



接口测试

张程
Email: bootan@cqu.edu.cn
QQ:80463125

基础知识



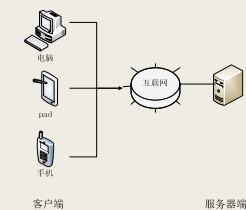
接口测试

- 软件接口的分类一般有如下几种情况：
 - 系统与系统之间的调用，如微信向用户提供统一的对外接口，程序员调用接口完成基于微信的小程序等；
 - 同一系统内部上层服务对下层服务的调用，如一个软件一般分为表示层，业务层和数据层，表示层调用业务层的接口来完成自己的工作，而业务层又会调用数据层的接口来实现相应的业务等。
- 接口测试(Interface Testing)就是针对系统组件间的接口进行的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。接口测试的重点是检查数据的交换、传递和控制管理过程，以及系统间的相互逻辑依赖关系等。也就是说，接口测试的重点是检查接口参数传递的正确性、接口功能实现的正确性、输出结果的正确性，以及对各种异常情况的容错处理的完整性和合理性。
- 接口测试以保证系统的正确和稳定为核心，其重要性主要体现在以下几个方面：
 - 能够及时发现bug，符合质量控制前移的理念。
 - 接口测试低成本高效益，因为接口测试可以自动化并且是持续集成的。
 - 接口测试从用户的角度对系统接口进行全面检测。实际项目中，接口测试会覆盖一定程度的业务逻辑。



HTTP工作原理

- 接口测试需要模拟浏览器发送Request至服务器和服务器返回Response到浏览器的整个过程



HTTP协议原理



- 客户端主要有两个职能：
 - 向服务器发送请求。
 - 接收服务器返回的报文并解释成友善的信息供我们阅读。
- 客户端大概有以下几类：浏览器、应用程序（桌面应用和APP应用）等
- 在谷歌浏览器地址栏中输入百度网址并回车，浏览器会做如下的处理：
 - 当人们在浏览器地址栏输入www.baidu.com的时候，浏览器发送一个Request 请求给服务器，要求服务器返回www.baidu.com的网站主页的HTML文件，接着服务器响应用户请求，把Response 文件对象发送回给浏览器。
 - 浏览器分析Response 中的HTML，发现其中引用了很多其他文件，比如images 文件、CSS 文件、JS 文件等，浏览器会自动再次发送Request 去获取网页中加载的图片文件、CSS 文件或者JS 文件。
 - 当网页中包含的所有文件都下载成功后，浏览器会根据HTML 语法结构，完整的显示出网页。
- HTTP协议详细规定了客户端与服务器之间互相通讯的规则，它主要解决了两个问题：
 - 如何定位资源资源？（URL）
 - 客户端与服务器间如何进行信息传递？（报文）



用Firefox浏览器抓取报文

- 打开Firefox浏览器，选择工具菜单→Web 开发者→切换工具箱



- 在浏览器的下方，将显示开发者工具窗口

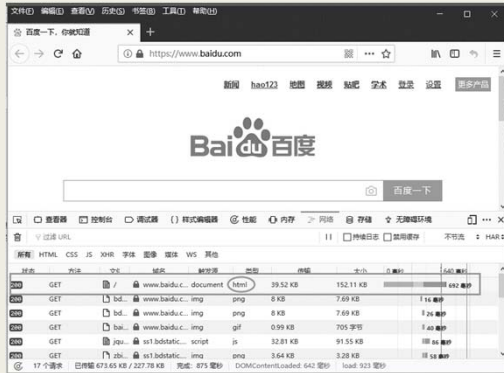


- 在开发者工具栏中，单击网络切换至网络页签





- 在浏览器地址栏中输入www.baidu.com，按回车键后，浏览器发送一个Request请求给服务器



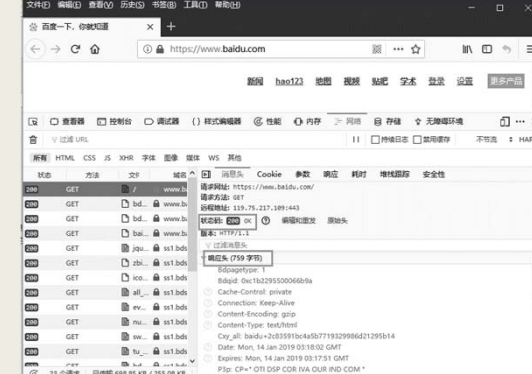
- 选中第一个主请求，可以看到请求的消息头



- 单击编辑和重发，就可以看到完整的HTTP请求报文

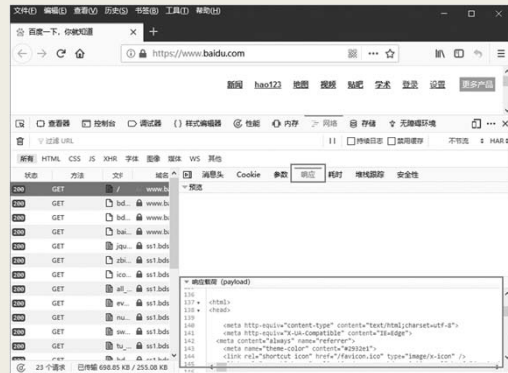


- 观察响应报文，响应报文的结构分为3个部分：状态行、响应头、响应主体





- 单击响应，就可以看到服务器返回的HTML源码



URL

- 统一资源定位符（Uniform Resource Locator，URL）是因特网的万维网服务程序上用于指定信息位置的表示方法。统一资源定位符URL用来标识万维网上的各种资源，使每一个资源在整个因特网的范围内具有唯一的标识符。
- URL的一般形式是：HTTP://<主机>[:<端口>]/<路径>，各部分含义如下：
 - HTTP：表示使用HTTP协议；
 - 主机：存放资源的主机域名或主机IP地址；
 - 端口：HTTP的默认端口号是80，通常可省略；
 - 路径：访问资源的路径。

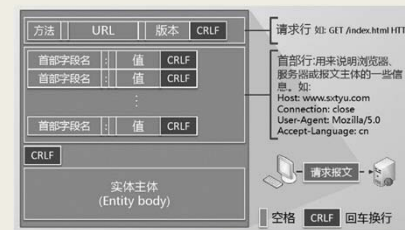


报文

- 报文的组成



- 请求报文
- 客户端向服务器发送请求时，会给服务器发送一个请求报文。请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。URL只是标识资源的位置，而HTTP报文用来提交和获取资源。客户端发送的HTTP请求消息，包括请求行、请求头部、空行和请求体4个部分。



请求报文格式



请求报文



■ 请求报文中的请求方法

序号	方法	描述
1	GET	请求读取一个Web页面
2	HEAD	请求读取一个Web页面的首部
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件），数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改
4	PUT	请求存储一个Web页面
5	DELETE	删除Web页面
6	CONNECT	HTTP1.1协议中预留给能够将连接改为管道方式的代理服务器
7	OPTIONS	允许客户端查看服务器的性能
8	TRACE	回显服务器收到的请求，主要用于测试或诊断

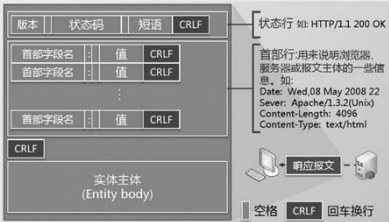


■ 请求报文中的Request属性

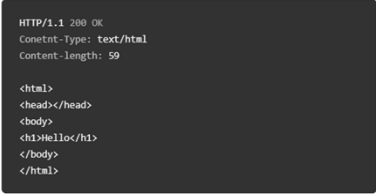
序号	请求头	描述
1	Host	对应网址URL中的Web名称和端口号，用于指定被请求资源的Internet主机和端口号，通常属于URL的一部分
2	User-Agent	告诉服务器客户端使用的操作系统和浏览器的名称、版本
3	Accept	指浏览器或其他客户端可以接受的文件类型，服务器可以根据它判断并返回适当的文件格式Accept: */*: 表示什么都可以接收Accept: image/gif: 表明客户端希望接受GIF图像格式的资源Accept: text/html: 表明客户端希望接受HTML文本
4	Accept-Language	指出浏览器可以接受的语言种类，如en或en-us指英语，zh或者zh-cn指中文
5	Accept-Encoding	指出浏览器可以接受的编码方式，编码方式不同于文件格式，它是为了压缩文件并加速文件传递速度。浏览器在接收到Web响应之后先解码，然后再检查文件格式，在许多情形下这可以减少大量的下载时间。
6	Referer	表明产生请求的网页来自于哪个URL，用户是从该 Referer页面访问到当前请求的页面。这个属性可以用来跟踪Web请求来自哪个页面，是从什么网站来的等。有时候遇到下载某网站图片，需要对应的referer，否则无法下载图片，那是因为人家做了防盗链，原理就是根据referer去判断是否是本网站的地址，如果不是，则拒绝，如果是，就可以下载。
7	Cookie	浏览器用这个属性向服务器发送Cookie，Cookie是在浏览器中寄存的小型数据体，它可以记载和服务器相关的用户信息，也可以用来实现会话功能。
8	Connection	keep-alive: 当一个网页打开完成后，客户端和服务端之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。close: 代表一个Request完成后，客户端和服务端之间用于传输HTTP数据的TCP连接会关闭，当客户端再次发送Request时，需要重新建立TCP连接。



- 响应报文中的Response属性
- HTTP响应报文由4部分组成，分别是：状态行、响应头部、空行、响应体



响应报文格式



响应报文示例



■ 响应报文响应类别

序号	分类	分类描述
1	1xx	进度通知，表示客户端的请求服务器正在处理
2	2xx	成功，表示客户端的请求服务器已经成功处理了
3	3xx	重定向，服务器通知客户端请求的资源已不存在
4	4xx	客户端发来的请求报文里有错误，比如语法错误或请求的资源不存在等
5	5xx	服务器端有错误，已经无法处理完客户端请求了



■ 响应报文常见状态码

状态码	名称	中文描述
200	OK	请求成功
301	Moved Permanently	资源被永久移动。请求的资源已被永久的移动到新URI，返回信息会包括新的URI，浏览器会自动定向到新URI
302	Found	资源临时移动。资源只是临时被移动，客户端应继续使用原有URI
403	Forbidden	没权限。服务器收到请求，但拒绝提供服务
404	Not Found	请求的资源不存在。遇到404首先检查请求URL是否正确
500	Internal Server Error	服务器内部错误，无法完成请求
503	Service Unavailable	由于超载或系统维护（一般是访问人数过多），服务器无法处理客户端的请求，通常这只是暂时状态



■ 响应报文中的响应头属性

序号	响应头	描述
1	Date	生成消息的具体时间和日期
2	Server	指明HTTP服务器的软件信息
3	Content-Type	Web服务器告诉浏览器自己响应的对象的类型和字符集
4	Content-length	指明实体正文的长度
5	Cache-Control	用来指定Response—Request遵循的缓存机制Public：可以被任何缓存所缓存Private：指响应信息的全部或部分用于单个用户，而不能用一个共享缓存来缓存。这可以让源服务器指示，响应的特定部分只用于一个用户，而对其他用户的请求则是一个不可靠的响应。No-cache：所有内容都不会被缓存，请求头里的no-cache表示浏览器不想读缓存，并不是说没有缓存。一般在浏览器按Ctrl+F5强制刷新时，请求头里也会有这个no-cache，表示跳过缓存直接请求服务器
6	Set-cookie	用于把cookie发送到客户端浏览器，每一个写入的cookie都会生成一个set-cookie
7	Last-modified	用于指示资源的最后修改日期和时间
8	Content-encoding	Web服务器表明自己使用了什么压缩方法(gzip, deflate)压缩响应中的对象

接口测试



为什么要做接口测试

- 接口测试不但可以将测试工作前置，它还可以解决其他方面的问题，比如在用户注册功能中，规定用户名为6~18个字符，可以包含字母（区分大小写）、数字、下划线。在做功能测试时，测试人员肯定会对用户名的组成规则进行测试，比如输入由20个字符组成的用户名、输入包含特殊字符的用户名等，但这些测试可能只是对前端的数据输入进行了合法性校验，而软件后端程序可能没有对接收到的数据做合法性校验。
- 相对UI测试来说，接口测试比较稳定，容易实现自动化持续集成，可以减少人工回归测试的人力成本与时间，缩短测试周期，支持后端快速发版需求。



接口测试的定义

- 接口测试主要用于检测外部系统与系统之间以及系统内部各个子系统之间的交互点。测试的重点是检查数据的交换、传递和控制管理过程，以及系统间的相互逻辑依赖关系等。
- 简单地说，接口测试就是通过测试不同情况下的输入参数与相对应的返回结果，来判断软件系统前后端之间的接口是否符合或满足相应的功能性、安全性要求。



接口测试实例分析

- 在做接口测试前，测试人员需要先拿到开发人员提供的接口文档。测试人员可以根据这个文档编写接口测试用例。要做好接口测试，首先要学会解析接口文档。一般接口文档会包含接口的地址、使用的方法（GET/POST/PUT）等、必填参数、非必填参数、参数长度、返回结果等，只有了解这些信息才能设计测试用例。

■ 接口文档解析

接口名称:	天天抽奖			
调用方式:	RESTFUL			
接口地址:	/appapi/hackDraw			
接口方法:	Post			
● 输入参数定义				
列名	字段名	类型	必填	备注
用户手机	mobilePhone	Number	是	只接受长度为 11 的整数
活动id	activityGuid	Number	是	只接受小于 1000 的整数
● 返回数据说明				
列名	字段名	类型	必填	备注
用户手机	mobilePhone	Number	是	
剩余抽奖次数	number	Number	是	每天限定 3 次抽奖机会
抽奖结果	successful	Varchar (10)	是	中奖=true; 未中奖=false

接口设计说明

从接口文档可以得到如下信息：

- (1) 接口的URL地址；
- (2) 接口的方法是post；
- (3) 接口有2个必填的参数，一个是手机号，一个是活动；
- (4) 对于手机号参数，其数据类型是数字，且限定为11个数字；
- (5) 对于活动ID参数，其数据类型也是数字，且是小于1000的数字；
- (6) 接口返回3个参数，用户手机号码、剩余抽奖次数、抽奖结果；
- (7) 返回的用户手机号码就是参与抽奖的用户手机号码；
- (8) 每天只有3次抽奖机会，抽一次少一次，当没有抽奖次数时，返回Number=0，并且抽奖结果不能为True；
- (9) 抽奖结果只能是True或者是False。

解析完接口文档后，测试人员基本上可以明确测试点和预期结果，为之后的测试用例设计做好准备。



■ 测试用例设计

- 接口功能比较单一，不需要很复杂的测试用例，只要准备好正常数据和异常数据及对应的各种返回结果即可。

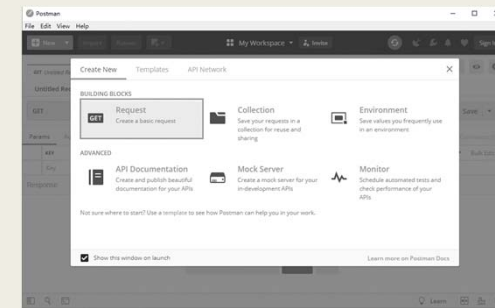
输入数据	预期输出结果
不填写任何参数	报错，缺少手机号和活动 ID
只填写手机号	报错，缺少活动 ID
只填写活动 ID	报错，缺少手机号
填写非 11 位的手机号	报错，手机号码不正确
填写 11 位带符号的手机号	报错，手机号码不正确
填写非数字的活动 ID	报错，活动 ID 不正确
填写大于等于 1000 的活动 ID	报错，活动 ID 不正确
填写小于等于 0 的活动 ID	报错，活动 ID 不正确
填写 1000 以内的非整数的活动 ID	报错，活动 ID 不正确
填写符合要求的手机号和活动 ID	抽奖成功，返回手机号与输入手机号一致，剩余抽奖次数为 2，抽奖结果 True/False
填写符合要求的手机号和活动 ID，第 3 次发送	抽奖成功，返回手机号与输入手机号一致，剩余抽奖次数为 1，抽奖结果 True/False
填写符合要求的手机号和活动 ID，第 3 次发送	抽奖成功，返回手机号与输入手机号一致，剩余抽奖次数为 0，抽奖结果 True/False
填写符合要求的手机号和活动 ID，第 4 次发送	抽奖失败，已经没有抽奖次数了
换一个符合要求的手机号和同样的活动 ID	抽奖成功，返回手机号与输入手机号一致，剩余抽奖次数为 2，抽奖结果 True/False
换一个符合要求的活动 ID 和同样的手机号	抽奖成功，返回手机号与输入手机号一致，剩余抽奖次数为 2，抽奖结果 True/False



接口测试工具-Postman

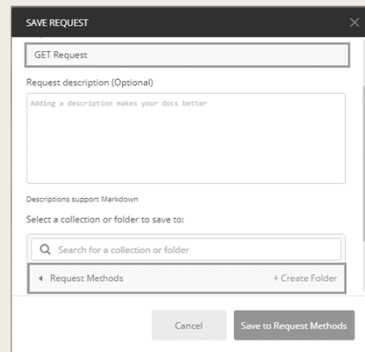
■ 使用Postman的基础功能

- (1) 在Postman界面中选择创建Request基础请求。除基础请求外还可以创建Collection（请求集合文件夹）、Environment（环境变量）、API Documents（API 文档）、Mock Server（模拟服务器）以及Monitor（监视器）。

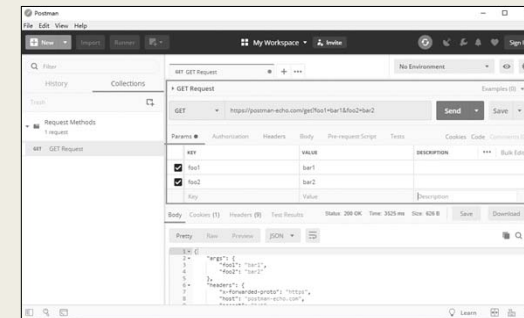




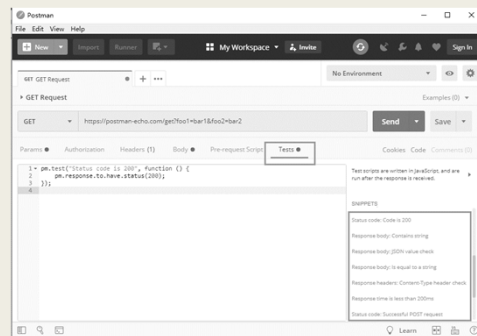
- (2) 在保存请求界面，输入请求名称GET Request、选择Request Methods创建新文件夹作为保存位置，单击保存按钮



- (3) 请求保存后，可以在该请求中继续添加URL。单击参数Params选项卡，输入参数及对应的参数值，可根据需要输入多个参数，这些参数会立即添加在URL链接上构成一个完整的GET请求。输入完成后单击发送send按钮发送请求，服务器响应并回显到界面下方区域



- (4) 一个完整的接口测试包括：请求→获取响应正文→断言，上一步读者已经知道了如何请求与获取响应，接下来使用 Postman进行断言。Tests选项卡是处理断言的地方，Postman人性化的提供了断言所需用的函数



- 在Tests界面选择合适的断言来实现断言场景。

```

2- pm.test("Status code is 200", function () {
3-   pm.response.to.have.status(200);
4- });
5-
6- pm.test("response body contains bar1", function () {
7-   pm.expect(pm.response.text()).to.include("bar1");
8- });
9-
10- pm.test("response host", function () {
11-   var jsonData = pm.response.json();
12-   pm.expect(jsonData.headers.host).to.eql("postman-echo.com");
13- });
14-

```

断言脚本

Status code：状态码，表示判断HTTP返回的状态；本例中第一条断言代码的含义是判断响应状态码是否为200，Status code is 200是断言名称，读者可以自行修改。

Response body：响应正文（Contains string）。本例中第二条断言代码的含义是判断响应的文本内容中是否包含字符串bar1。Response host：JSON值检查（JSON value check），解析响应JSON数据，判断host的值是否与postman-echo.com匹配。本例中第三条断言代码的作用是对JSON字符串进行解析。原始代码如下：

```

pm.test("Your test name", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.value).to.eql(100);
});

```

其中jsonData变量是解析后的json对象，在JS中，一个JSON对象获取其属性的值可以直接使用jsonData.value。这里把代码修改为如下形式来判断第3个场景：

```

pm.test("response host", function () {
  var jsonData = pm.response.json();
  pm.expect(jsonData.headers.host).to.eql("postman-echo.com");
});

```


重庆大学
CHONGQING UNIVERSITY

- (5) 本例中共创建了3个Tests断言, 创建完成后, 单击send按钮发送请求, 在响应区内可以看到断言全部通过

GET 请求的参数随地址栏传递给服务器, POST请求相对于GET 请求多了个Body 部分, Body用来设置POST请求的参数
 form-data: 是一种表单格式, 它会将表单的数据处理为一条消息, 如: form-data; name="file"; filename="chrome.png", 其含义是将数据传递给服务器。
 x-www-form-urlencoded: 浏览器的原生form表单, 以如下的数据格式 foo1=bar1&foo2=bar2将数据传递给服务器。
 raw: 可以发送任意格式的接口数据, 可以text、json、xml、html等。
 binary: HTTP请求中的Content-Type:application/octet-stream, 表示只可以发送二进制数据。通常用于文件的上传。
 使用POST方法时需要注意POST请求参数的格式。



断言响应结果



POST请求

重庆大学
CHONGQING UNIVERSITY

- Postman进阶使用
- 1.设置环境变量
 - (1) 单击Postman界面左上角的New→Environment创建环境变量, 添加环境变量后单击Add按钮



新建环境变量

重庆大学
CHONGQING UNIVERSITY

- Postman进阶使用
- 1.设置环境变量
 - (2) 环境变量添加后, 在使用这些键值的时候只需要加上两个花括号来引用key, 同时在右下角下拉列表中选择需要的环境就可以了



切换环境

重庆大学
CHONGQING UNIVERSITY

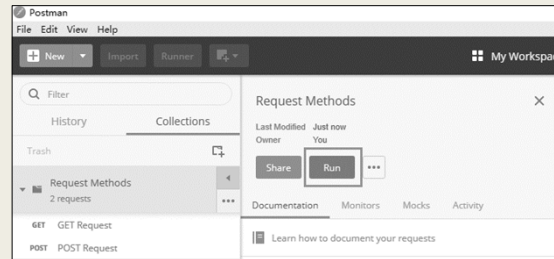
- Postman进阶使用
- 2.使用Collections管理用例
 - (1) 创建一个名为Request Methods的集合, 包含两个请求

新建环境变量





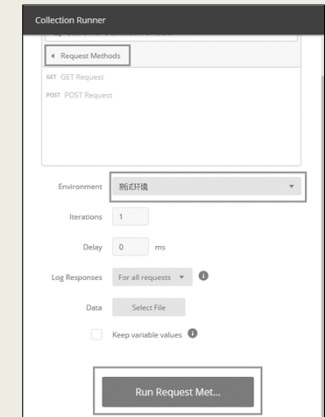
- Postman进阶使用
- 2.使用Collections管理用例
 - (2) 运行Collection, 单击图中的Run按钮, 一次执行整个Collection里的所有用例



运行collections



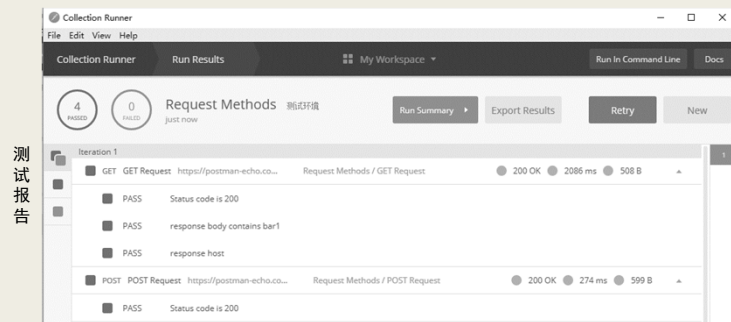
- Postman进阶使用
- 2.使用Collections管理用例
 - (3) 进入Collection Runner界面, 选择Request Methods集合



运行collection runner设置



- Postman进阶使用
- 2.使用Collections管理用例
 - (4) 运行完成后, 即可得出一个简易的聚合报告



测试报告



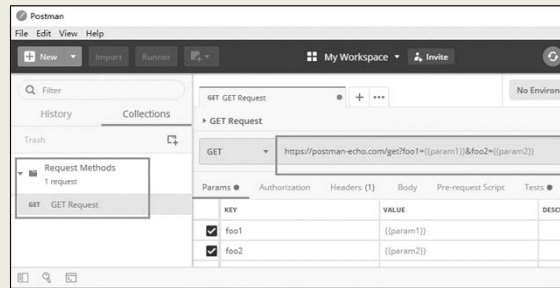
- Postman进阶使用
- 3.选取外部文件作为数据源
 - (1) 下面以Postman Echo中的GET方法作为示例。首先创建一个名为data.csv的数据文件作为源数据。

param1	Param2
test1	user1
test2	User2

数据文件内容

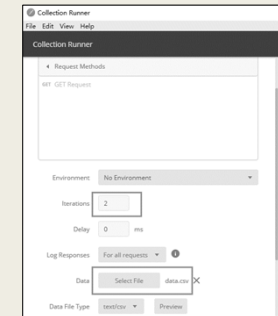


- Postman进阶使用
- 3.选取外部文件作为数据源
 - (2) 在Request Methods下添加GET请求, 将URL中的常量值用csv文件中的参数来代替



参数化

- Postman进阶使用
- 3.选取外部文件作为数据源
 - (3) 保存修改结果, 调用Runner模块运行此集合, 选择外部数据文件, Iterations运行次数会自动匹配外部数据文件中的数据条数



运行集合设置



- Postman进阶使用
- 3.选取外部文件作为数据源
 - (4) 运行完成后查看报告, 可以看出共有两次循环, 第一次循环取数据表中的第一条数据, 第二次循环取数据表中的第二条数据

运行结果

