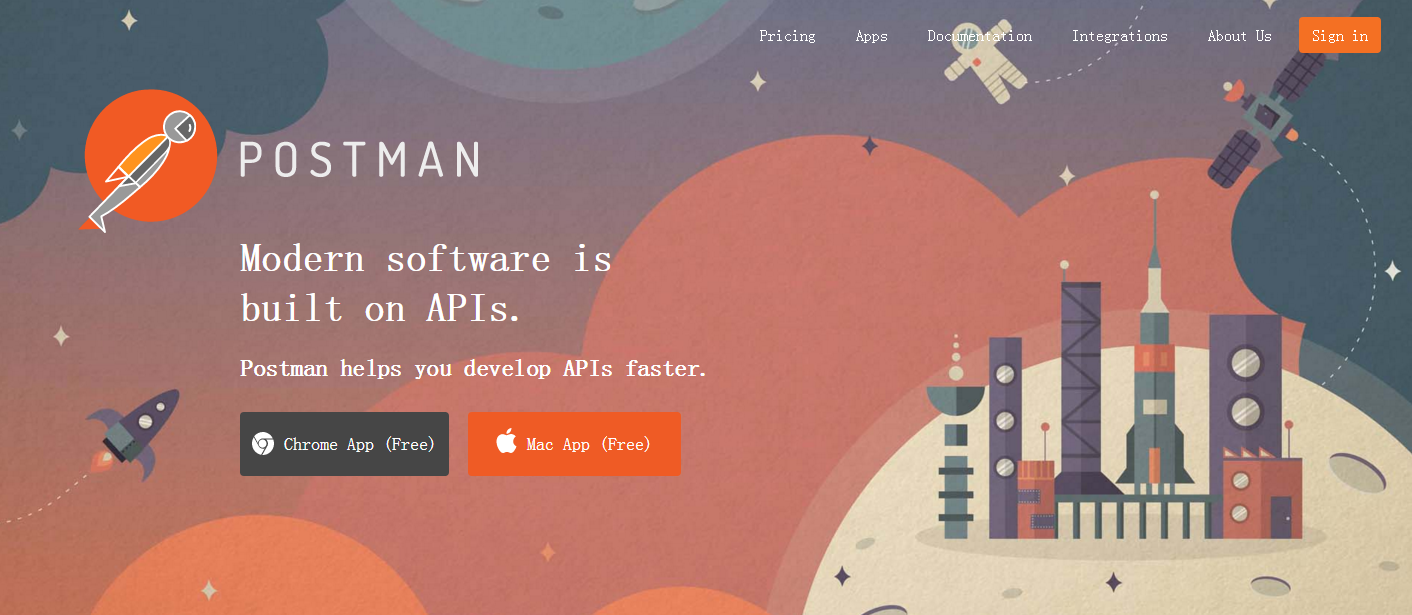
# 准备工作：

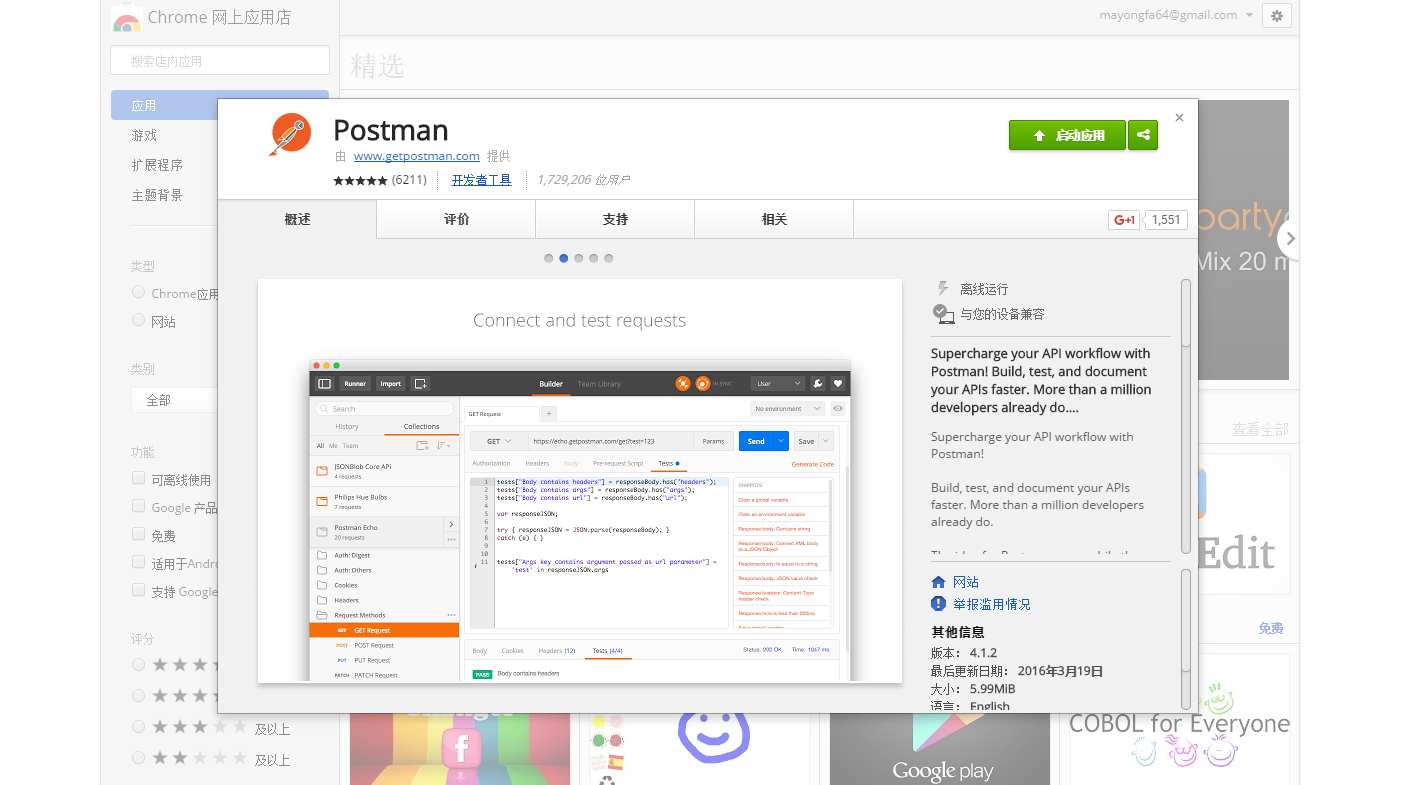
## postman安装步骤

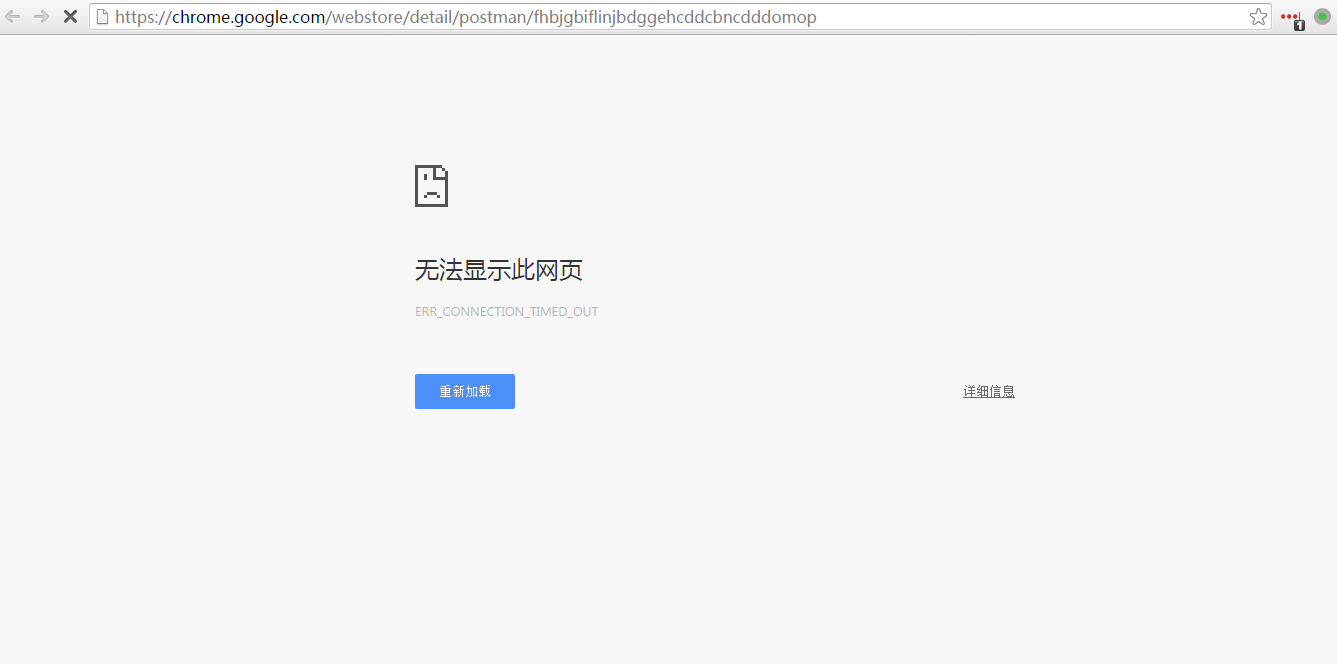
**(1). 官网安装（别看）**

* 打开官网，[https://www.getpostman.com](https://www.getpostman.com/)



* 点击那个灰灰色的「Chrome App (Free)」按钮。正常情况会跳转到Chrome网上应用店界面，但是，由于，嗯，你懂的！你一般看到的是第二张图片的样子，到处充满着忧伤...

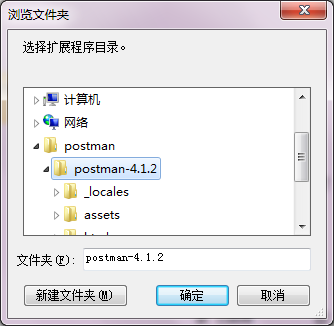
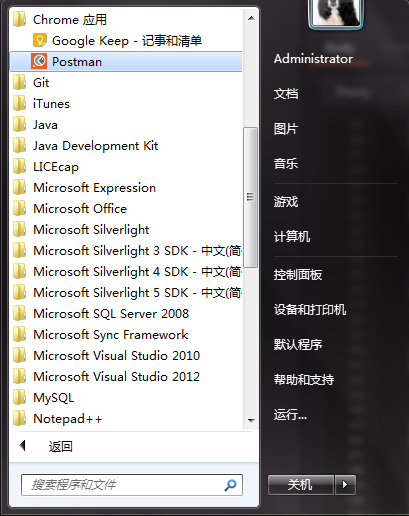




先请擦掉眼泪，不要忧伤，我们依然可以好好的。

**2. 非官网安装**

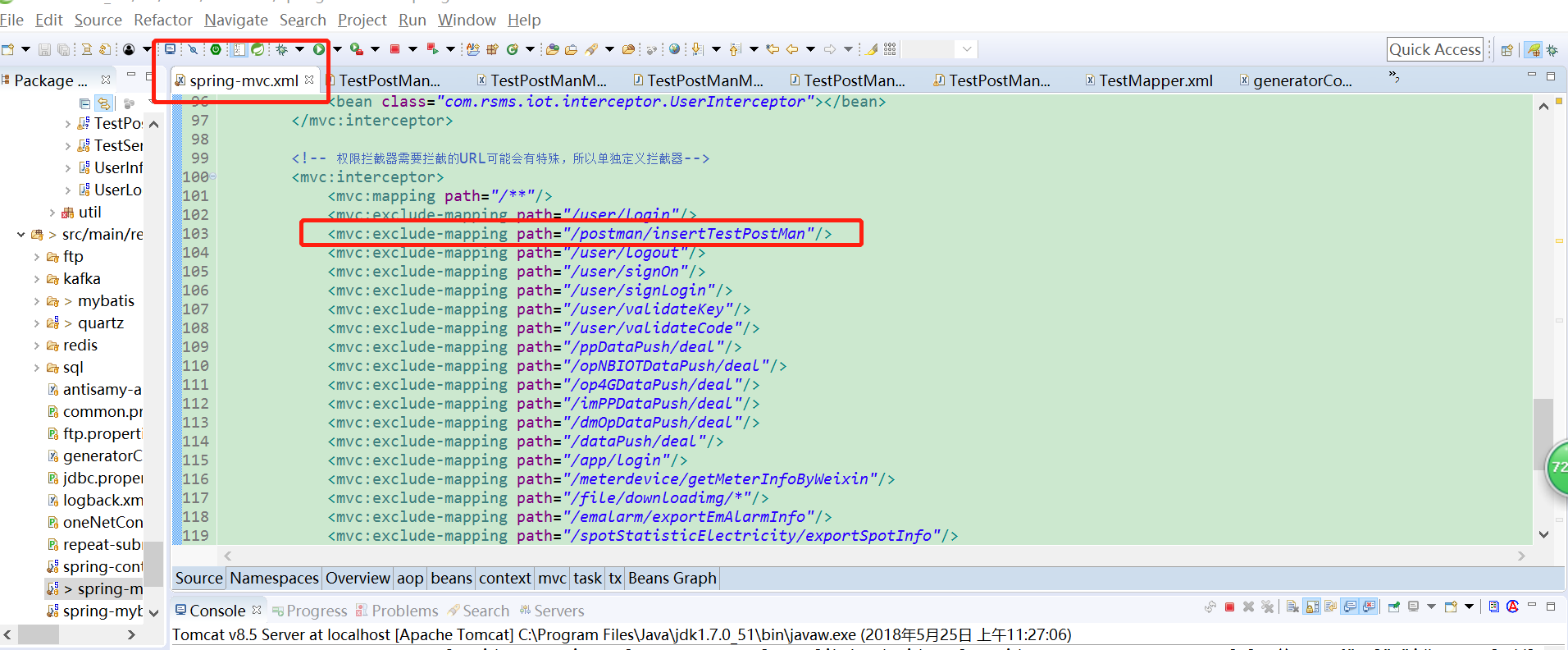
这是一种直接通过打包已经安装的扩展程序的方式，来进行我认为的「非法安装」，但没办法，只能这样。

* 我会给你一个安装包，见附件。你应该下载下来，解压缩到你喜欢的位置。
* 打开 Chrome 浏览器的「扩展程序」  
  
* 点击「加载已解压的扩展程序...」按钮，找到你刚刚下载的安装包的位置，点击确定。  
  
* 你去看看 Windows 的开始菜单，会突然发现多了一个「Chrome 应用」的文件夹，你点击里边的 Postman 试一下嘛。  
  

好了，到这里通过「非法安装」安装的 Postman 已经完毕。

## 2．在spring-mvc.xml文件中将所需要测试的接口地址添加上

## 



[**postman进行http接口测试**](https://www.cnblogs.com/ceshi2016/p/6613256.html)

## 优点：

1、支持用例管理

2、支持get、post、文件上传、响应验证、变量管理、环境参数管理等功能

3、支持批量运行

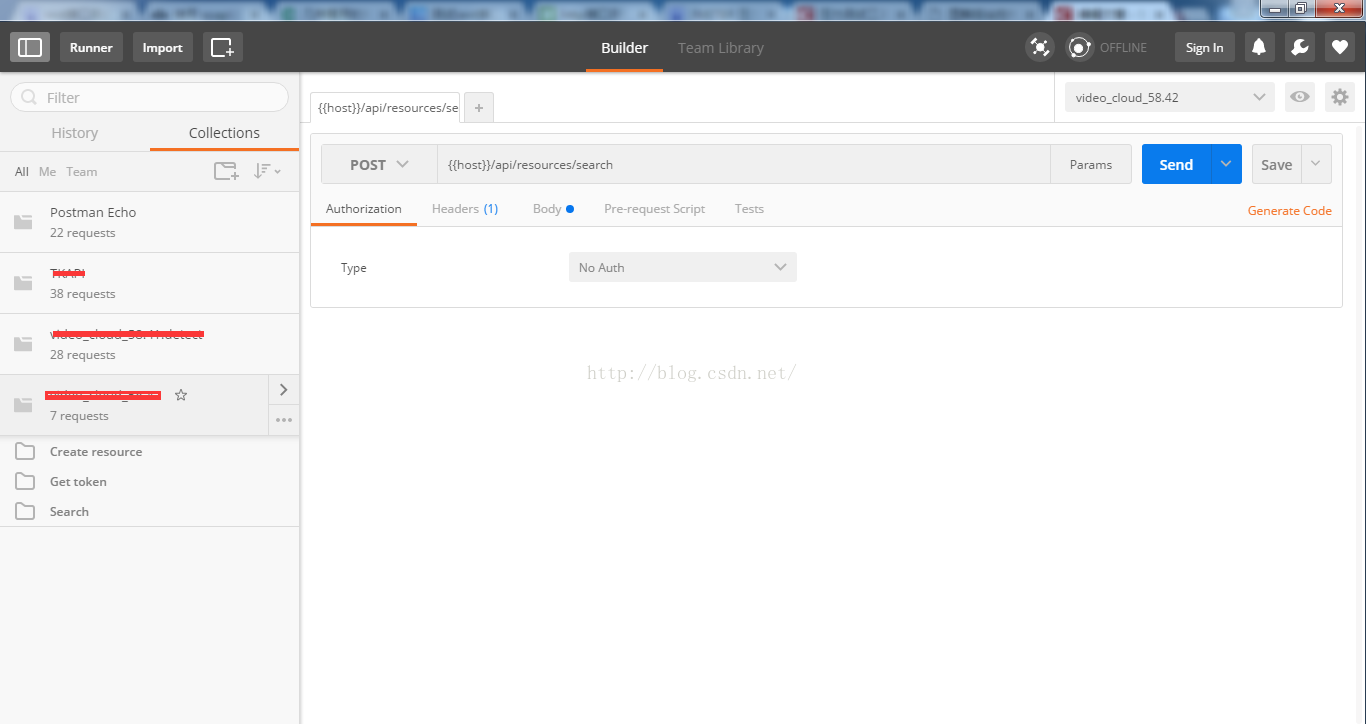
4、支持用例导出、导入

5、支持云端保存用例【付费用户】

可以说POSTMAN满足了HTTP接口测试的大部分功能，只有少部分的功能不被支持，比如：请求流程的控制；前面说了这么多，接下来我们就看看POSTMAN的安装与使用吧。

## 1、什么是POSTMAN

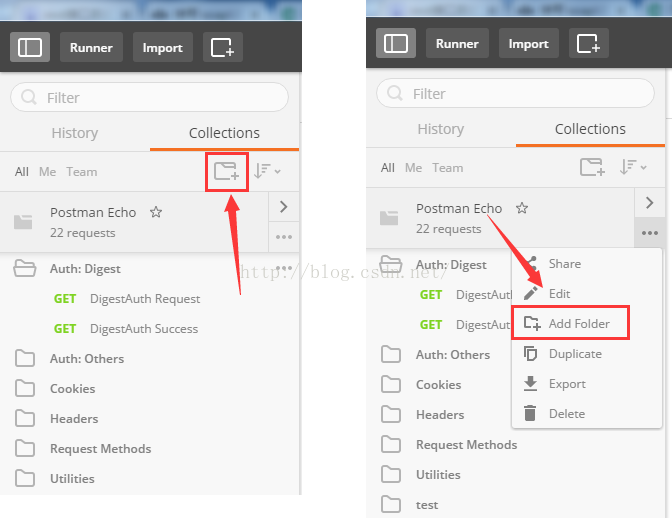
POSTMAN是一个Chrome的一个插件工具，我们可以通过Chrome的应用商店进行搜索并安装，安装完成会在桌面上显示一个postman的图标，每次点击这个图标就可以启动POSTNA的界面。



启动过后就是上面的界面了，左边是用来管理用例的目录结构，右边是具体某个用例的请求内容的参数及响应内容；默认的postman会自带一个demo的项目叫“POSTMAN Echo”，里面有各种场景的用例demo，对于新上手的同学可以通过查看这些demo用例来学习如何使用POSTMAN。

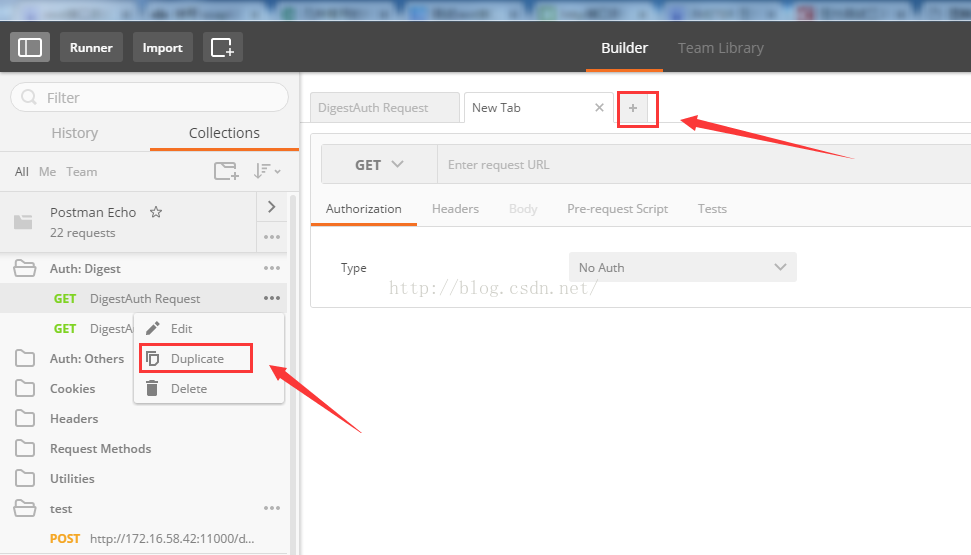
## 2、新建一个项目

直接点击左边栏上面的添加目录图标来新增一个根目录，这样就等于新建了一个项目，我们可以把一个项目或一个模块的用例都存放在这个目录之下，并且在根目录之下我们还可以在建立子目录来进行功能用例的细分，具体见下图。



## 2、新增一个用例

创建了项目目录后我们就可以新建用例了，具体是点击右侧区域的+号来新增一个空用例的模板，也可以通过复制一个已有用例来达到新建一个用例的目的，2种方法见下：

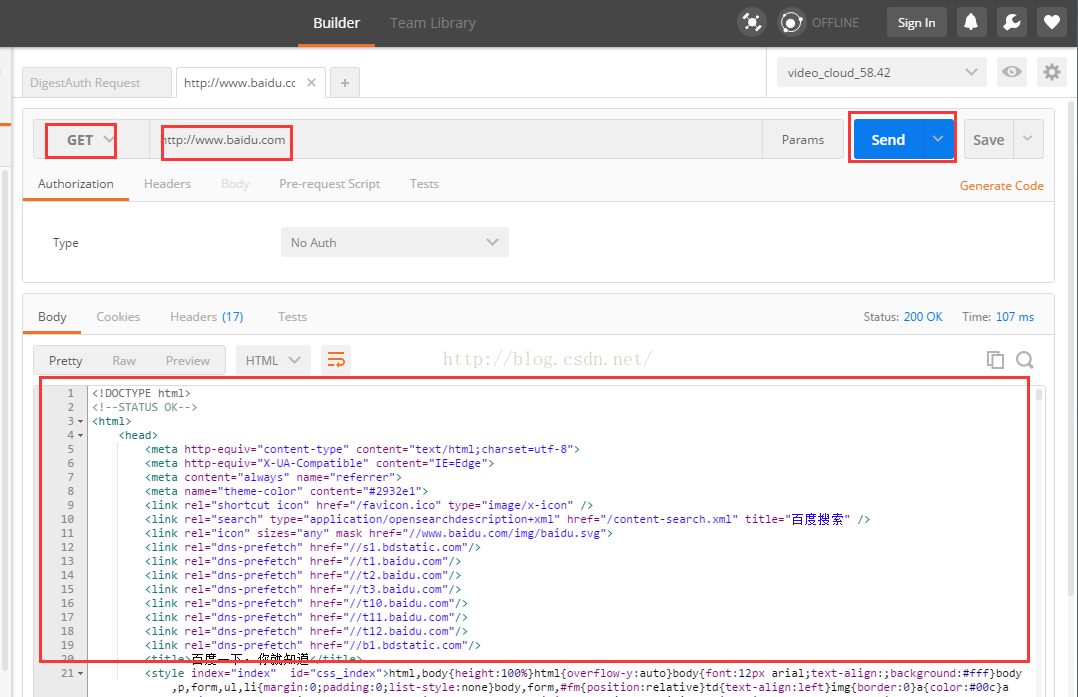


## 3、添加请求信息

新建的用例请求内容为空，我们需要添加相应的请求信息，这部分的操作都在右侧的信息区域，一般流程如下：

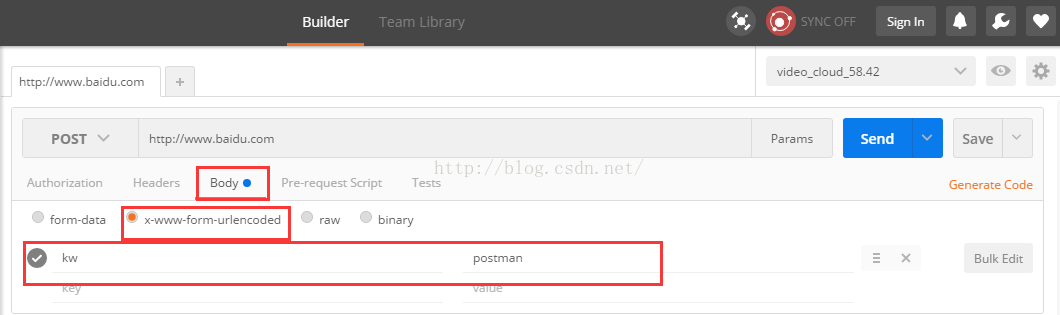
1. 选择一个请求方法，如：get或post
2. 填写请求的url，如：<http://localhost:8080/rsms_2.0/user/validateKey>
3. 如果是get则请求参数直接写在url后，用？连接
4. 如果是post则请求添加在body中
5. 点击“send”发送请求
6. 查看请求响应内容

## 4、get请求参数

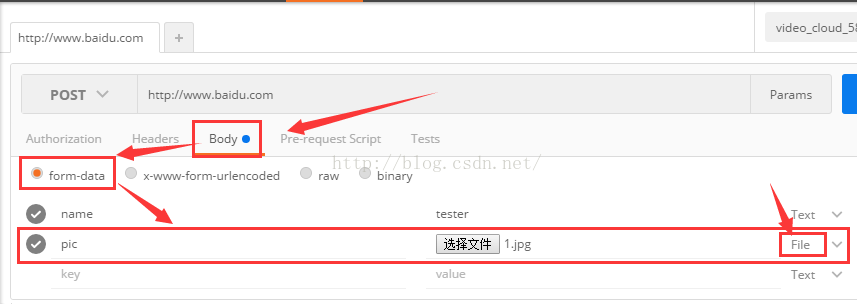


## 5、post请求参数

post请求的主要的特点是把请求数据放在body中，而非url后



上面的样例是post方式传输普通参数，如果我们需要发送带文件的请求时，就要改下请求格式了，具体如下：

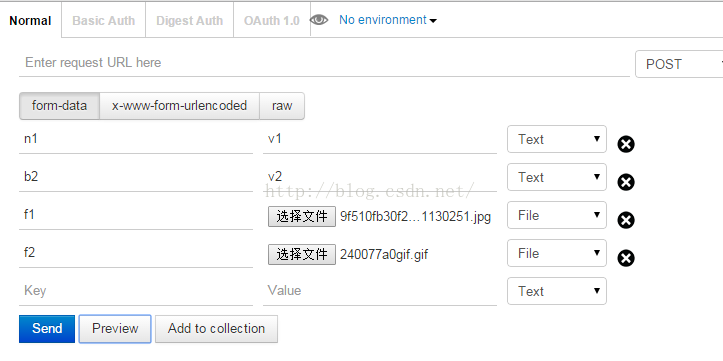


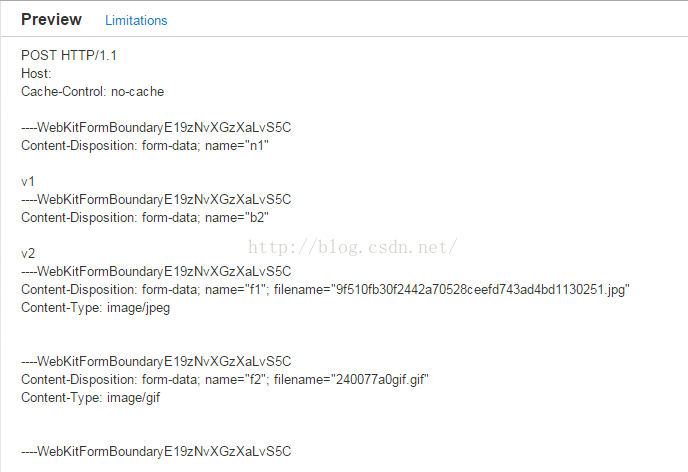
注意上面标红框的内容，都是必须要对应上。

## form-data、x-www-form-urlencoded、raw、binary的区别

### 1、form-data:

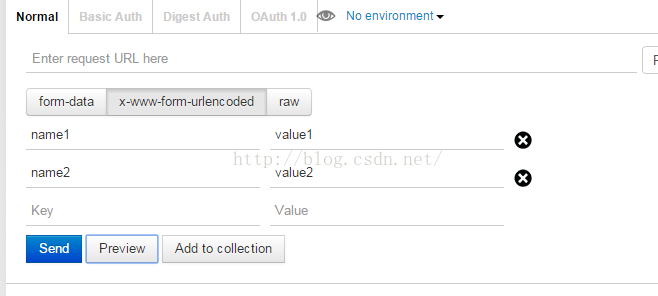
就是http请求中的**multipart/form-data**,它会将表单的数据处理为一条消息，以标签为单元，用分隔符分开。既可以上传键值对，也可以上传文件。当上传的字段是文件时，会有Content-Type来说明文件类型；content-disposition，用来说明字段的一些信息；由于有boundary隔离，所以multipart/form-data既可以上传文件，也可以上传键值对，它采用了键值对的方式，所以可以上传多个文件。

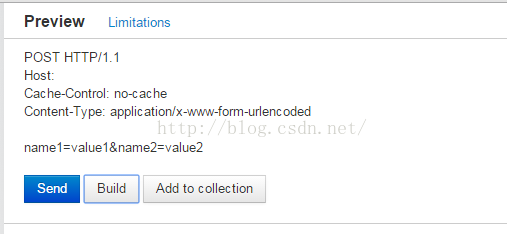




### 2、x-www-form-urlencoded：

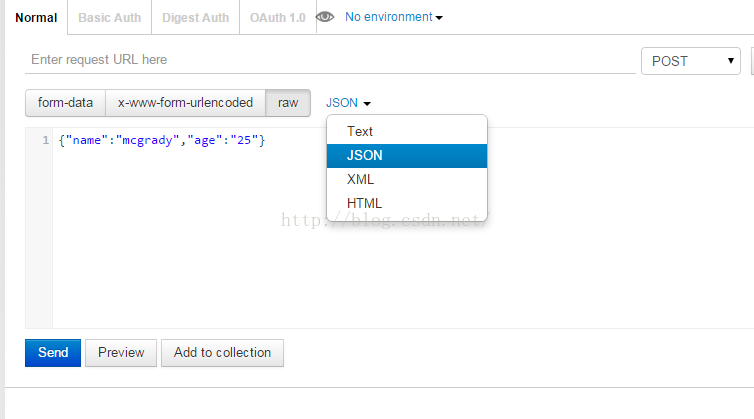
             就是**application/x-www-from-urlencoded**,会将表单内的数据转换为键值对，比如,name=java&age = 23

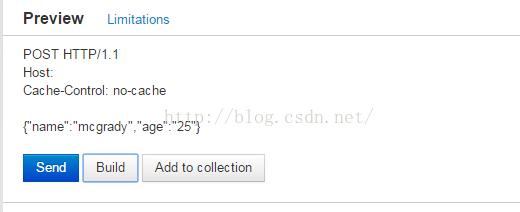




### 3、raw

            可以上传任意格式的文本，可以上传text、json、xml、html等





### 4、binary

          相当于**Content-Type:application/octet-stream**,从字面意思得知，只可以上传二进制数据，通常用来上传文件，由于没有键值，所以，一次只能上传一个文件。

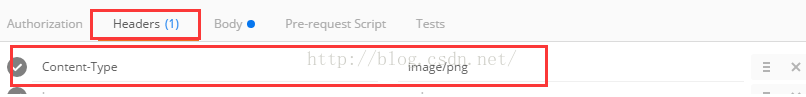
* **multipart/form-data与x-www-form-urlencoded区别**

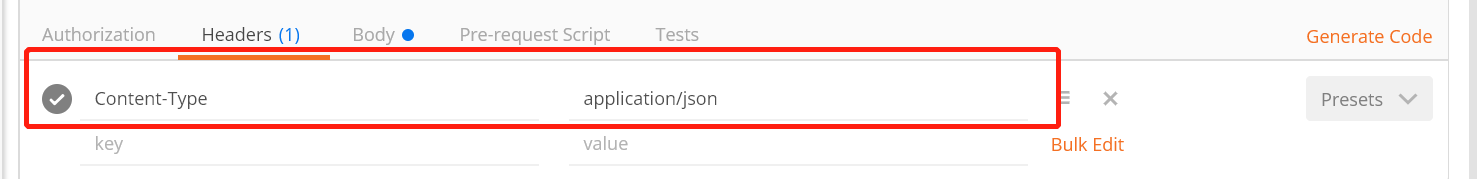
multipart/form-data：既可以上传文件等二进制数据，也可以上传表单键值对，只是最后会转化为一条信息；

x-www-form-urlencoded：只能上传键值对，并且键值对都是间隔分开的。

## 6、添加头信息

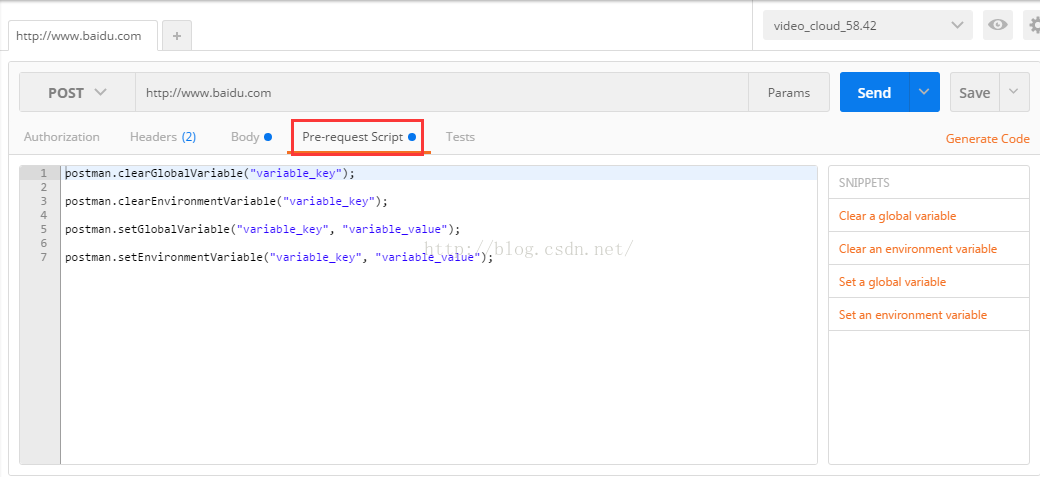
有些时候请求时还需要一些特定的头信息，postman同样可以完美的支持，直接点击Headers标签就可以进行请求头的信息设置





**7、预处理和结果检查（Pre-request Script/Tests）**

预处理主要是对一些环境变量之类的进行设置，相当于数据初始化；如图：



响应处理就是对响应结果进行分析和验证，比如检查code是不是200，内容是不是等于具体某个值，是否包含特定的值等等。



因为预处理和结果检查都是使用js作为脚本语言，所以你还可以进行任意的js可以实现的场景，来辅助测试。

**8、全局变量与环境变量**

全局变量我们可以自己在预处理和结果处理2个脚本环境里进行赋值，在具体的测试数据里我们就可以直接使用，具体的使用方法是为：{{variable\_key}}；比如你在脚本中可以设置全局变量：

**[javascript]** [view plain](http://blog.csdn.net/five3/article/details/53021084) [copy](http://blog.csdn.net/five3/article/details/53021084)

[在CODE上查看代码片](https://code.csdn.net/snippets/1964639)

1. postman.setGlobalVariable("username", "tester");

那么在用例数据项里面我就可以这样使用，{{username}}，用来代表具体的tester值，具体见下图

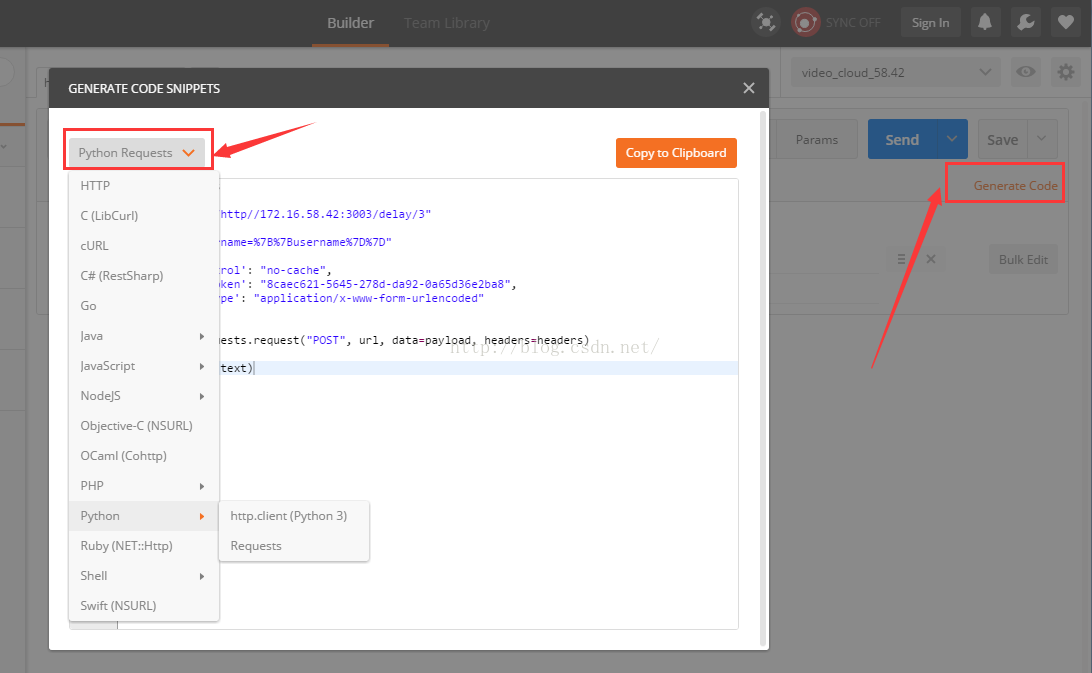


而环境变量的设置可使用与全局变量基本一样，只是环境变量我们还有另外一个入口可以进行设置，那就是环境配置管理中，我们可以预先建立若干和与环境相关的一套变量，根据实际的测试需求在执行前选择对应的环境变量模板，这样可以快速切换测试服务器与线上服务器之前的环境差异。比如：配置2套环境变量模板，一套url是测试环境，另一套为线上环境，根据测试对象不同我们选择不同的环境变量模板就行了，而不再需要修改测试数据中的url了。

上面我们就把请求的host提取出来，然后在不同环境变量模板里使用不同的url值，后面我们就可以通过选择不同的环境变量模板来进行对应的请求测试

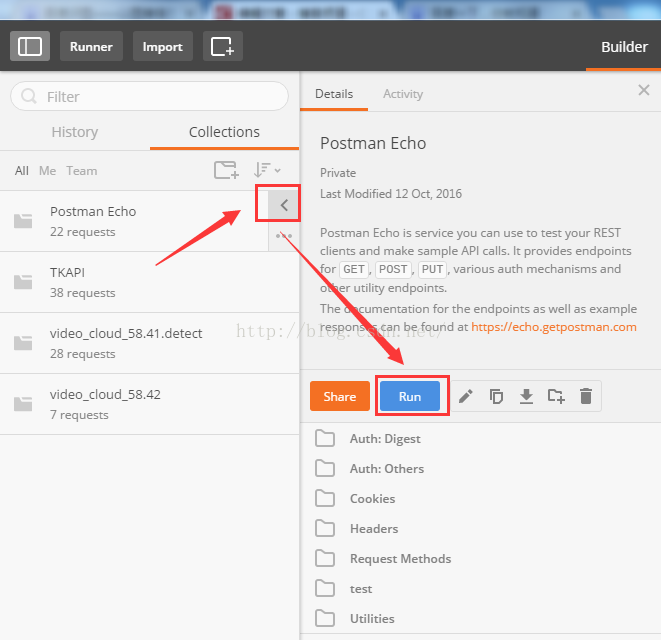
**9、导出用例为代码**

POSTMAN还有一个很赞的地方就是导出用例为CODE，即如果你编写好了用例之后可以通过点击“Generate  Code”来一键生成代码，并且还有好多语言和类库可以选择，帧的是棒棒哒！

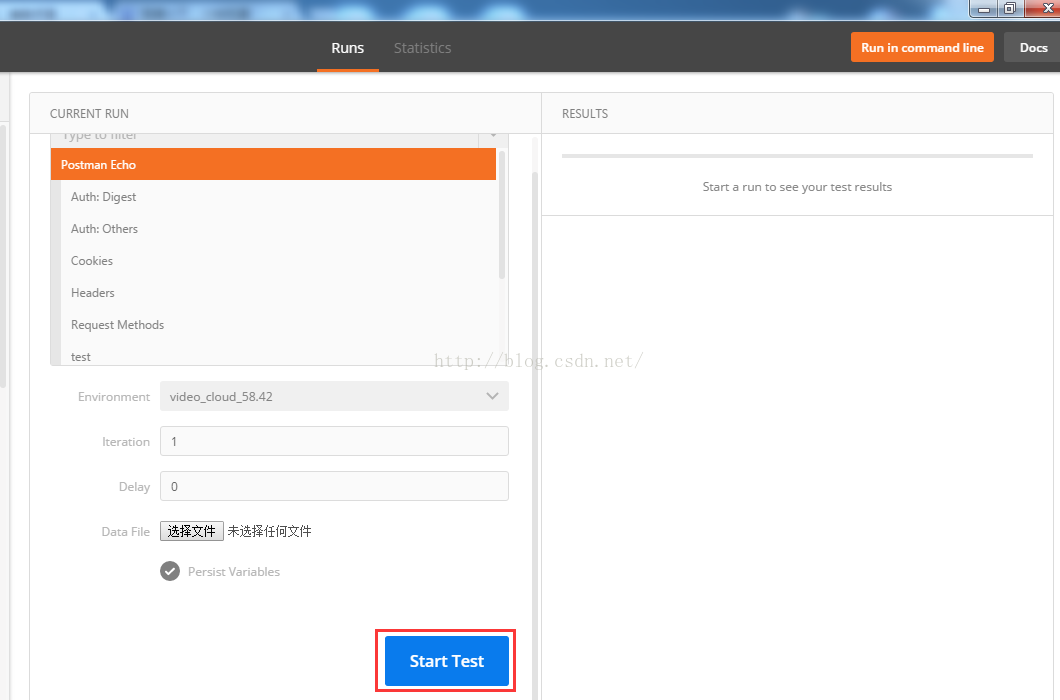


**10、批量执行用例**

最后我们再来看看POSTMAN的批量执行功能，这个功能由单独的runner来负责的，我们需要在另外的界面进行操作，具体如下：



依次点击上面的按钮就会出现runer界面，如下直接点击“Start Test”即可



# 实例

选择post方式，填写请求地址<http://localhost:8080/rsms_2.0/postman/insertTestPostMan>，在Headers中添加如下内容，在body中添加需要的参数，点击send.这样这个请求就发送出去啦

# 

# 

# 

# 

# 