Wisdom RESTClient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；

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，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
```

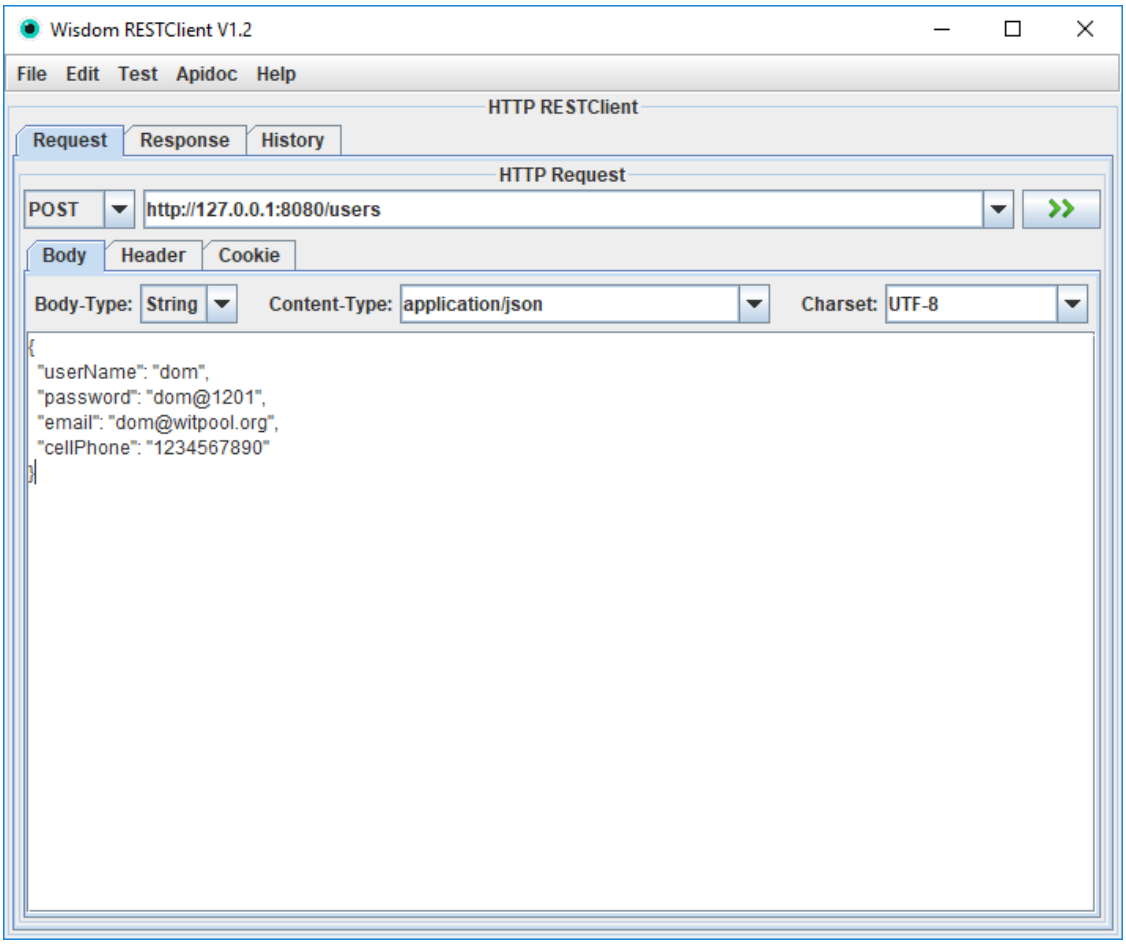
2.1.6 填写 Cookie

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。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：



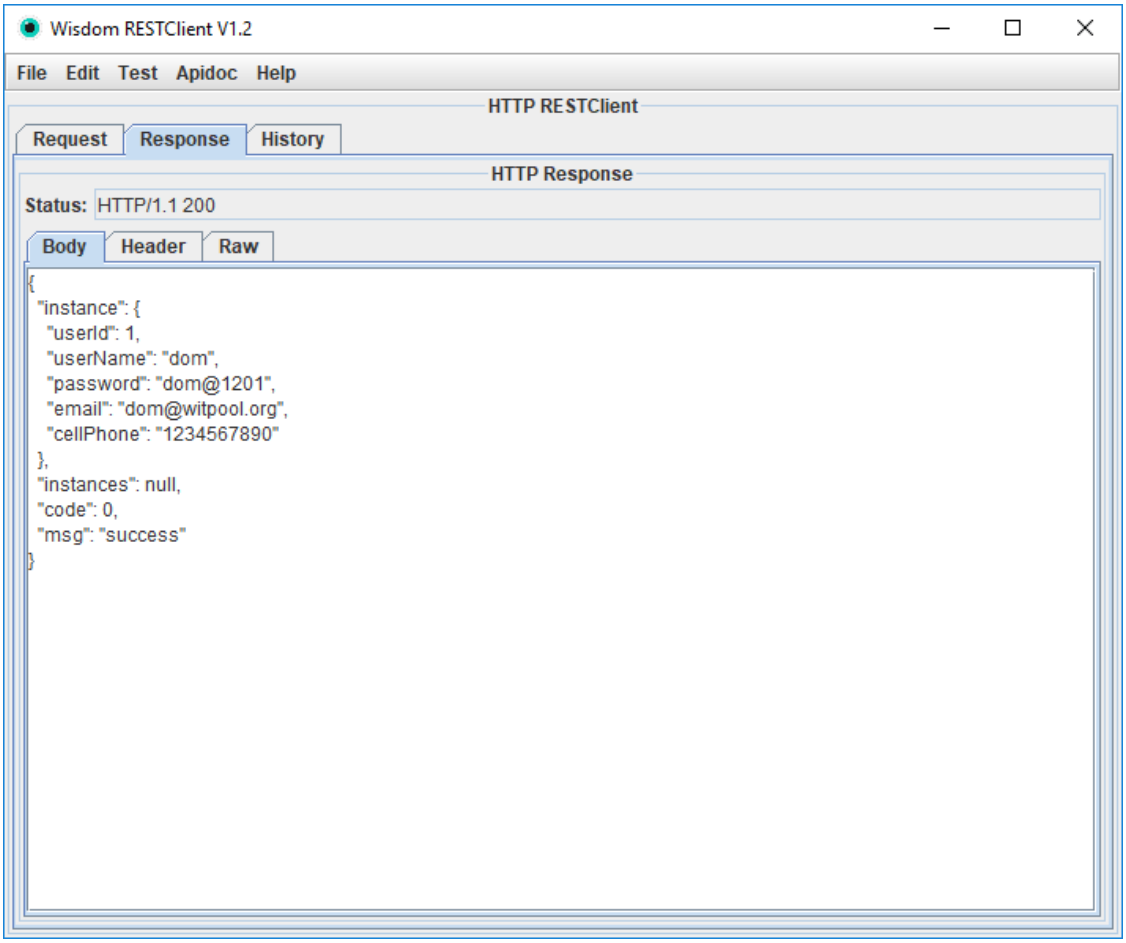
2.2 响应视图中返回 REST API 响应的数据

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：

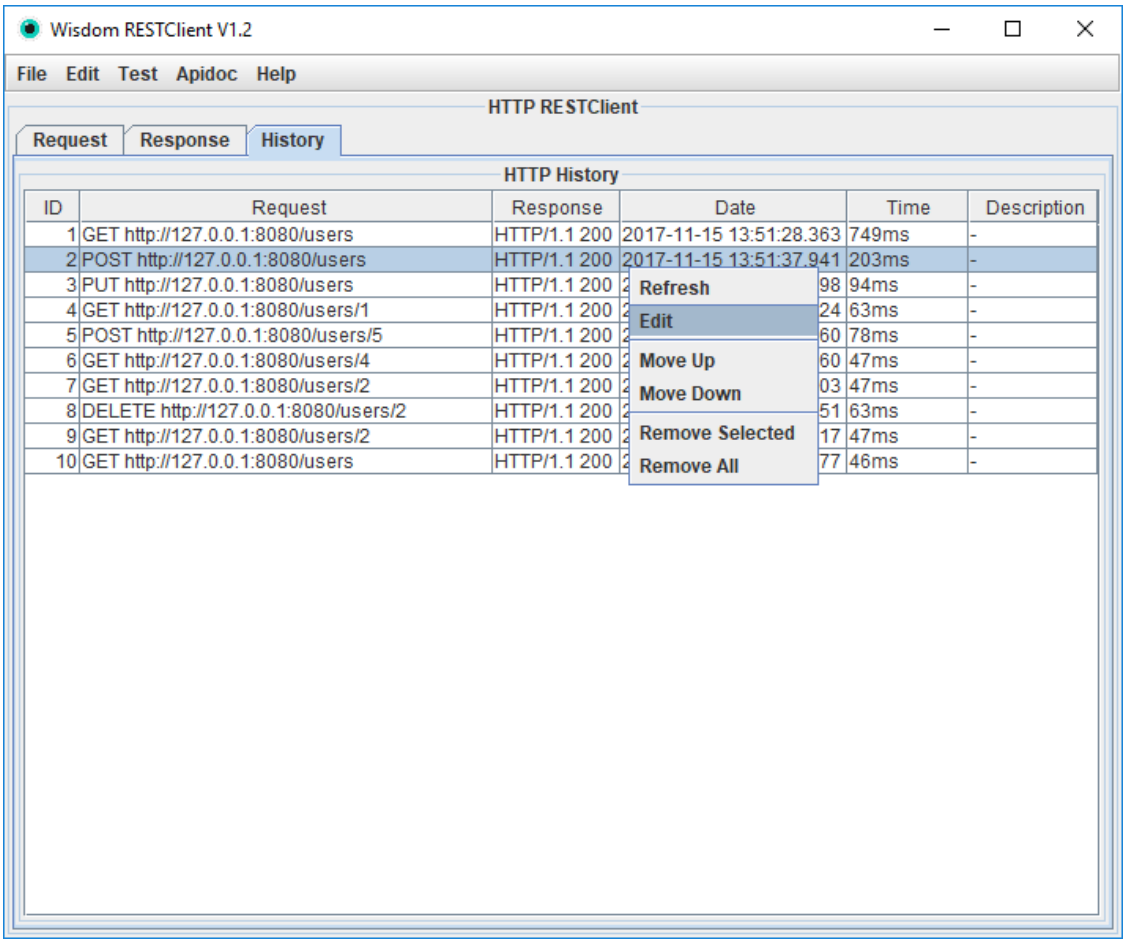


2.3 历史视图中记录测试过的 REST API

在历史视图中可以对 API 进行的可视化编辑如下：

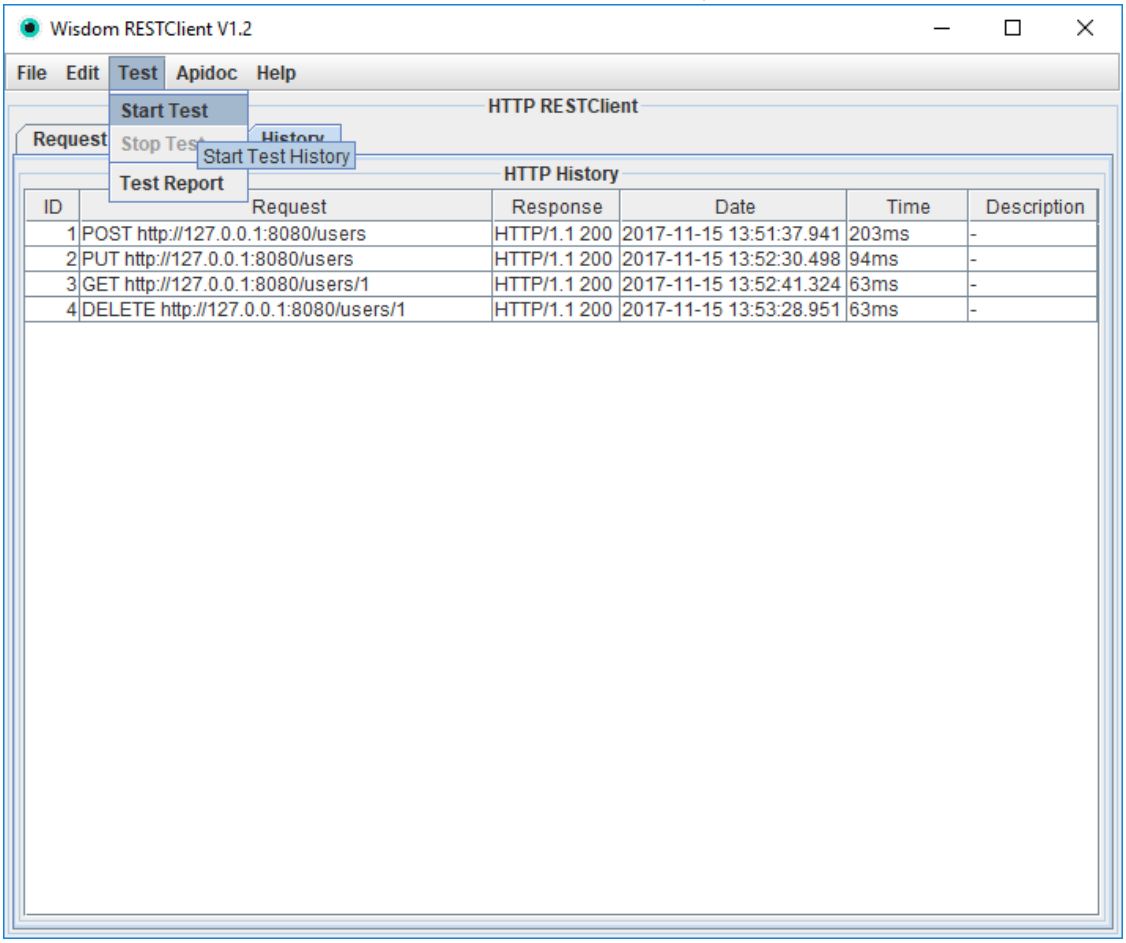
- 刷新 API
- 对选中的 API 进行顺序调整
- 删除选中的 API 或者清空全部历史 API
- 可以编辑选中的 API

历史 API 可视化编辑的快捷菜单如图所示：


Wisdom RESTClient V1.2
File Edit Test Apidoc Help
HTTP RESTClient
Request Response History
HTTP History
ID Request Response Date Time Description
1 GET http://127.0.0.1:8080/users HTTP/1.1 200 2017-11-15 13:51:28.363 749ms -
2 POST http://127.0.0.1:8080/users HTTP/1.1 200 2017-11-15 13:51:37.941 203ms -
3 PUT http://127.0.0.1:8080/users HTTP/1.1 200 98 94ms -
4 GET http://127.0.0.1:8080/users/1 HTTP/1.1 200 24 63ms -
5 POST http://127.0.0.1:8080/users/5 HTTP/1.1 200 60 78ms -
6 GET http://127.0.0.1:8080/users/4 HTTP/1.1 200 60 47ms -
7 GET http://127.0.0.1:8080/users/2 HTTP/1.1 200 03 47ms -
8 DELETE http://127.0.0.1:8080/users/2 HTTP/1.1 200 51 63ms -
9 GET http://127.0.0.1:8080/users/2 HTTP/1.1 200 17 47ms -
10 GET http://127.0.0.1:8080/users HTTP/1.1 200 77 46ms -
Refresh
Edit
Move Up
Move Down
Remove Selected
Remove All

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`

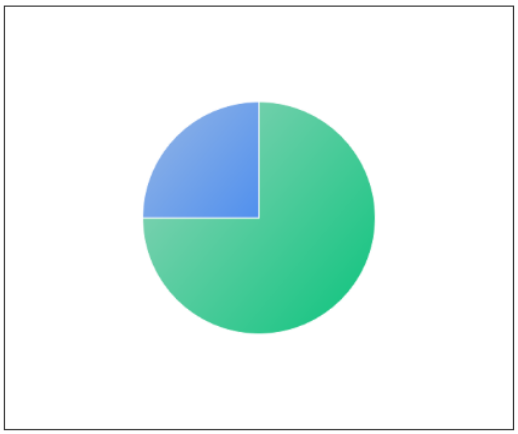


记录的历史 API 测试完成后，在 Windows 系统中会使用默认的浏览器打开测试报告。其他系统可以根据提示框中的报告路径，手动打开测试报告。 测试报告如图所示：

Test Report

This report is generated by Wisdom RESTClient.

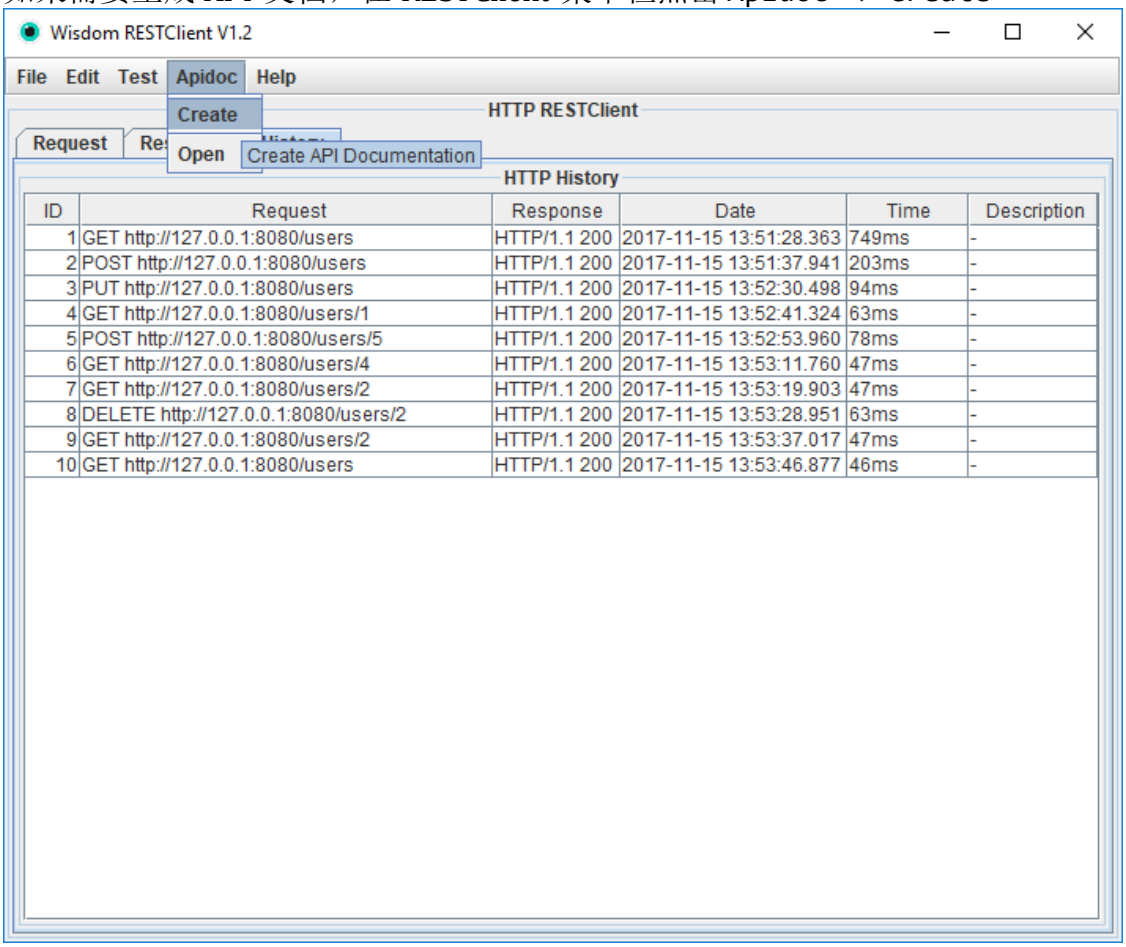
Total	4
Passes	3
Failures	1
Errors	0



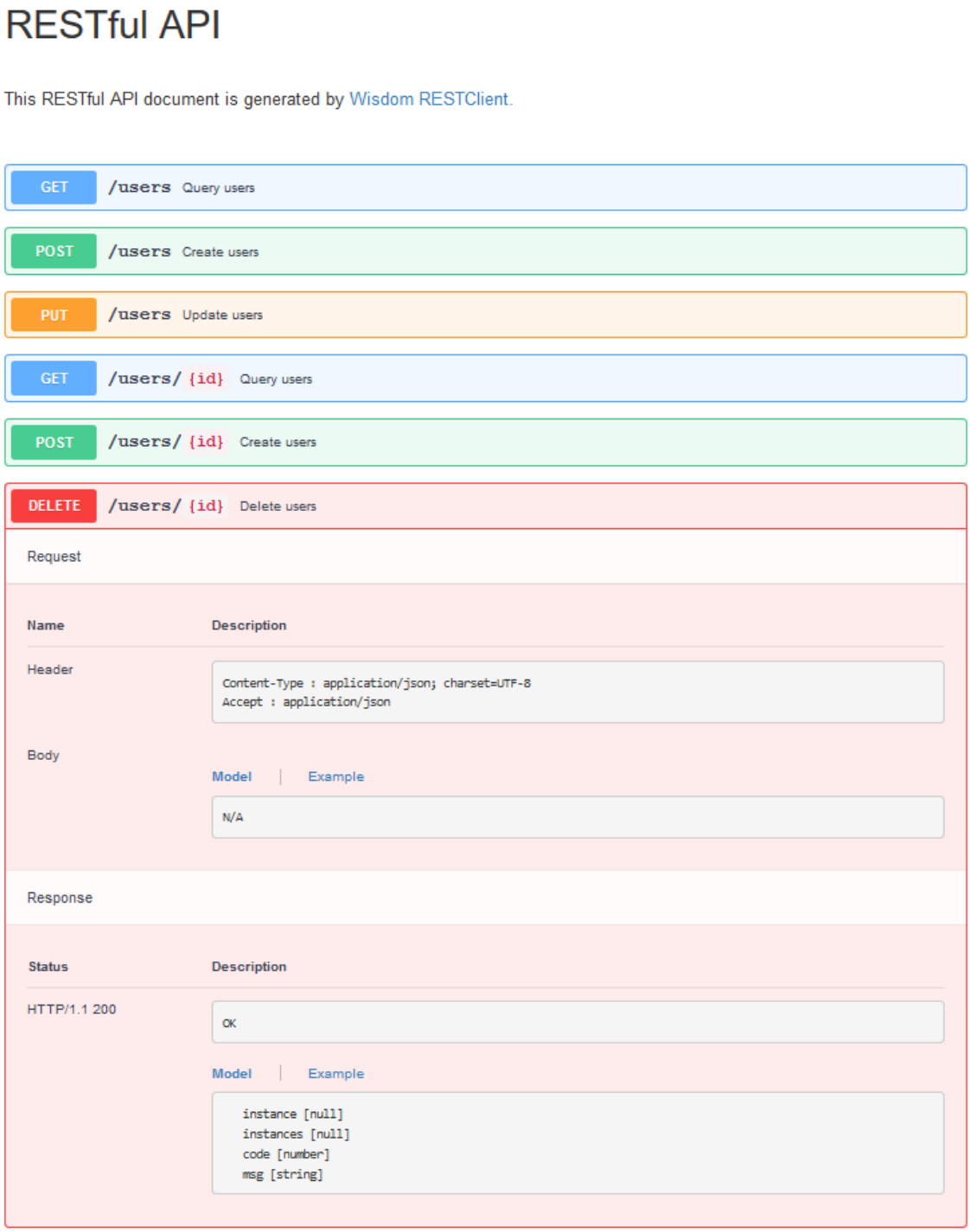
ID	Request	Response	Date	Time	Description	Result	Cause
1	POST http://127.0.0.1:8080/users	HTTP/1.1 200	2018-11-23 13:41:50.910	60 ms	-	FAILURE	The cause of the error/failure: 10004 -- Response body is inconsistent with the previous one. Please compare the response body in log file with the historical case.
2	PUT http://127.0.0.1:8080/users	HTTP/1.1 200	2018-11-23 13:41:50.971	8 ms	-	PASS	-
3	GET http://127.0.0.1:8080/users/1	HTTP/1.1 200	2018-11-23 13:41:50.979	7 ms	-	PASS	-
4	DELETE http://127.0.0.1:8080/users/1	HTTP/1.1 200	2018-11-23 13:41:50.987	6 ms	-	PASS	-

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：



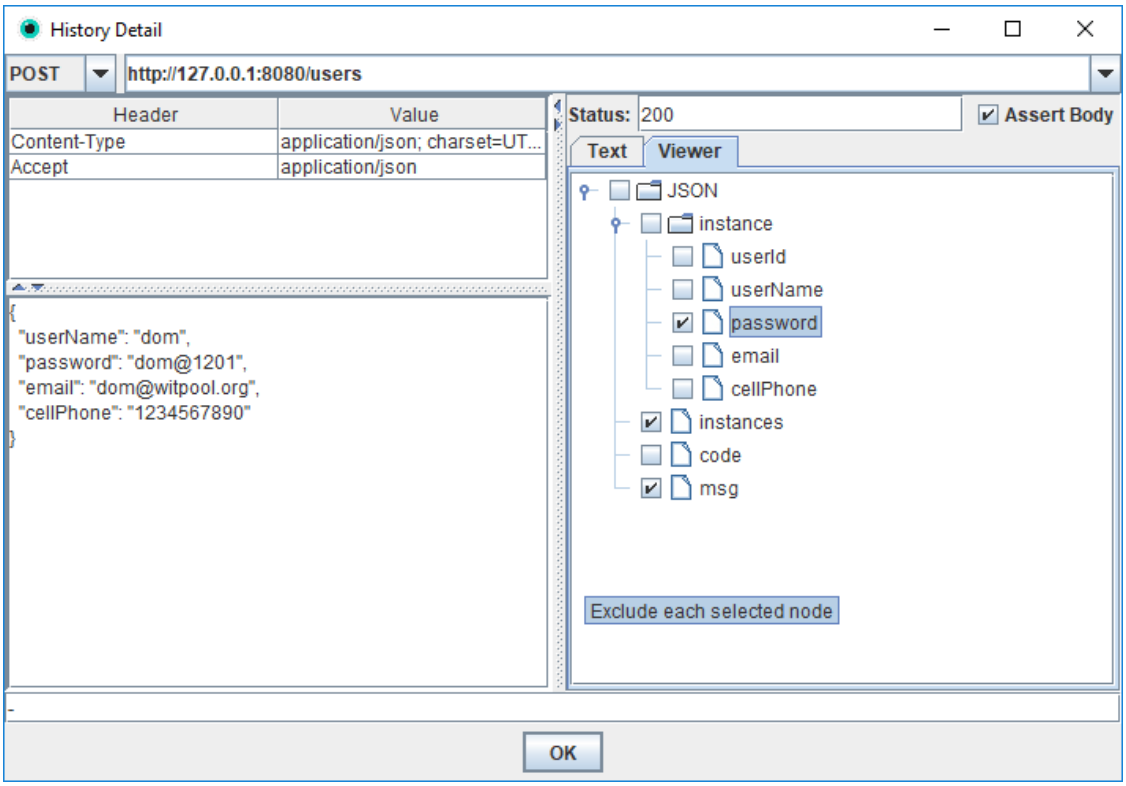
2.6 对历史 REST API 进行编辑

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。

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。

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来获得更多的使用示例和帮助。