[**Wisdom RESTClient**](https://github.com/wisdom-projects/rest-client) 一款自动化测试REST API的工具，它可以自动化测试RESTful API并生成精美的测试报告，同时基于测试过的历史API，可以生成精美的RESTful API文档。

# 1. 使用RESTClient前的准备工作

## 1.1 下载RESTClient

JAR包： [restclient.jar](https://github.com/Wisdom-Projects/rest-client/blob/master/tools)；

## 1.2 使用前安装Java

支持的Java版本 **>=1.7**

## 1.3 启动RESTClient软件

双击[restclient.jar](https://github.com/Wisdom-Projects/rest-client/blob/master/tools)，或者执行命令java -jar restclient.jar启动RESTClient软件。

RESTClient主窗体包含：

* 请求视图（Request）
* 响应视图（Response）
* 历史视图（History）
* 菜单栏（File, Edit, Test, Apidoc, Help）

# 2. 使用RESTClient测试REST API步骤

## 2.1 请求视图中输入REST API所需的请求数据

在请求视图中对所测试的REST API输入的数据详情如下：

### 2.1.1 选择请求方法

RESTClient支持请求方法详情如下：

|  |  |  |
| --- | --- | --- |
| 方法名 | 操作 | 备注 |
| GET | 查询 | 无需要填写请求体 |
| POST | 添加 |  |
| PUT | 修改 |  |
| DELETE | 删除 | 无需要填写请求体 |

### 2.1.2 输入访问REST API的URL

* URL格式： HTTP协议://主机名:端口号/路径
* URL示例： http://restclient.cn:8080/restapi

### 2.1.3 输入请求体(Body)

如果选择的请求方法是**POST**或者**PUT**则可以填写请求体，**其他方法则无需填写**。

#### 2.1.3.1 选择请求体类型（Body-Type）

* **字符串(String)**
* 直接在请求体的文本框中填写字符串；
* **文件(File)**
* 浏览并选择地文本文件，文件内容会被读取并作为请求体。

#### 2.1.3.2 选择内容类型（Content-Type）

根据REST API消息体类型，对照下表，选择跟API匹配的内容类型，如果表中的内容类型都不是API所需要的类型，可以直接在内容类型文本框中**输入所需类型**。 常见的内容类型详情如下：

|  |  |
| --- | --- |
| 内容类型（Content-Type） | 数据格式 |
| application/json | JSON |
| application/xml | XML |
| application/x-www-form-urlencoded | Form表单 |
| text/plain | 纯文本 |
| text/xml | XML文本 |
| text/html | HTML文本 |
| multipart/form-data | 用于上传文件 |
| application/xhtml+xml | XHTML |

### 2.1.4 选择字符集(Charset）

默认字符集是**UTF-8**，可以选择REST API所需要的字符集，如果下拉列表里的字符集都不是API所需要的，可以直接在字符集文本框中**输入所需的字符集**。

### 2.1.5 填写消息头(Header）

可以根据REST API定义要求，以键值对的形式添加相应的消息头。 Header键值对示例：

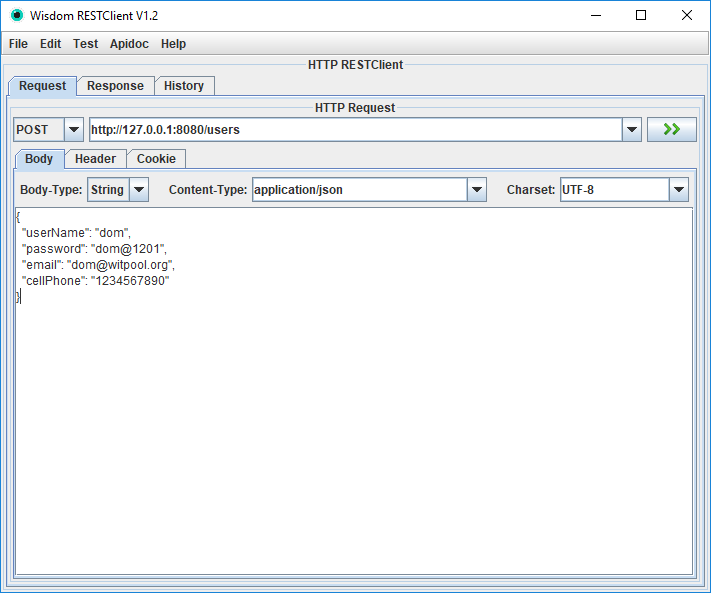
Key ： Accept  
Value ： application/json

### 2.1.6 填写Cookie

可以根据REST API定义要求，以键值对的形式添加相应的Cookie。 如果API需要登录认证，请先使用浏览器完成API登录认证成功后，将浏览器生成的JSESSIONID填写到Cookie中，这样就可以无需登录认证，直接访问REST API了，免登陆使用详情[**参考资料**](http://blog.wdom.net/article/9)。 Cookie键值对示例：

Key ：JSESSIONID   
Value : MY0REST1COOKIE2DEMO3

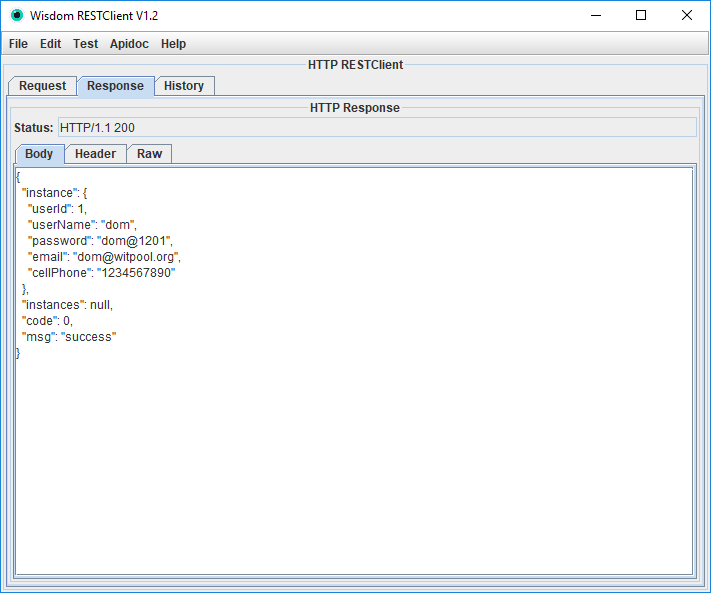
### 2.1.7 完整的请求数据示例

填写完请求数据后点击Start按钮会触发API请求，在请求视图中输入完整的请求数据如图所示： 

## 2.2 响应视图中返回REST API响应的数据

REST API请求完成后得到响应数据如下：

* 响应状态码（Status）
* 响应消息体（Body）
* 响应消息头（Header）
* 原始的响应数据（Raw）

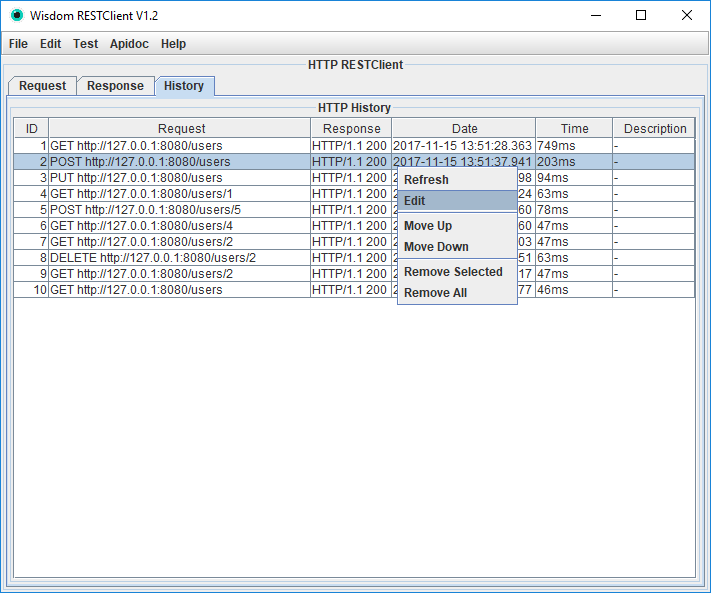
响应数据如图所示： 

## 2.3 历史视图中记录测试过的REST API

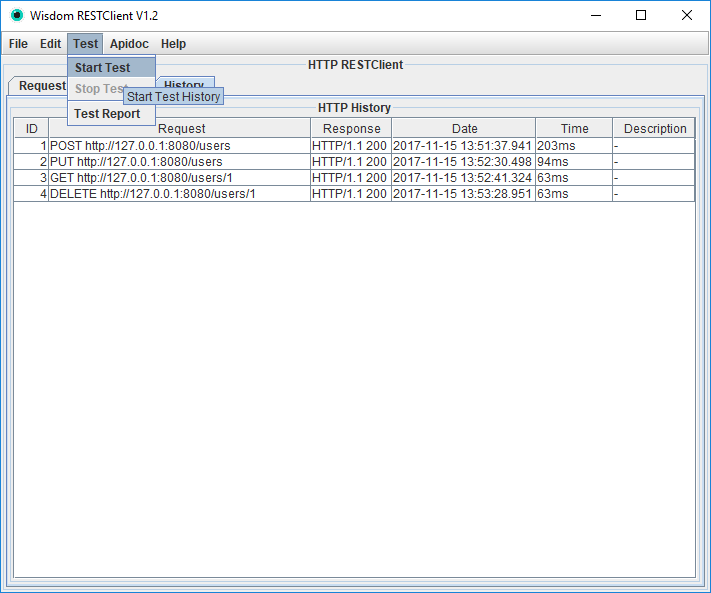
在历史视图中可以对API进行的可视化编辑如下：

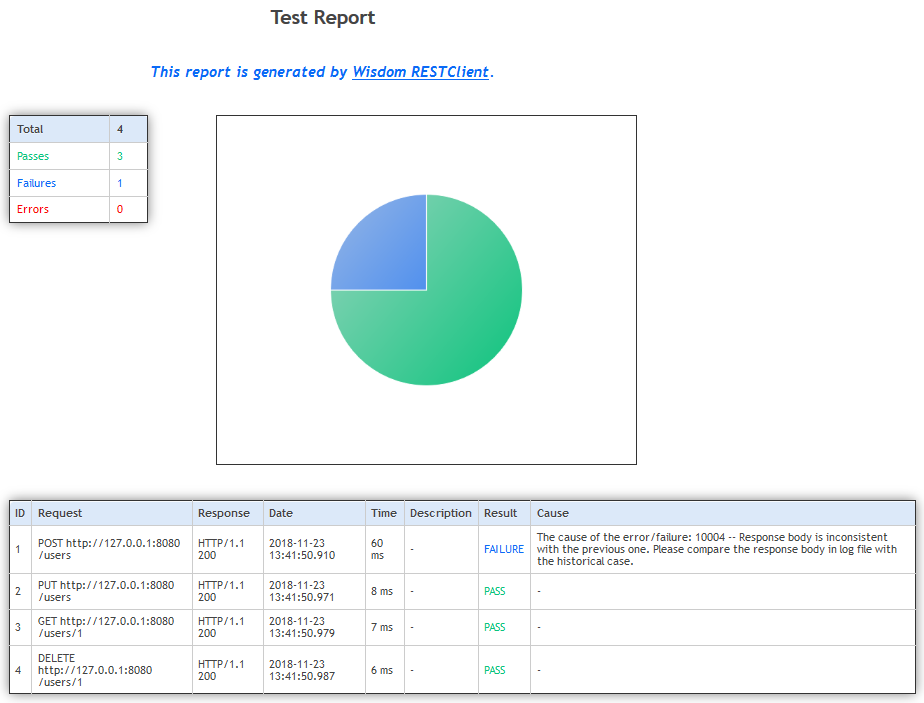
* 刷新API
* 对选中的API进行顺序调整
* 删除选中的API或者清空全部历史API
* 可以编辑选中的API

历史API可视化编辑的快捷菜单如图所示：

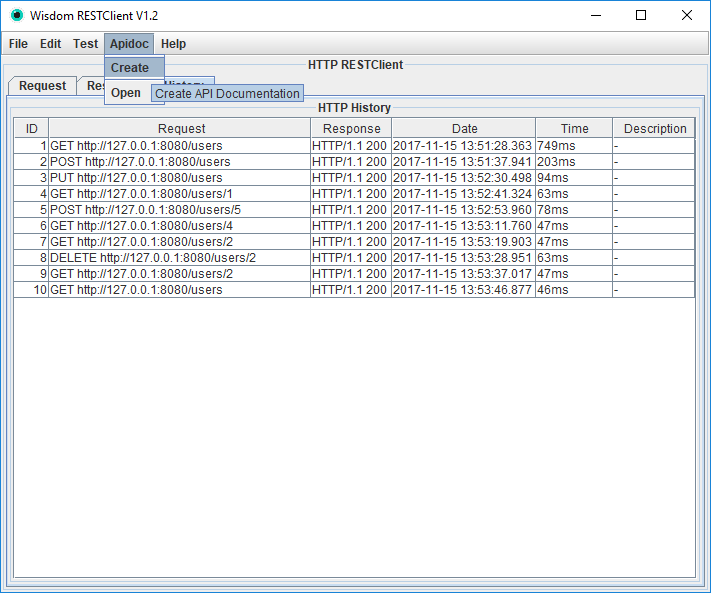


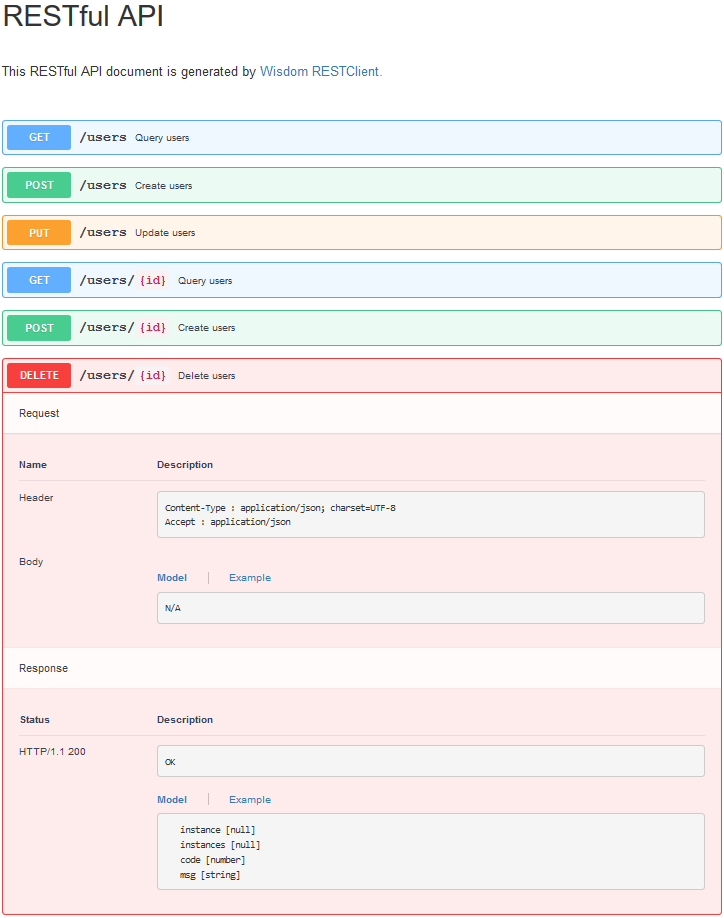
## 2.4 对历史REST API进行再测试

如果需要对历史API进行再测试，在RESTClient菜单栏点击 Test => Start Test 

记录的历史API测试完成后，在Windows系统中会使用默认的浏览器打开测试报告。其他系统可以根据提示框中的报告路径，手动打开测试报告。 测试报告如图所示： 

## 2.5 对历史REST API生成API文档

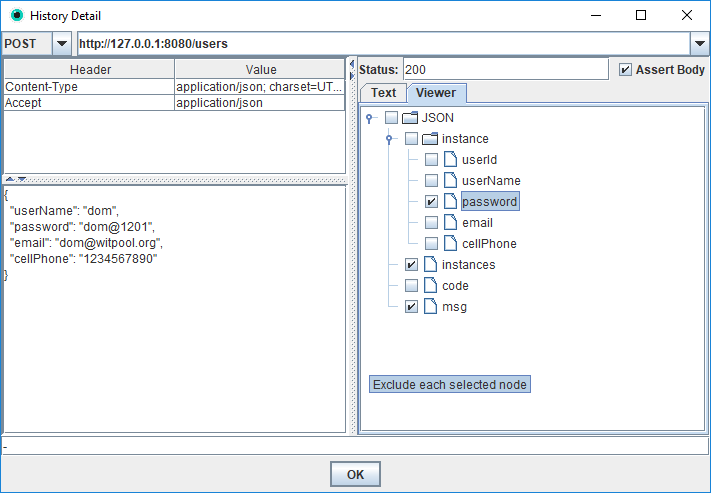
如果需要生成API文档，在RESTClient菜单栏点击 Apidoc => Create 

API文档生成完成后，在Windows系统中会使用默认的浏览器打开API文档。其他系统可以根据提示框中的文档路径，手动打开API文档。 API文档如图所示： 

## 2.6 对历史REST API进行编辑

为了满足API再测试要求或者满足API文档数据要求，可以对API进行如下操作：

* 调整API顺序
* 删除冗余的、废弃的API
* 对API进行可视化编辑

历史视图中选中API，快捷菜单中选择Edit打开API编辑窗体，如图所示： 

在API编辑窗体中，可以编辑如下内容：

* 请求方法
* 请求URL
* 请求头（Header）
* 请求体（Body）
* 响应状态码（Status）
* 响应的消息体（Text视图）
* 是否校验返回的消息体（Assert Body）

默认勾选了Assert Body，API再测试会对返回的消息体进行完整匹配校验，如果不需要对返回的消息体进行匹配校验，可以去勾选。

如果返回的消息体中的某些JSON节点不需要进行再测试匹配校验，可以在Viewer视图上勾选排除这些节点，这样API再测试只对未排除的节点进行匹配校验。

## 2.7 定制API文档

如果生成的API文档不能满足要求，需要改动，可以修改数据文件work/apidoc/js/apidata.js来定制API文档，API定制详情可以[**参考资料**](http://blog.wdom.net/article/10)。

## 2.8 通过命令行（CLI）方式使用RESTClient实现自动化测试REST API

RESTClient支持通过执行命令的方式启动和再测试API以及生成API文档，RESTClient CLI使用详情[**参考资料**](http://blog.wdom.net/article/6)。

通过CLI方式，这样很容易在**Jenkins**中定时执行命令来调度RESTClient进行API再测试，从而实现**自动化测试REST API**和生成REST API文档。

# 3. 问题咨询与帮助

使用RESTClient过程中遇到问题可以查看RESTClient日志文件：work/log/rest-client.log，这样很容易排查出问题的具体原因。

更多的RESTClient使用示例，请参考[**相关的技术资料**](http://blog.wdom.net/tag/RESTClient)来获得更多的使用示例和帮助。