Postman使用方法

**1.POSTMan的安装**

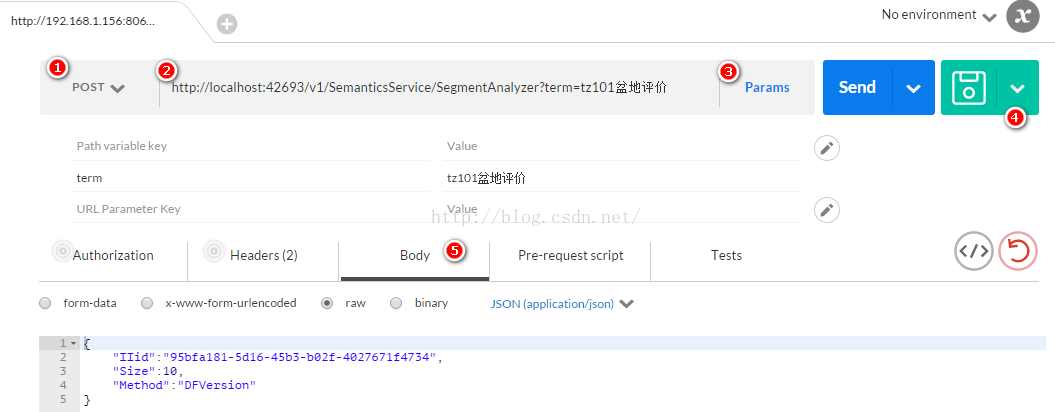
http://blog.csdn.net/shiyaru1314/article/details/49303317

亲身测试过，是可以安装成功的哦

PostMan 作为API测试的可视化工具，对于无页面的服务来说是十分方便的。

**2.经常使用的功能**

**2.1 开发过程中的测试：类似单元测试**

****

看图说话：

1.指的的是发送请求的方式，有很多种，点击下拉列表，一看就知道了。一般就用Get和Post

2.是服务地址

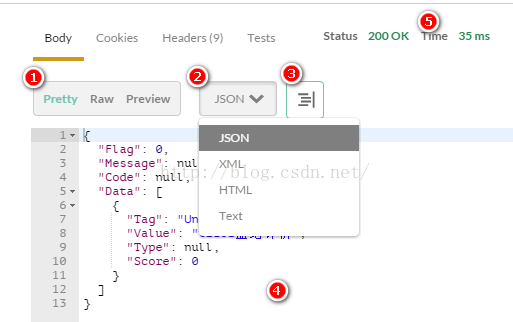
3.简单的参数的构造

4.可以保存当前所有的设置，也可以从这里添加到collection中

5.可以构造请求体，请求体的格式也可以是多种多样的哦，这里所用的是Json格式的，格式多样

比如也可以是Text格式

通常简单的测试单个的功能，使用上面的方式就可以了。



1.格式的形式

2.数据的展示形式

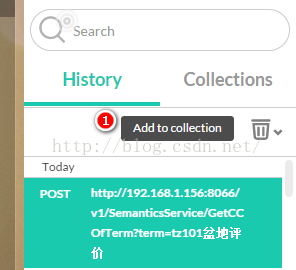
3.可以去除一些无用的数据，简单化

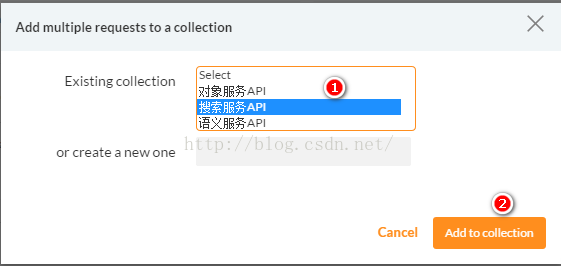
4.数据展示空间

5.执行时间，一般200ms以内，算是比较好的实现

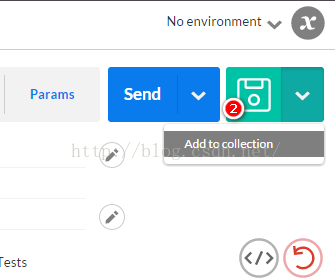
**2.2开发之后的测试，或是做当前API接口的展示作用**

第一种添加到collection的方法



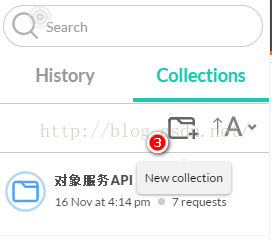


第二种添加到collection的方法（上面的点击Save as就跳转到下面的窗体）





第三种添加的方法

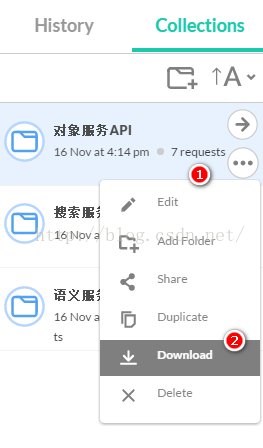


这种一般是建一个壳子，再将功能添加到壳子下面便于查找

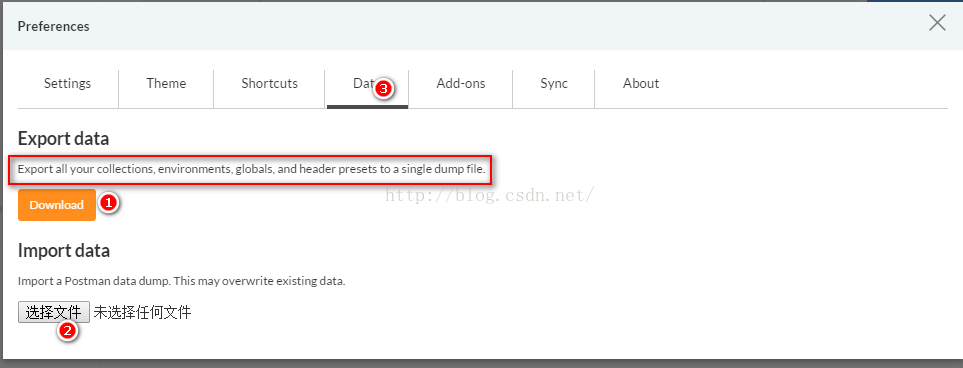
添加完所有的功能，并将其划分到不同的类别中，一重要动作是必须执行的，那就是

**3.将当前做的保存到文件中**

这样就可以共享，大家一起来测试API服务的功能，参数结构也都确定，只需修改其值就能够达到测试的目的。总之感觉非常好用的。



点击setting里面，也可以进行文件的导入导出，这里可以导入到一个文件当中



导出的文件长什么样子。要不要看看



**4.新版的postman V4.6中可以使用test功能**

这给我们提供了自动化测试的方便

首先需要写测试用例

var responseJson;

try{

    tests[" body has Count"] = responseBody.has("count");

 tests[" body has Docs"] = responseBody.has("metadatas");

responseJson = JSON.parse(responseBody);

tests[" doc0 contains iiid"] = "iiid" in  responseJson.Data.metadatas[0];

}

catch(e){

}

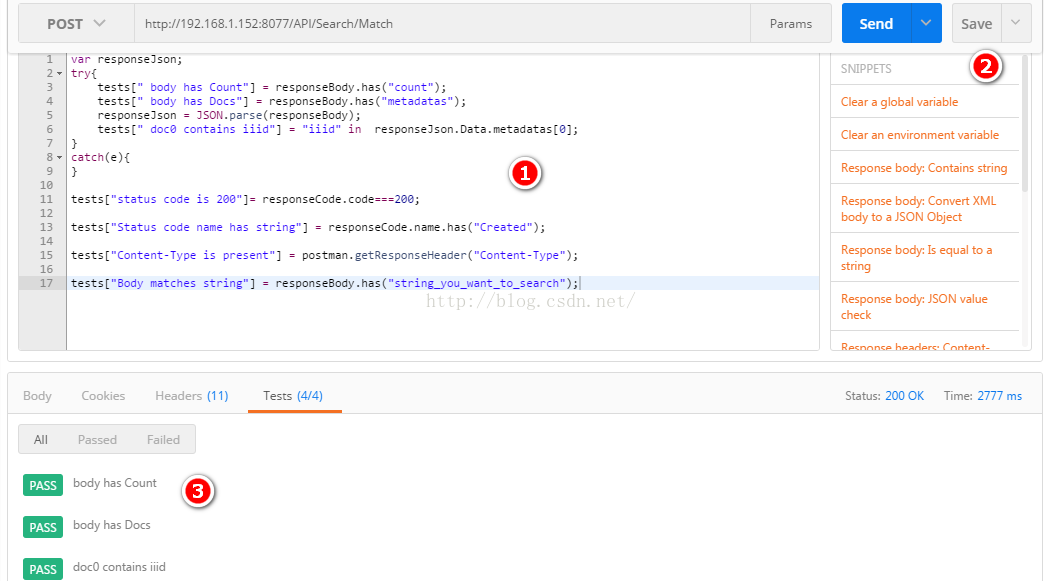
tests["status code is 200"]= responseCode.code===200;

tests["Status code name has string"] = responseCode.name.has("Created");

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

tests["Body matches string"] = responseBody.has("string\_you\_want\_to\_search");

如上是一些简单的测试代码示例，主要就是一些判断，判断结果和预期结果是否一致



上图就是新版本的postman的test功能，

其中

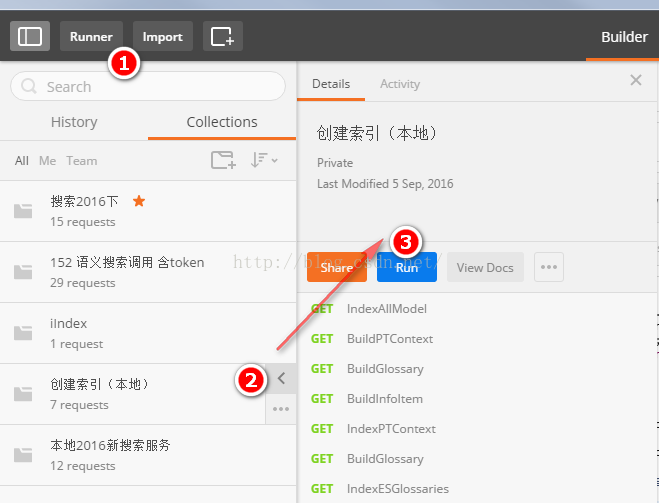
* 部分1是测试代码书写位置
* 部分2是一些常见的测试代码的自动生成
* 部分3是执行完毕的测试结果，pass代表通过

以上只是单个服务测试的结果

我们还可以进行批量的测试

点击系统左上角的"runner"按钮，会打开一个新的页面，会包含所有的collections

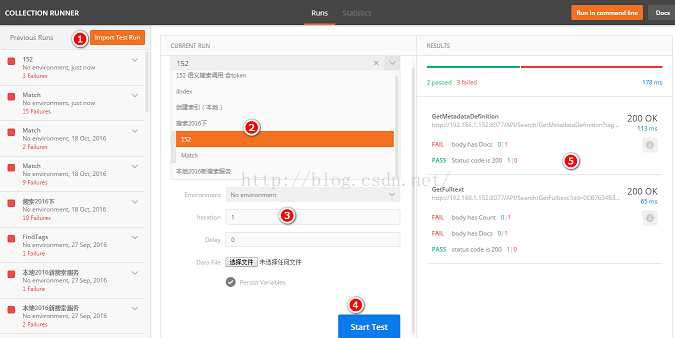
也可以在单个集合的包文件夹上点击按钮 “<”



可以直接点击左上角1的“runner”按钮，针对全部的collections

可以点击某个collection包文件夹的“>”按钮，然后再点击“run”按钮。针对单个collection

点击之后打开一个新的页面如下



1.可以再这个页面导入测试的用例

2.选择需要测试的内容

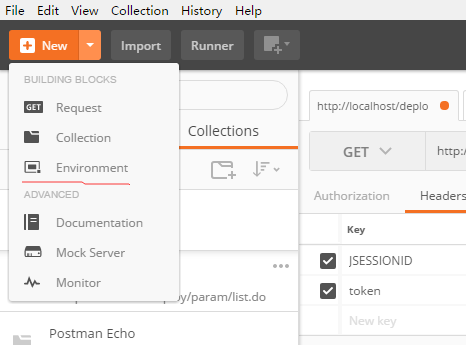
3.设置迭代次数，测试执行次数

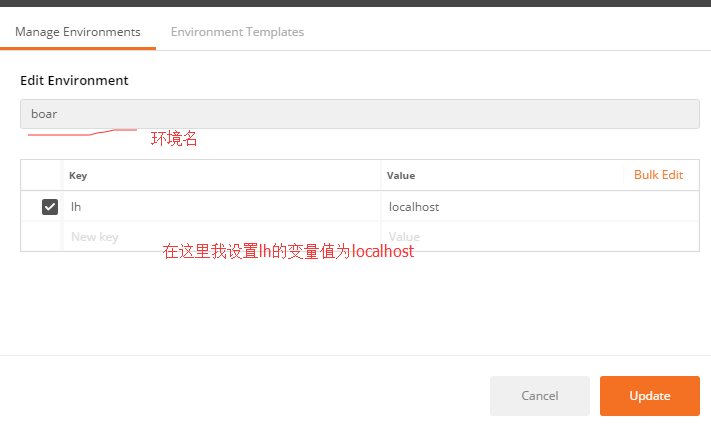
4.点击“start test”按钮执行测试

5.测试结果展示，pass表示通过 fall表示失败

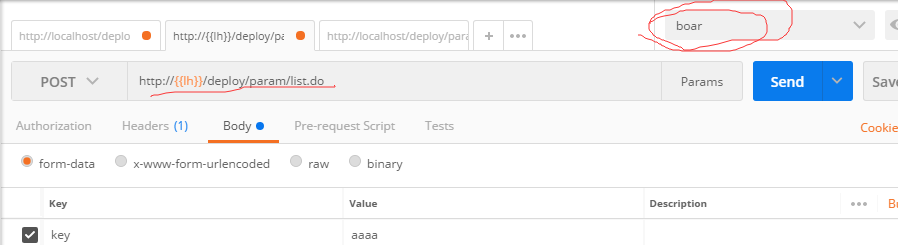
**5.添加环境变量**

方法1：





使用：



方法2：

