

Advantech AE Technical Share Document

Date	2018/6/22	SR#	1-3477572285
Category	■FAQ □SOP	Related OS	N/A
Abstract	How to use RESTful API to access iRTU devices		
Keyword	RESTful, API, browser, https		
Related Product	ADAM-3600, ECU-1251, ECU-1152		

■ **Problem Description:**

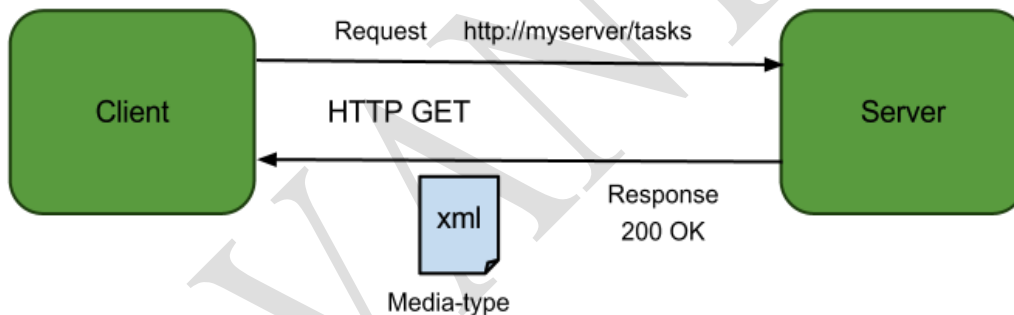
This document shows how to use RESTful API to access iRTU devices.

■ **Answer:**

RESTful is a Request-Response protocol in Client-Server network architecture. It composed by URL, HTTP Method, and Content.

For example,

- URL: http://myserver/tasks
- HTTP Methods: POST, GET, PUT, PATCH, DELETE
- Content: XML, JSON



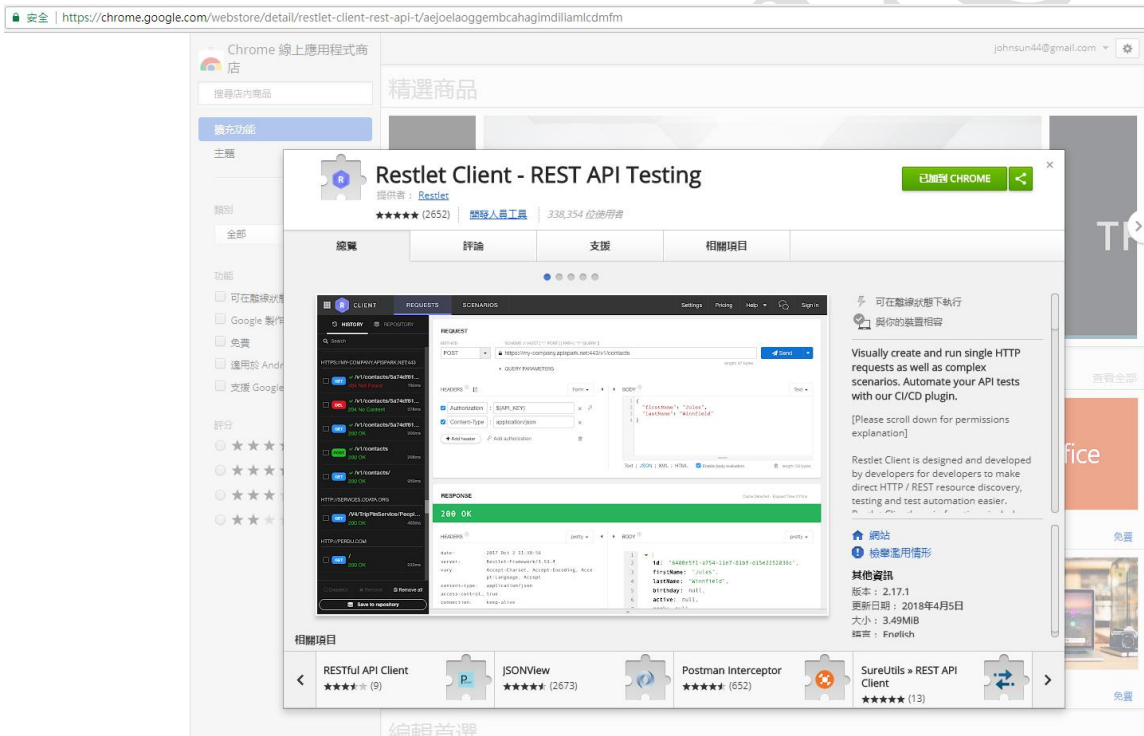
Using browser to open a website is also using RESTful. For ADAM-3600, it is provided several useful RESTful APIs, and the method is listed as below.

Method	Function used in ADAM-3600
GET	Get Tag date/IO data
	Get version
	Get data logger information
	Get update information
PUT	Login/Logout
	Update Tag date/IO data (RW)

PATCH	Calibration
	Time Setting
	Data logger query
POST	Firmware upload
	Firmware update
	Log message

I. Install REST API Testing Tool

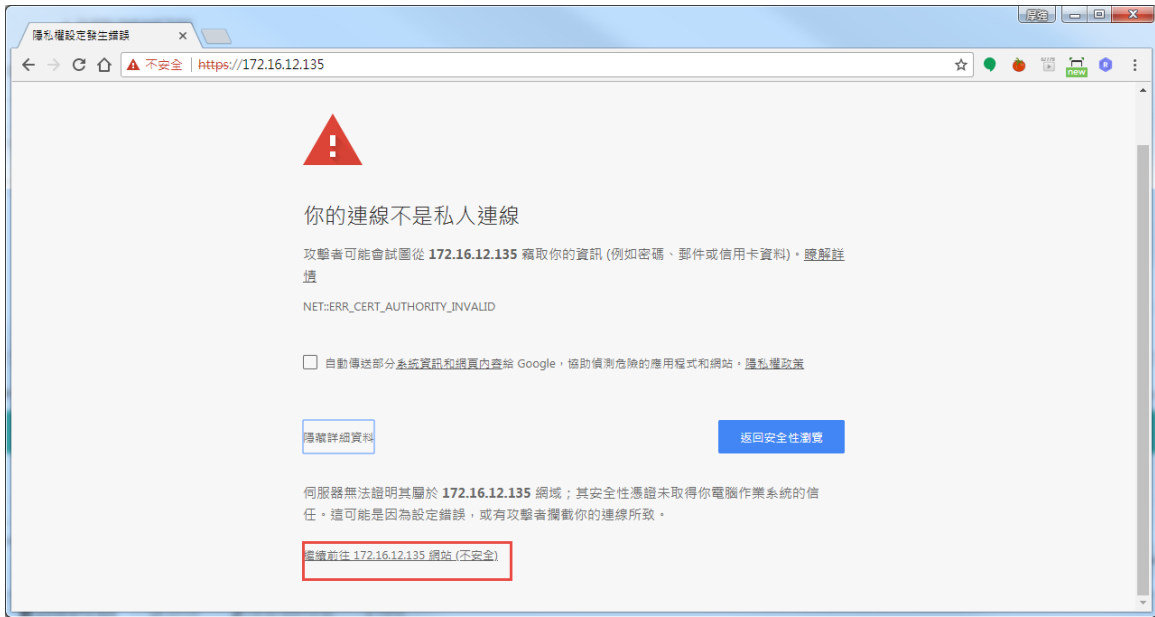
There is one Tool for testing REST API called as Restlet Client. User could search and install it in Google Chrome Plug-in.



II. Add the ADAM-3600 webpage into safety website

Because of https protocol license authentication limitation, user needs to open ADAM-3600 webpage in the browser first.

Add the ADAM-3600 webpage (ex. <https://172.16.12.141/login.html>) into safety website



User may close the webpage once after adding the webpage into trust websites list.

III. Examples of ADAM-3600 RESTful Command

3.1 Log in

3.1.1 Refer to “ADAM-3600 RESTful Specification”. Find URL, Method, and Content.

http://support.advantech.com.tw/Support/DownloadSRDetail_New.aspx?SR_ID=1-1KPLJQG&Doc_Source=Download

2.10 Log in/log out

2.10.1 Log in

`/sys/log_in`

Description	Log in the device for configuration or image updating.
URL Structure	https://10.0.0.1/sys/log_in/
HTTP Method	PUT :
GET	None
PUT	Request: PUT /sys/log_in [Example]: ● Request: PUT /sys/log_in Content-type: application/json { "password":"00000000" } Response: { "sesion_id": "c9f4baf91d3e4ed7cfb18e598c5711f5", }

3.1.2 Select the method as “**PUT**”, and key in the URL based on the ADAM-3600 IP

“https://172.16.12.141/sys/log_in”.

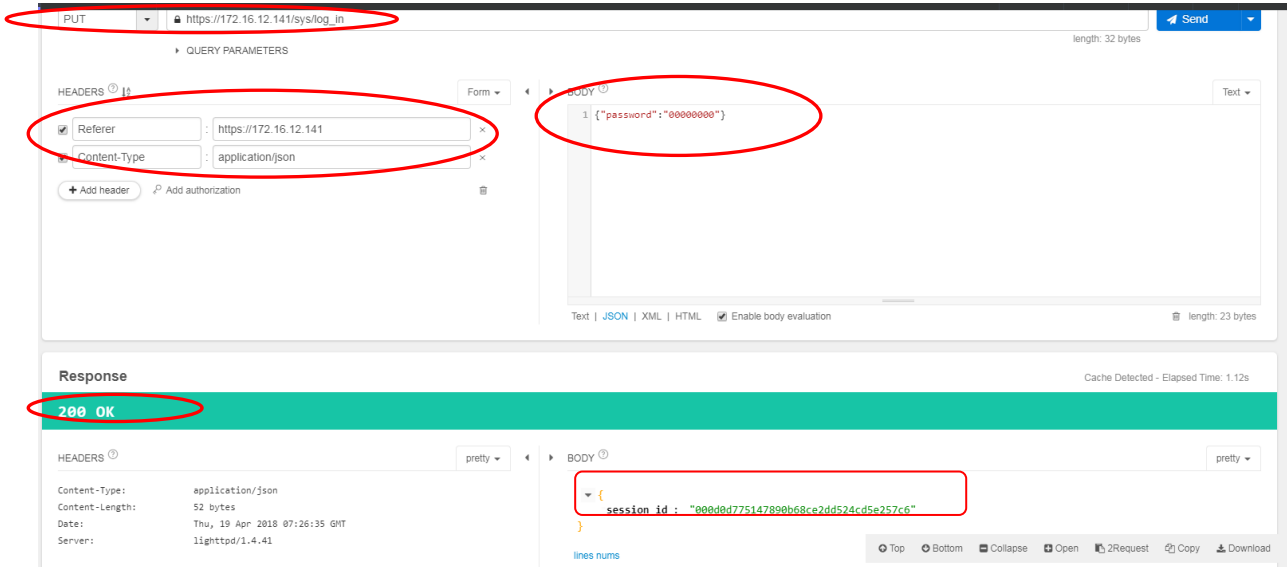
3.1.3 Select HTTP Header includes **Referer** “https://172.16.12.141/” and **Content-Type** “application/json”.

3.1.4 Fill the content with users’ password {“password”: “00000000”} in **BODY**.

3.1.5 Press **Send** button.

3.1.6 User may get the Response **200 OK** and **session id**. It shows logging in successfully.

3.1.7 Copy the **session id** for further applications.



3.2 Get Tags’ information

3.2.1 Refer to “ADAM-3600 RESTful Specification”. Find URL, Method, and Content.

- 2.8 Tag Information

- 2.8.1 System Tag - Data Acquisition

/data/tags/

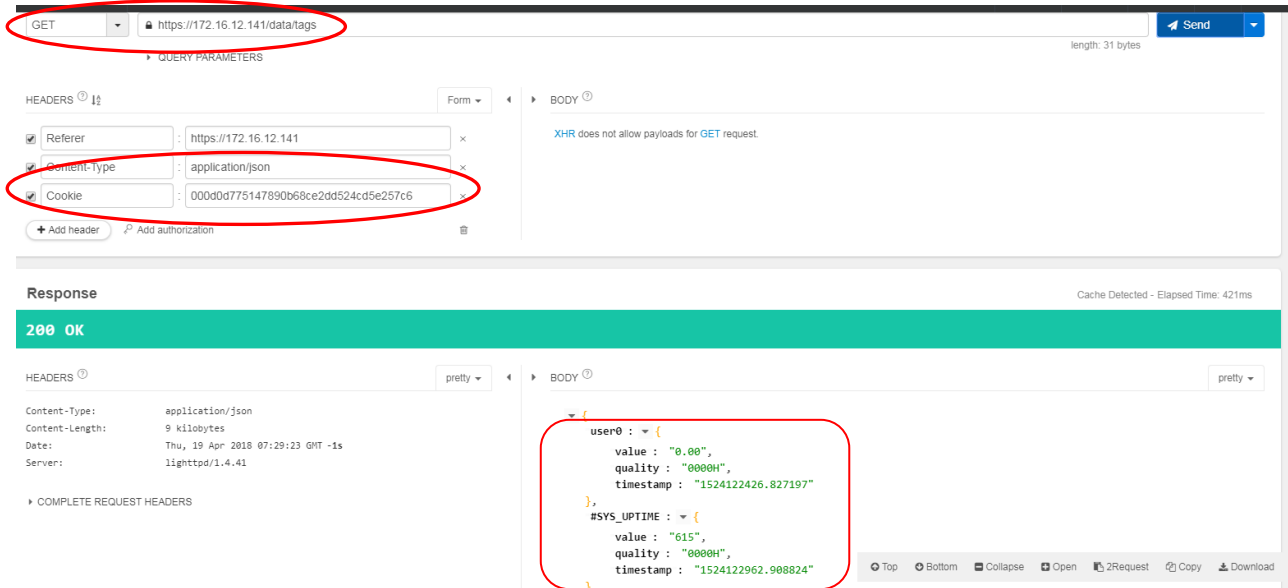
Description	Retrieves information about the <code>digitaloutput</code> value resource on specific slot.
URL Structure	<code>https://10.0.0.1/data/tags</code> System tags can't enter next level.
HTTP Method	GET: Returns the representation of all of tag resources include value, quality and timestamp
GET	Multi-Tag Request: GET /data/tags/ [Example]: <ul style="list-style-type: none"> Request : GET /data/tags/ Content-type: application/json Response: 200 OK <pre>{ "#SYS_UPTIME": { "value": "733", "quality": "0000H", "timestamp": "1421395474.600190" }, }</pre>

3.2.2 Select the method as “GET”, and key in the URL based on the ADAM-3600 IP “https://172.16.12.141/data/tags”.

3.2.3 Select HTTP Header includes **Referer** “https://172.16.12.141/”, **Content-Type** “application/json”, **Cookie** “session id”.

3.2.4 Press **Send** button.

3.2.5 User may get the Response **200 OK** and **Tags** information.



3.3 Change DO value

3.3.1 Refer to “ADAM-3600 RESTful Specification”. Find URL, Method, and Content.

<p>PUT</p>	<p>Single Channel Request : PUT /data/do_value/slot_index/ch_num [Example]: ● Request: PUT /data/do_value/slot_0/ch_2 Content-type: application/json <pre>{ "Md":0, }</pre> Response: 200 OK <pre>{ "Val":0, }</pre> Response: 200 OK</p>
<p>PATCH</p>	<p></p>

● JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital input configurations	DOVal	Array

● Resource value definitions:

Field	Abbreviation	Data Type	Property	Description				
Channel Number	Ch	Number	R	0, 1, ...: Digital output channel number				
Mode	Md	Number	RW	Digital output mode				
				<table border="1"> <tr> <td>0</td> <td>DO</td> </tr> <tr> <td>1</td> <td>Pulse Output</td> </tr> </table>	0	DO	1	Pulse Output
0	DO							
1	Pulse Output							
Signal Logic Status	Stat	Number	R	1, 0: Output signal is Logic High or Low				
Channel Value	Val	Number	RW	DO measurement data				

3.3.2 Select the method as “**PUT**”, and key in the URL based on the ADAM-3600 IP “https://172.16.12.141/data/do_value/slot_0/ch_0”.

3.3.3 Select HTTP Header includes **Referer** “https://172.16.12.141/”, **Content-Type** “application/json”, **Cookie** “session id”.

3.3.4 Fill the content with the modified channel value {“val”: “1”} in **BODY**.

3.3.5 Press **Send** button.

3.3.6 User may get the Response **200 OK** and **Write Success**.

3.3.7 User could check if the channel 0 has been changed.

The screenshot shows a REST client interface with the following details:

- Method:** PUT (circled in red)
- URL:** https://172.16.12.141/data/do_value/slot_0/ch_0
- Headers:**
 - Referer: https://172.16.12.141
 - Cookie: 000d0d775147890b68ce2dd524cd5e257c6
 - Content-Type: application/json
- Body:** {"val": "1"}
- Response:** 200 OK (highlighted in green)
- Response Headers:**
 - Content-Type: application/json
 - Content-Length: 27 bytes
 - Date: Thu, 19 Apr 2018 08:16:10 GMT -1s
 - Server: lighttpd/1.4.41
- Response Body:** {"Success": "Write Success"} (circled in red)