

IHRM项目实战——基于Postman

项目目标

- 掌握Postman如何实现关联
- 掌握如何为项目创建测试用例结构
- 掌握如何为项目设置环境变量
- 掌握在项目中如何对接口响应数据进行断言
- 掌握在项目中如何实现接口关联
- 掌握在项目中如何使用Newman生成HTML测试报告
- 掌握如何使用Postman实现接口测试
- 掌握接口测试流程

业务测试前置技术——Postman关联

1 学习目标

1. 掌握Postman如何实现关联

2 为什么需要关联?

当请求之间存在依赖关系，比如一个请求的输入参数是另一个请求返回的数据，这时候就需要用到关联处理。

例如：IHRM添加员工接口，前置条件为【已登录】状态。

- (1) IHRM登录接口
- (2) IHRM添加员工接口

3 如何实现关联?

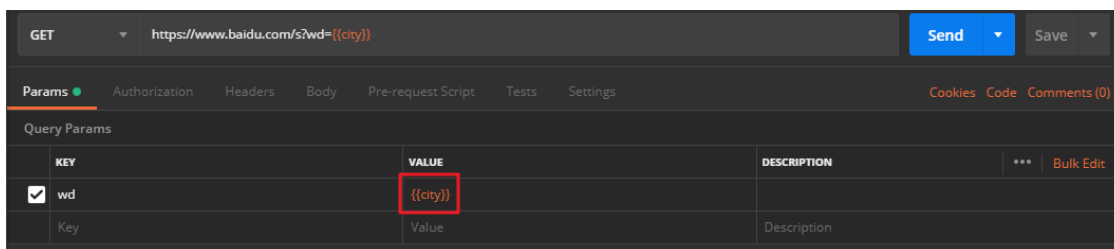
- 通过【Tests】标签编写Javascript脚本，获取被依赖请求的响应结果数据

```
// 获取json格式响应数据
var jsonData = pm.response.json();
console.info(jsonData);
```

- 通过全局变量在请求之间传递数据

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

- 依赖请求直接获取全局变量的值作为输入数据



4 实战案例

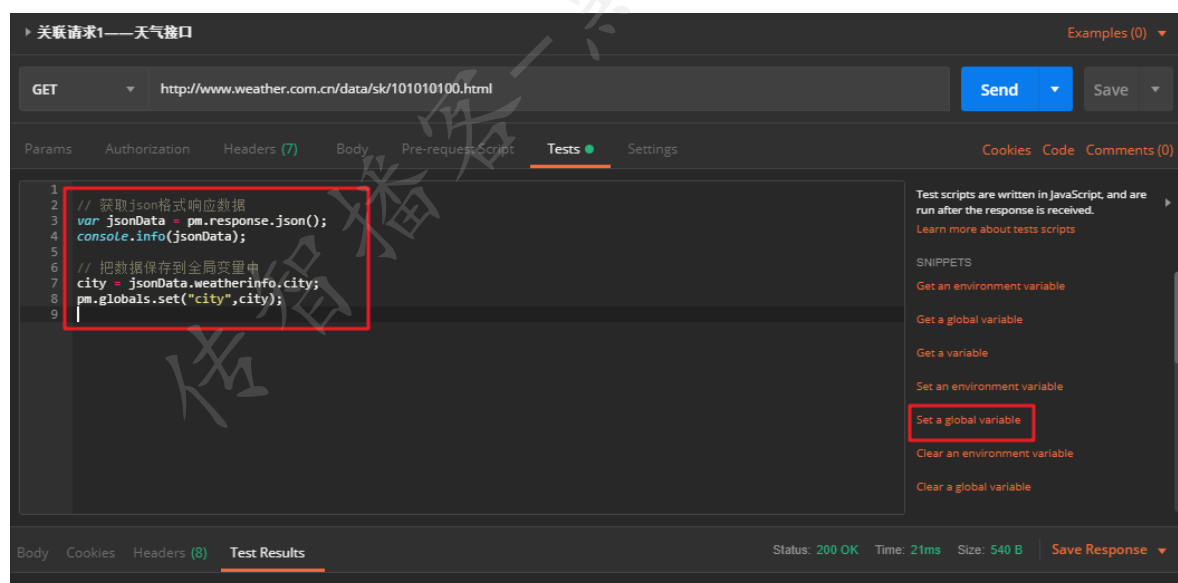
需求:

1. 请求获取天气接口, <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>, 把获取到的城市名称作为请求参数

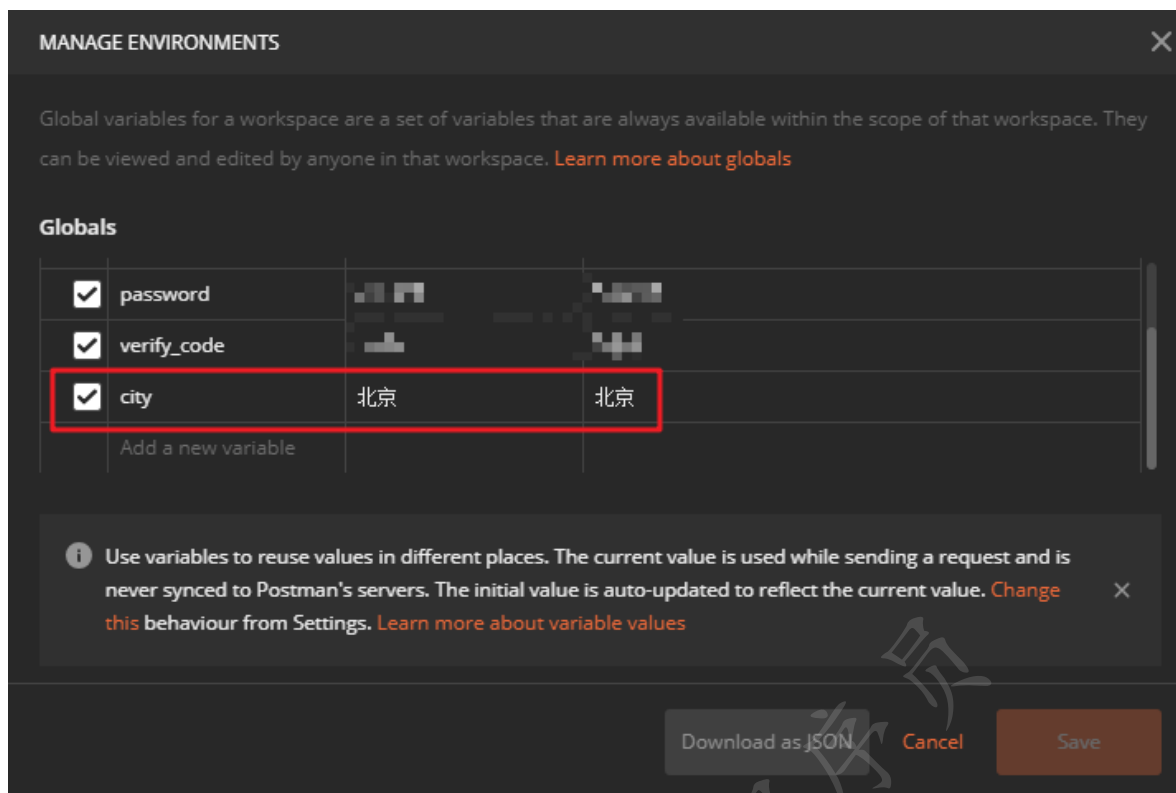
需求分析:

1. 新建一个获取天气信息接口:
 - (1) 请求方式=`get`
 - (2) 请求地址=<http://www.weather.com.cn/data/sk/101010100.html>
2. 在天气接口请求【Tests】标签中设计测试脚本存储返回数据中的【城市】信息到系统全局变量`city`中
3. 新建一个百度接口请求:
 - (1) 请求方式=`get`
 - (2) 请求地址=<http://www.baidu.com/s>
 - (3) 请求参数`wd={{city}}`

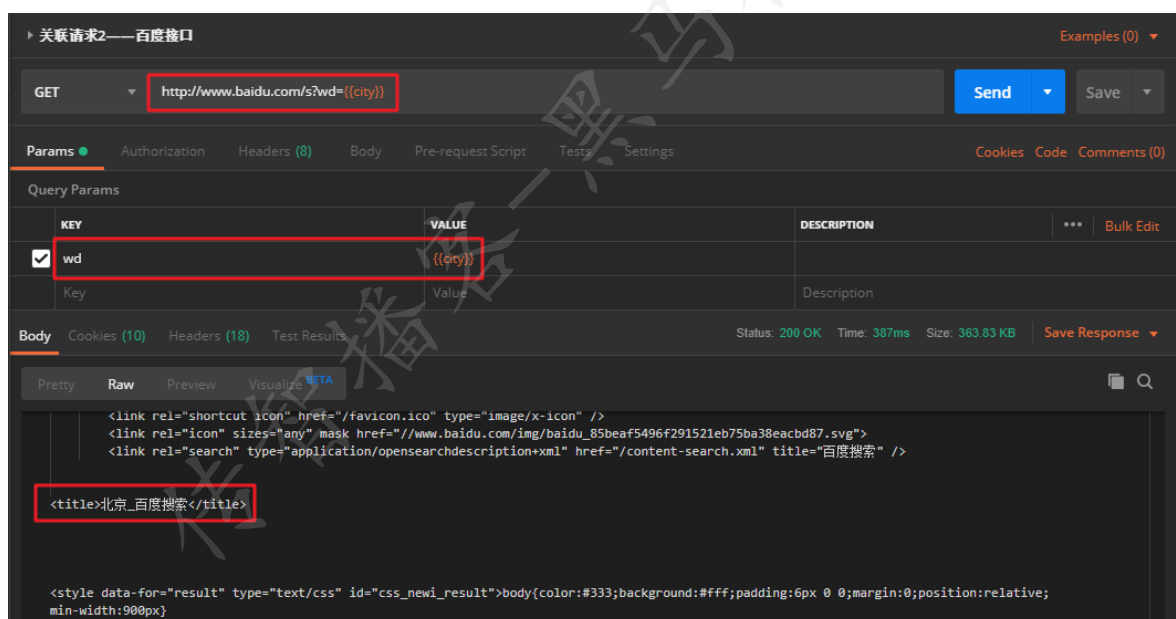
4.1 构建天气接口请求



4.2 存储天气接口返回数据到系统全局变量



4.3 构建百度接口请求

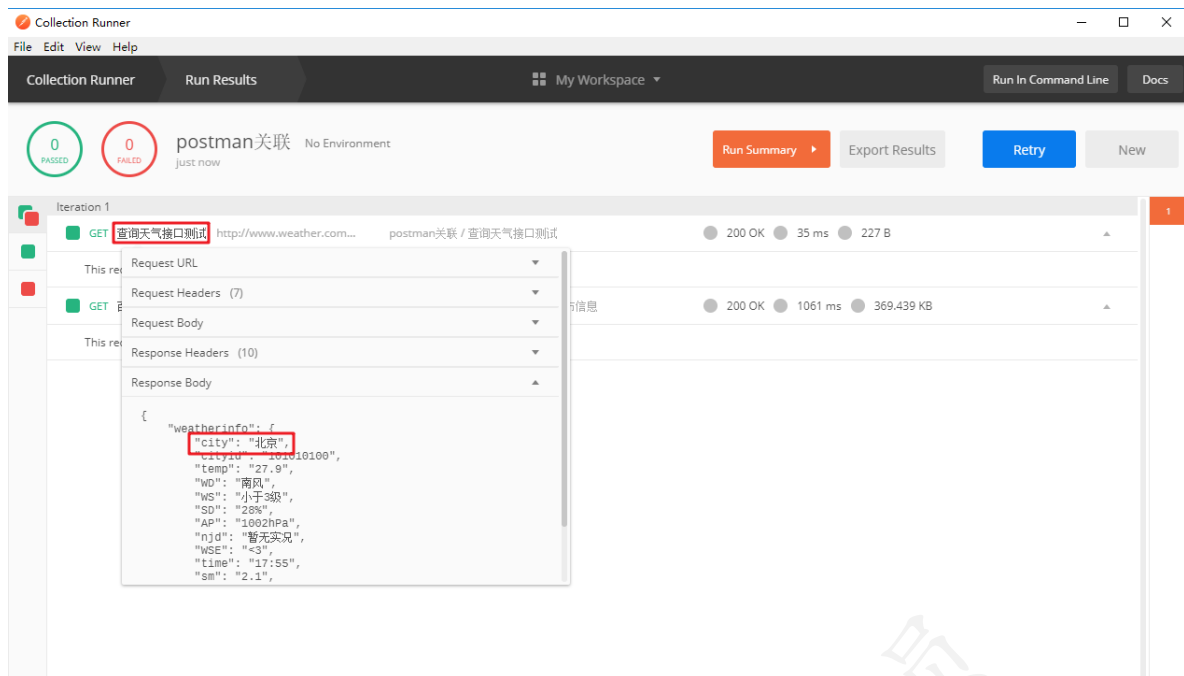


5 目标验收

练习:

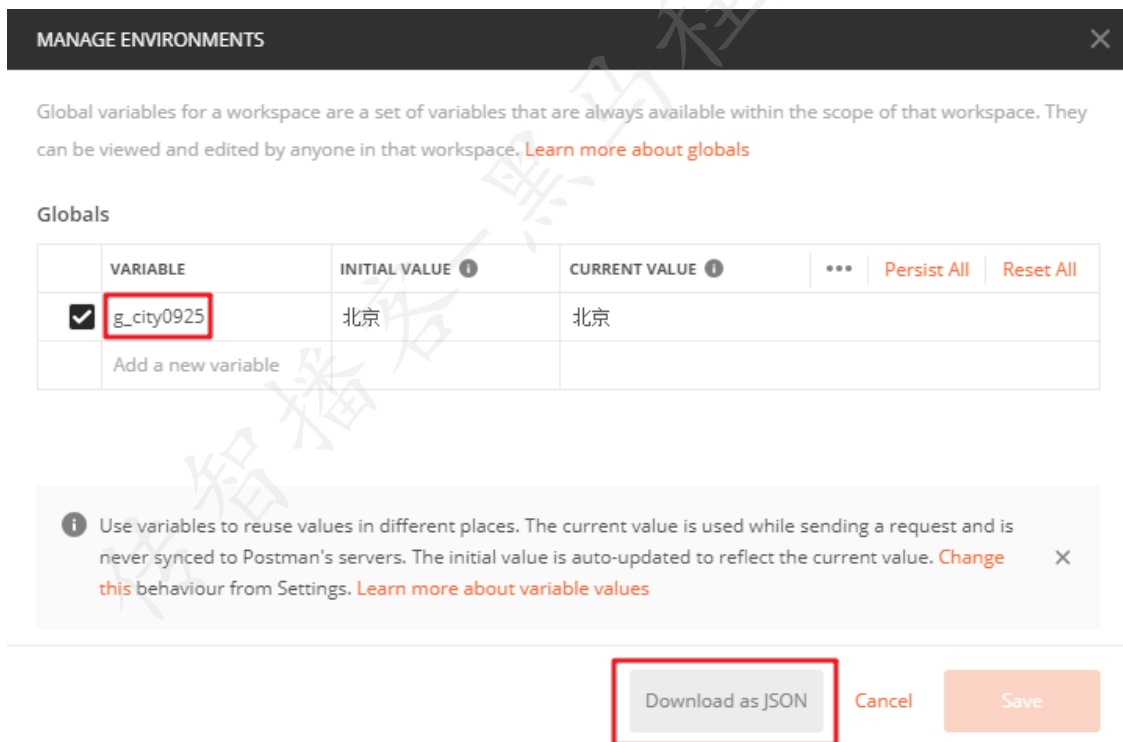
1. 请求获取天气接口, <http://www.weather.com.cn/data/sk/101010100.html>
2. 获取返回结果中的城市名称
3. 调用百度搜索接口: <http://www.baidu.com/s>, 把获取到的城市名称作为请求参数

5.1 Postman批量执行

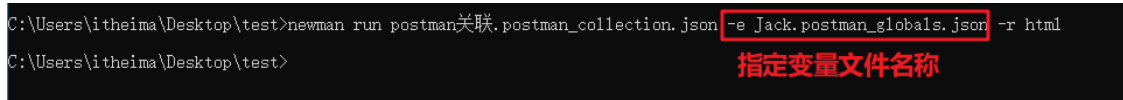


5.2 Newman命令行执行

- 第1步：导出postman脚本与环境变量



- 第2步：命令行执行



- 第3步：查看运行报告

Newman Report

Collection
Description
Time
Exported with

postman关联
postman关联知识点讲解
Wed Sep 25 2019 15:51:27 GMT+0800 (GMT+08:00)
Newman v4.5.4

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

Total run duration
Total data received
Average response time

1223ms
360.81KB (approx)
523ms

Total Failures 0

Requests

查询天气接口测试

Description
Method
URL
Mean time per request

获取天气信息，并存储返回数据的地区信息到全局变量
GET
<http://www.weather.com.cn/data/sk/101010100.html>
163ms

6 实际应用

IHRM添加员工接口，前置条件为【已登录】状态。

- (1) IHRM登录接口
- (2) IHRM添加员工接口



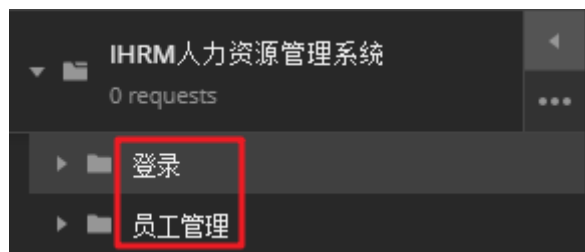
基于Postman工具实现接口自动化

1 初始化工作

1.1 创建测试用例结构

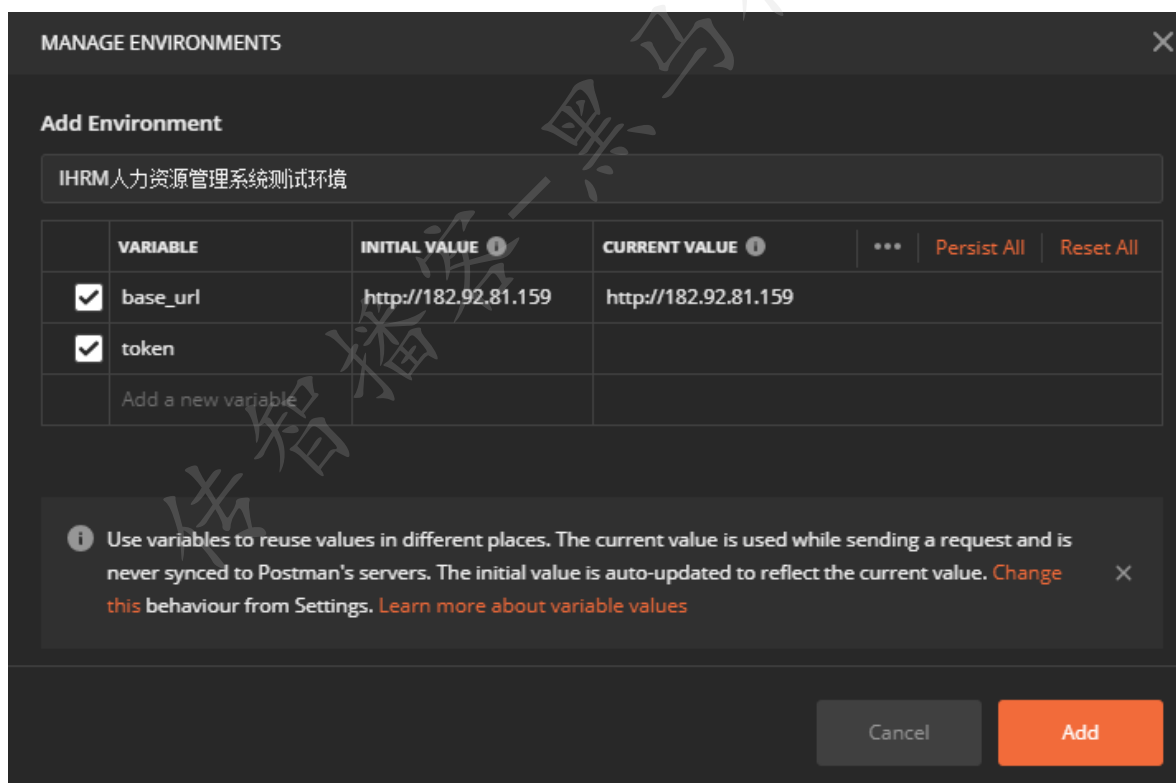
- 创建测试集：IHRM人力资源管理系统
- 创建目录

- 登录模块
- 员工管理模块



1.2 设置环境变量

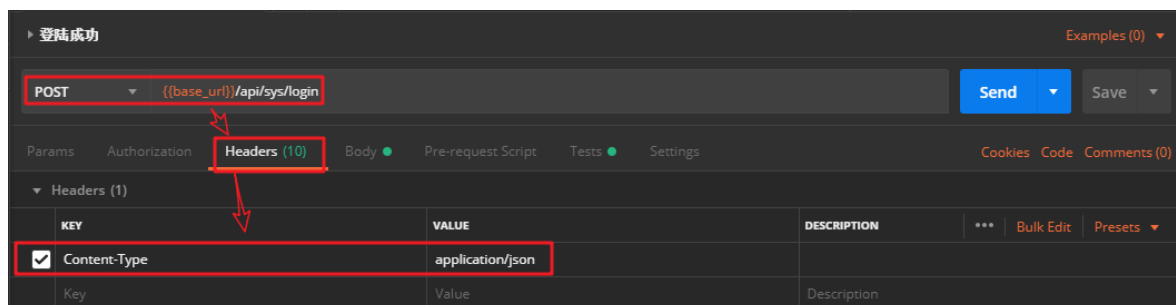
- 开发环境
 - base_url=<http://www.ihrm.dev/>
 - token=xxx
- 测试环境
 - base_url=<http://182.92.81.159>
 - token=xxx
- 生产环境
 - base_url=<http://www.ihrm.prod/>
 - token=xxx



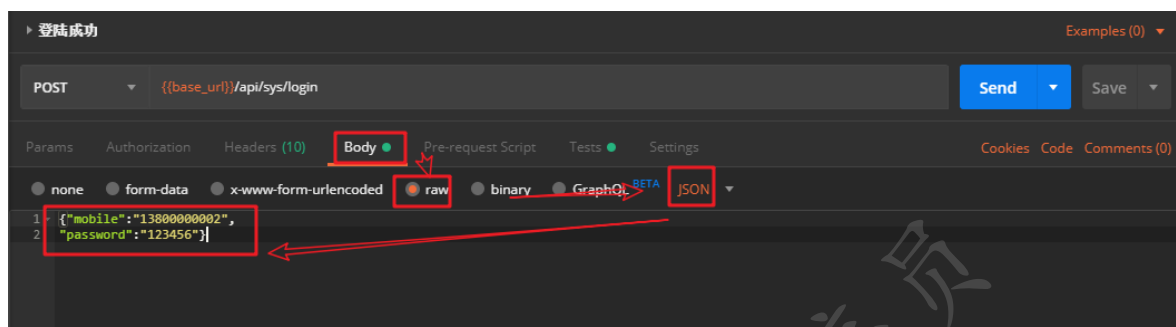
2 实现测试用例

2.1 登录-登录成功

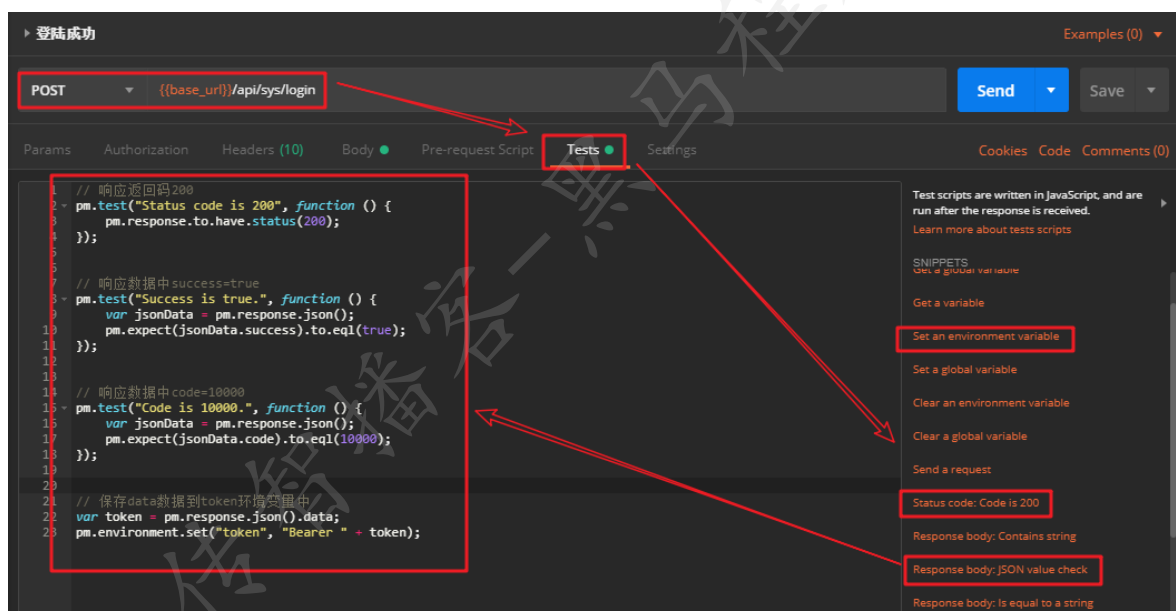
2.1.1 设置Headers



2.1.2 设置Body



2.1.3 编写测试脚本



```
// 响应返回码200
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

// 响应数据中success=true
pm.test("Success is true.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.success).to.eql(true);
});

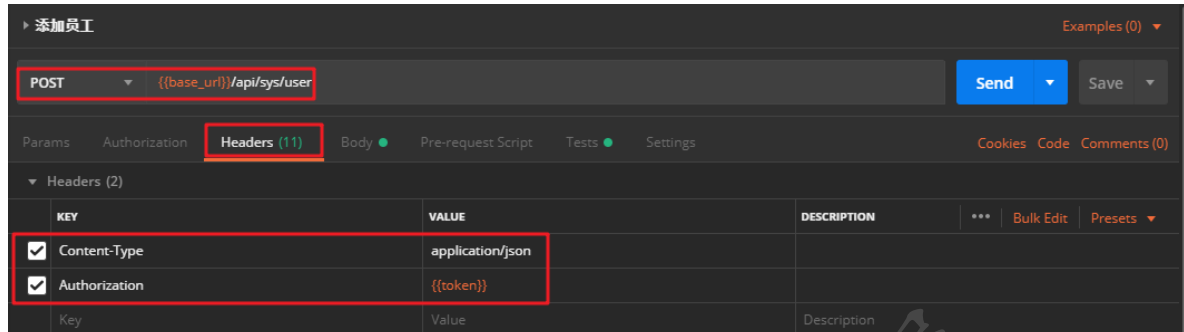
// 响应数据中code=10000
pm.test("Code is 10000.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.code).to.eql(10000);
});

// 保存data数据到token环境变量中
var token = pm.response.json().data;
pm.environment.set("token", "Bearer " + token);
```

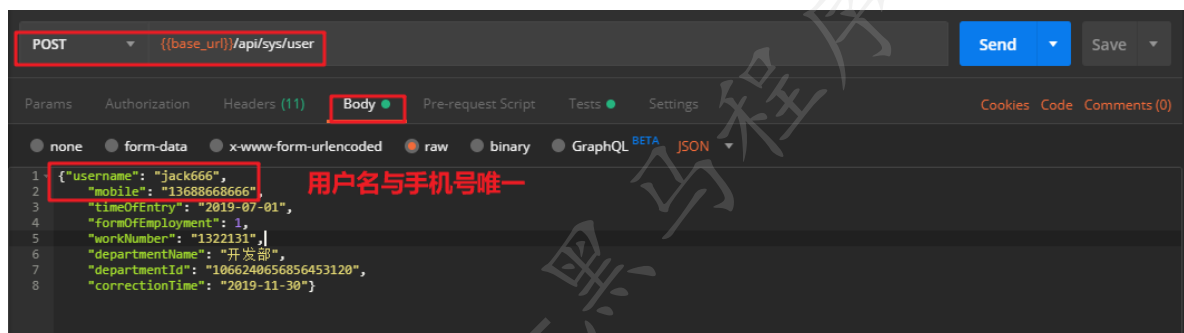
```
// 保存data数据到token环境变量中
var token = pm.response.json().data;
pm.environment.set("token", "Bearer " + token);
```

2.2 员工管理-添加员工

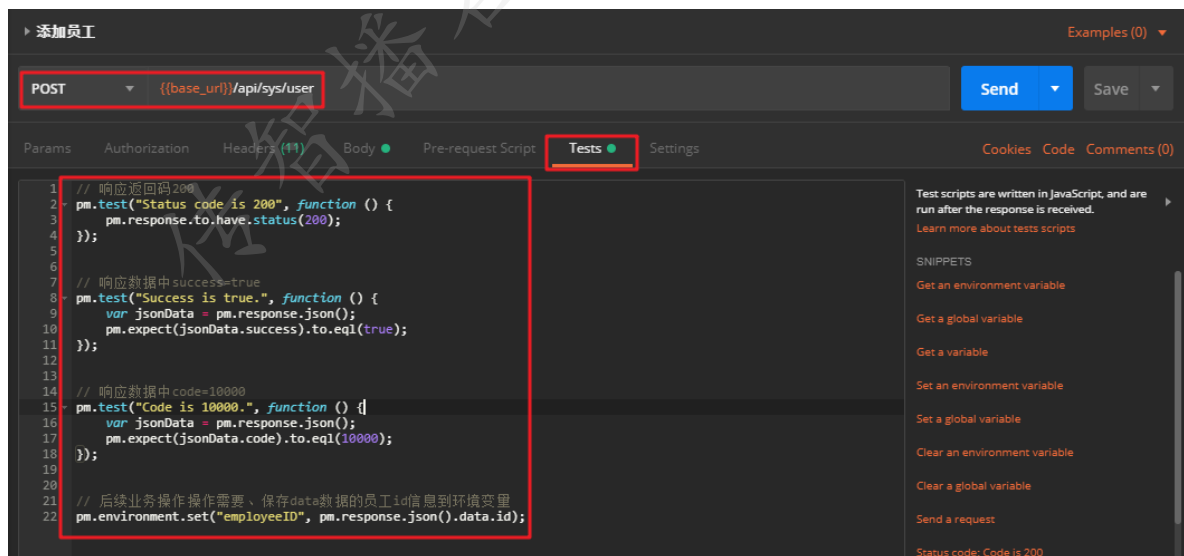
2.2.1 设置Headers



2.2.2 设置Body



2.2.3 编写测试脚本



```
// 响应返回码200
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});

// 响应数据中success=true
pm.test("Success is true.", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.success).to.eql(true);
});
```



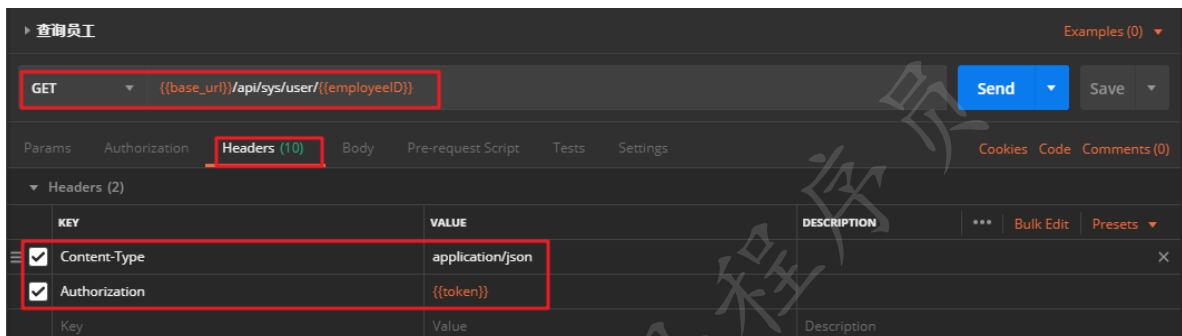
```
});

// 响应数据中code=10000
pm.test("Code is 10000.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.code).to.eql(10000);
});

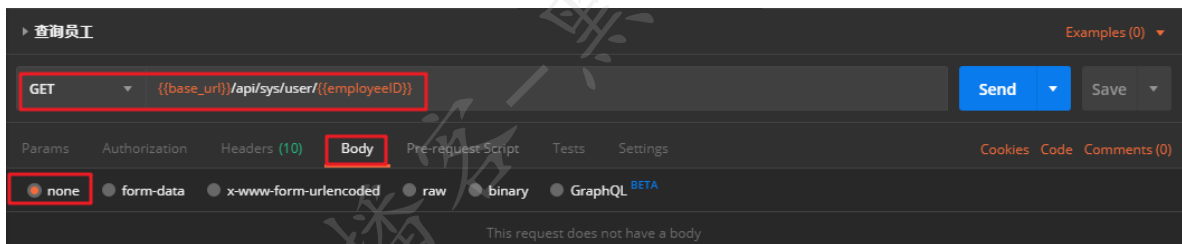
// 后续业务操作需要、保存data数据的员工id信息到环境变量
pm.environment.set("employeeID", pm.response.json().data.id);
```

2.3 员工管理-查询员工

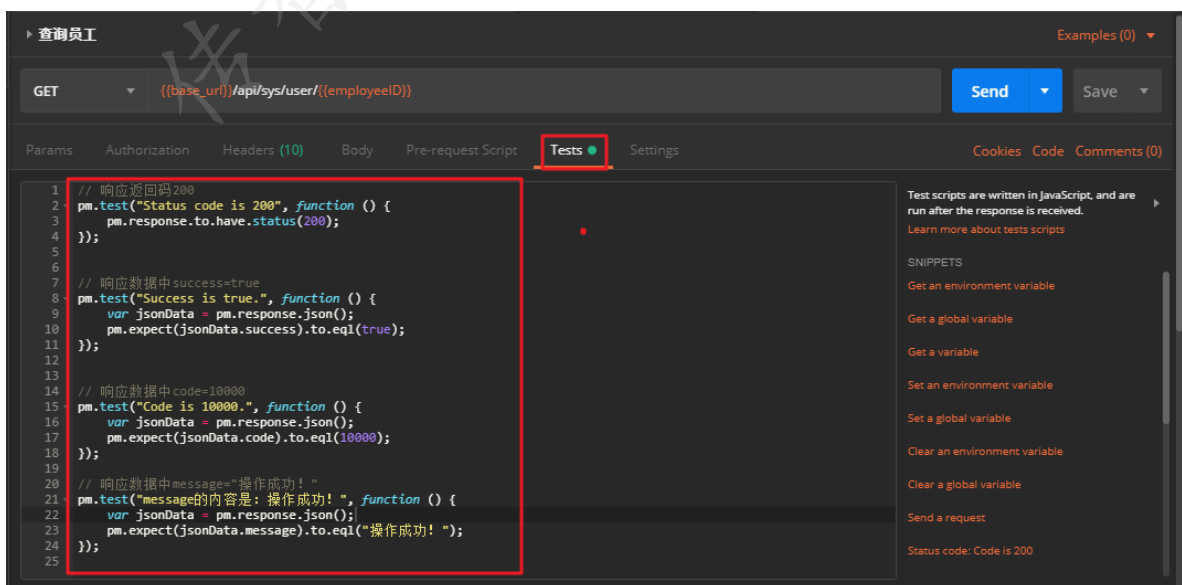
2.3.1 设置Headers



2.3.2 设置Body



2.3.3 编写测试脚本



```
// 响应返回码200
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
```

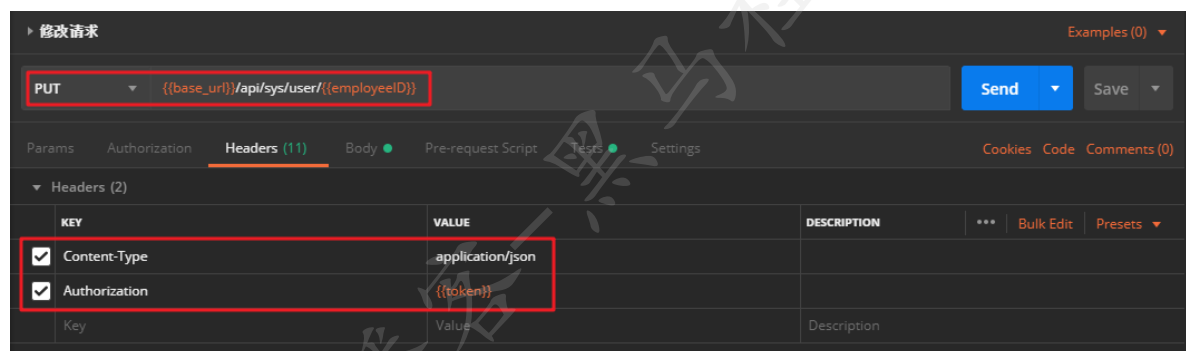
```
// 响应数据中success=true
pm.test("Success is true.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.success).to.eql(true);
});

// 响应数据中code=10000
pm.test("Code is 10000.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.code).to.eql(10000);
});

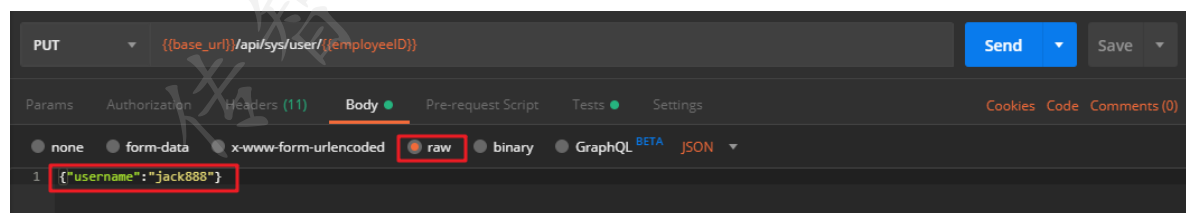
// 响应数据中message="操作成功！"
pm.test("message的内容是：操作成功！", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.message).to.eql("操作成功！");
});
```

2.4 员工管理-修改员工

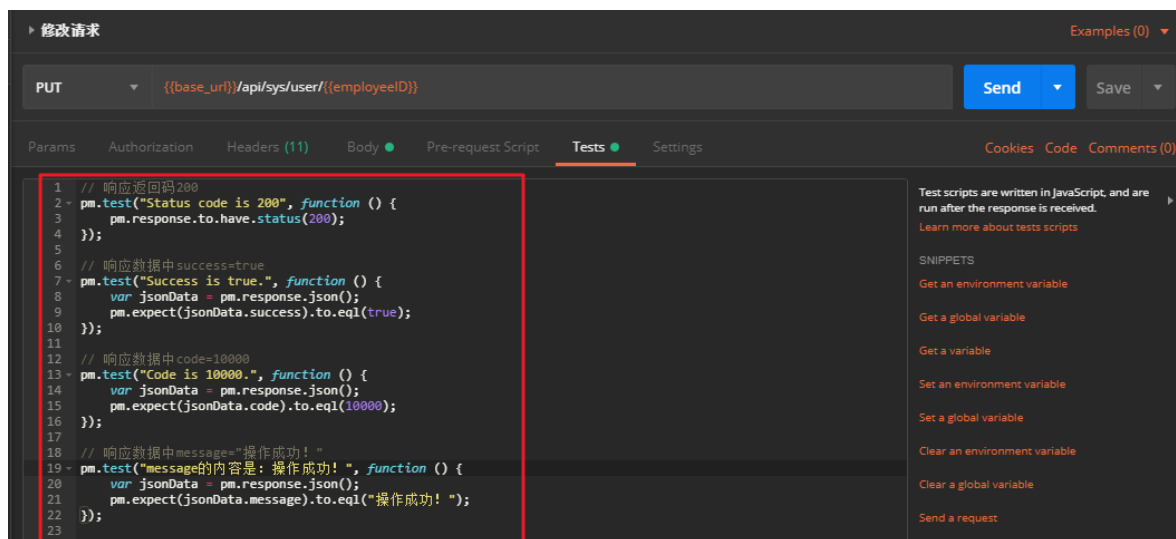
2.4.1 设置Headers



2.4.2 设置Body



2.4.3 编写测试脚本



```
// 响应返回码200
pm.test("Status code is 200", function () {
  pm.response.to.have.status(200);
});

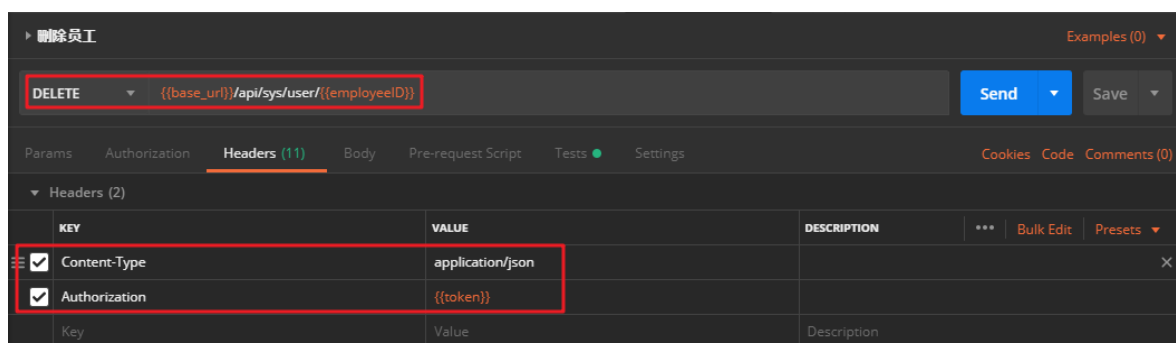
// 响应数据中success=true
pm.test("Success is true.", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.success).to.eql(true);
});

// 响应数据中code=10000
pm.test("Code is 10000.", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.code).to.eql(10000);
});

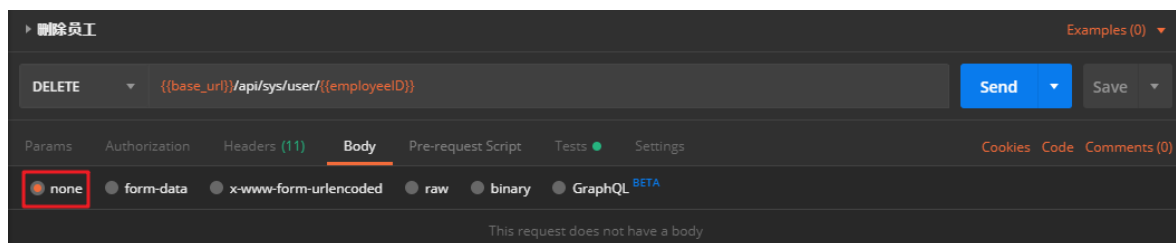
// 响应数据中message="操作成功!"
pm.test("message的内容是: 操作成功!", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.message).to.eql("操作成功!");
});
```

2.5 员工管理-删除员工

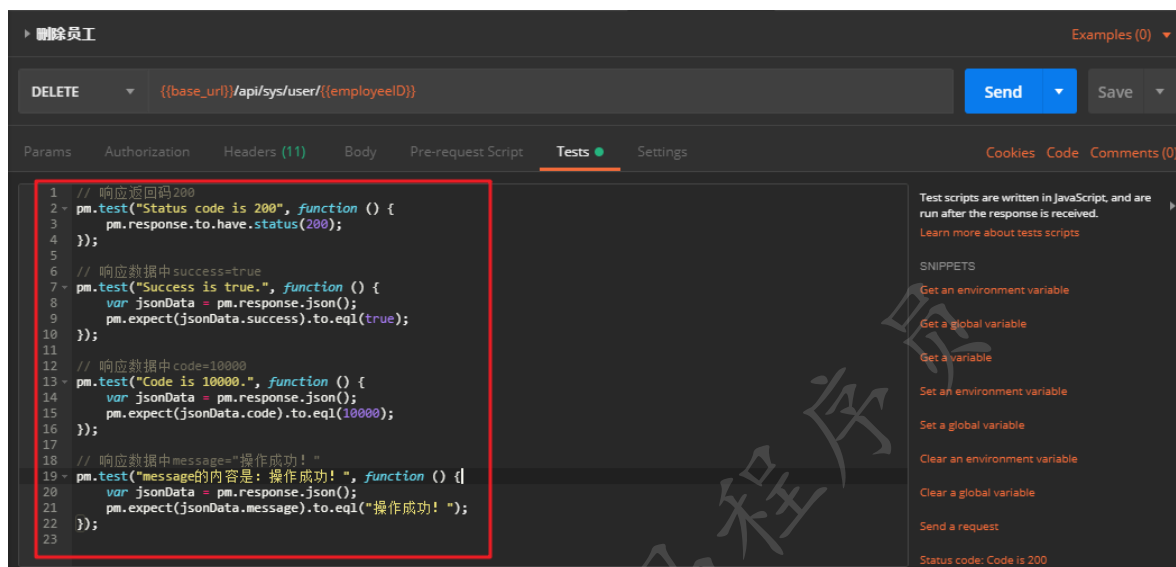
2.5.1 设置Headers



2.5.2 设置Body



2.5.3 编写测试脚本



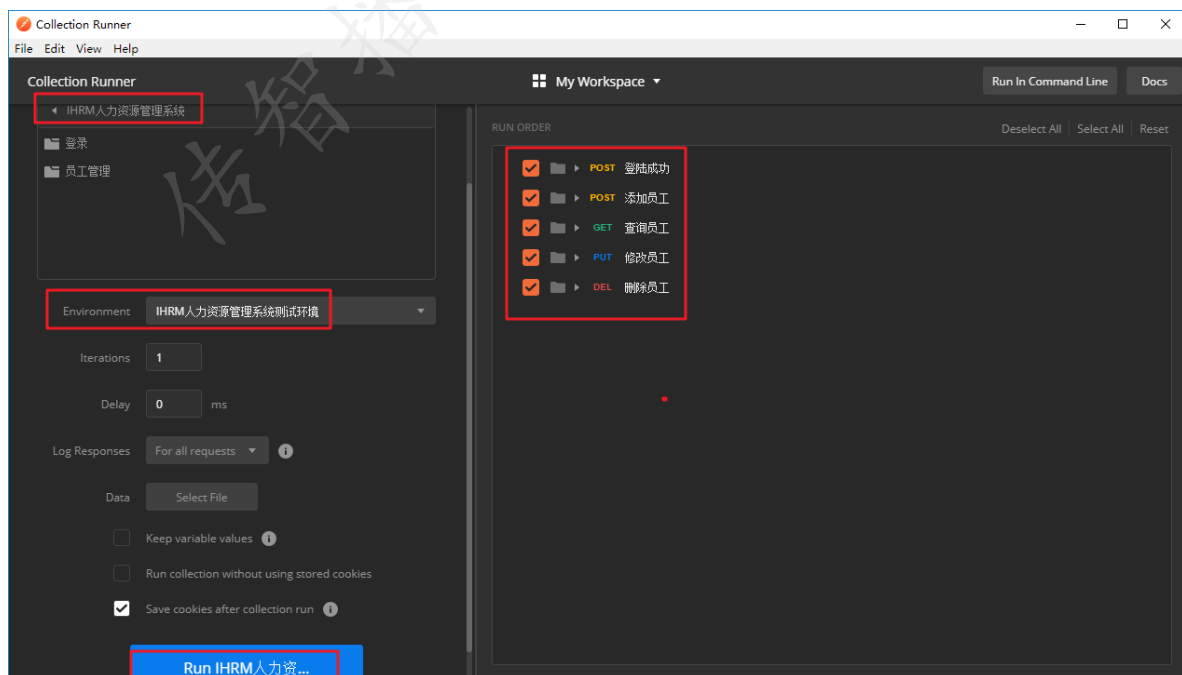
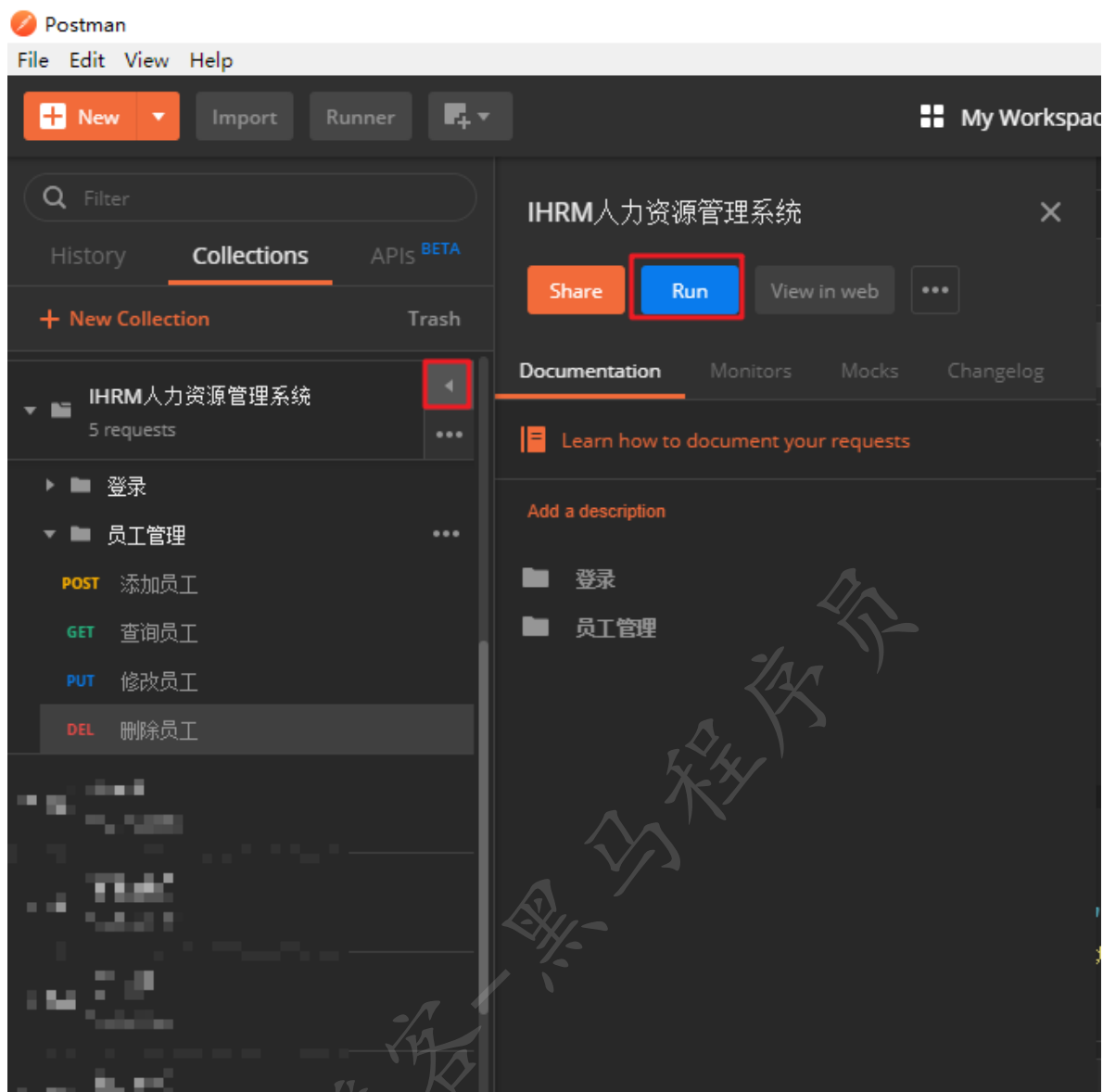
```
// 响应返回码200
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

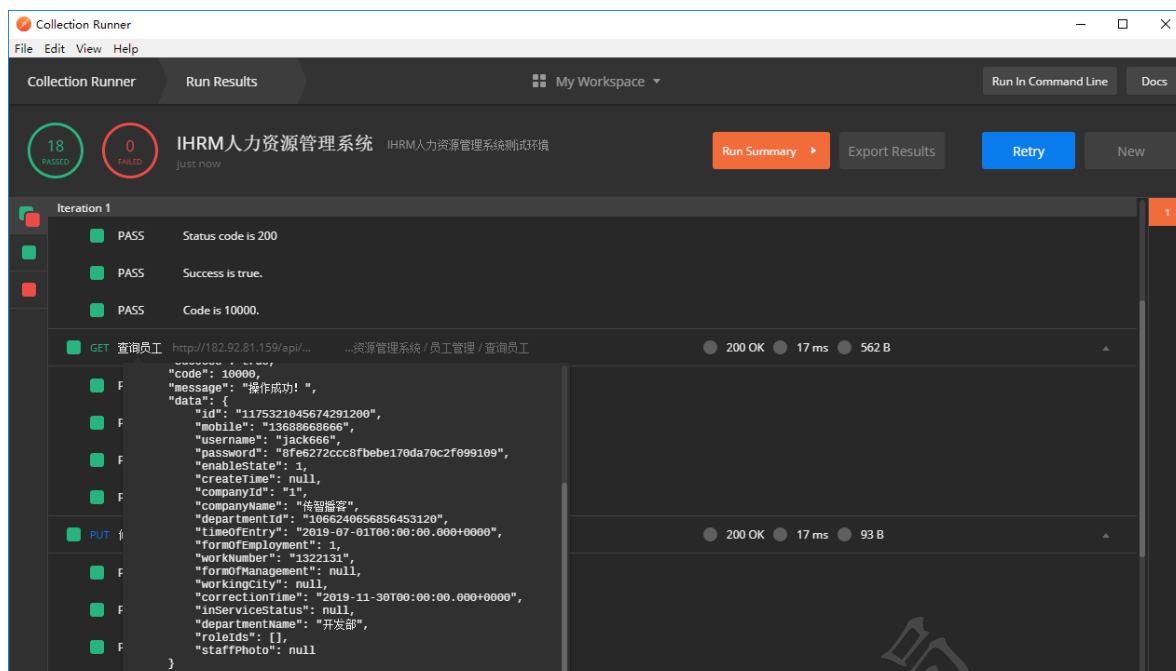
// 响应数据中success=true
pm.test("Success is true.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.success).to.eql(true);
});

// 响应数据中code=10000
pm.test("Code is 10000.", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.code).to.eql(10000);
});

// 响应数据中message="操作成功！"
pm.test("message的内容是：操作成功！", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.message).to.eql("操作成功！");
});
```

3 自动化测试



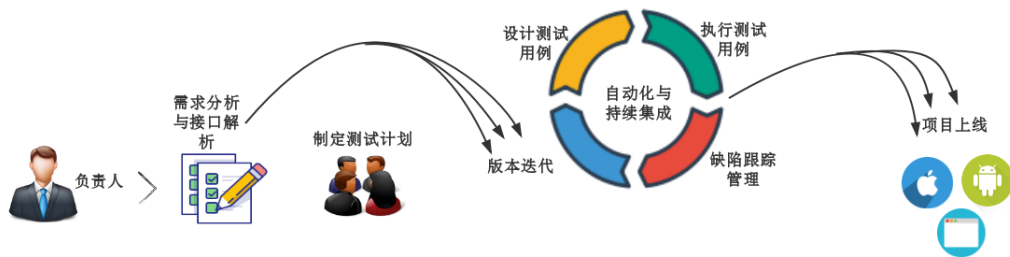


IHRM项目实战

项目启动



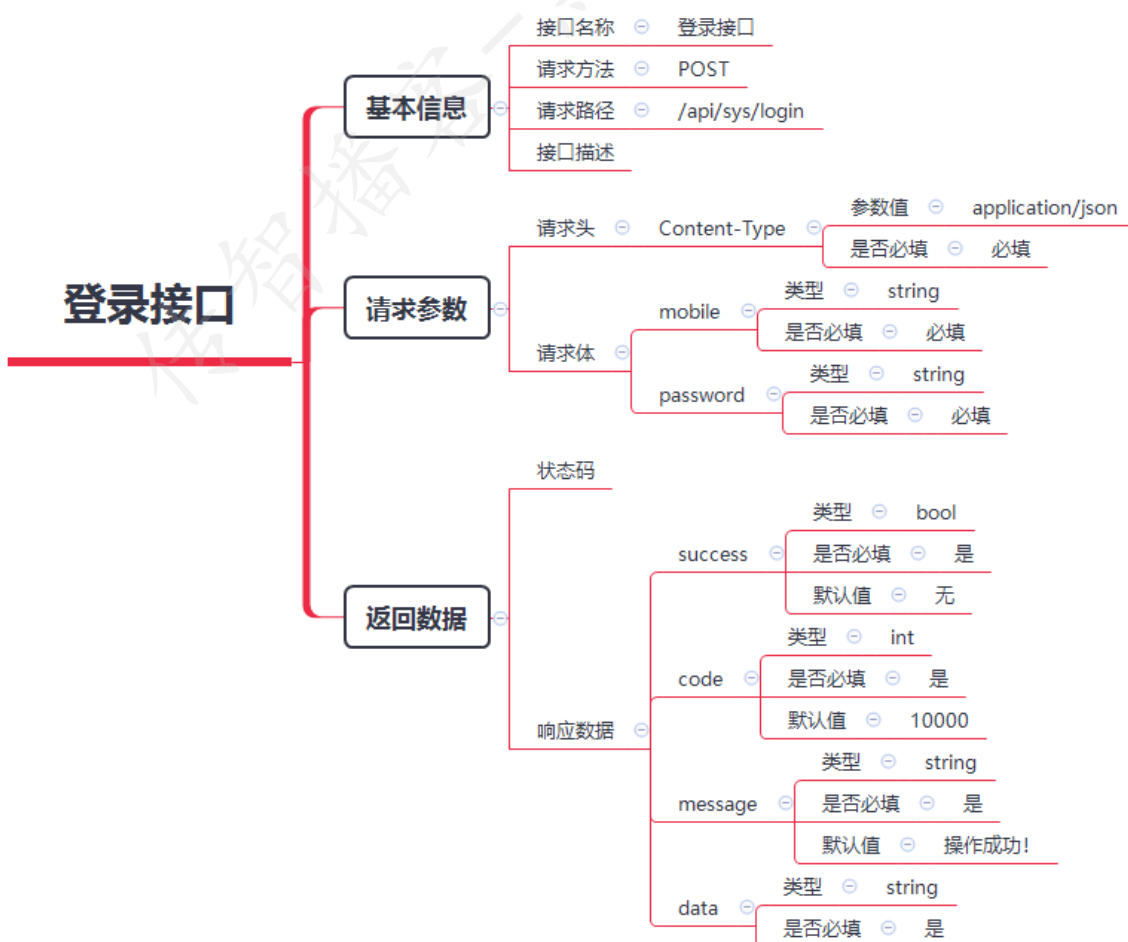
接口测试流程



- 需求分析与 **接口文档** 解析
- 制定 **接口** 测试计划
- 设计 **接口** 测试用例与评审
- 执行 **接口** 测试用例与缺陷跟踪管理
- 编写 **接口** 测试报告

1 接口文档 解析

1.1 登录



1.2 添加员工

添加员工接口测试设计

正常情况

成功添加员工

前置条件 登录

请求方式 POST

请求地址 /api/sys/user

```
{
  "username": "tom",
  "mobile": "13012340001",
  "timeOfEntry": "2019-07-01",
  "formOfEmployment": 1,
  "workNumber": "1322131",
  "departmentName": "开发部",
  "departmentId": "1066240656856453120",
  "correctionTime": "2019-11-30"
}
```

请求数据

```
{
  "success": true,
  "code": 10000,
  "message": "操作成功!",
  "data": {
    "id": "1174155068198678528"
  }
}
```

响应结果

异常情况

必填项校验

姓名 不输入

手机 不输入

工号 不输入

入职时间 正确输入

聘用形式

部门 正确输入

管理形式

工作城市 正确输入

转正时间

可选参数校验

内容 姓名不存在

姓名 类型

长度

内容 手机已存在

手机 类型

长度

内容 工号已存在

工号 类型

长度

内容

入职时间 类型

长度

内容

聘用形式 类型

长度

内容

部门 类型

长度

内容

管理形式 类型

长度

内容

工作城市 类型

长度

内容

转正时间 类型

长度

参数异常校验

1.3 查询员工

查询员工

正常情况

获取员工信息成功

前置条件 登录

请求方式 GET

请求地址 /api/sys/user/target 目标值, 如本查询接口中的ID"1175223466613600256"

```
{
  "success": true,
  "code": 10000,
  "message": "操作成功!",
  "data": {
    "id": "1175223466613600256",
    "mobile": "13488888887",
    "username": "jack33",
    "password": "d49a5fcfad8be6cdfa3bbe3fe7e14aa2",
    "enableState": 1,
    "createTime": null,
    "companyId": "1",
    "companyName": "传智播客",
    "departmentId": "1066240656856453120",
    "timeOfEntry": "2019-07-01T00:00:00.000+0000",
    "formOfEmployment": 1,
    "workNumber": "1322131",
    "formOfManagement": null,
    "workingCity": null,
    "correctionTime": "2019-11-30T00:00:00.000+0000",
    "inServiceStatus": null,
    "departmentName": "开发部",
    "roleIds": [],
    "staffPhoto": null
  }
}
```

响应结果

请求参数 ID不存在"1175223466613600257"

```
{
  "success": false,
  "code": 99999,
  "message": "抱歉, 系统繁忙, 请稍后重试!",
  "data": null
}
```

相应结果

请求参数 ID为空

```
{
  "success": false,
  "code": 99999,
  "message": "抱歉, 系统繁忙, 请稍后重试!",
  "data": null
}
```

相应结果

请求参数 ID为负数"-1175223466613600256"

```
{
  "success": false,
  "code": 99999,
  "message": "抱歉, 系统繁忙, 请稍后重试!",
  "data": null
}
```

相应结果

请求参数 ID为非法字符"jack666"

```
{
  "success": false,
  "code": 99999,
  "message": "抱歉, 系统繁忙, 请稍后重试!",
  "data": null
}
```

相应结果

异常情况

获取不存在的用户信息

不传递参数

传递参数为负数

传递参数为非法字符

1.4 修改员工

修改员工

正常情况

员工【ID】信息不可修改

前置条件 登录

请求方式 PUT

请求地址 /api/sys/user/:target

请求数据 {"id":"123"}

响应结果 {
 "success": false,
 "code": 99999,
 "message": "抱歉，系统繁忙，请稍后重试！",
 "data": null
}

成功修改员工【姓名】信息

请求数据 {"fullName":"tom-old"} API文档bug

请求数据 {"username":"tom-new"}

响应结果 {
 "success": true,
 "code": 10000,
 "message": "操作成功！",
 "data": {
 "id": "1175242529167437824"
 }
}

员工【手机】信息不可修改

请求数据 {"mobilePhone":"13812345678"} API文档bug

请求数据 {"mobile":"13812345678"}

响应结果 {
 "success": false,
 "code": 99999,
 "message": "抱歉，系统繁忙，请稍后重试！",
 "data": null
}

成功修改员工【入职时间】信息

成功修改员工【聘用形式】信息

成功修改员工【工号】信息

成功修改员工【部门】信息

成功修改员工【管理形式】信息

成功修改员工【工作城市】信息

成功修改员工【转正时间】信息

异常情况

修改员工【姓名】信息失败

长度异常

类型异常

不输入

修改员工【手机】信息失败

长度异常

类型异常

不输入

修改员工【入职时间】失败

长度异常

类型异常

不输入

修改员工【聘用形式】信息失败

修改员工【工号】信息失败

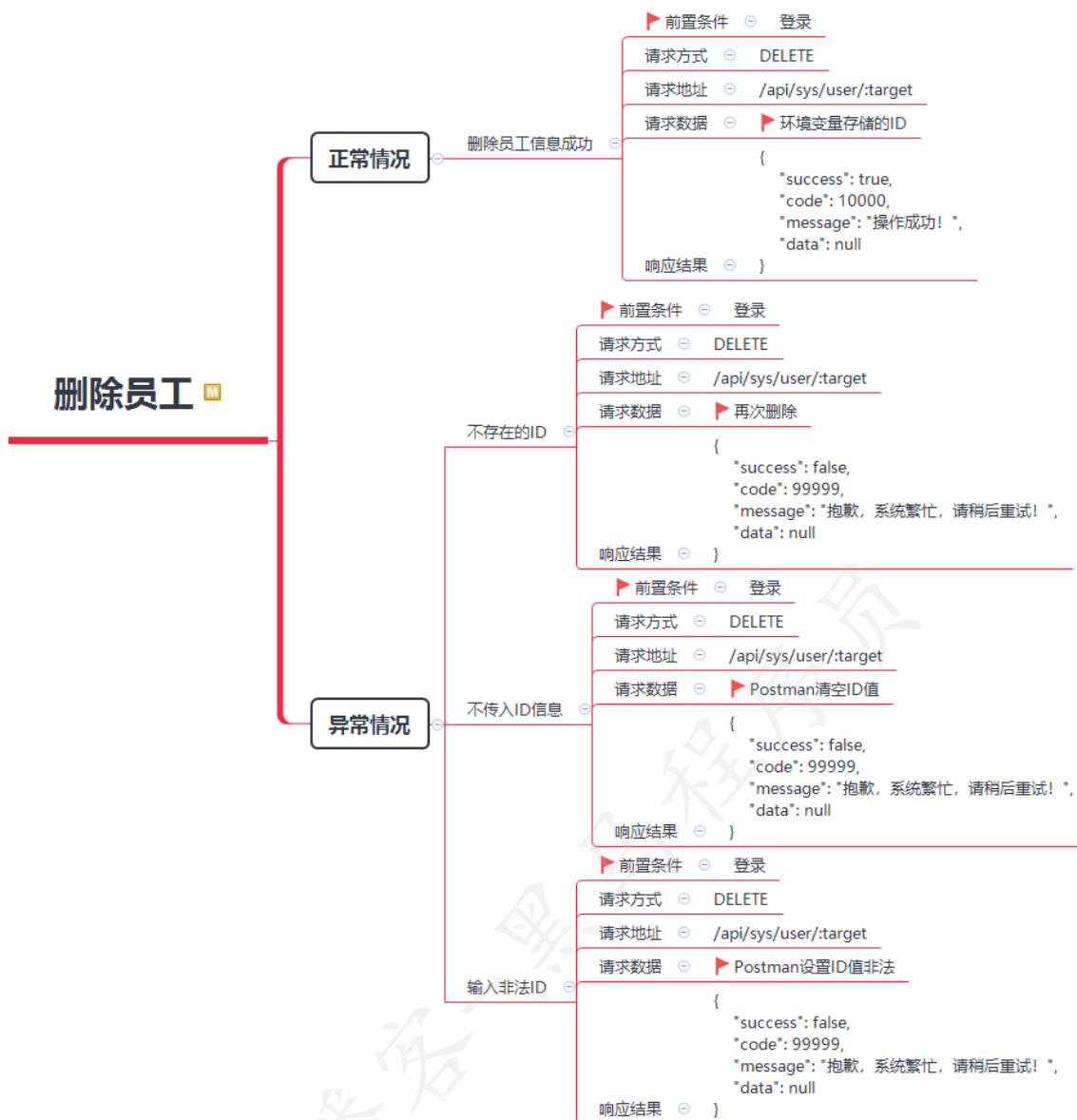
修改员工【部门】信息失败

修改员工【管理形式】信息失败

修改员工【工作城市】信息失败

修改员工【转正时间】信息失败

1.5 删除员工



2 制定 接口 测试计划

项目	内容
测试目标	基于员工ID获取员工信息
测试范围	测试接口是否支持ID, 并且有预期返回结果, 预期结果中有返回员工手机号、用户名、部门等
技术	利用有效和无效的数据来编写用例并执行测试以核实以下内容: (1) 在使用有效数据时, 得到预期结果; (2) 在使用无效数据时, 返回响应的错误消息或警告消息。
接口Case示例	ID=1148850279369996666
完成标准	实现按员工ID的查询功能 所发现严重程度为S1, S2, S3级的BUG已全部解决
测试重点和优先级	重点为员工ID查询, 主要验证接口参数与返回数据的正确性

3 设计 接口 测试用例与评审

3.1 登录

ID	优先级	用例标题	接口名称	请求URL	请求类型	请求头	请求参数	响应状态码	响应数据
Tpshop-LoginAPI-001	P1	POST请求——登录成功	登录	/api/sys/login	POST	Content-Type=application/json	{"mobile":"13800000002", "password":"123456"}	200	{ 'success': true, 'code': 10000, 'message': '操作成功!', 'data': '4719f69e-2ee4-426f-b216-a6c25f8a6475' }
Tpshop-LoginAPI-002	P2	POST请求——用户名错误	登录	/api/sys/login	POST	Content-Type=application/json	{"mobile":"13800000666", "password":"123456"}	200	{ 'success': false, 'code': 20001, 'message': '用户名或密码错误', 'data': null }
Tpshop-LoginAPI-003	P2	POST请求——用户名为空	登录	/api/sys/login	POST	Content-Type=application/json	{"mobile":""," "password":"123456"}	200	{ 'success': false, 'code': 20001, 'message': '用户名或密码错误', 'data': null }
Tpshop-LoginAPI-004	P2	POST请求——密码错误	登录	/api/sys/login	POST	Content-Type=application/json	{"mobile":"13800000002", "password":"error"}	200	{ 'success': false, 'code': 20001, 'message': '用户名或密码错误', 'data': null }

3.2 添加员工

	A	B	C	D	E	F	G	H	I	J	K
	ID	优先级	用例标题	前置条件	接口名称	请求URL	请求类型	请求头	请求数据	响应状态码	响应数据
1	IHRM-API-addUser-001	P1	POST请求——成功添加员工	登录状态	添加员工	/api/sys/user	POST	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	{ "username": "jack", "mobile": "13488888888", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "1322131", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }	200	{ "success": true, "code": 10000, "message": "操作成功", "data": { "id": "1174163739225944064"} }
2	IHRM-API-addUser-002	P2	POST请求——必填项校验——姓名	登录状态	添加员工	/api/sys/user	POST	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	{ "username": "", "mobile": "13488888888", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "1322131", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }	200	{ "success": false, "code": 99999, "message": "抱歉，系统忙，请稍后再试！", "data": null }
3	IHRM-API-addUser-003	P2	POST请求——必填项校验——手机	登录状态	添加员工	/api/sys/user	POST	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	{ "username": "jack2", "mobile": "", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "1322131", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }	200	{ "success": false, "code": 99999, "message": "抱歉，系统忙，请稍后再试！", "data": null }
4									{ "username": "jack3", "mobile": "", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "1322131", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }		

3.3 查询员工

	A	B	C	D	E	F	G	H	I	J	K
	ID	优先级	用例标题	前置条件	接口名称	请求URL	请求类型	请求头	请求数据	响应状态码	响应
1	IHRM-API-getUser-001	P1	GET请求——获取员工信息成功	登录状态	查询员工	/api/sys/user/target 说明: target = 存在的员工ID (501175223466613600256)	GET	{ "Content-Type": "application/json", "Authorization":"Bearer xxx" }	/	200	{ "success": true, "code": 10000, "message": "操作成功!", "data": { "id": "1175223466613600256" ,"mobile": "13488888887", "username": "jack33", "password": "d49a5fcfad8b" ,"enableState": 1, "createTime": null, }
2	IHRM-API-getUser-002	P2	GET请求——获取不存在的信息	登录状态	查询员工	/api/sys/user/target 说明: target = 不存在的员工ID (501175223466613600257)	GET	{ "Content-Type": "application/json", "Authorization":"Bearer xxx" }	/	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙." ,"data": null }
3	IHRM-API-getUser-003	P2	GET请求——不传递参数	登录状态	查询员工	/api/sys/user/target 说明: target = 不输入	GET	{ "Content-Type": "application/json", "Authorization":"Bearer xxx" }	/	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙." ,"data": null }
4	IHRM-API-getUser-004	P2	GET请求——传递参数为负数	登录状态	查询员工	/api/sys/user/target 说明: target = 员工ID为负数 (-107125223466613600256)	GET	{ "Content-Type": "application/json", "Authorization":"Bearer xxx" }	/	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙." ,"data": null }

3.4 修改员工

[illegible]

3.5 删除员工

	A	B	C	D	E	F	G	H	I	J	K
	ID	优先级	用例标题	前置条件	接口名称	请求URL	请求类型	请求头	请求数据	响应状态码	响应数据
1	IHRM-API-deleteUser-001	P1	DELETE请求——删除员工信息成功	登录状态	删除员工	/api/sys/user/target	DELETE	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	通过环境产品 传递ID信息	200	{ "success": true, "code": 10000, "message": "操作成功!", "data": null }
2	IHRM-API-deleteUser-002	P2	DELETE请求——不存在的ID	登录状态	删除员工	/api/sys/user/target	DELETE	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	重复删除	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙, 请稍后 "data": null }
3	IHRM-API-deleteUser-003	P2	DELETE请求——不传入ID信息	登录状态	删除员工	/api/sys/user/target	DELETE	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	手动修改 postman设置 中id值	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙, 请稍后 "data": null }
4	IHRM-API-deleteUser-004	P2	DELETE请求——输入非法ID	登录状态	删除员工	/api/sys/user/target	DELETE	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	手动在 postman设置 中输入非法 id值	200	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙, 请稍后 "data": null }
5											

4 执行 接口 测试用例与缺陷跟踪管理

4.1 执行测试用例

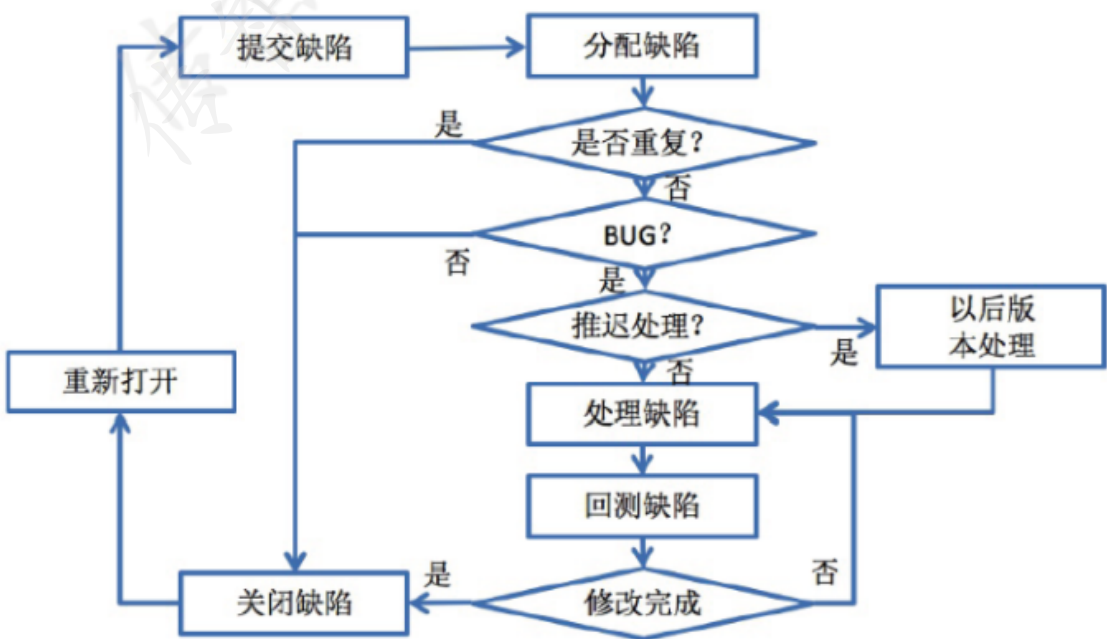
- 通过 Postman工具执行

4.2 缺陷跟踪管理

4.2.1 缺陷报告

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	ID	模块	严重程度	优先级	Bug类型	状态	Bug标题	前置条件	请求URL	请求类型	请求头	请求数据	预期结果	实际结果
1	001	添加员工	S2	Priority 1	代码错误	新建	手机号码必填项未做验证	登录	/api/sys/user	POST	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	{ "username": "jack2", "mobile": "", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "1322131", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙, 请稍后重试!", "data": null }	{ "success": true, "code": 10000, "message": "操作成功!", "data": { "id": "117416449695202304" }
2	002	添加员工	S2	Priority 1	代码错误	新建	工号必填项未做验证	登录	/api/sys/user	POST	{ "Content-Type": "application/json", "Authorization":"Bearer xxx"} }	{ "username": "jack3", "mobile": "13477777777", "timeOfEntry": "2019-07-01", "formOfEmployment": 1, "workNumber": "", "departmentName": "开发部", "departmentId": "1066240656856453120", "correctionTime": "2019-11-30"} }	{ "success": false, "code": 99999, "message": "抱歉, 系统繁忙, 请稍后重试!", "data": null }	{ "success": true, "code": 10000, "message": "操作成功!", "data": { "id": "1174164710341537792" }
3														

4.2.2 跟踪缺陷



5 编写 接口 测试报告

`newman run` 导出的postman脚本json文件 `-r html -e` 导出的环境变量json文件

参数说明:

`-e source` : 指定环境变量文件的路径

Newman Report

Collection
Description
Time
Exported with

postman关联
postman关联知识点讲解
Wed Sep 25 2019 15:51:27 GMT+0800 (GMT+08:00)
Newman v4.5.4

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

Total run duration
Total data received
Average response time

1223ms
360.81KB (approx)
523ms

Total Failures 0

Requests

查询天气接口测试	
Description	获取天气信息，并存储返回数据的地区信息到全局变量
Method	GET
URL	http://www.weather.com.cn/data/sk/101010100.html
Mean time per request	163ms

6 IHRM项目总结汇报



每次被老师叫起来回答问题
就感觉自己好像
在万人礼堂演讲

今日总结

- 掌握Postman如何实现关联
- 掌握如何为项目创建测试用例结构
- 掌握如何为项目设置环境变量
- 掌握在项目中如何对接口响应数据进行断言
- 掌握在项目中如何实现接口关联
- 掌握在项目中如何使用Newman生成HTML测试报告
- 掌握如何使用Postman实现接口测试
- 掌握接口测试流程

作业及要求

作业：

1. 梳理接口项目今日知识点，以xmind形式输出（必做题）
2. 以word文档的形式提交今日项目心得体会（必做题）
3. 基于postman实现IHRM项目登录和员工管理模块的自动化测试(选做题)

要求：

1. 作业需要上交，命名格式：
姓名_作业_接口解析，例如：张三_作业_接口解析；
姓名_作业_接口测试用例设计，例如：张三_作业_测试用例设计；
2. 作业提交时间截止到晚上21:00。

说明：

1. 各小组成员作业上交至组长处；
2. 组长汇总作业后打包提交至班长；
3. 班长将全班作业汇总后提交至老师。

