证码的接口，需要在URL中设置随机数
- 请求路径：`http://localhost/index.php?m=Home&c=User&a=verify&r=0.6569606479434671`
- 分析：在JS中如何生成随机数？`var r = Math.random();` // 返回[0, 1)之间的随机数

实现截图



5. Postman关联

当请求之间有依赖关系，比如一个请求的入参是另一个请求返回的数据，这时候就需要用到关联处理。

5.1 如何实现？

- 要获取某个请求的响应结果数据，可以通过在“Tests”标签页中编写JavaScript脚本来实现

```
// 获取JSON格式的响应数据
var jsonData = pm.response.json();
```

- 多个请求之间可以使用全局变量或环境变量来传递数据

```
// 把数据保存到全局变量中
pm.globals.set('city', city);
```

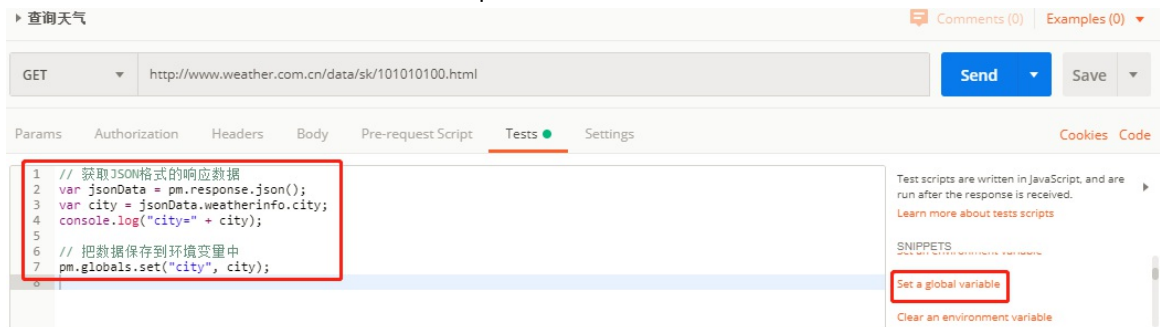
- 获取某个全局变量的值：`{{var_name}}`

5.2 案例

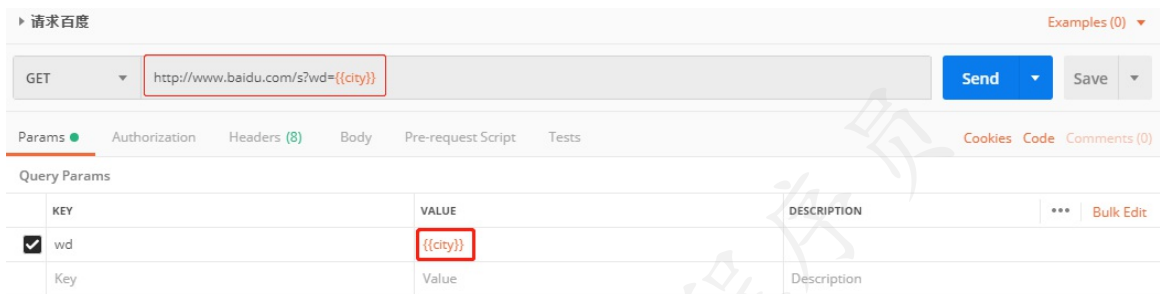
1. 请求获取天气的接口，<http://www.weather.com.cn/data/sk/101010100.html>
2. 获取返回结果中的城市名称
3. 调用百度搜索接口：`http://www.baidu.com/s?wd=北京`，把获取到的城市名称作为请求参数

实现步骤截图

1. 在第一个请求的'Tests'标签页中编写JavaScript脚本



2. 在第二个请求中使用全局变量

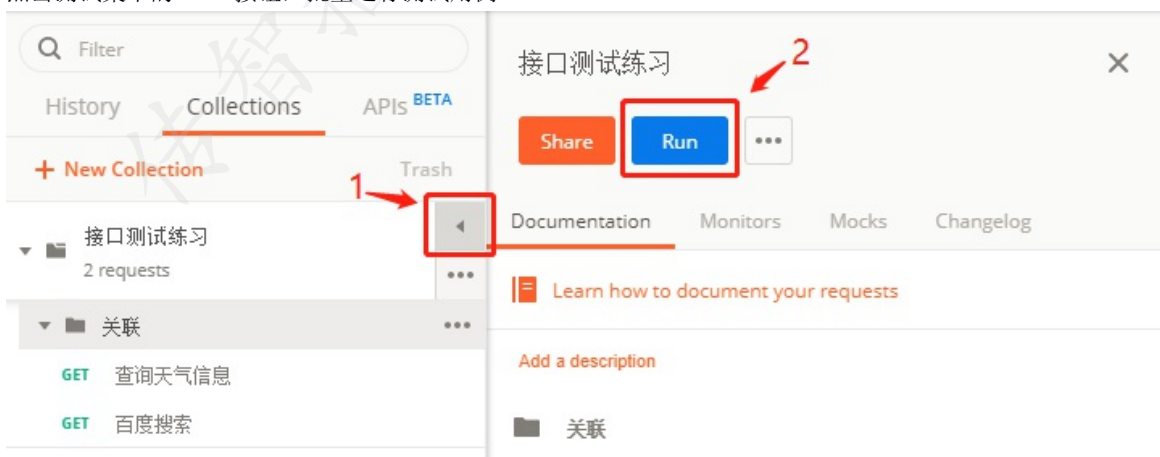


6. 批量执行测试用例

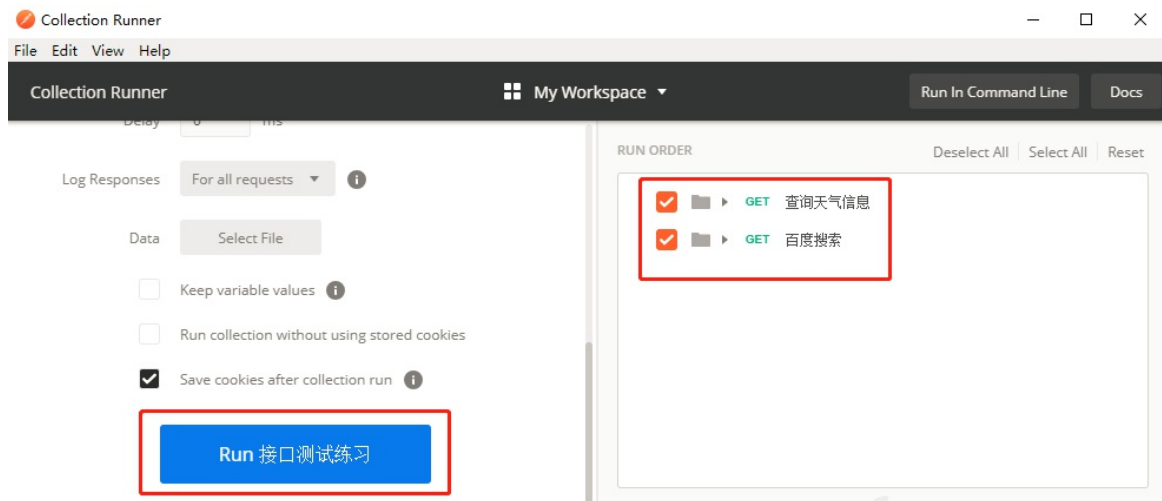
要想批量执行测试用例，需要先把接口请求添加到测试集中，通过运行测试集的方式就可以实现批量运行测试用例。

操作步骤

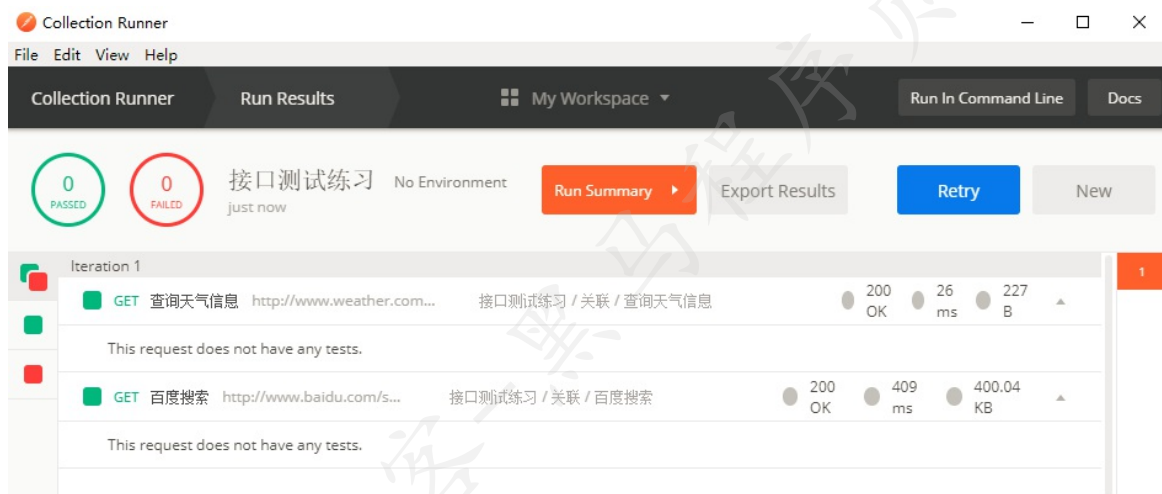
1. 点击测试集中的“Run”按钮，批量运行测试用例



2. 弹出Collection Runner窗口，点击运行按钮



3. 查看测试结果



7. 读取外部文件实现参数化

应用场景：针对于某个接口，如果有大量的测试数据需要批量验证，可以把测试数据保存到数据文件中，通过参数化的方式来实现。

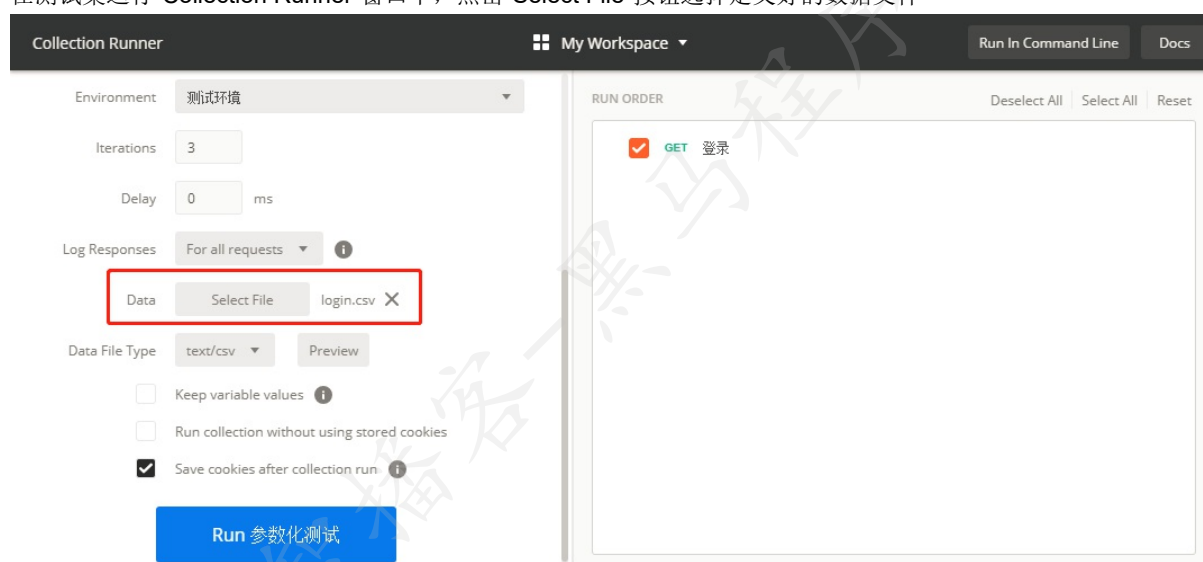
7.1 定义数据文件

常用的数据文件格式有：CSV、JSON



7.2 选择数据文件

在测试集运行“Collection Runner”窗口中，点击“Select File”按钮选择定义好的数据文件



7.3 获取数据文件中的参数

在不同的地方获取参数的方法是不同的，介绍两种获取参数的方法：

- 请求参数可以通过`{{var_name}}`来引用
- 断言中的数据，由于是在Tests中编写脚本，并不能直接通过`{{var_name}}`来引用，可以通过内置变量`data`来获取

```
// 获取数据文件中的用户名参数  
var username = data.username;
```

7.4 案例

- 需求：批量查询手机号归属地和所属运营商信息，并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

定义数据文件

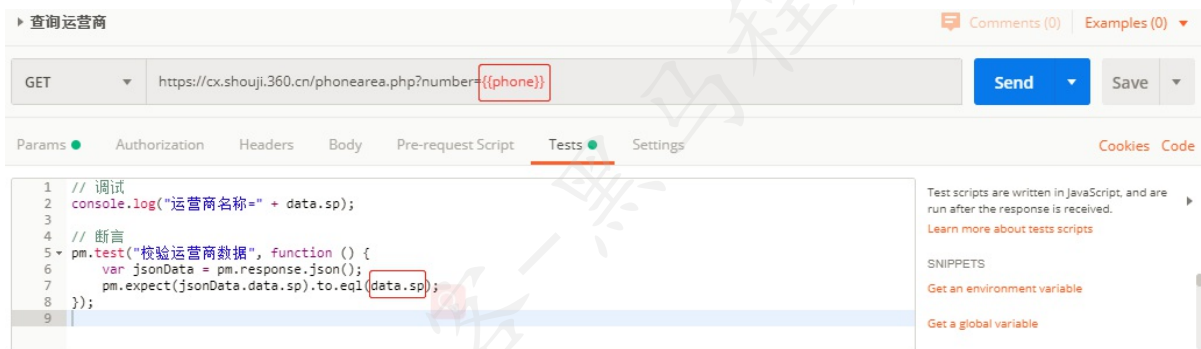
- CSV格式

```
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"}
]
```

实现步骤截图



Postman测试报告

目标

1. 掌握如何搭建Newman的相关环境
2. 掌握如何使用Newman生成HTML测试报告

1. Newman

Newman是一款基于nodejs开发的可以运行Postman脚本的工具，并可以生成测试报告

2. 环境准备

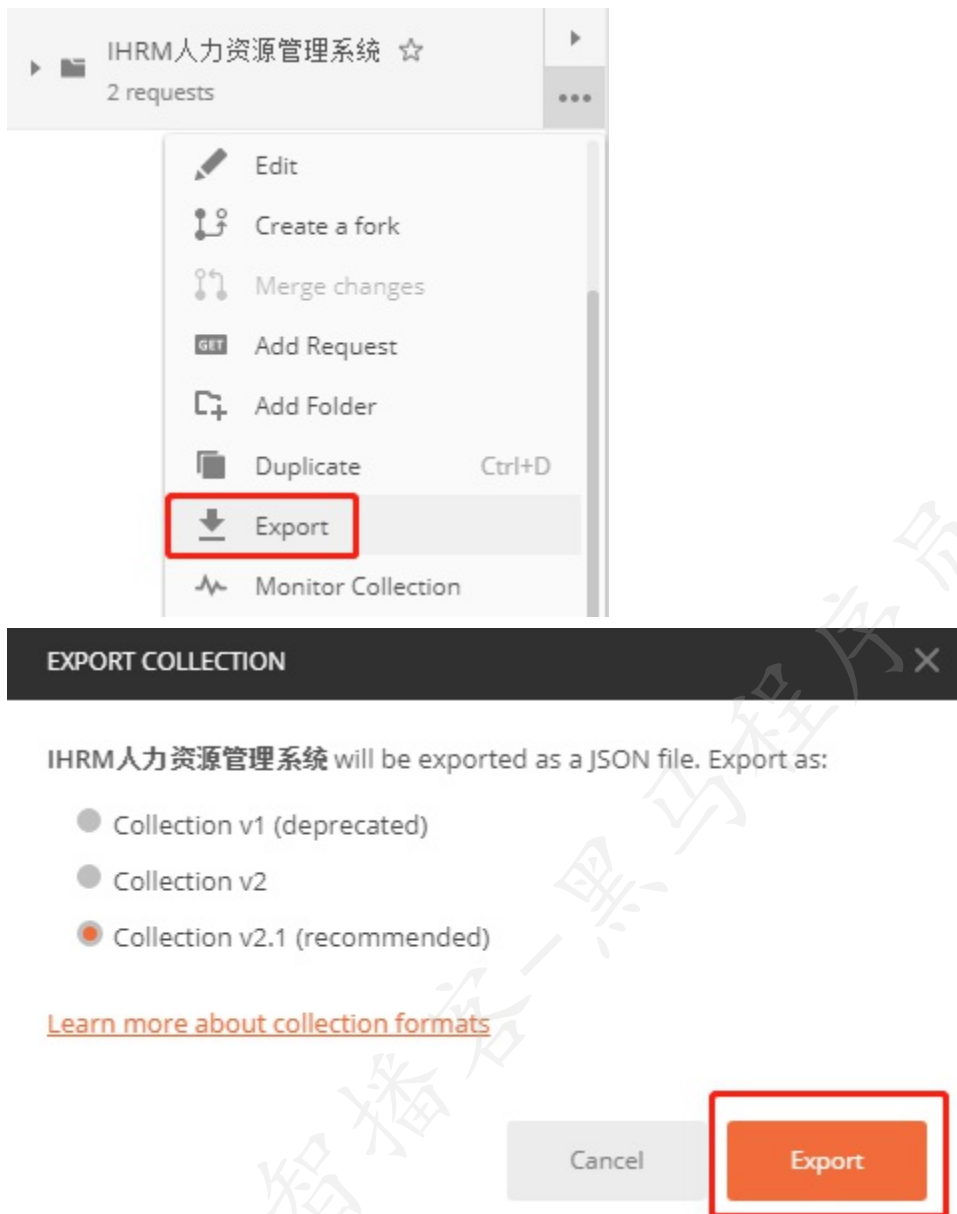
2.1 安装nodejs

- 下载地址: <http://nodejs.cn/download/>
- 如果下载的是安装包，下载后双击安装即可
- 如果下载的是压缩包（.zip），解压到某个目录后，需要配置环境变量
- 校验：打开cmd，输入 `node -v`，看到输出node的版本信息，即代表安装成功

2.2 安装newman

1. 安装newman
 - 打开cmd输入: `npm install -g newman`
 - 校验：安装完成后，输入 `newman -v` 命令查看版本信息，检测是否安装成功
2. 安装newman-reporter-html
 - 打开cmd输入: `npm install -g newman-reporter-html`

3. 导出测试集数据



4. 执行测试

使用newman命令，运行导出的测试集脚本，打开cmd输入：

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

eg:
newman run demo.postman_collection.json -r html
newman run demo.postman_collection.json -r html --reporter-html-export report.html
```

命令说明

- run xxx.json: 表示要执行的postman脚本，即导出的测试集数据
- -e source: 用来指定环境变量文件的路径
- -d source: 用来指定测试数据文件的路径

- **-r html**: 用来指定生成的数据结果的类型，如json、html等
- **--reporter-html-export source**: 指定html报告生成的路径；如果不设置该参数，则会在当前目录下生成一个名字为newman的文件夹

查看测试报告

命令运行完成后，打开生成的HTML文件即可看到测试报告内容

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	http://www.weather.com.cn/data/sk/101010100.html
Mean time per request	94ms
Mean size per request	227B
Total passed tests	0
Total failed tests	0
Status code	200

项目实战

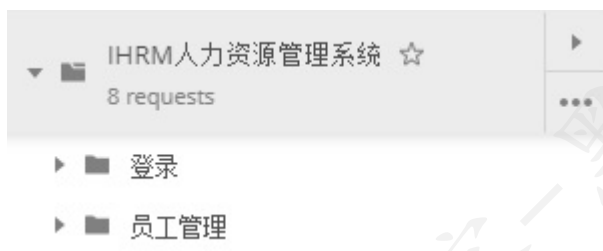
目标

1. 掌握如何使用Postman实现接口测试

1. 初始化工作

1.1 创建测试用例结构

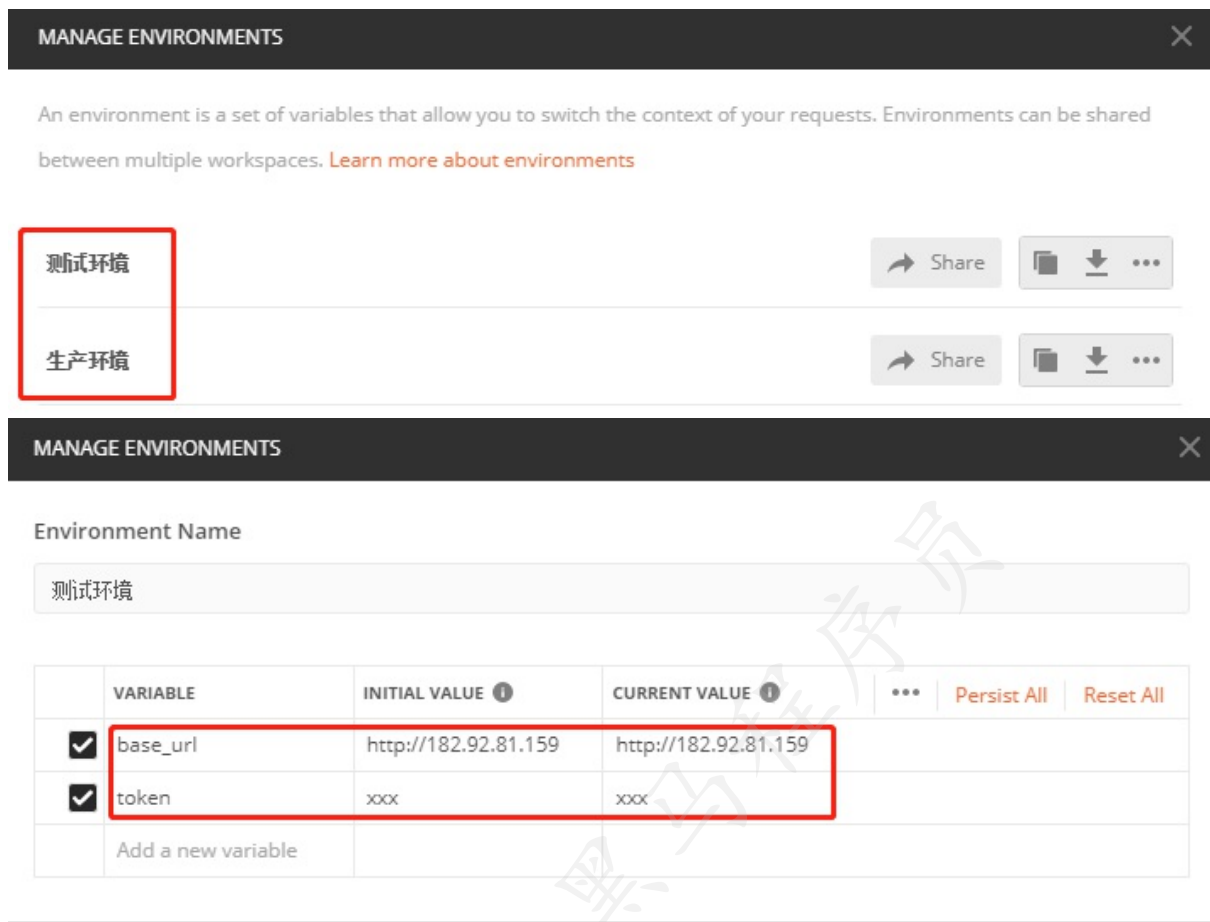
1. 创建测试集：IHRM人力资源管理系统
2. 创建目录
 - 登录模块
 - 员工管理模块



1.2 设置环境变量

在实际的工作中，我们既要对测试环境进行测试，也要对生产环境进行测试。

- 测试环境
 - `base_url=http://ihrm-test.itheima.net`
 - `token=xxx`
- 生产环境
 - `base_url=http://www.myhrm.com`
 - `token=xxx`



2. 实现测试用例

根据编写的测试用例文档，使用Postman实现测试用例

2.1 登录-登录成功

操作步骤

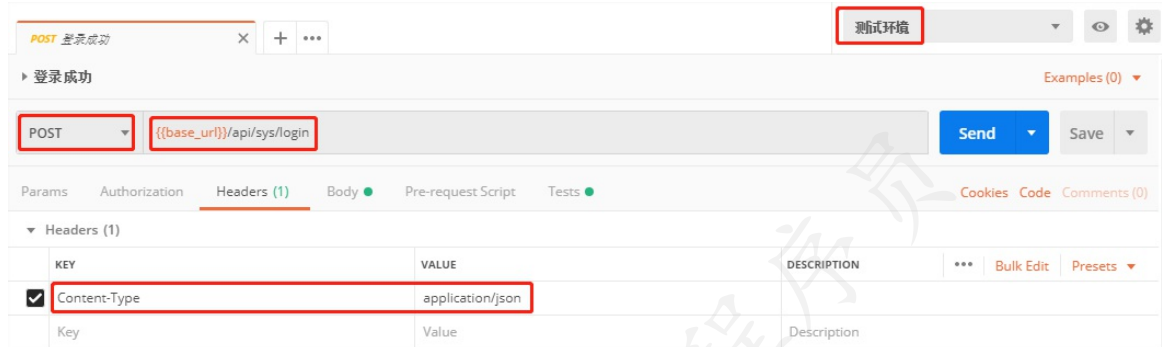
1. 在‘登录’目录下，添加‘登录成功’的请求
2. 填写请求数据：请求方式、请求URL、请求头、请求体
3. 在‘Tests’标签页中，编写测试脚本：断言、业务数据处理
4. 发送请求，调试脚本

实现

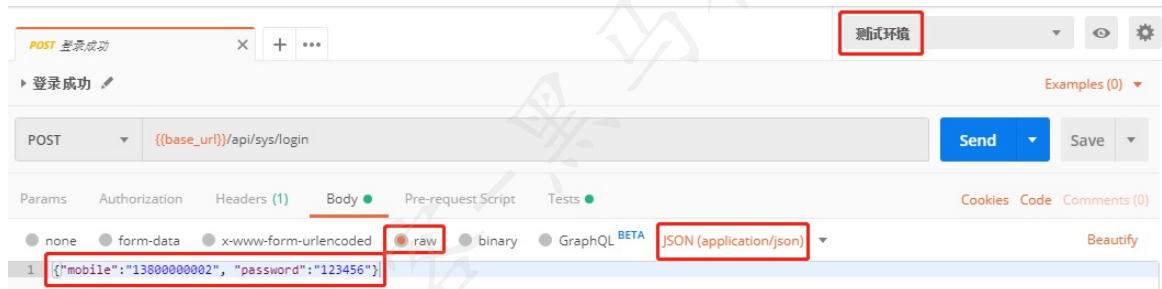
1. 添加‘登录成功’的请求



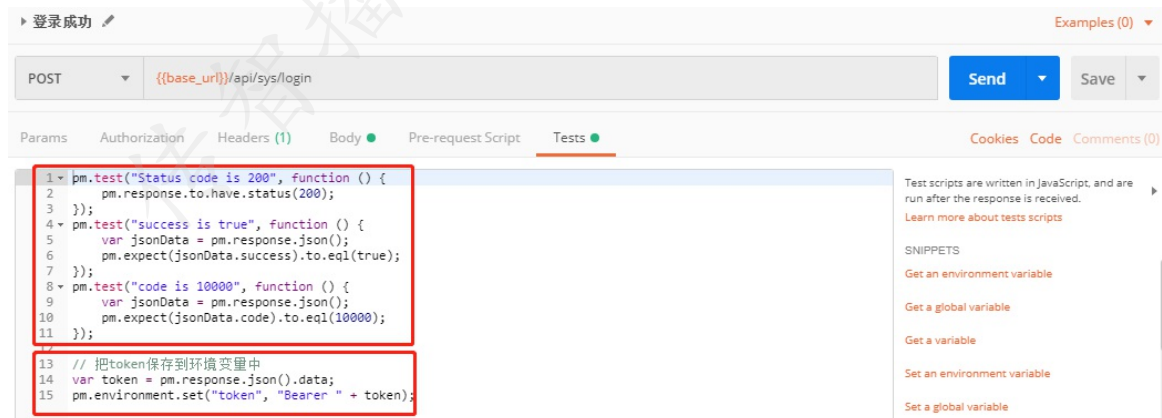
2. 填写请求数据：请求方式、请求URL



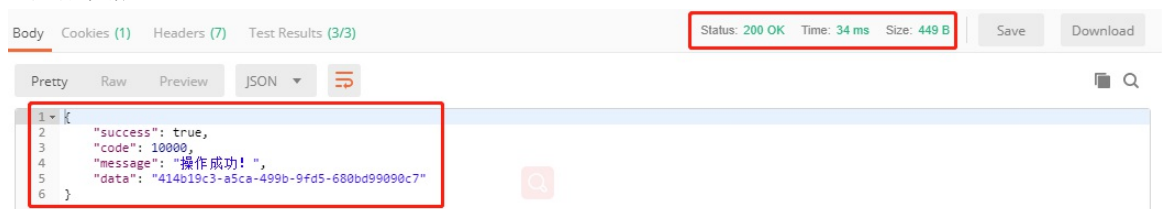
3. 填写请求数据：请求头



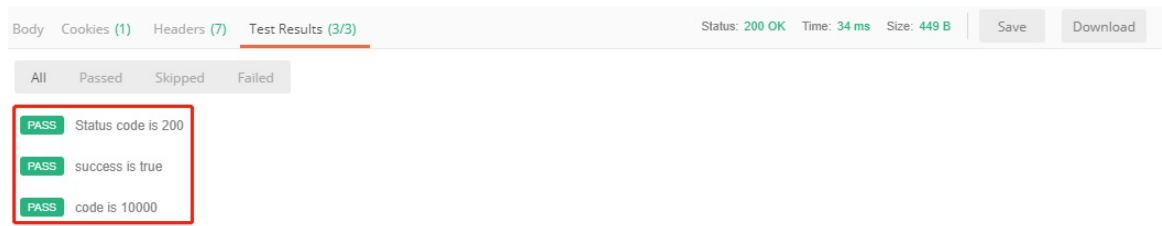
4. 编写测试脚本



5. 响应体数据



6. 断言结果



3. 生成测试报告

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

操作步骤

1. 导出测试集数据
2. 导出环境变量数据
3. 执行命令：

```
newman run ihrm.postman_collection.json -e dev.postman_environment.json -r html --reporter-html-export report.html
```