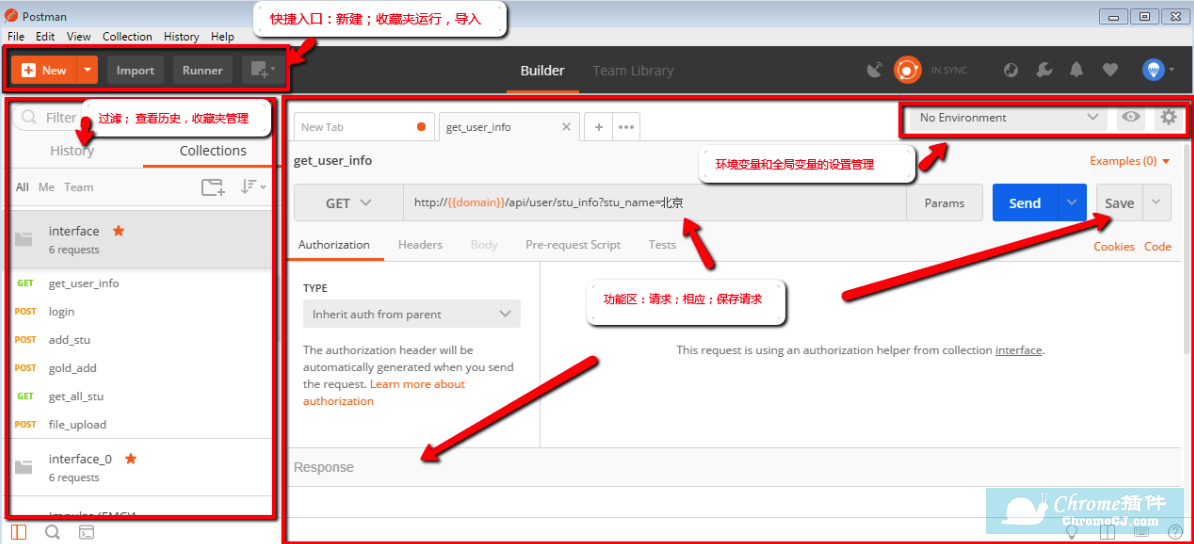
postman接口测试教程与实例分享

在之前的[Postman](http://chromecj.com/web-development/2017-09/60.html" \t "_blank)教程中我们已经介绍了基础的功能，我们做一些简单的复习后，就要进入我们今天的主题：Postman接口测试教程与实例分享。

postman 的界面图



**各个功能区的使用如下：**

* **快捷区**： 快捷区提供常用的操作入口，包括运行收藏夹的一组测试数据，导入别人共享的收藏夹测试数据（Import from file, Import from folder, Import from link等），或新建请求、收藏夹、环境变量等。
* **侧边栏**： 包括搜索栏， Request 请求的历史记录和收藏夹管理。
* **功能区**： Request 请求设置，查看 Response 响应结果和测试结果，可以将请求保存到收藏夹。
* **设置区**：设置和管理环境变量和全局变量。

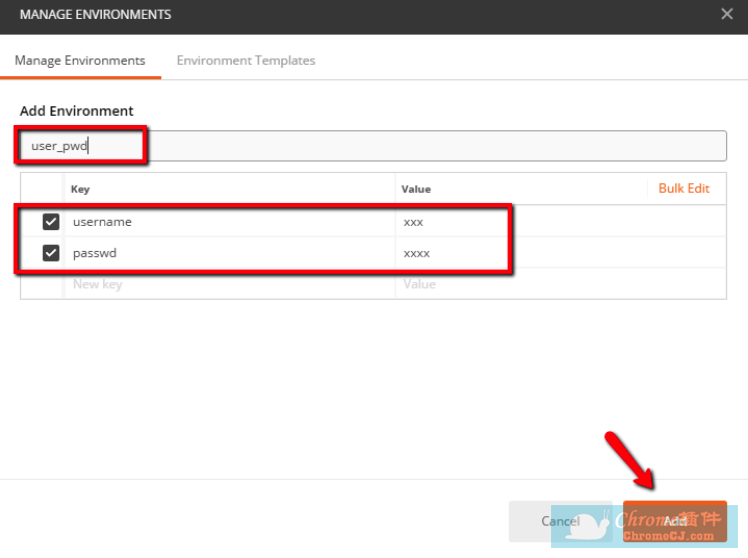
一、环境变量和全局变量的设置：

**a. 环境变量**的设置方法如下图。点击设置一个环境变量，名为"user\_pwd", 里面设置username 和passwd的值，然后在请求中应用这个环境变量"user\_pwd"即可。

环境变量可以使用在以下地方：

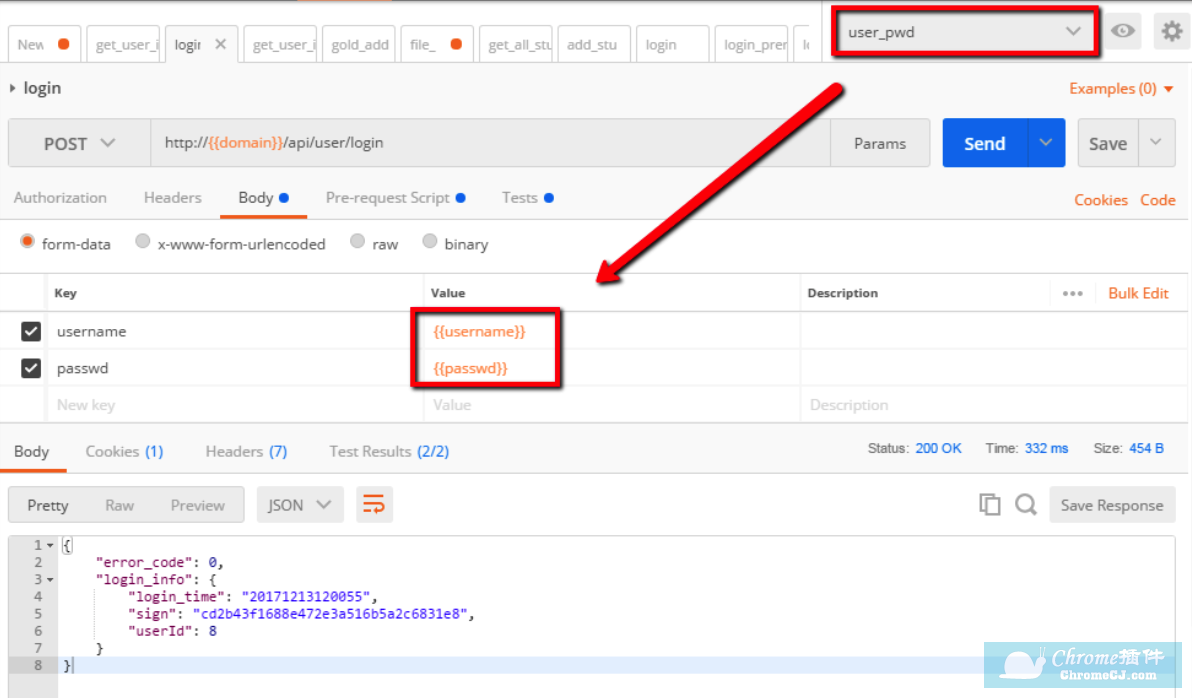
* URL
* URL params
* Header values
* form-data/url-encoded values
* Raw body content
* Helper fields

**注意：在你要使用的变量名上附上双花括号，一个请求只能应用一个环境变量。**



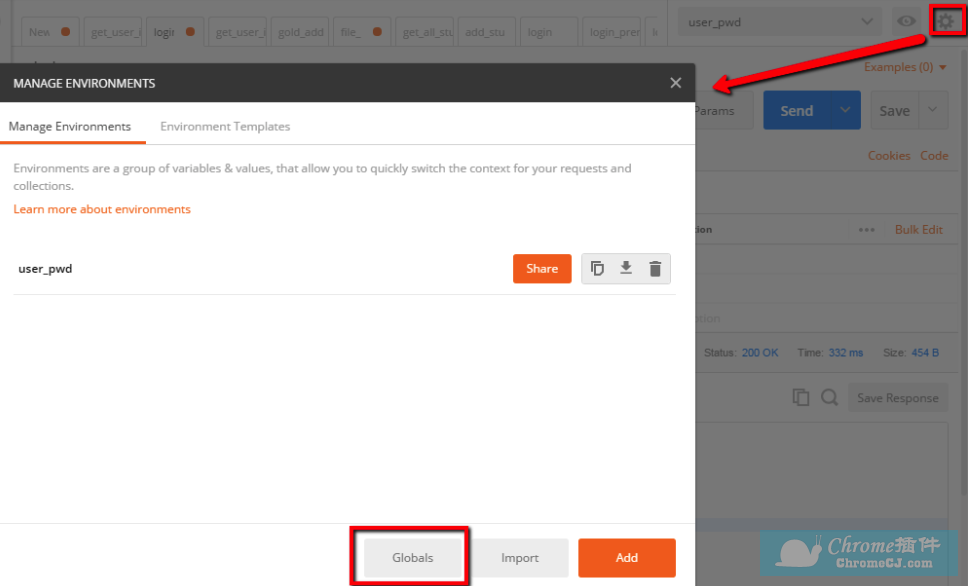
下图"username" 和"passwd" 的value值应用了环境变量**"user\_pwd"**，所以在body 中，只需在key对应的value那写上变量**{{username}}, {{passwd}}**。

**注意：一个项目需要测试好几个环境的情况，就可以对不同的环境设置一个环境变量，比如对“test environment”设置一个环境变量，对“stage environment”设置一个环境变量，对“production 环境”设置一个环境变量。**



 b. **全局变量**的设置方法类似，如下图，点击**"Globals"** 后就会出现添加环境变量类似的页面，进行变量和值的设置。

**注意：全局变量设置就应用于整个收藏夹（Collection）中的请求，不用像环境变量一样需要选取。**



二、用Postman做接口测试的实例

接口测试中常用的请求为GET 和POST，以下均以这两种请求为例。

**GET 和POST 的区别：**

* GET 使用URL 或Cookie 传参，而POST将数据放在Body 中。
* GET的URL 在长度上会有限制，而POST没有。
* POST比GET相对安全，因为在地址栏不可见。
* 一般POST请求用来获取数据，POST请求用来发送数据。

**对于上面的区别，其实第一点POST也可以将数据放在URL里，GET请求其实也没有长度限制，POST请求看起来是隐式的，但是可以通过抓包拿到参数。**

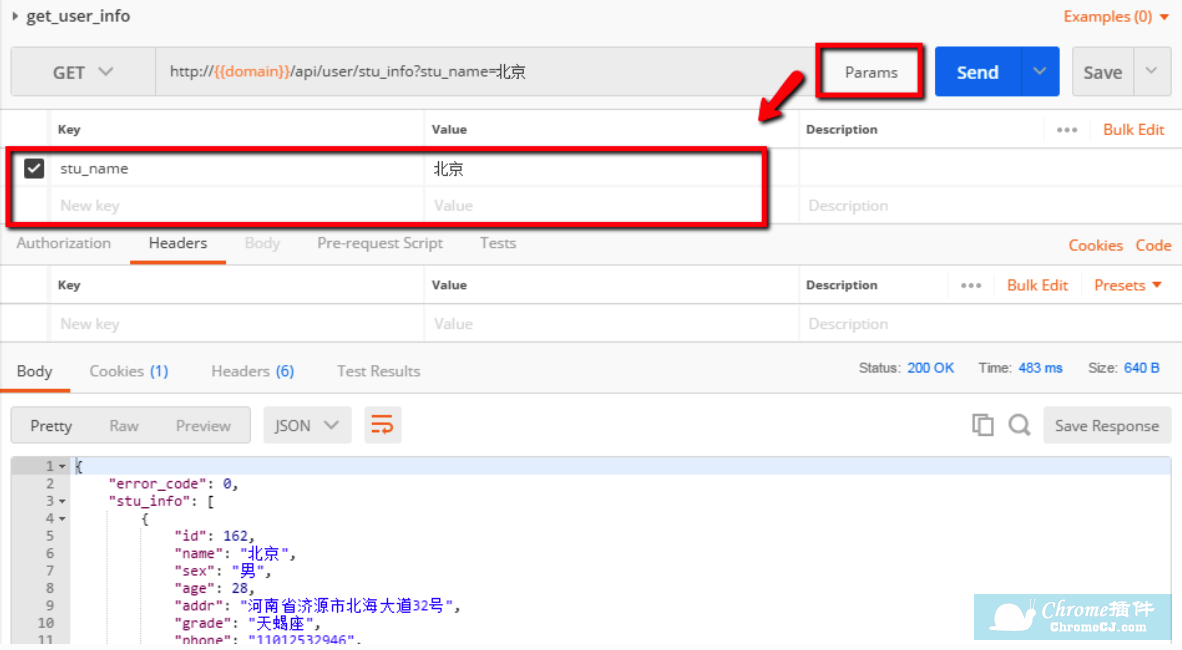
**1. GET 请求：**

平常我们拿一个url 访问页面，就是所谓的get 请求。

**例1.（stu\_info 接口：获取名字为xx 的信息）**

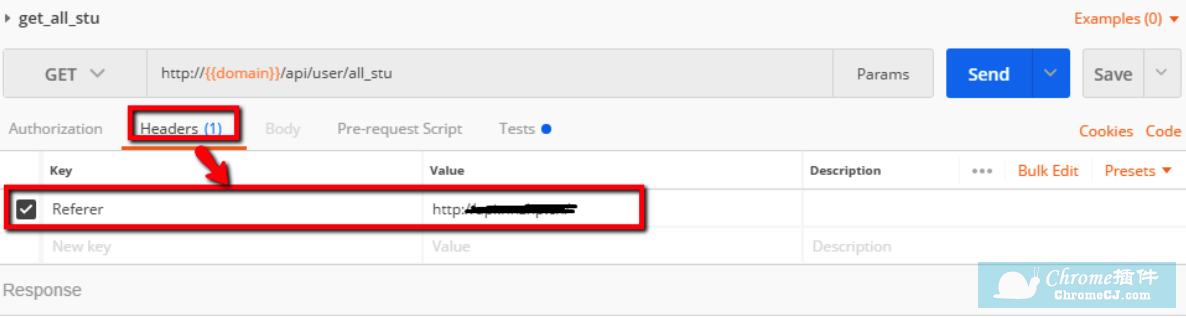
将请求方式设为GET, 同时输入完整的url， 跟访问浏览器一样， 或者输入接口**url**, 点击**"Params"** 输入需要的key 以及value， 点击"Send"， 即可在"Response" 中查看返回的结果。

**注意： get 请求的内容不能放在body里，并且长度有限制。由于对domain 设置了全局变量，所以，url 中用变量{{domain}} 做了替换，以下的请求中都应用了这个全局变量。**



**例2. （all\_stu 接口：获取所有用户的信息）**

除了例1这种GET请求，还有些接口发送GET请求时除了发送key-value, 还要将Headers的信息也一起发送过去，这样就可以借助Postman来实现。以下图为例，需要在Headers 里添加**"Referer"**信息，以获取所有用户的信息。



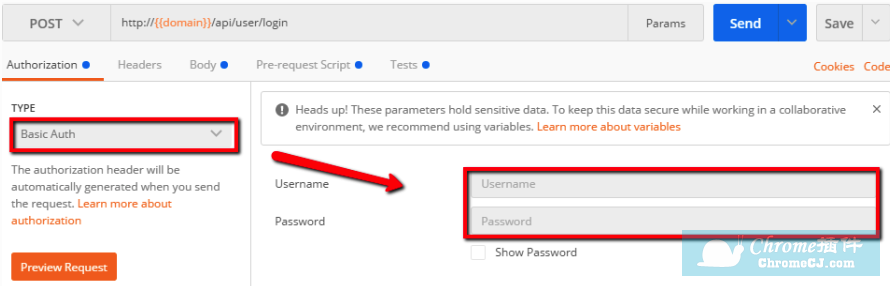
**2. POST请求：**

POST 请求不能像GET一样直接在浏览器输入就可以请求，需要借助工具来完成。

**使用方法：**选择请求方式为post，输入请求的url，以及输入必要的**"Authorization" , "header"** 和**"Body"** 数据。post请求可以发送key-value, json格式，file等。

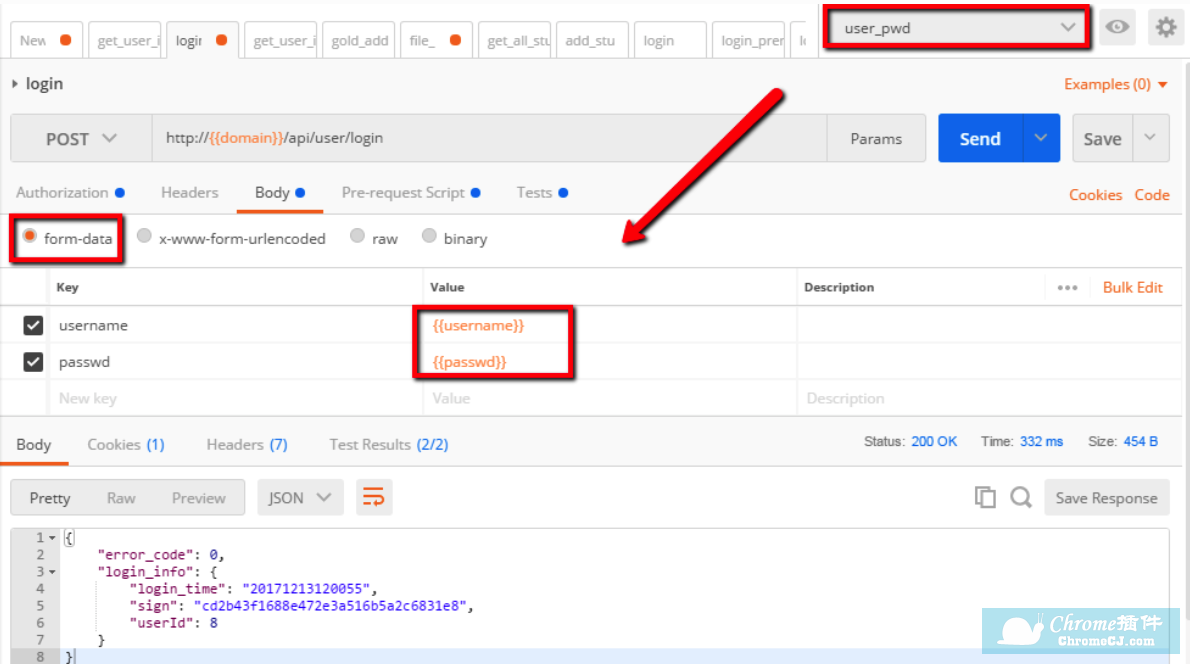
对于**"Authorization"**的使用，遇到比较多的是**Type** 为**"Basic Auth"**, 然后设置对应的**Username 和Password**。

这里的username, password的值可以通过设置环境变量的方式取得。



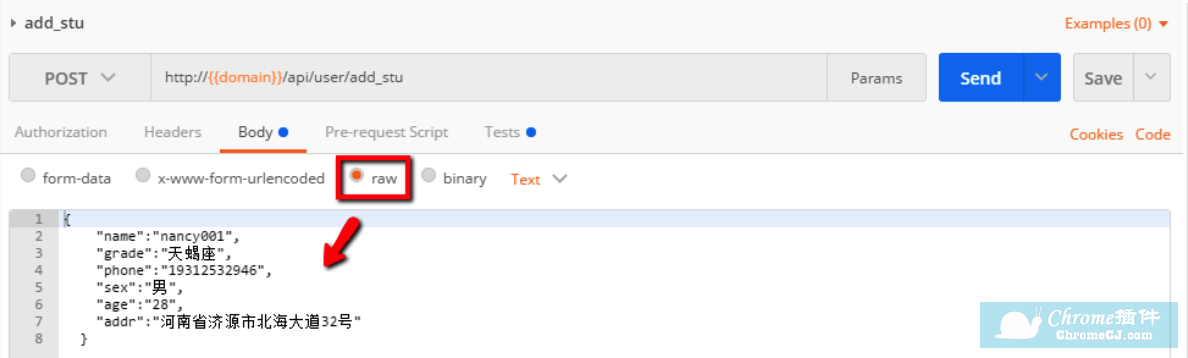
**a. 用Postman 发送key-value 的请求：**

以login 接口为例，在Body 中选取**"form-data"** 格式，输入所需的key-value, 选取对应的环境变量。



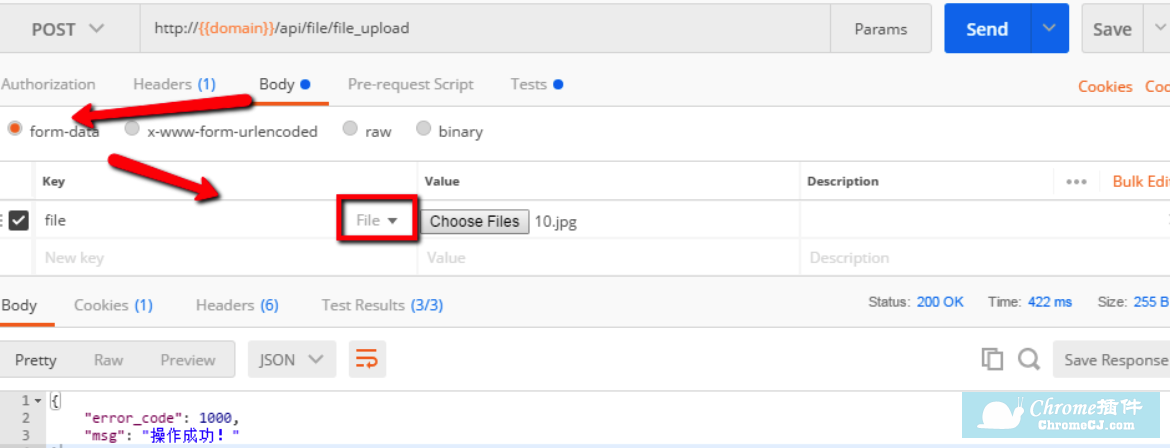
**b. 用Postman 发送json格式的请求：**

以add user 接口为例，在Body 中选取**"raw"** 格式，根据接口文档输入json 数据， 有需要应用环境变量的选取环境变量。



**c. 用Postman 发送文件的请求：**

以file upload 接口为例，在Body 中选取**"form-data"** 格式，在key 里输入"file"，在右边的下拉里选取类型为"File"，点击"Choose Files" 就可以上传本地文件了。



 三、Pre-requestScript 的使用

 对于环境变量和全局变量的使用，除了上面所讲的方法外，也可以用Pre-requestScript 方法。

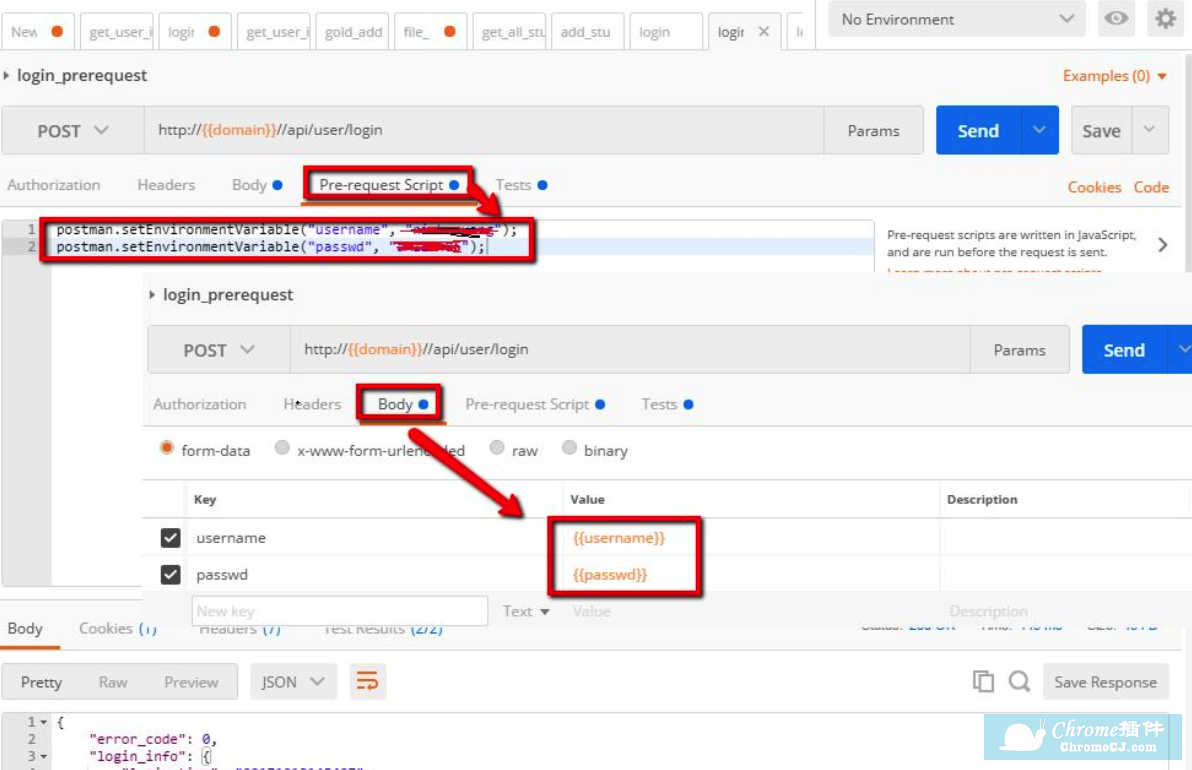
以login接口为例，在"Pre-requestScript"中设置环境变量 **"username", "passwd"**,在Body 中选取**"form-data"** 格式，输入所需的key-value, value即为变量**{{username}}, {{passwd}}**。

postman. setEnvironmentVariable (“key”, “value”);

postman. setGlobalVariable (“key”, “value”);

getEnvironmentVariable ("key");//获取key的环境变量

getGlobalVariable(“key”);//获取key的全局变量

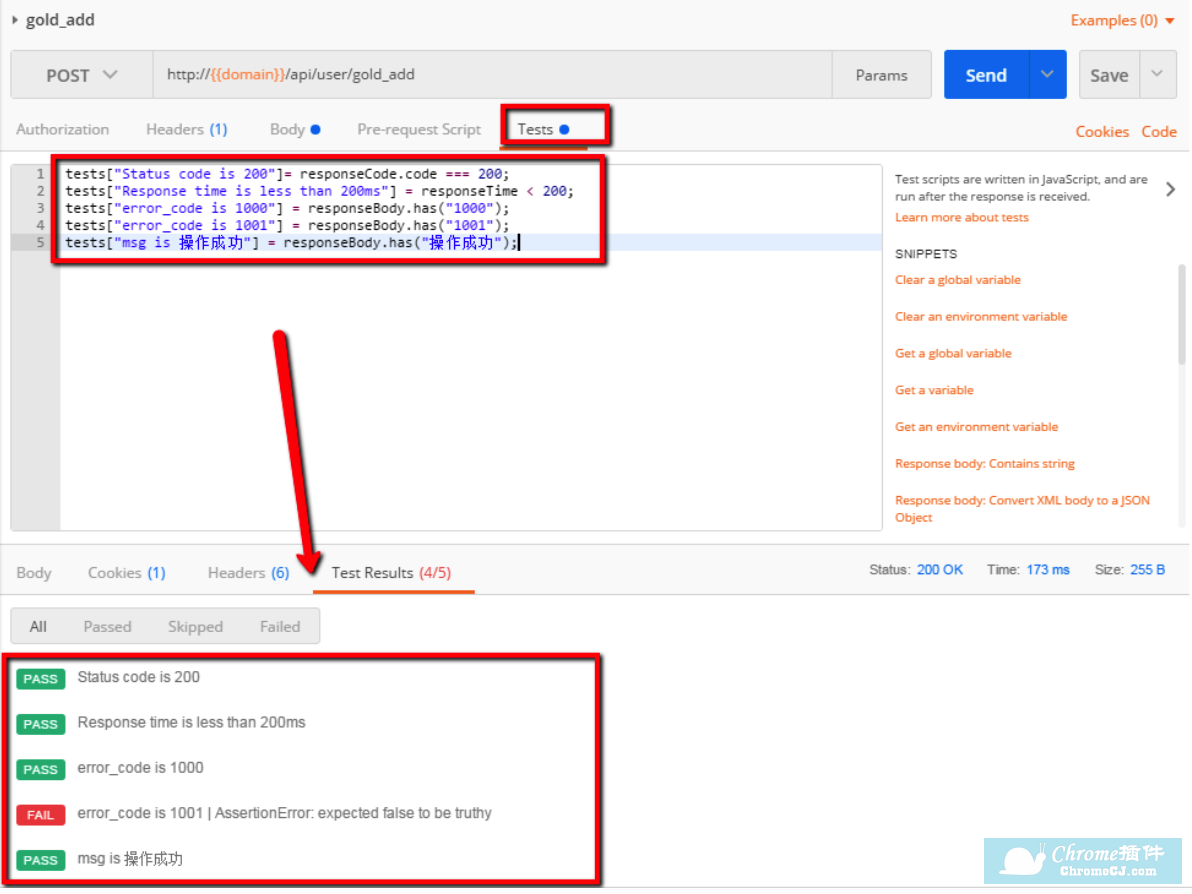


 四、Tests 的使用

**1. Tests 作为测试用例的应用**

Tests 主要用来设计用例，比如要测试返回结果是否含有某一字符串，就可以用到 Tests。以gold\_add 接口为例，来编写测试用例，分别测试返回的结果是否含有，如下所示：

响应成功返回PASS, 失败返回FAIL。



**常用tests如下：**

1.检查response body中是否包含某个string  
tests["Body matches string"] = responseBody.has("string\_you\_want\_to\_search");

**注意："Body matches string" 需唯一。**

2.检测JSON中的某个值是否等于预期的值

var data = JSON.parse(responseBody);

tests["Your test name"] = data.value === 100;

JSON.parse()方法，把json字符串转化为对象。parse()会进行json格式的检查是一个安全的函数。

如：检查json中某个数组元素的个数(这里检测programs的长度)

var data = JSON.parse(responseBody);

tests["program's lenght"] = data.programs.length === 5;

3.检查response body是否与某个string相等  
4.转换XML body为JSON对象  
var jsonObject = xml2Json(responseBody);

tests["Body is correct"] = responseBody === "response\_body\_string";

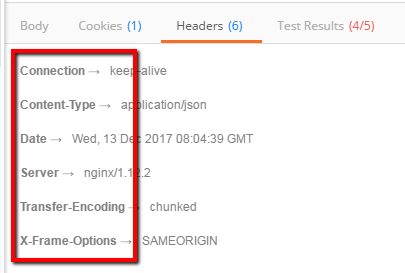
5.测试response Headers中的某个元素是否存在(如:Content-Type)

tests["Content-Type is present"] = postman.getResponseHeader("Content-Type");

//getResponseHeader()方法会返回header的值，如果该值存在

或者：

tests["Content-Type is present"] = responseHeaders.hasOwnProperty("Content-Type");  
上面的方法，不区分大小写。下面的方法，要区分大小写。



6.验证Status code的值

tests["Status code is 200"] = responseCode.code === 200;

7.验证Response time是否小于某个值  
tests["Response time is less than 200ms"] = responseTime < 200;

8.name是否包含某个值  
tests["Status code name has string"] = responseCode.name.has("Created");

9.POST 请求的状态响应码是否是某个值  
tests["Successful POST request"] = responseCode.code === 201 || responseCode.code === 202;

10.很小的JSON数据验证器

var schema = {

"items": {

"type": "boolean"

}

};

var data1 = [true, false];

var data2 = [true, 123];

console.log(tv4.error);

tests["Valid Data1"] = tv4.validate(data1, schema);

tests["Valid Data2"] = tv4.validate(data2, schema);

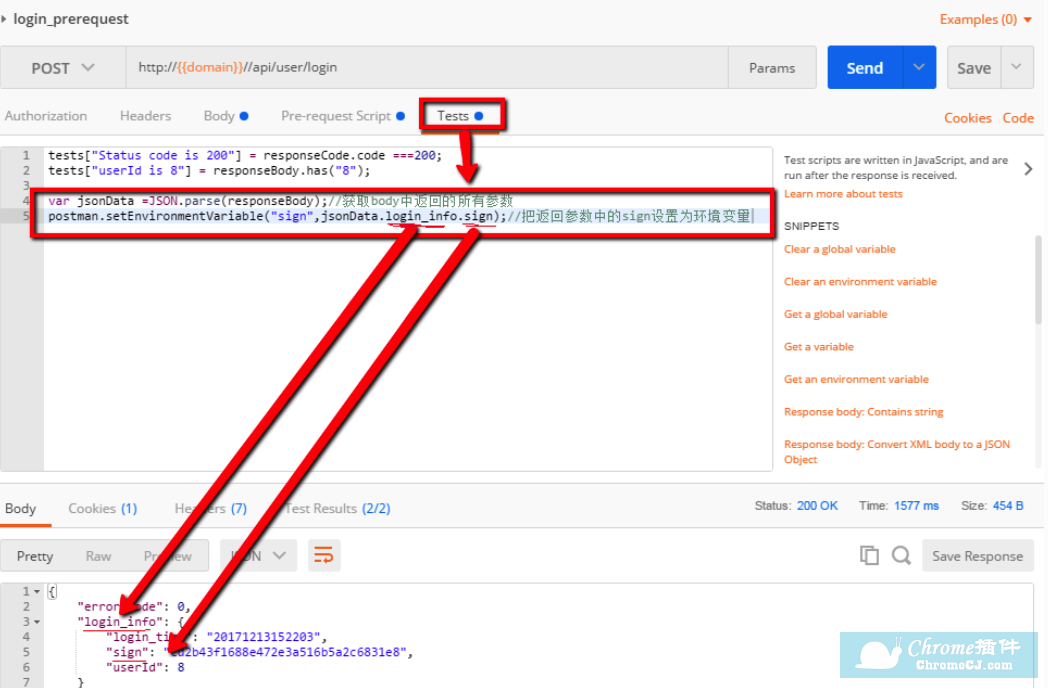
**2. 设置环境变量，建立多个接口间的关联**

以gold\_add 接口为例，由于该接口有权限验证，需要admin用户才可以做操作，需要添加cookie，所以需要将login 接口关联起来使用。

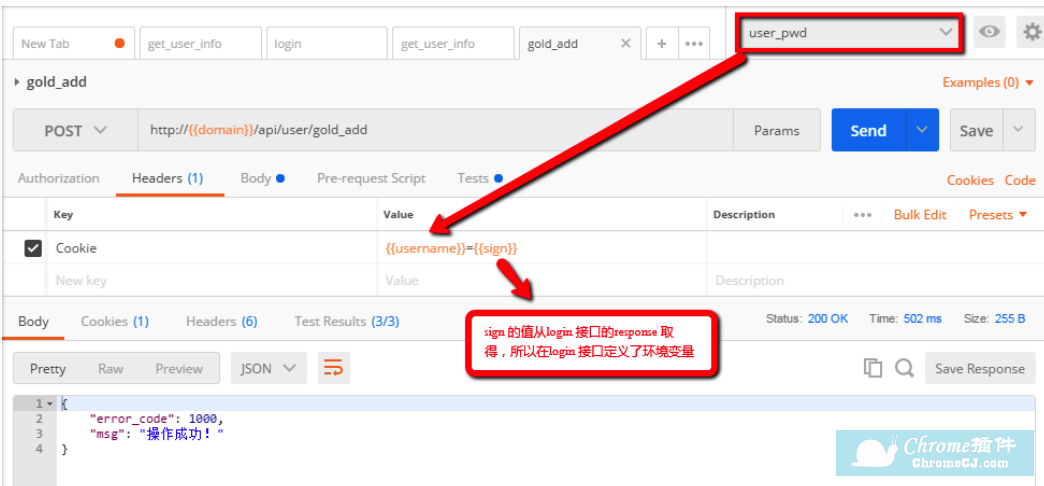
在login 请求中， 在"Tests"中加入

var jsonData =JSON.parse(responseBody);//获取body中返回的所有参数  
postman.setEnvironmentVariable("sign",jsonData.login\_info.sign);//把返回参数中的sign设置为环境变量

这样sign 就可以作为环境变量，应用于gold\_add接口。



在gold\_add 接口中，{{username}} 由于在环境变量"user\_pwd" 中已经设置，可以直接输入变量名, {{sign}} 动态取得login接口中 Response 的 "sign" 值，{{sign}}已在上面login 请求的"Tests" 中定义。



以上就是postman接口测试教程与实例分享总结。更多postman使用方法请关注我们chrome插件网。