# 接口测试-项目实战

## Postman读取外部数据文件（参数化）

当 http请求，使用的 数据有较高相似度，相同的请求时，考虑使用参数化（将数据组织到数据文件中）。

**数据文件简介**

* **CSV：**

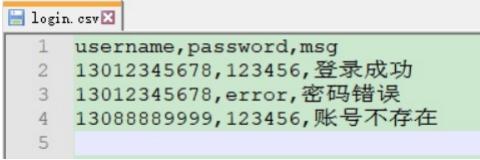
 优点：

 数据组织形式简单，适用于大量数据的场合。

 缺点：

1. 不支持 bool类型数据。（数据被 postman读入后，自动添加 “” 包裹bool值。） 2. 不支持 多参、少参、无参、错误参数 的接口测试。

3. 不支持复杂数据类型。（如 嵌套字典、列表等）



* **JSON ：**

 优点：

1. 支持 bool 类型。

2. 支持 多参、少参、无参、错误参数。

3. 支持复杂数据类型。

 缺点：

 对于相同数据量，json数据文件大小远大于 CSV文件。



**导入外部数据文件**

**CSV文件**

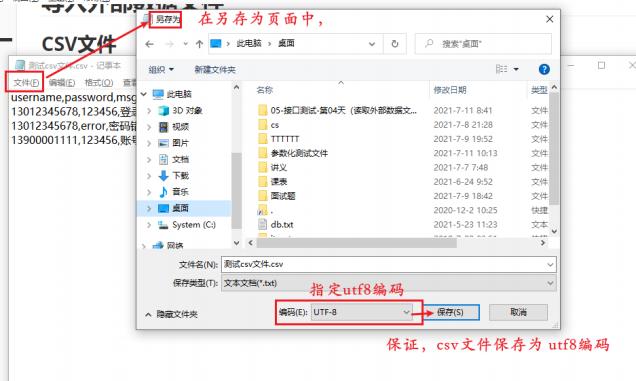
1. 创建 xxx.csv 文件。

2. 将数据写入到 csv文件中。

 第一行 写入 的是 数据对应的 “字段名”。

 从第二行向后依次是对应的数值，数据间用 英文逗号隔分。



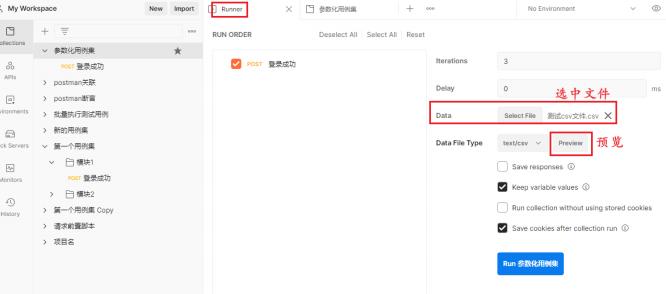


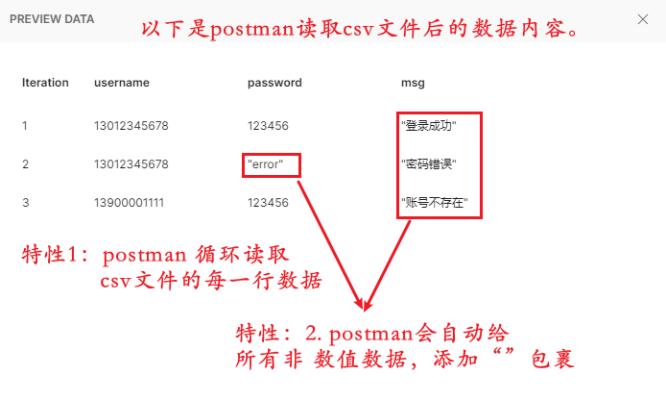
3. 在 Postman 中，选中使用数据文件的 用例集，导入数据文件。

1. 点击用例集名称，使用 Run 按钮，进入 “Runner” 页面。

2. 使用 “Select File” 按钮选择 xxx.csv 文件。

3. 点击 预览按钮，校验数据文件是否正确。





**JSON文件**

1. 创建 xxx.json 数据文件

2. 在 数据文件中，按json 语法写入json数据。postman要求，json格式的数据文件，数据内容的最外层，必须是 [ ] 。内部所有的数据用 {} 存储。（ 建议：使用 json.cn在线网站编写。）

3. 在 Postman 中，选中使用数据文件的 用例集，导入数据文件。

1. 点击用例集名称，使用 Run 按钮，进入 “Runner” 页面。

2. 使用 “Select File” 按钮选择 xxx.json 文件。

3. 点击 预览按钮，校验数据文件是否正确。



**读取数据文件数据**

**理论**

**根据使用数据 位置 不同，有两种获取数据文件中数据的方法。**

 第一种：**请求参数**（请求行、请求头、请求体）中，使用 数据文件中 的数据

 **使用 {{}}** 包裹 csv 文件 字段名 或 json 文件中的 key

 如： {{username}} 或 {{password}}

 第二种：**代码**（断言、请求前置脚本）中，使用 数据文件中 的数据

 需要借助 postman 提供的 **关键字data** 点 csv 文件的字段名 或 json文件的 key

 如： data.username 或 data.password

**案例**

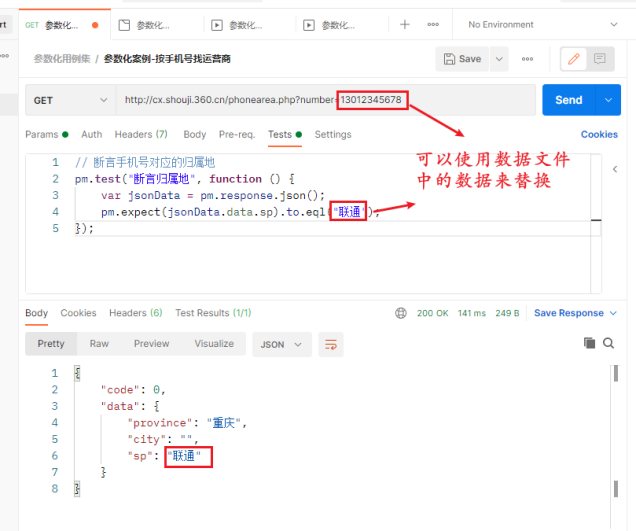
需求：批量查询手机号归属地、所属运营商，校验运营商数据正确性

接口： <http://cx.shouji.360.cn/phonearea.php?number=13012345678>

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

1. 测试单个接口，工作中 参数和 响应体 数据对应关系。



2. 写好数据文件，按 csv 或 json 对应语法的语法格式组织数据。



3. 替换 上述用例中的 两处：

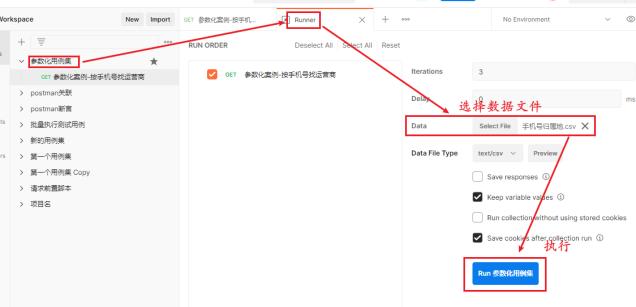
1. 归属地：在 代码中。 用 data.字段名 替换

2. 手机号：在 查询参数中。 用 {{字段名}} 替换。

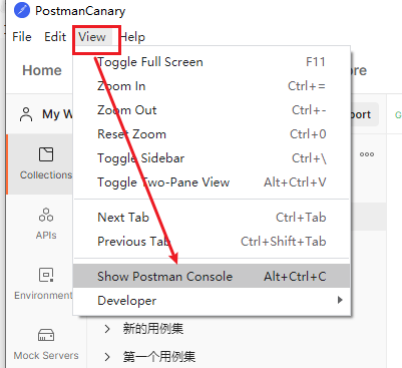
4. 强调：一旦替换完成，Send 就不再能正常获取 响应数据。（数据必须从数据文件中读取，才能正常发送请 求，Send 不具备读取数据文件的功能）



5. 选择使用数据文件的用例集， 进入 Runner 页面。导入数据文件。运行。



6. 如果有异常，借助 postman 的控制台，查看。





小结：借助 数据文件，实现 **“数据驱动”。**—— 有多少条数据，对应就有多少个 http请求。

**生成测试报告**

步骤：

1. 批量执行测试用例（借助数据文件）

2. 导出 测试用例集 （建议与 数据文件 放在一起）

3. 执行生成测试报告的命令

|  |
| --- |
| 1 newman run 用例集名称 .json -d 数据文件名 .csv/.json -r html --reporter-html-export 报告  名称 .html |

**iHRM项目实战**

**简介**

 功能模块



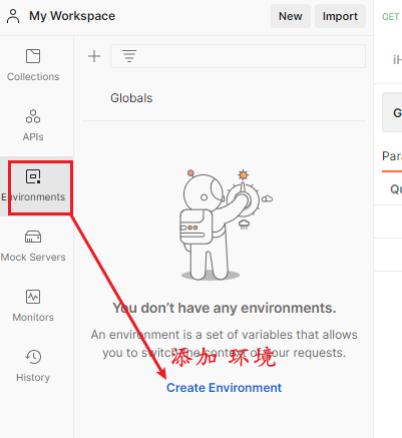
 技术架构

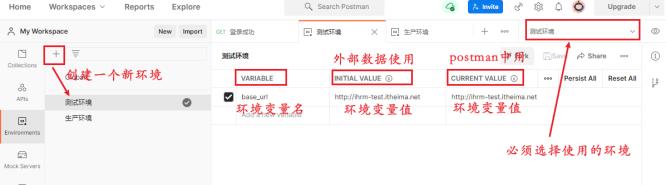


**新建用例集**



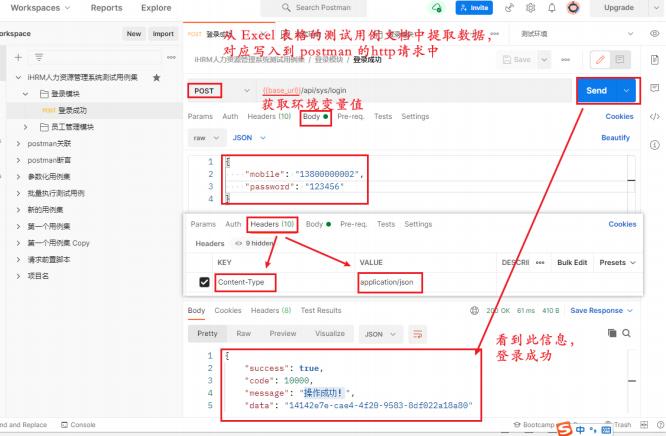
**创建环境变量**



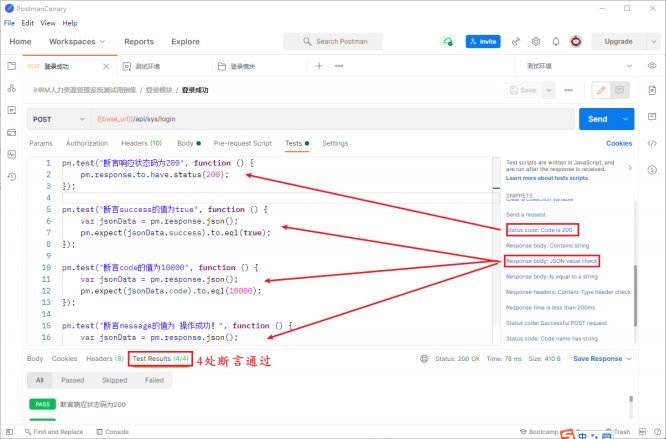


**登录成功接口**

**请求接口**



**添加断言**



**其他接口共性分析**

1. 由于是同一个接口，因此 他们的 请求方法、 URL、请求头 完全一致。

2. 请求体（请求数据），各不相同。

3. 预期结果，大致分为 3 类：

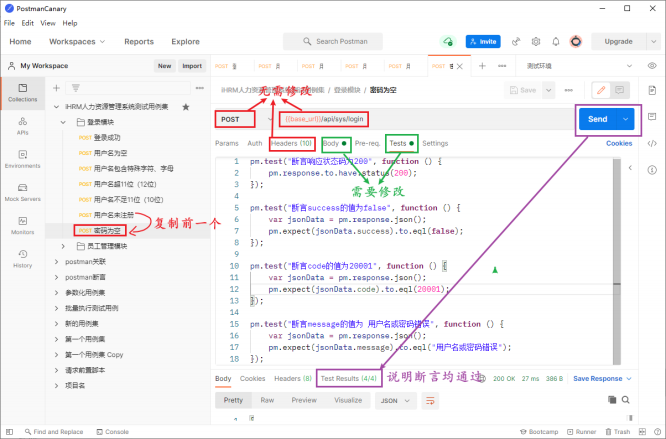
1. 操作成功！ —— 登录成功、多参、全部参数

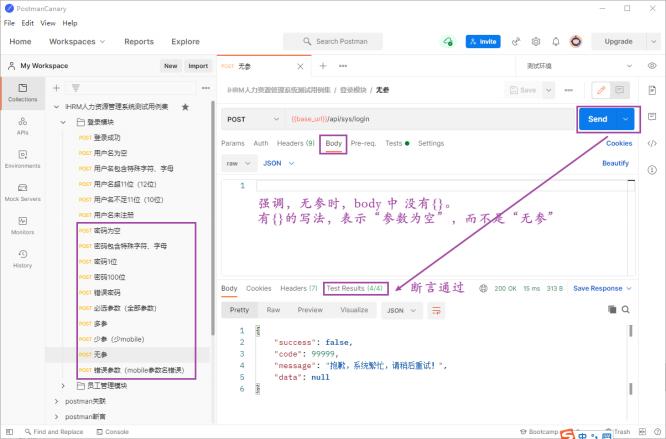
2. 抱歉系统繁忙。。。 —— 无参

3. 用户名密码错误！ —— 其他情况

**实现其他接口**







**员工管理模块**

**总析**

1. **登录成功，返回的 “令牌” ，被 添加、查询、修改、删除 员工接口依赖。**

2. **添加员工成功，返回的 “员工id”，被 查询、修改、删除 员工接口依赖。**

以上的依赖关系，借助 postman关联 和 环境变量 来实现！！

**提取令牌**

1. 发送 http 登录请求，在响应体中 得到 data值。

2. 将 data值 拼接 “Bearer” 和 “空格” ，组成一个 合法令牌。

3. 将 拼接后的令牌 环境变量中。（在 Tests 中通过代码写入）

// 拼接 data的值 ，添加 Bearer 和 空格。组成合法令牌

// 获取 响应体 , 保存在 jsonData变量中。

var jsonData = pm.response.json()

4

// 提取 data的值。保存在 tmp\_token 中。

var tmp\_token = jsonData.data

7

// 拼接 Bearer 和 空格 到 data值的前面。 "Bearer" + " " + tmp\_token

pm.environment.set("env\_token", "Bearer"+" "+tmp\_token)

1

2

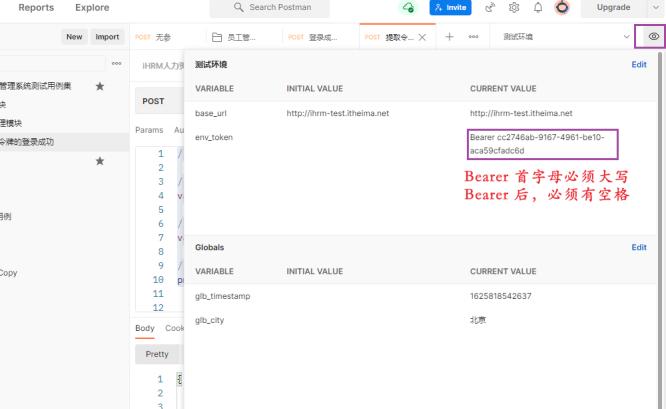
3

8

9

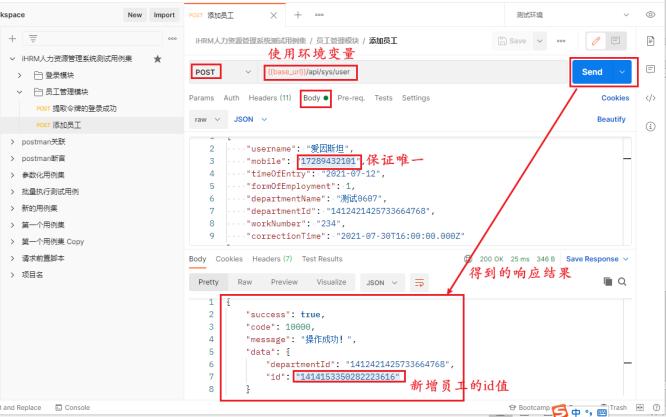
5

6



**添加员工**





**提取添加员工的id**

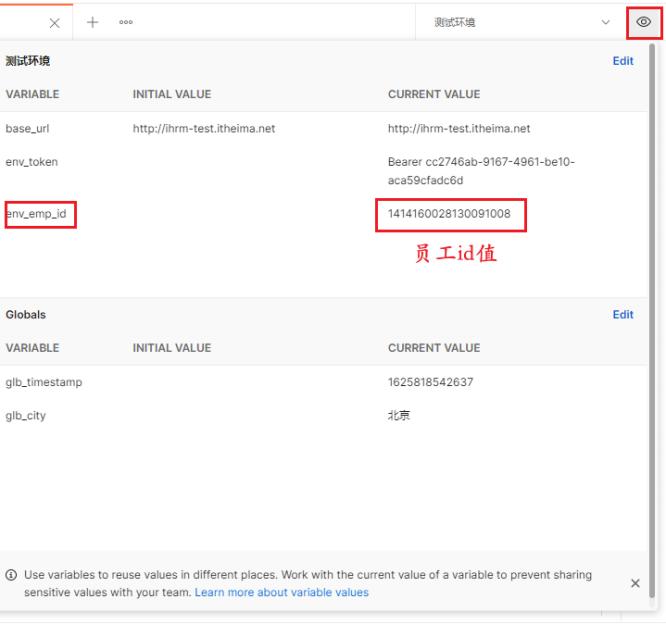
1. 获取 添加员工成功的 data下面的 id值。

2. 将 员工id，设置 环境变量（代码写入 Tests 中）

|  |
| --- |
| 1 // 获取添加员工的 响应体  2 var jsonData = pm.response.json()  3 // 提取员工id  4 var tmp\_id = jsonData.data.id  5 // 设置到环境变量。  6 pm.environment.set("env\_emp\_id", tmp\_id) |

3. 发送 Send 之前，注意先修改 手机号，保证手机号唯一！

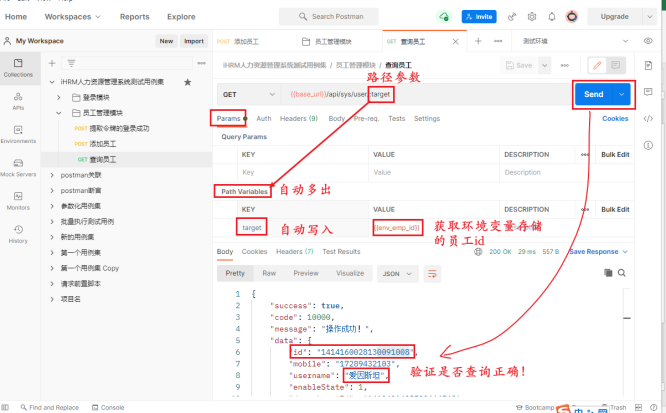
4. 点击 “ 眼睛” 图标 查看。



**查询员工**

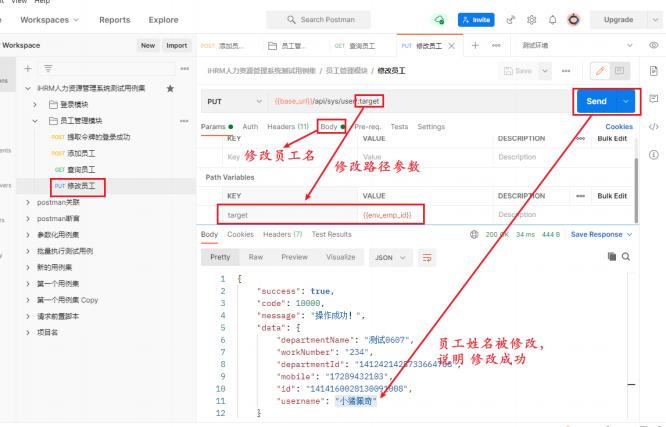
将测试用例中的，请求方法、 URL、请求头，写入到 postman对应请求中。

给 :target 对应 的 路径参数，填充上 “员工id”的值。（ {{员工id环境变量}} ）



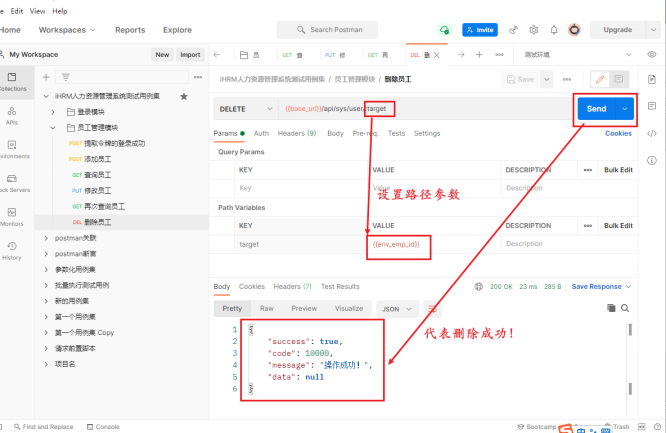
**其他员工操作**

修改员工

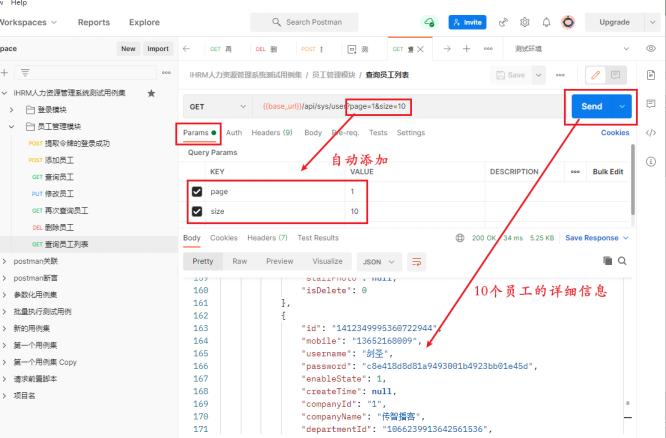


删除员工





查询员工列表：



**批量运行测试用例**



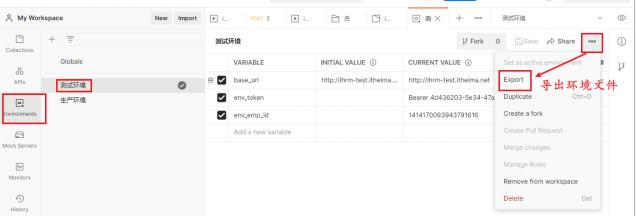
1. 给 每个http请求添加 断言（成功断言）

2. 点击用例集名称， run ，进入 Runner 页面。批量执行。

3. 导出 用例集。

**生成测试报告**

1. 将环境导出成一个 环境文件。



2. 将 环境文件，与测试用例集文件，放置到同一个目录下。

3. 执行命令：

|  |
| --- |
| 1 newman run 测试用例集文件 .json -e 环境文件 .json -r html --reporter-html-export 测试报告  名 .html  2 # 示例 ：  3 newman run iHRM人力资源管理系统测试用例集 .postman\_collection.json -e 测试环  境 .postman\_environment.json -r html --reporter-html-export iHRM的测试报告 .html |