

VSCode的REST Client指南, 超好用的HTTP客户端工具

1 简介

在做Web应用开发的时候, 经常需要测试Web接口, 就需要一些客户端来发送HTTP请求到服务端。常用的客户端工具有Postman, curl等, 经过多次使用, 我觉得VSCode的REST Client还是非常好用的, 介绍给大家。

它是VSCode的一个插件, 要安装才能使用:



REST Client humao.rest-client

Huachao Mao | 1,432,232 | ★★★★★ | Repository | License

REST Client for Visual Studio Code

Install

主要特性有:

在文本编辑器中就可以直接发送请求和查看返回;

可以直接发送curl命令;

可以将HTTP请求复制为curl命令, 这样可以在其它linux终端上执行;

可以管理请求历史: 保存、查看、清除等;

多个请求在一个文件上, 用###分割;

可以保存返回体到本地;

生成多种语言的客户端代码, 如python/java等;

Cookies功能;

最重要的一点: 极其强大的变量管理功能。

接下来将通过示例介绍这些特性。

2 基本功能

我先新建一个文件pkslow.http, 后缀要是http或rest才可以被识别。

2.1 链接即请求

只要一个链接就可以了:

```
pkslow.http U ×
pkslow.http > ...
Send Request
1 https://www.pkslow.com
2
```

不用其它内容, 它就会显示Send Request, 然后点击它, 就能发送GET请求了。

2.2 多个请求

如果要多个请求, 就通过###来分割。我们接着定义一个更完整的GET请求吧。

```
https://www.pkslow.com
###
GET https://www.pkslow.com HTTP/1.1
```

点击Send Request, 会发送请求, 返回体在右边:

```
pkslow.http U ×
pkslow.http > GET /
Send Request
1 https://www.pkslow.com
2
3 ###
Send Request
4 GET https://www.pkslow.com HTTP/1.1
5
6
7
8
```

Response(136ms) ×

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.18.0
3 Date: Sat, 29 May 2021 08:13:58 GMT
4 Content-Type: text/html; charset=utf-8
5 Transfer-Encoding: chunked
6 Connection: close
7 Access-Control-Allow-Headers: Content-Type, ADMIN-Authorization, API-Authorization
8 Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS
9 Access-Control-Allow-Credentials: true
```

POST请求:

```
POST https://www.pkslow.com/user HTTP/1.1
content-type: application/json

{
  "name": "larry",
  "age": 18
}
```

2.3 将请求复制为curl命令

选中请求, 右键然后选择如下:

```
###
```

Send Request

```
GET https://www.pkslow.com HTTP/1.1
```

Open in Integrated Terminal

Go to Definition F12

Go to References ⌘F12

Peek

Find All References ⌘⇧F12

Change All Occurrences ⌘⇧F2

Open Changes

Cut ⌘X

Copy ⌘C

Paste ⌘V

Send Request ⌘⇧R

Generate Code Snippet ⌘⇧C

Copy Request As cURL

结果如下：

```
curl --request GET \  
--url https://www.pkslow.com/ \  
--header 'user-agent: vscode-restclient'
```

2.4 支持发送curl

可以支持运行curl命令，如下：

###

Send Request

```
curl --request GET \  
--url https://www.pkslow.com/ \  
--header 'user-agent: vscode-restclient'
```

2.5 将请求转化为客户端代码

操作如下：

Open in Integrated Terminal	⇧ ⌘ O
Go to Definition	F12
Go to References	⇧ F12
Peek	▸
Find All References	⇧ ⌘ F12
Change All Occurrences	⌘ F2
Open Changes	▸
Cut	⌘ X
Copy	⌘ C
Paste	⌘ V
Send Request	⇧ ⌘ R
Generate Code Snippet	⇧ ⌘ C
Copy Request As cURL	
Command Palette...	⇧ ⌘ P

我们选择python，结果如下：

```
import requests  
  
url = "https://www.pkslow.com/"  
headers = {'user-agent': 'vscode-restclient'}  
response = requests.request("GET", url, headers=headers)  
print(response.text)
```

3 强大的变量

将这个变量的所有功能一起讨论，是因为我觉得它太强大了。也是我选择它很重要的一个理由。

关于变量：

支持自定义变量和环境变量；

不同环境变量定义及环境切换功能；

共享变量；

可以将变量放在任何地方：URL、Header、body等；

获取其它请求的变量；

提供系统动态变量：

```
{{ $guid }}  
  
{{ $randomInt min max }}  
  
{{ $timestamp [offset option] }}  
  
{{ $datetime rfc1123|iso8601 [offset option] }}  
{{ $localDatetime rfc1123|iso8601 [offset option] }}  
  
{{ $processEnv [%] envVarName }}  
{{ $dotenv [%] variableName }}  
  
{{ $aadToken [new] [public|cn|de|us|ppe] [<domain|tenantId>] [aud:<domain|tenantId>] }}
```

3.1 文件中自定义变量

通过@命令来定义变量，然后通过两个大括号{{}}来引用变量：

```
###  
@pkslowHost = www.pkslow.com  
  
GET {{pkslowHost}} HTTP/1.1
```

3.2 配置多环境变量

在vscode的设置文件settings.json配置多环境变量：

```
"rest-client.environmentVariables": {  
  "$shared": {  
    "username": "pkslow",  
    "password": "123456"  
  },  
  "local": {  
    "hostname": "localhost:8080",  
    "password": "{{ $shared password }}"  
  },  
  "production": {  
    "hostname": "localhost:8081",  
    "password": "{{ $shared password }}"  
  }  
}
```

添加了两个环境local和production，其中\$shared的变量是大家都可以引用的。

引用变量hostname和password如下：

```
###  
POST http://{{hostname}}/auth/login HTTP/1.1
```

```
Content-Type: application/json
```

```
{
  "username": "pkslow",
  "password": "{{password}}"
}
```

在vscode的右下角切换环境:



3.3 获取其它请求返回体的值为变量

设想我们经常遇到的一个场景, 先通过login请求来获取Token, 然后再把拿到的Token作为请求头发送其它请求。这个Token肯定是每次登陆都不一样的, 如果每次都要获取, 然后复制, 再粘贴到后面请求中, 就会非常麻烦。REST Client为这种场景提供了便利的办法, 我们来看看如何使用吧:

先通过登陆请求获取Token:

```
###
# @name loginAdmin
POST http://{{hostname}}/auth/login HTTP/1.1
Content-Type: application/json

{
  "username": "pkslow",
  "password": "{{password}}"
}
```

我们添加多了一行:

```
# @name loginAdmin
```

给它定义一个名字, 后续就可以引用它的返回体了。

返回体内容如下:

```
eyJhbGciOiJIUzI1NiIsInR5cGU6IjwiZXN0b3kiLCJyb2x1cyI6WyJST0xFX0FETU01I0sImIhdCI6MTYyMjI3OTMxMCwiZXhwIjoxNjIyMjc5OTUwFQ.h-fwUEOPx_ttt1B0R8cXMHJw
```

所以我们要引用返回的所有内容, 如下:

```
###
@token = {{loginAdmin.response.body.*}}
GET http://{{hostname}}/admin HTTP/1.1
Authorization: Bearer {{token}}
```

通过{{loginAdmin.response.body.*}}来引用变量。格式如下:

{{requestName.(response|request).(body|headers).(.*|JSONPath|XPath|Header Name)}}

所以如果返回体内容是:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cGU6IjwiZXN0b3kiLCJyb2x1cyI6WyJST0xFX0FETU01I0sImIhdCI6MTYyMjI3OTMxMCwiZXhwIjoxNjIyMjc5OTUwFQ.h-fwUEOPx_ttt1B0R8cXMHJw"
}
```

就通过{{loginAdmin.response.body.\$.token}}来引用变量。

4 总结

我把我觉得比较有用的功能都讲解了一下, 更多请参考:GitHub vscode REST Client