

Advantech Energy Solutions

# Open RESTFul API Specification

For ADAM-3600 RESTful Web Service

An.Xin  
2019-12-30

RECORD OF CHANGES

Version	Date	Chapter	Description	Reviser
1.0	2015-1-14		Initial Version.	An Xin
1.1	2015-6-01		Ver.1.1	An Xin
1.2	2015-9-21		Ver.1.2	An Xin
1.3	2016-11-18		Ver.1.3	An Xin
1.4	2017-08-23		Ver1.4	An.Xin
1.5	2018-04-23		Ver1.5	An.Xin
1.6	2018-11-30		Ver1.6	An.Xin
1.7	2019-12-30	2.9	Ver1.7	An.Xin

# Content

1	Web Service Uniform Interface.....	4
1.1	RESTful Web Service Uniform Interface.....	4
1.2	GET.....	4
1.3	PUT.....	4
1.4	PATCH.....	4
1.5	POST.....	5
1.6	Safe and Idempotent.....	5
1.7	HTTPS Response Status-Code.....	5
2	Web Service API Definition.....	7
2.1	URLOverview.....	7
2.2	Device Information.....	9
2.3	Device Control.....	13
2.3.1	Restart.....	13
2.3.2	Calibration.....	13
2.4	Analog Input.....	16
2.5	Analog output.....	21
2.6	Digital input.....	24
2.7	Digital output.....	27
2.8	Tag Information.....	31
2.8.1	System Tag - Data Acquisition.....	31
2.8.2	User Tag/IO Tag/Calc Tag - Data Acquisition.....	34
2.9	Data Logger.....	40
2.10	Log in/log out.....	46
2.10.1	Log in.....	46
2.10.2	Log out.....	47
2.11	Image Update.....	48
2.11.1	Image File Information Verification.....	48
2.11.2	Image Upload.....	48
2.11.3	Image Update.....	50
2.11.4	Get information in update process.....	51
2.11.5	Image version information.....	51
2.12	System log file.....	52
	Get syslog of device.....	52
2.13	Network.....	54

# 1 Web Service Uniform Interface

## 1.1 RESTful Web Service Uniform Interface

In REST, resources (a resource is any information that you want to make available to others) are identified by unique URI. User agents (a.k.a. web browsers) only interact with resources using the prescribed HTTP verbs. The main verbs (GET, PUT, POST, and DELETE) are what we call the uniform interface. Resources can be read or written to using the HTTP verbs. GET to read resource, PUT is typically used for modifying an existing resource, POST indicates the desire to perform batch operation, DELETE indicates that a client wishes to delete a resource.

## 1.2 GET

The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which shall be returned as the entity in the response and not the source text of the process, unless that text happens to be the output of the process.

## 1.3 PUT

PUT can be used when the client is sending data to the the server and the client is determining the URI for the newly created resource.

*The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI.*

- client must supply the ID
- if the resource exists, **\*replace\*** it with the inbound data
- if it doesn't exist, **create** a new one (assuming you can do that)

## 1.4 PATCH

[RFC5786](#) defines the new HTTP/1.1 [RFC2616] method, PATCH, which is used to apply partial modifications to a resource.

PATCH can be used when the client is sending one or more changes to be applied by the the server.

*The PATCH method requests that a set of changes described in the request entity be applied to the resource identified by the Request-URI. The set of changes is represented in a format called a "patch document"...*

The point is that PATCH is used to doing some kind of **'partial' update**.

## 1.5 POST

POST can be used when the client is sending data to the server and the server will decide the URI for the newly created resource.

*"The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line."*

This is what most of us think of when we talk about **"creating data"** on a web server.

## 1.6 Safe and Idempotent

Methods can also have the property of "idempotence" in that (aside from error or expiration issues) the side-effects of  $N > 0$  identical requests is the same as for a single request. The methods GET, HEAD, PUT and DELETE share this property. Also, the methods OPTIONS and TRACE SHOULD NOT have side effects, and so are inherently idempotent.

	Safe?	Idempotent?
GET	Y	Y
POST	N	N
PATCH	N	N
PUT	N	Y
DELETE	N	Y

## 1.7 HTTPS Response Status-Code

[RFC2616] The Status-Code element is a 3-digit integer result code of the attempt to understand and satisfy the request.

The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. There are 5 values for the first digit:

- 1xx: Informational - Request received, continuing process

- 2xx: Success - The action was successfully received, understood, and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

The status codes supported by ADAM web server are described below.

Status-Code	Reason-Phrase	Conditions
<b>200</b>	OK	The request succeeded, and that the requested information is in the response. This is the most common status code to receive.
<b>400</b>	Bad Request	The request could not be understood by the server due to malformed syntax.. <ul style="list-style-type: none"> <li>● The value in Content-length header does not match with the real data length.</li> <li>● Invalid JOSN objects or format.</li> <li>● Invalid multipart contents.</li> </ul>
<b>403</b>	Forbidden	The server refuses to fulfill the request due to authentication error, such as invalid cookie.
<b>404</b>	Not Found	The requested resource (URI) does not exist on the server.
<b>405</b>	Method Not Allowed	The request method (POST or GET) is not allowed on the requested resource.
<b>411</b>	Length Required	The required Content-length header is missing.
<b>500</b>	Internal Server Error	Memory access errors occurred on the server when processed the file uploaded by client. Fail to reply the JSON-format message due to out of buffer size, or others.
<b>503</b>	Service Unavailable	The login list is full.

## 2 Web Service API Definition

### 2.1 URLOverview

URI			HTTP Methods				Description	Applicable
			GET	PUT	PATCH	POST		
<b>/ data</b>			<input checked="" type="checkbox"/>				The entry URI for user's HTML	ADAM-3600 Series
<b>/sys/log_in</b>				<input checked="" type="checkbox"/>			Log in for update data	
<b>/sys/log_out</b>				<input checked="" type="checkbox"/>			Log out	
<b>/sys/version</b>			<input checked="" type="checkbox"/>				Get all versions on device	
<b>/sys/file_verify</b>			<input checked="" type="checkbox"/>				Get upload file total size and total count, and each file size and file name	
<b>/sys/upload</b>						<input checked="" type="checkbox"/>	Upload files for update	
<b>/sys/update</b>						<input checked="" type="checkbox"/>	Firmware update	
<b>/sys/update_info</b>			<input checked="" type="checkbox"/>				Get infos in the process of update	
<b>/sys/log_create</b>						<input checked="" type="checkbox"/>	Start to get syslog messages	
<b>/sys/log_message</b>			<input checked="" type="checkbox"/>				Get syslog messages	
<b>/sys/control</b>					<input checked="" type="checkbox"/>		Calibration, restart	
<b>/data/tags</b>	<b>/value</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Tag value acquisition	
	<b>/quality</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Tag quality acquisition	
	<b>/timestamp</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Tag timestamp acquisition	
<b>/data /gprs_info</b>	<b>/value</b>		<input checked="" type="checkbox"/>					

	<b>/quality</b>		<input checked="" type="checkbox"/>				
	<b>/timestamp</b>		<input checked="" type="checkbox"/>				
<b>/data /di_value</b>	<b>/slot_x</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Data values of digital input channels
		<b>/ch_x</b>					
<b>/data /do_value</b>	<b>/slot_x</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Data values of digital output channels
		<b>/ch_x</b>					
<b>/data /ai_value</b>	<b>/slot_x</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Data values of analog output channels
		<b>/ch_x</b>					
<b>/data /ao_value</b>	<b>/slot_x</b>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			Data values of analog input channels
		<b>/ch_x</b>					
<b>/data /device_info</b>	<b>/slot_x</b>		<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		Device information, such as module name, firmware version, time, date
<b>/data /datalogger</b>			<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>		Datalogger query

Note: All method (GET/ PUT/POST/PATCH) must be used with cookie, seen in 2.10.1 Log in chapter.



## 2.2 Device Information

[/ data / device\\_info / slot\\_index](#)

Description	Retrieves the device information including the I/O module slots.
URL Structure	<p><b>https://10.0.0.1/data/device_info</b></p> <p><b>https://10.0.0.1/ data/device_info /slot_index</b></p> <p>where <b>index = 0</b> : the on board module</p> <p><b>1 ~</b> : the identifier of I/O extension slot</p>
HTTP Method	<p>GET: Returns the representation of all of I/O module slots information resource.</p> <p>PATCH: Set date/time/timezone.</p>
GET	<p>Multi Slot Request:</p> <p><b>GET / data/device_info</b></p> <p>Single Slot Request:</p> <p><b>GET / data/device_info /slot_index</b></p> <p>[Example]:</p> <ul style="list-style-type: none"> <li>Request : <b>GET / data/device_info</b></li> </ul> <p>Content-type: application/json</p> <p>Response: 200 OK</p> <pre>{   "slot_0": {     "SL":      0,     "Id":      "ADAM-3600-C2GL1A1E",     "DIn":     8,     "DOn":     4,     "AIn":     8,     "AOn":     0,     "FwVer":   "01010168",     "Tm":      "2016-04-07T03:34:36+08:00",     "Tz":      "/Asia/Shanghai"   },   "slot_1": {     "SL":      1,     "Id":      "ADAM-3617",     "DIn":     0,     "DOn":     0,     "AIn":     4,     "AOn":     0,     "FwVer":   "01010170"   }, }</pre>

	<pre> "slot_2": {     "SL":      2,     "Id":      "ADAM-3618",     "DIn":     0,     "DOn":     0,     "AIn":     4,     "AOn":     0,     "FwVer":   "01010173" }, "slot_4": {     "SL":      4,     "Id":      "ADAM-3624",     "DIn":     0,     "DOn":     0,     "AIn":     0,     "AOn":     4,     "FwVer":   "01010176" } } </pre> <ul style="list-style-type: none"> <li>Request : <b>GET /data/device_info/slot_0</b></li> </ul> <p>Content-type: application/json  Response: 200 OK</p> <pre> {     "SL":0,     "Id": "ADAM-3600-C2GL1A1E",     "DIn":      8,     "DOn":      4,     "AIn":      8,     "AOn":      0,     "FwVer":    "01010168",     "Tm":       "2016-04-07T03:38:29+08:00",     "Tz":       "/Asia/Shanghai" } </pre>
PUT	None
POST	None
PATCH	Request: <b>PATCH /data/device_info/slot_0</b>  [Example]: <ul style="list-style-type: none"> <li>Request: <b>PATCH /data/device_info/slot_0</b>, configure time/date/time zone.</li> </ul> Content-type: application/json

```
{
    "Tm": "2016-04-07T03:38:29+08:00",
    "Tz": "/Asia/Shanghai"
}
```

Response: 200 OK

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description						
Slot Number	SL	Number	R	0, 1 ~: Slot number.						
Model ID	Id	String	R	Model ID. For example, <table border="1" data-bbox="933 757 1533 904"> <tr> <td>“ADAM-3600-C2G L1A1E”</td> <td>slot_0: core module</td> </tr> <tr> <td>“ADAM-3617”</td> <td>slot_1 ~: extension slots</td> </tr> <tr> <td>“ADAM-3656”</td> <td>slot_1 ~: extension slots</td> </tr> </table>	“ADAM-3600-C2G L1A1E”	slot_0: core module	“ADAM-3617”	slot_1 ~: extension slots	“ADAM-3656”	slot_1 ~: extension slots
“ADAM-3600-C2G L1A1E”	slot_0: core module									
“ADAM-3617”	slot_1 ~: extension slots									
“ADAM-3656”	slot_1 ~: extension slots									
Total DI Number	DIn	Number	R	0, 1 ~: Digital input channel number.						
Total DO Number	DOn	Number	R	0, 1 ~: Digital output channel number.						
Total AI Number	AIn	Number	R	0, 1 ~: Analog input channel number.						
Total AO Number	Aon	Number	R	0, 1 ~: Analog output channel number.						
Firmware Version	FwVer	String	R	Version information of main firmware image.						
Local Time and Date	Tm	String	RW	<b>Note: Only for slot_0 (on board device)</b>						

Following ISO 8601 which is endorsed by W3C and RFC 3339.

Complete date plus hours, minutes and seconds:

**YYYY-MM-DDThh:mm:ssTZD**

where:

YYYY = four-digit year

MM = two-digit month (01=January, etc.)

DD = two-digit day of month (01 through 31)

hh = two digits of hour (00 through 23) (am/pm

NOT allowed)

mm = two digits of minute (00 through 59)

ss = two digits of second (00 through 59)

TZD = time zone designator (Z or +hh:mm or

-hh:mm)

For example,

**“1994-11-05T08:15:30-05:00”** corresponds to November

5, 1994, 8:15:30 am, US Eastern Standard Time.

Time zone

Tz

String

RW

Note: Only for slot\_0 (on board device)  
/Asia/Shanghai.

Remarks

## 2.3 Device Control

`/ sys / control`

### 2.3.1 Restart

`/ sys / control / rst`

Description	The system can be controlled by command objects.
URL Structure	<b>https://10.0.0.1/sys/control/rst</b>
HTTP Method	PATCH: Send the control command to module.
GET	None
PUT	None
PATCH	Request: <b>PATCH /sys/control/rst</b>  [Example]: <ul style="list-style-type: none"> <li>Request: <b>PATCH /sys/control</b>, Restart the module.</li> </ul> Content-type: application/json <pre>{   "Rst": "1" }</pre> Response: 200 OK

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
System Restart	Rst	String	W	<a href="#">1</a> : Restart the system

Remarks

### 2.3.2 Calibration

`/ sys / control / cali`

Description	The system can be controlled by command objects.
URL Structure	<b>https://10.0.0.1/sys/control/cali</b>
HTTP Method	PATCH: Send the control command to module.

GET	None
PUT	None
PATCH	<p>Request: <b>PATCH /sys/control/cali</b></p> <p>[Example]:</p> <ul style="list-style-type: none"> <li>Request: <b>PATCH /sys/control/cali</b>, calibrate the AI/AO module.</li> </ul> <p>Content-type: application/json</p> <pre>{   "Mid": "1",   "Cid": "1",   "Rng": "1",   "Calmd ": "1",   "Iot ": "1", }</pre> <p>Response: 200 OK</p>

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description																										
Module Id	Mid	String	W	0~4: Module id																										
Channel Id	Cid	String	W	0 ~7:Channel id.																										
Range Code	Rng	String	W	Range code <table border="1"> <tr><td><i>1</i></td><td><i>V_Neg10To10</i></td></tr> <tr><td><i>3</i></td><td><i>V_Neg2pt5To2pt5</i></td></tr> <tr><td><i>7</i></td><td><i>V_0To10</i></td></tr> <tr><td><i>33</i></td><td><i>mA_0To20</i></td></tr> <tr><td><i>34</i></td><td><i>mA_4To20</i></td></tr> <tr><td><i>0x8028</i></td><td><i>Btype_200To1820C</i></td></tr> <tr><td><i>0x8020</i></td><td><i>Etype_Neg270To1000C</i></td></tr> <tr><td><i>0x801D</i></td><td><i>Jtype_Neg210To1200C</i></td></tr> <tr><td><i>0x801E</i></td><td><i>Ktype_Neg270To1372C</i></td></tr> <tr><td><i>0x8029</i></td><td><i>Ntype_Neg100To100C</i></td></tr> <tr><td><i>0x8025</i></td><td><i>Rtype_0To1768C</i></td></tr> <tr><td><i>0x8026</i></td><td><i>Stype_0To1768C</i></td></tr> <tr><td><i>0x801F</i></td><td><i>Ttype_Neg270To400C</i></td></tr> </table>	<i>1</i>	<i>V_Neg10To10</i>	<i>3</i>	<i>V_Neg2pt5To2pt5</i>	<i>7</i>	<i>V_0To10</i>	<i>33</i>	<i>mA_0To20</i>	<i>34</i>	<i>mA_4To20</i>	<i>0x8028</i>	<i>Btype_200To1820C</i>	<i>0x8020</i>	<i>Etype_Neg270To1000C</i>	<i>0x801D</i>	<i>Jtype_Neg210To1200C</i>	<i>0x801E</i>	<i>Ktype_Neg270To1372C</i>	<i>0x8029</i>	<i>Ntype_Neg100To100C</i>	<i>0x8025</i>	<i>Rtype_0To1768C</i>	<i>0x8026</i>	<i>Stype_0To1768C</i>	<i>0x801F</i>	<i>Ttype_Neg270To400C</i>
<i>1</i>	<i>V_Neg10To10</i>																													
<i>3</i>	<i>V_Neg2pt5To2pt5</i>																													
<i>7</i>	<i>V_0To10</i>																													
<i>33</i>	<i>mA_0To20</i>																													
<i>34</i>	<i>mA_4To20</i>																													
<i>0x8028</i>	<i>Btype_200To1820C</i>																													
<i>0x8020</i>	<i>Etype_Neg270To1000C</i>																													
<i>0x801D</i>	<i>Jtype_Neg210To1200C</i>																													
<i>0x801E</i>	<i>Ktype_Neg270To1372C</i>																													
<i>0x8029</i>	<i>Ntype_Neg100To100C</i>																													
<i>0x8025</i>	<i>Rtype_0To1768C</i>																													
<i>0x8026</i>	<i>Stype_0To1768C</i>																													
<i>0x801F</i>	<i>Ttype_Neg270To400C</i>																													
Calibration mode	Calmd	String	W	Calibration mode <table border="1"> <tr><td><i>1</i></td><td><i>SaveCaliValToFlash</i></td></tr> <tr><td><i>1</i></td><td><i>AoSaveCaliValToFlash</i></td></tr> <tr><td><i>3</i></td><td><i>WriteFacCaliValToUser</i></td></tr> <tr><td><i>0x5500</i></td><td><i>ZeroCalibration</i></td></tr> <tr><td><i>0xAA00</i></td><td><i>SpanCalibration</i></td></tr> <tr><td><i>0x55</i></td><td><i>AoOutputSpanValue</i></td></tr> </table>	<i>1</i>	<i>SaveCaliValToFlash</i>	<i>1</i>	<i>AoSaveCaliValToFlash</i>	<i>3</i>	<i>WriteFacCaliValToUser</i>	<i>0x5500</i>	<i>ZeroCalibration</i>	<i>0xAA00</i>	<i>SpanCalibration</i>	<i>0x55</i>	<i>AoOutputSpanValue</i>														
<i>1</i>	<i>SaveCaliValToFlash</i>																													
<i>1</i>	<i>AoSaveCaliValToFlash</i>																													
<i>3</i>	<i>WriteFacCaliValToUser</i>																													
<i>0x5500</i>	<i>ZeroCalibration</i>																													
<i>0xAA00</i>	<i>SpanCalibration</i>																													
<i>0x55</i>	<i>AoOutputSpanValue</i>																													

				<i>0xAA</i>	<i>AoCalibreateSpanValue</i>	
IO type	Iot	String	W	Range code		
				<i>0</i>	AI	
				<i>1</i>	AO	
Remarks						

## 2.4 Analog Input

### AI - Data Acquisition

`/ data / ai_value / slot_index / ch_num`

Description	Retrieves information about the analog input value resource on specific slot.
URL Structure	<b><code>https://10.0.0.1/data/ai_value/slot_index</code></b> <b><code>https://10.0.0.1/data/ai_value/slot_index/ch_num</code></b> where <code>index = 0</code> : the core module <code>1 ~</code> : the identifier of I/O extension slot where <code>num = 0 ~</code> : the channel number
HTTP Method	GET: Returns the representation of all of analog input value resource. PUT: None PATCH: Apply partial modifications to analog input value resource.
GET	Multi Channel Request: <b><code>GET /data/ai_value/slot_index</code></b> Single Channel Request: <b><code>GET /data/ai_value/slot_index/ch_num</code></b>  [Example]: <ul style="list-style-type: none"><li>Request : <b><code>GET /data/ai_value/slot_0</code></b></li></ul> Content-type: application/json Response: 200 OK <pre>{   "AIVal": [     {       "Ch":0,       "En":0,       "Rng":1,       "Val":32765,       "Eg":0,       "Evt":0,       "LoA": 0,       "HiA": 0,       "HVal":0,       "HEg":0,       "LVal":0,       "LEg":0,       "SVal":0,</pre>



```
"ClrH": 0,  
"ClrL": 0  
},  
{  
"Ch":1,  
"En":0,  
"Rng":1,  
"Val":32765,  
"Eg":0,  
"Evt":0,  
"LoA":0,  
"HiA":0,  
"HVal":0,  
"HEg":0,  
"LVal":0,  
"LEg":0,  
"SVal":0,  
"ClrH": 0,  
"ClrL": 0  
},  
{  
"Ch":2,  
"En":0,  
"Rng":1,  
"Val":32765,  
"Eg":0,  
"Evt":0,  
"LoA":0,  
"HiA":0,  
"HVal":0,  
"HEg":0,  
"LVal":0,  
"LEg":0,  
"SVal":0,  
"ClrH": 0,  
"ClrL": 0  
},  
{  
"Ch":3,  
"En":0,  
"Rng":1,  
"Val":32765,  
"Eg":0,  
"Evt":0,
```

```
"LoA":0,  
"HiA":0,  
"HVal":0,  
"HEg":0,  
"LVal":0,  
"LEg":0,  
"SVal":0,  
"ClrH": 0,  
"ClrL": 0  
},  
{  
"Ch":4,  
"En":0,  
"Rng":1,  
"Val":32765,  
"Eg":0,  
"Evt":0,  
"LoA":0,  
"HiA":0,  
"HVal":0,  
"HEg":0,  
"LVal":0,  
"LEg":0,  
"SVal":0,  
"ClrH": 0,  
"ClrL": 0  
}  
]  
}
```

- Request : **GET /data/ai\_value/slot\_0/ch\_2**

Content-type: application/json

Response: 200 OK

```
{  
"Ch":2,  
"En":0,  
"Rng":1,  
"Val":32765,  
"Eg":0,  
"Evt":0,  
"LoA":0,  
"HiA":0,  
"HVal":0,
```

```

    "HEg":0,
    "LVal":0,
    "LEg":0,
    "SVal":0,
    "ClrH": 0,
    "ClrL": 0
  }

```

PUT

Single Channel Request:  
**PUT /data/ai\_value/slot\_index/ch\_num**

[Example]:

- Request: **PUT /data/ai\_value/slot\_0/ch\_3**

Content-type: application/json

```

{
  "Rng": "1"
}

```

Response: 200 OK

PATCH

- JSON array name definition:

Field	Abbreviation	Data Type
Array of Analog input configurations	AIVal	Array

- Resource value definitions (Total channels = AI channel number + 1 average channel):

Field	Abbreviation	Data Type	Property	Description										
Channel Number	Ch	Number	R	0, 1, ...: Analog input channel number. Note for the average channel: The average channel number for a 4-ch AI module is 4.										
Input Range	Rng	Number	RW	Analog input range. <table border="1" data-bbox="925 1478 1324 1724"> <thead> <tr> <th>Range code</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>+/- 10 V</td> </tr> <tr> <td>3</td> <td>+/- 2.5 V</td> </tr> <tr> <td>33</td> <td>4 ~ 20 mA</td> </tr> <tr> <td>34</td> <td>0 ~ 20 mA</td> </tr> </tbody> </table>	Range code		1	+/- 10 V	3	+/- 2.5 V	33	4 ~ 20 mA	34	0 ~ 20 mA
Range code														
1	+/- 10 V													
3	+/- 2.5 V													
33	4 ~ 20 mA													
34	0 ~ 20 mA													
Channel Enable	En*	Number	R	1 / 0: Enable / Disable AI conversion Notice: Average channel is read only. When channel mask of average is not 0, the value is 1.										
Channel Raw Value	Val	Number	R	0 ~ 65535 :AI measurement data (Raw data)										
Channel Engineering data	Eg*	Number	R	AI engineering data, the value is 1/1000 scale. For example,										

1630 → 1.63

Channel Event Status	Evt	Number	R	AI statuses
Low Alarm Status	LoA*	Number	R	Low alarm status Read 1 : low alarm occurred. 0 : not occurred Write 0 : clear the low alarm status
High Alarm Status	HiA*	Number	R	High alarm status Read 1 : high alarm occurred. 0 : not occurred Write 0 : clear the high alarm status
Maximum Value	AI Raw HVal*	Number	R	AI max. measurement data (Raw data)
Maximum Engineering data	AI HEg*	Number	R	AI max. engineering data, the value is 1/1000 scale For example, 10200 → 10.2
Minimum Value	AI Raw LVal*	Number	R	AI min. measurement data (Raw data)
Minimum Engineering data	AI Leg*	Number	R	AI min. engineering data, the value is 1/1000 scale For example, 250 → 0.25
Channel After Scaling	Raw Value SVal*	Number	R	0 ~ 65535 : AI measurement data (Raw data) after scaling
Clear Value	Maximum AI ClrH*	Number	R	1 : Clear the Maximum AI value
Clear Value	Minimum AI ClrL*	Number	R	1 : Clear the Minimum AI value
Remarks	*: Unused in ADAM-3600, just keep consistent with other products.			

## 2.5 Analog output

### AO - Data Acquisition

`/ data / ao_value / slot_index / ch_num`

Description	Retrieves information about the analog input value resource on specific slot.
URL Structure	<b><code>https://10.0.0.1/data/ao_value/slot_index</code></b> <b><code>https://10.0.0.1/data/ao_value/slot_index/ch_num</code></b> where <b>index = 0</b> : the core module <b>1 ~</b> : the identifier of I/O extension slot where <b>num = 0 ~</b> : the channel number
HTTP Method	GET: Returns the representation of all of analog output value resource. PUT: None PATCH: Apply partial modifications to analog input value resource.
GET	Multi Channel Request: <b>GET /data/ao_value/slot_index</b> Single Channel Request: <b>GET /data/ao_value/slot_index/ch_num</b>  [Example]: ● Request : <b>GET /data/ao_value/slot_0</b>  Content-type: application/json Response: 200 OK <pre>{   "AIVal": [     {       "Ch":0,       "Rng":1,       "Val":148,     },     {       "Ch":1,       "Rng":1,       "Val":0,     },     {       "Ch":2,       "Rng":1,       "Val":0,     }   ] }</pre>

```

    },
    {
      "Ch":3,
      "Rng":1,
      "Val":0,
    }
  ]
}

```

- Request : **GET /data/ao\_value/slot\_0/ch\_2**

Content-type: application/json

Response: 200 OK

```

{
  "Ch":2,
  "Rng":328,
  "Val":0,
}

```

PUT

Single Channel Request:

**PUT /data/ao\_value/slot\_index/ch\_num**

[Example]:

- Request: **PUT /data/ao\_value/slot\_0/ch\_3**

Content-type: application/json

```

{
  "Rng":"1",
}
{
  "Val": "65535",
}

```

Response: 200 OK

PATCH

- JSON array name definition:

Field	Abbreviation	Data Type
Array of Analog input configurations	AOVal	Array

- Resource value definitions (Total channels = AO channel number + 1 average channel):

Field	Abbreviation	Data Type	Property	Description
Channel Number	Ch	Number	R	0, 1, ...: Analog output channel number. Note for the average channel: The average channel number for a 4-ch AI module is 4.

Input Range	Rng	Number	RW	Analog input range.	
				Range code	
				7	0~ 10 V
				33	4 ~ 20 mA
				34	0 ~ 20 mA
Channel Raw Value	Val	Number	RW	0 ~ 65535 :AO data (Raw data)	
Remarks					

## 2.6 Digital input

### DI - Data Acquisition

/ data / di\_value / slot\_index / ch\_num

Description	Retrieves information about the digital input value resource on specific slot.
URL Structure	<b>https://10.0.0.1/data/di_value/slot_index</b> <b>https://10.0.0.1/ data/di_value/slot_index/ch_num</b> where index = 0 : the core module 1 ~ : the identifier of I/O extension slot where num = 0 ~ : the channel number
HTTP Method	GET: Returns the representation of all of digital input value resource. PUT: Replace all of digital input value resource PATCH: Apply partial modifications to digital input value resource.
GET	Multi Channel Request: <b>GET /data/di_value/slot_index</b> Single Channel Request: <b>GET /data/di_value/slot_index/ch_num</b>  [Example]: ● Request : <b>GET /data/di_value/slot_0</b>  Content-type: application/json Response: 200 OK { "DIVal": [ { "Ch":0, "Md":0, "Stat":1, "Val":1, "Cnting":0, "ClrCnt":0, "OvLch": 0 }, { "Ch":1, "Md":0, "Stat":0, "Val":1, "Cnting":0, "ClrCnt":0, "OvLch": 0 } ] }



```
"Val":0,
  "Cnting":0,
  "ClrCnt":0,
  "OvLch": 0
},
{
  "Ch":2,
  "Md":0,
  "Stat":0,
  "Val":0,
  "Cnting":1,
  "ClrCnt":0,
  "OvLch": 0
},
{
  "Ch":3,
  "Md":0,
  "Stat":0,
  "Val":1,
  "Cnting":0,
  "ClrCnt":0,
  "OvLch": 0
}
]
}
```

● Request : **GET /di\_value/slot\_0/ch\_2**

Content-type: application/json

Response: 200 OK

```
{
  "Ch":2,
  "Md":0,
  "Stat":1,
  "Val":1,
  "Cnting":0,
  "ClrCnt":0,
  "OvLch": 0
}
```

PUT

Single Channel Request:  
**PUT /data/di\_value/slot\_index/ch\_num**

[Example]:

● Request: **PUT /data/di\_value/slot\_0/ch\_2**

Content-type: application/json

```
{
  "Md": "1",
}
```

Response: 200 OK

```
{
  "Cnting": "1",
}
```

Response: 200 OK

PATCH

● JSON array name definition:

Field	Abbreviation	Data Type
Array of Digital input configurations	DIVal	Array

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description						
Channel Number	Ch	Number	R	0, 1, ...: Digital input channel number.						
Mode	Md	Number	RW	Digital input mode. <table border="1" data-bbox="906 1108 1276 1209"> <tr> <td>0</td> <td>DI</td> </tr> <tr> <td>1</td> <td>Counter</td> </tr> </table>	0	DI	1	Counter		
0	DI									
1	Counter									
Signal Logic Status	Stat	Number	R	1, 0: Input signal is Logic High or Low.						
Channel Value	Val	Number	R	DI measurement data <table border="1" data-bbox="906 1355 1476 1500"> <thead> <tr> <th>Input Mode</th> <th>Value Description</th> </tr> </thead> <tbody> <tr> <td>DI</td> <td>Logic status of DI</td> </tr> <tr> <td>Counter</td> <td>Counter value</td> </tr> </tbody> </table>	Input Mode	Value Description	DI	Logic status of DI	Counter	Counter value
Input Mode	Value Description									
DI	Logic status of DI									
Counter	Counter value									
Start Counter	Cnting	Number	RW	Start/Stop counter counting Read 1 : counter is counting 0 : not counting Write 1 : start counting 0 : stop counting						
Clear Counter	ClrCnt	Number	W	1 : Clear the counter value						
Get/Clear Overflow or Latch Status	Counter OvLch*	Number	RW	counter overflow or latch status Read 1 : overflow/latch occurred. 0 : no overflow or latch Write 0 : clear the overflow or latch status						

Remarks

\*: Unused in ADAM-3600, just keep consistent with other products.

## 2.7 Digital output

### DO - Data Acquisition

/ data / do\_value / slot\_index / ch\_num

Description	Retrieves information about the digital input value resource on specific slot.
URL Structure	<b>https://10.0.0.1/data/do_value/slot_index</b> <b>https://10.0.0.1/data/do_value/slot_index/ch_num</b> where index = 0 : the core module 1 ~ : the identifier of I/O extension slot where num = 0 ~ : the channel number
HTTP Method	GET: Returns the representation of all of digital output value resource. PUT: Replace all of digital output value resource PATCH: Apply partial modifications to digital output value resource.
GET	Multi Channel Request: <b>GET /data/do_value/slot_index</b> Single Channel Request: <b>GET /data/do_value/slot_index/ch_num</b>  [Example]: ● Request : <b>GET /data/do_value/slot_0</b>  Content-type: application/json Response: 200 OK { "DOVal": [ { "Ch":0, "Md":0, "Stat":1, "Val":1, "PsCtn":0, "PsStop":0, "PsLo" : 5000, "PsHi" : 5000, "PsIV": 0 }, { "Ch":1, "Md":0, 

```
"Stat":0,  
"Val":0,  
"PsCtn":0,  
"PsStop":0,  
"PsLo" : 5000,  
"PsHi" : 5000,  
"PsIV": 0  
},  
{  
"Ch":2,  
"Md":0,  
"Stat":0,  
"Val":0,  
"PsCtn":0,  
"PsStop":0,  
"PsLo" : 5000,  
"PsHi" : 5000,  
"PsIV": 0  
},  
{  
"Ch":3,  
"Md":0,  
"Stat":0,  
"Val":1,  
"PsCtn":0,  
"PsStop":0,  
"PsLo" : 5000,  
"PsHi" : 5000,  
"PsIV": 0  
}  
]  
}
```

- Request : **GET /data/do\_value/slot\_0/ch\_2**

Content-type: application/json

Response: 200 OK

```
{  
"Ch":2,  
"Md":0,  
"Stat":1,  
"Val":1,  
"PsCtn":0,  
"PsStop":0,
```

	<pre> “PsLo” : 5000, “PsHi” : 5000, "PsIV": 0 } </pre>
PUT	<p>Single Channel Request:</p> <p><b>PUT /data/do_value/slot_index/ch_num</b></p> <p>[Example]:</p> <ul style="list-style-type: none"> <li>Request: <b>PUT /data/do_value/slot_0/ch_2</b></li> </ul> <p>Content-type: application/json</p> <pre> {   "Md": "0", } </pre> <p>Response: 200 OK</p> <pre> {   "Val": "0", } </pre> <p>Response: 200 OK</p>
PATCH	

- 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" data-bbox="845 1556 1220 1668"> <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 <table border="1" data-bbox="869 1803 1476 2027"> <thead> <tr> <th>Output Mode</th> <th>Value Description</th> </tr> </thead> <tbody> <tr> <td>DO</td> <td>Get the current signal status or set its status</td> </tr> <tr> <td>Pulse Output</td> <td>Get or set the absolute pulse count value 0-2<sup>32</sup></td> </tr> </tbody> </table>	Output Mode	Value Description	DO	Get the current signal status or set its status	Pulse Output	Get or set the absolute pulse count value 0-2 <sup>32</sup>
Output Mode	Value Description									
DO	Get the current signal status or set its status									
Pulse Output	Get or set the absolute pulse count value 0-2 <sup>32</sup>									

Pulse Continue State	Output	PsCtn	Number	RW	1 / 0: Pulse outputting is continuous or not. Write 1: start pulse out 0: stop pulse out
Stop Pulse Output		PsStop*	Number	W	1: Stop the pulse outputting. (Continue is disabled, Absolute and incremental values are reset to zero. DO signal status is set to logic low.)
Low Signal Width		PsLo	Number	RW	Low signal width of pulse 1 - 65535 (0.1 ms)
High Signal Width		PsHi	Number	RW	High signal width of pulse 1 - 65535 (0.1 ms).
Incremental Output Value	Pulse	PsIV*	Number	RW	Incremental Pulse Output Value
Remarks	*: Unused in ADAM-3600, just keep consistent with other products.				

## 2.8 Tag Information

### 2.8.1 System Tag - Data Acquisition

`/ data / tags /`

Description	Retrieves information about the digitaloutput value resource on specific slot.
URL Structure	<b>https://10.0.0.1/data/tags</b> 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: <b>GET / data/tags/</b>  [Example]: <ul style="list-style-type: none"><li>● Request : GET /data/ tags / Content-type: application/json Response: 200 OK<pre>{   "#SYS_UPTIME": {     "value": "733",     "quality": "0000H",     "timestamp": "1421395474.600190"   },   "#SYS_CURRENT_TIME": {     "value": "1421395474",     "quality": "0000H",     "timestamp": "1421395474.600212"   },   "#SYS_CPU_FREQ": {     "value": "629145600",     "quality": "0000H",     "timestamp": "1421395474.600214"   },   "#SYS_MEM_SIZE": {     "value": "242.18 MB",     "quality": "0000H",     "timestamp": "1421395474.600668"   },   "#SYS_CPU_USED": {</pre></li></ul>

```
"value": "39.01%",  
"quality": "0000H",  
"timestamp": "1421395474.600673"  
},  
"#SYS_MEM_USED": {  
"value": "22.70%",  
"quality": "0000H",  
"timestamp": "1421395474.601017"  
},  
"#SYS_TFCARD_CAPACITY": {  
"value": "954.00 MB",  
"quality": "0000H",  
"timestamp": "1421395474.677785"  
},  
"#SYS_TFCARD_FREE_SPACE": {  
"value": "98.13 MB",  
"quality": "0000H",  
"timestamp": "1421395474.677894"  
},  
"#SYS_SDCARD_CAPACITY": {  
"value": "0 Bytes",  
"quality": "0000H",  
"timestamp": "1421395474.699519"  
},  
"#SYS_SDCARD_FREE_SPACE": {  
"value": "0 Bytes",  
"quality": "0000H",  
"timestamp": "1421395474.733687"  
},  
"#SYS_NODE_ID": {  
"value": "15",  
"quality": "0000H",  
"timestamp": "1421395474.736068"  
},  
"#SYS_COM_COUNT": {  
"value": "0.01",  
"quality": "0000H",  
"timestamp": "1421395474.736074"  
},  
"#SYS_LAN_COUNT": {  
"value": "0",  
"quality": "0000H",  
"timestamp": "1421395474.736076"  
},  
}
```



	<pre> "#ICDM_COM1_SCORE": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_COM2_SCORE": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_COM3_SCORE": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_LAN1_SCORE": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_LAN1_LINK": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_LAN2_SCORE": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" }, "#ICDM_LAN2_LINK": {   "value": "0",   "quality": "ffffH",   "timestamp": "0.000000" } } </pre>
--	--

PUT	None
-----	------

--	--

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
value	value	String	R Tag value	
quality	quality	String	R Tag quality.	

	0000H	OK		
	0001H	QLTY_OVER_RANGE		
	0002H	QLTY_UNDER_RANGE		
	0003H	QLTY_OPEN_LOOP		
	0004H	QLTY_SHORTED_LOOP		
	8001H	QLTY_RESTART		
	8002H	QLTY_DRIVER_LOAD_FAILED		
	8003H	QLTY_PORT_OPEN_FAILED		
	8004H	QLTY_DEVICE_ERROR		
	8005H	QLTY_CONVERSION_CODE_ERROR		
	8006H	QLTY_CONVERSION_ERROR		
	8007H	QLTY_DATA_TYPE_ERROR		
	8008H	QLTY_DATA_SIZE_ERROR		
	8009H	QLTY_DATA_ERROR		
	800aH	QLTY_CHECKSUM_ERROR		
	800bH	QLTY_MSG_SEQ_ERROR		
	8080H	QLTY_NO_OWNER		
	8081H	QLTY_DISCONNECTED		
	8100H	QLTY_DRIVER_SEPCIFIED		
timestamp	timestamp	String	R	Tag timestamp
Remark				

### 2.8.2 User Tag/IO Tag/Calc Tag - Data Acquisition

/ data / tags / **tag\_name** / value

/ data / tags / **tag\_name** / quality

/ data / tags / **tag\_name** / timestamp

Description	Retrieves information about the digitaloutput value resource on specific slot.
URL Structure	<b>https://10.0.0.1/data/tags/ tag_name /value/</b> <b>https://10.0.0.1/data/tags/ tag_name /quality/</b> <b>https://10.0.0.1/data/tags/tag_name /timestamp/</b> where <b>tag_name</b> : user tag name defined in DataCenter
HTTP Method	GET: Returns the representation of all of tag resources include value, quality and timestamp

GET

Multi-Tag Request:

**GET / data/tags/**

Single Channel Request:

**GET /data/ tags / tag\_name /**

Single Resource Request:

**GET /data/tags/ tag\_name /value/**

**GET /data/tags/ tag\_name /quality/**

**GET /data/ tags / tag\_name /timestamp/**

[Example]:

- Request : GET /data/ tags /  
Content-type: application/json  
Response: 200 OK

```
{
  "#SYS_UPTIME": {
    "value": "733",
    "quality": "0000H",
    "timestamp": "1421395474.600190"
  },
  "#SYS_CURRENT_TIME": {
    "value": "1421395474",
    "quality": "0000H",
    "timestamp": "1421395474.600212"
  },
  "#SYS_CPU_FREQ": {
    "value": "629145600",
    "quality": "0000H",
    "timestamp": "1421395474.600214"
  },
  "#SYS_MEM_SIZE": {
    "value": "242.18 MB",
    "quality": "0000H",
    "timestamp": "1421395474.600668"
  },
  "#SYS_CPU_USED": {
    "value": "39.01%",
    "quality": "0000H",
    "timestamp": "1421395474.600673"
  },
  "#SYS_MEM_USED": {
    "value": "22.70%",
    "quality": "0000H",
    "timestamp": "1421395474.601017"
  },
}
```

```
"#SYS_TFCARD_CAPACITY": {
  "value": "954.00 MB",
  "quality": "0000H",
  "timestamp": "1421395474.677785"
},
"#SYS_TFCARD_FREE_SPACE": {
  "value": "98.13 MB",
  "quality": "0000H",
  "timestamp": "1421395474.677894"
},
"#SYS_SDCARD_CAPACITY": {
  "value": "0 Bytes",
  "quality": "0000H",
  "timestamp": "1421395474.699519"
},
"#SYS_SDCARD_FREE_SPACE": {
  "value": "0 Bytes",
  "quality": "0000H",
  "timestamp": "1421395474.733687"
},
"#SYS_NODE_ID": {
  "value": "15",
  "quality": "0000H",
  "timestamp": "1421395474.736068"
},
"#SYS_COM_COUNT": {
  "value": "0.01",
  "quality": "0000H",
  "timestamp": "1421395474.736074"
},
"#SYS_LAN_COUNT": {
  "value": "0",
  "quality": "0000H",
  "timestamp": "1421395474.736076"
},
"#ICDM_COM1_SCORE": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"#ICDM_COM2_SCORE": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
}
```

```
},
"#ICDM_COM3_SCORE": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"#ICDM_LAN1_SCORE": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"#ICDM_LAN1_LINK": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"#ICDM_LAN2_SCORE": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"#ICDM_LAN2_LINK": {
  "value": "0",
  "quality": "ffffH",
  "timestamp": "0.000000"
},
"ai0": {
  "value": "0.00",
  "quality": "0000H",
  "timestamp": "1425970361.870002"
},
"ai1": {
  "value": "0.00",
  "quality": "0000H",
  "timestamp": "1425970361.870002"
},
"ai2": {
  "value": "0.00",
  "quality": "0000H",
  "timestamp": "1425970361.870002"
},
"ai3": {
  "value": "0.00",
  "quality": "0000H",
```

```

        "timestamp": "1425970361.870002"
    },
}

```

- Request : GET /data/ tags /ai0 /  
Content-type: application/json  
Response: 200 OK

```

{
    "value": "0.00",
    "quality": "0000H",
    "timestamp": "1425970541.870002"
}

```

PUT

Single Channel Request:  
**PUT** /data/tags/ [tag\\_name](#) /value/  
[Example]:

- Request: **PUT** /data/tags/ai0/value  
Content-type: application/json

```

{
    "value": "2.00",
}

```

Response: 200 OK

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
value	value	String	RW	Tag value
quality	quality	String	R	Tag quality.
				0000H OK
				0001H QLTY_OVER_RANGE
				0002H QLTY_UNDER_RANGE
				0003H QLTY_OPEN_LOOP
				0004H QLTY_SHORTED_LOOP
				8001H QLTY_RESTART
				8002H QLTY_DRIVER_LOAD_FAILED
				8003H QLTY_PORT_OPEN_FAILED
				8004H QLTY_DEVICE_ERROR
				8005H QLTY_CONVERSION_CODE_ERROR
				8006H QLTY_CONVERSION_ERROR
				8007H QLTY_DATA_TYPE_ERROR
				8008H QLTY_DATA_SIZE_ERROR

				8009H	QLTY_DATA_ERROR		
				800aH	QLTY_CHECKSUM_ERROR		
				800bH	QLTY_MSG_SEQ_ERROR		
				8080H	QLTY_NO_OWNER		
				8081H	QLTY_DISCONNECTED		
				8100H	QLTY_DRIVER_SEPCIFIED		
timestamp	timestamp	String	R	Tag timestamp			
Remarks							

## 2.9 Data Logger

**/ data / datalogger / tagname**

Description	Retrieves all tag names recorded in data logger..
URL Structure	<b>https://10.0.0.1/data/datalogger/tagname</b>
HTTP Method	GET: Returns the tag names recorded in datalogger.
GET	<p>Request: <b>GET / data/datalogger/tagname</b></p> <p>[Example]:</p> <ul style="list-style-type: none"> <li>Request : <b>GET / data/ datalogger/tagname</b></li> </ul> <p>Content-type: application/json Response: 200 OK</p> <pre>{   "datalogger_tagname": [     "#SYS_CPU_USED",     "#SYS_TFCARD_CAPACITY",     "#SYS_NODE_ID",     "#SYS_LAN_COUNT",     "#SYS_UPTIME",     "#SYS_MAC_LAN1",     "#SYS_MAC_LAN2",     "#MOBILE_MPN",     "#GPS_LATITUDE",   ] }</pre>
PUT	None
POST	None
PATCH	None
Remarks	

**/ data / datalogger**

Description	Start to query all tags recorded in data logger..
URL Structure	<b>https://10.0.0.1/data/datalogger</b>
	POST: Returns the time stamp of the beginning point of the query.
GET	None



PUT	None
POST	<p>Request: <b>POST / data/datalogger</b></p> <p>[Example]: Start to query Statistics data</p> <ul style="list-style-type: none"> <li>Request : <b>POST</b> <b>/ data/datalogger</b> Content-type: application/json</li> </ul> <pre>{   "Tn": "#SYS_CPU_FREQ",   "St": "1577899541",   "Et": "1577975941",   "Tp": "-1" }</pre> <p>Response: 200 OK</p> <pre>{   "taskid": "0 " }</pre>

PATCH	None
-------	------

● Resource value definitions:

Field	Abbreviation	Data Type	Property	Description								
Tag Name	Tn	String	RW	Tag name in datalogger								
Start time	St	String	RW	Start time of time. Unix timestamp 0: The first record time								
Ebd time	Et	String	RW	End time of time. Unix timestamp 0: The latest record time								
Query Type	Tp	String	RW	Query type								
				<table border="1"> <tr> <td>"-1"</td> <td>Historical query</td> </tr> <tr> <td>"0"</td> <td>Query in minutes</td> </tr> <tr> <td>"1"</td> <td>Query in hour</td> </tr> <tr> <td>"2"</td> <td>Query in day</td> </tr> </table>	"-1"	Historical query	"0"	Query in minutes	"1"	Query in hour	"2"	Query in day
"-1"	Historical query											
"0"	Query in minutes											
"1"	Query in hour											
"2"	Query in day											

Remarks	
---------	--

**/ data / datalogger / TaskID**

Description	Get the status of the inquiries
URL Structure	<b>https://10.0.0.1/data/datalogger/ TaskID</b> <b>Where:</b> TaskID: returns by post request: / data/datalogger
	GET: Returns the query status.
GET	Request: <b>GET / data/datalogger/TaskID</b>

[Example]: Get query status

● Request : **GET**

**/ data/datalogger/0**

Content-type: application/json

Response: 200 OK

If the query is not finished,

```
{
  "status": "querying",
  "timestamp": "xxxxxxx",
  "count": "4",
}
```

If the query is timeout

```
{
  "taskid": "xxx",
  "status": "timeout",
  "records": [
    {
      "timestamp": "1523424600",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    },
    {
      "timestamp": "1523424720",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    },
    {
      "timestamp": "1523424900",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    }
  ],
}
```

```
{
  "timestamp": "1523424960",
  "quality": "-32640",
  "partial": "0",
  "last": "0.0000",
  "min": "0.0000",
  "max": "0.0000",
  "avg": "0.0000"
},
{
  "timestamp": "1523425020",
  "quality": "-32640",
  "partial": "0",
  "last": "0.0000",
  "min": "0.0000",
  "max": "0.0000",
  "avg": "0.0000"
}
],
"count": "5"
```

}  
If the query is completed

```
{
  "taskid": "xxx",
  "status": "finish",
  "records": [
    {
      "timestamp": "1523424600",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    },
    {
      "timestamp": "1523424720",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    }
  ]
},
```

```

    {
      "timestamp": "1523424900",
      "quality": "-32640",
      "partial": "1",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    },
    {
      "timestamp": "1523424960",
      "quality": "-32640",
      "partial": "0",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    },
    {
      "timestamp": "1523425020",
      "quality": "-32640",
      "partial": "0",
      "last": "0.0000",
      "min": "0.0000",
      "max": "0.0000",
      "avg": "0.0000"
    }
  ],
  "count": "5"
}

```

[Example]: Query Historical data

- Request : **GET**

/ **data/ datalogger/0**

```

{
  "taskid": "xxx",
  "status": "finish",
  "records": [
    {
      "timestamp": "1523424652.0000",
      "value": "0.0000",
      "quality": "-32640"
    },
    {
      "timestamp": "1523424653.0000",

```

```

        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424654.0000",
        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424655.0000",
        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424656.0000",
        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424657.0000",
        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424658.0000",
        "value": "0.0000",
        "quality": "-32640"
    },
    {
        "timestamp": "1523424659.0000",
        "value": "0.0000",
        "quality": "-32640"
    }
    ],
    "count": "8"
}

```

PUT None

POST Request: POST / **data/datalogger/0**

[Example]: Send cancel command to stop query

- Request : **POST**

/ **data/datalogger/0**

Content-type: application/json

	<pre>{   "status": "cancel" }</pre> <p>Response: 200 OK</p>																																						
PATCH	None																																						
<ul style="list-style-type: none"> <li>Query value definitions :</li> </ul> <table border="1"> <thead> <tr> <th>Field</th> <th>Abbreviation</th> <th>Data Type</th> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Tag Name</td> <td>Tn</td> <td>String</td> <td>RW</td> <td>Tag name in datalogger</td> </tr> <tr> <td>Start time</td> <td>St</td> <td>String</td> <td>RW</td> <td>Start time of time. Unix timestamp 0: The first record time</td> </tr> <tr> <td>Ebd time</td> <td>Et</td> <td>String</td> <td>RW</td> <td>End time of time. Unix timestamp 0: The latest record time</td> </tr> <tr> <td>Query Type</td> <td>Tp</td> <td>String</td> <td>RW</td> <td>Query type</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td> <table border="1"> <tr> <td>"-1"</td> <td>Historical query</td> </tr> <tr> <td>"0"</td> <td>Query in minutes</td> </tr> <tr> <td>"1"</td> <td>Query in hour</td> </tr> <tr> <td>"2"</td> <td>Query in day</td> </tr> </table> </td> </tr> </tbody> </table>		Field	Abbreviation	Data Type	Property	Description	Tag Name	Tn	String	RW	Tag name in datalogger	Start time	St	String	RW	Start time of time. Unix timestamp 0: The first record time	Ebd time	Et	String	RW	End time of time. Unix timestamp 0: The latest record time	Query Type	Tp	String	RW	Query type					<table border="1"> <tr> <td>"-1"</td> <td>Historical query</td> </tr> <tr> <td>"0"</td> <td>Query in minutes</td> </tr> <tr> <td>"1"</td> <td>Query in hour</td> </tr> <tr> <td>"2"</td> <td>Query in day</td> </tr> </table>	"-1"	Historical query	"0"	Query in minutes	"1"	Query in hour	"2"	Query in day
Field	Abbreviation	Data Type	Property	Description																																			
Tag Name	Tn	String	RW	Tag name in datalogger																																			
Start time	St	String	RW	Start time of time. Unix timestamp 0: The first record time																																			
Ebd time	Et	String	RW	End time of time. Unix timestamp 0: The latest record time																																			
Query Type	Tp	String	RW	Query type																																			
				<table border="1"> <tr> <td>"-1"</td> <td>Historical query</td> </tr> <tr> <td>"0"</td> <td>Query in minutes</td> </tr> <tr> <td>"1"</td> <td>Query in hour</td> </tr> <tr> <td>"2"</td> <td>Query in day</td> </tr> </table>	"-1"	Historical query	"0"	Query in minutes	"1"	Query in hour	"2"	Query in day																											
"-1"	Historical query																																						
"0"	Query in minutes																																						
"1"	Query in hour																																						
"2"	Query in day																																						
Remarks	<p><b>Timeout:</b> Timeout for querying data logger is 20 seconds, if the interval between two GET requests is more than 20 seconds , the query will be canceled, data that has been queried can be get.</p> <p><b>Cancel:</b> If the query is canceled, data that has been queried can be get.</p> <p><b>Note:</b> In both above cases, the query result may be incomplete.</p>																																						

## 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	<b>https://10.0.0.1/sys/log_in/</b>
HTTP Method	PUT:
GET	None
PUT	Request: <b>PUT</b> /sys/log_in [Example]: <ul style="list-style-type: none"> <li>Request: <b>PUT</b> /sys/log_in</li> </ul> Content-type: application/json

	<pre>{   "password":"00000000" }</pre> <p>Response:</p> <pre>{   "sesion_id": "c9f4baf91d3e4ed7cfb18e598c5711f5", }</pre>			
<ul style="list-style-type: none"> <li>Resource value definitions :</li> </ul>				
Field	Abbreviation	Data Type	Property	Description
password	password	String	RW	User password
Session index	session_id	String	R	Session index for cookies
Remarks	<p>Session_id will be used in cookie for GET/PUT/POST/PATCH method.</p> <p>Cookie: ADAMSID=c9f4baf91d3e4ed7cfb18e598c5711f5</p>			

### 2.10.2 Log out

**/sys/log\_out**

Description	Log out the device
URL Structure	<b>https://10.0.0.1/sys/log_out/</b>
HTTP Method	PUT
GET	None
PUT	<ul style="list-style-type: none"> <li>Request: <b>PUT</b> /sys/log_out</li> <li>Content-type: application/json</li> </ul> <p>Response:</p> <pre>{   "Success":"Log out " }</pre>
<ul style="list-style-type: none"> <li>Resource value definitions :</li> </ul>	
Remarks	

## 2.11 Image Update

### 2.11.1 Image File Information Verification

**/sys/file\_verify**

Description	Get image file information from user.
URL Structure	<b>https://10.0.0.1/sys/file_verify/</b>
HTTP Method	POST
GET	None
PUT	None
POST	Request: <b>POST</b> /sys/file_verify [Example]: <ul style="list-style-type: none"><li>● Request: <b>POST</b> /sys/file_verify Content-type: application/json</li></ul> Response: <pre>{   "size": "5375",   "count": "2"   "detail": [     {       "name": "manifest.xml",       "size": "5007",     },     {       "name": "checksum.md5",       "size": "368",     },   ] }</pre>
	● Resource value definitions :
Remarks	This request must be executed before /sys/upload

### 2.11.2 Image Upload

**/sys/upload**

The standard way to upload files in a web application is to use a form with a special multipart/form-data encoding.

<RFC1521> In the case of multiple part entities, in which one or more different sets



of data are combined in a single body, a "multipart" Content-Type field must appear in the entity's header. The body must then contain one or more "body parts," each preceded by an encapsulation boundary, and the last one followed by a closing boundary. Each part starts with an encapsulation boundary, and then contains a body part consisting of header area, a blank line, and a body area.

Each body part is preceded by an encapsulation boundary. The encapsulation boundary **MUST NOT** appear inside any of the encapsulated parts. Thus, it is crucial that the composing agent be able to choose and specify the unique boundary that will separate the parts.

Encapsulation boundaries must not appear within the encapsulations, and must be no longer than 70 characters, not counting the two leading hyphens.

The encapsulation boundary following the last body part is a distinguished delimiter that indicates that no further body parts will follow. Such a delimiter is identical to the previous delimiters, with the addition of two more hyphens at the end of the line:

--gc0p4Jq0M2Yt08jU534c0p--

<RFC2388> "Multipart/form-data" can be used for forms that are presented using representations other than HTML (spreadsheets, Portable Document Format, etc), and for transport using other means than electronic mail or HTTP. This document defines the representation of form values independently of the application for which it is used.

"Multipart/form-data" contains a series of parts. Each part is expected to contain a content-disposition header [RFC 2183] where the disposition type is "form-data", and where the disposition contains an (additional) parameter of "name", where the value of that parameter is the original field name in the form. For example, a part might contain a header:

Content-Disposition: form-data; name="user"

with the value corresponding to the entry of the "user" field.

Field names originally in non-ASCII character sets may be encoded within the value of the "name" parameter using the standard method described in RFC 2047.

Description	Log in the device for configuration or image updating.
URL Structure	<b>https://10.0.0.1/sys/upload/</b>
HTTP Method	<b>POST:</b>
GET	None
PUT	None
POST	Request: <b>POST</b> /sys/upload [Example]: ● Request: <b>POST</b> /sys/upload Content-type: application/json  Content-Type:multipart/form-data;

	<pre> boundary=-----WebKitFormBoundaryuTETT0zNHgTxZV1W  ...  -----WebKitFormBoundaryuTETT0zNHgTxZV1W  Content-Disposition:form-data; name="ADAM-3600-image-1.1.2.bin"; filename="ADAM-3600-image-1.1.2.bin"  Content-Type: text/plain  &lt; data&gt;  -----WebKitFormBoundaryuTETT0zNHgTxZV1W-- </pre> <p>Response: 200 OK</p>
<ul style="list-style-type: none"> <li>● Resource value definitions :</li> </ul>	
Remarks	Must log in first

### 2.11.3 Image Update

#### **/sys/update**

Description	Begin to update the files of the ADAM-3600.
URL Structure	<b>https://10.0.0.1/sys/update/</b>
HTTP Method	POST
GET	None
PUT	None
POST	Request: <b>POST</b> /sys/update [Example]: <ul style="list-style-type: none"> <li>● Request: <b>POST</b> /sys/update</li> </ul> Content-type: application/json  Response: <pre> {   "Notice": "updating" } </pre>
<ul style="list-style-type: none"> <li>● Resource value definitions :</li> </ul>	
Remarks	Must log in first

### 2.11.4 Get information in update process

**/sys/update\_info**

Description	Get the updating notice information
URL Structure	<b>https://10.0.0.1/sys/update_info</b>
HTTP Method	GET
GET	Request: <b>GET</b> /sys/update_info [Example]: <ul style="list-style-type: none"><li>● Request: <b>GET</b> /sys/update_info</li></ul> Content-type: application/json  Response: { "updateinfo": "xxxxxxxxx..." }  xxxxxxxxx...: Update notice information
PUT	None
POST	None
● Resource value definitions :	
Remarks	

### 2.11.5 Image version information

**/sys/version?filename=/tmp/manifest.xml**

Description	Show the version information of the update package for ADAM-3600.
URL Structure	<b>https://10.0.0.1/sys/ version?filename=/tmp/manifest.xml</b>
HTTP Method	GET
GET	Request: <b>GET</b> /sys/version?filename=/tmp/manifest.xml [Example]: <ul style="list-style-type: none"><li>● Request: <b>GET</b> /sys/version?filename=/tmp/manifest.xml</li></ul> Content-type: application/json
PUT	None
POST	None
● Resource value definitions :	
Remarks	

## 2.12 System log file

### Get syslog of device

#### **/sys/log\_create**

Description	Create a thread to get the syslog of ADAM-3600.			
URL Structure	<b>https://10.0.0.1/sys/log_create</b>			
HTTP Method	GET			
GET	Request: <b>GET</b> /sys/log_create [Example]: ● Request: <b>GET</b> /sys/log_create Content-type: application/json  Response: { "TS":"1431773198" }			
PUT	None			
POST	None			
● Resource value definitions :				
Field	Abbreviation	Data Type	Property	Description
TimeStamp	TS	String	R	The timestamp to create the get syslog message thread
Remarks				

#### **/sys/log\_message**

Description	Get the syslog messages of ADAM-3600 between twice requests.
URL Structure	<b>https://10.0.0.1/sys/log_message</b>
HTTP Method	POST
GET	None
PUT	None
POST	Request: <b>POST</b> /sys/log_message  [Example]: ● Request: <b>POST</b> /sys/log_message Content-type: application/json Request:

```

{
  "TS":"1431773198"
}
Response:
{
  " syslog_info ":"xxxxxxx..."
}

xxxxxxx...: the syslog messages of ADAM-3600 between twice
requests.

```

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
TimeStamp	TS	String	R	The timestamp to create the get syslog message thread

Remarks

## 2.13 Network

`/data/gprs_info`

Description	Retrieves information about the GPRS information on specific slot.
URL	<b>https://10.0.0.1/data/gprs_info</b>
Structure	<b>https://10.0.0.1/data/gprs_info/xxxx</b>
HTTP Method	GET: Returns the representation of GPRS information.
GET	<p>Multi Request:  <b>GET / data/gprs_info/</b></p> <p>Single information Request:  <b>GET / data/gprs_info/xxxx</b></p> <p>xxxx: name of some gprs information, such as: PUB_IP</p> <p>[Example]:</p> <ul style="list-style-type: none"> <li>● Request : GET /data/ gprs_info  Content-type: application/json  Response: 200 OK</li> </ul> <pre>{     "MOBILE_MNO":      "No GPRS Service",     "MOBILE_MNT":      "No GPRS Service",     "MOBILE_MPN":      "0",     "MOBILE_MDT":      "0 Bytes",     "MOBILE_SIGNAL_QUALITY":  "Not Connected(0)",     "PUB_IP":          "0.0.0.0" }</pre> <ul style="list-style-type: none"> <li>● Request : GET /data/ gprs_info/ PUB_IP</li> </ul> <p>Content-type: application/json  Response: 200 OK</p> <pre>{     "0.0.0.0" }</pre>
PUT	None

● Resource value definitions :

Field	Abbreviation	Data Type	Property	Description
Mobile network operator	MOBILE_MNO	String	R	Mobile operator: China Mobile China Unicom China Telecom China Tietong
Mobile network type	MOBILE_MNT	String	R	Mobile network type: 2G, 3G, 4G
Mobile Phone Number	MOBILE_MPN	String	R	Mobile Phone Number 86xxxxxxxxxxx
Mobile data traffic	MOBILE_MDT	String	R	Mobile data traffic
Mobile signal quality	MOBILE_SIGN AL_QUALITY	String	R	Mobile signal quality 0-100
IP address	PUB_IP	String	R	IP address of the public network
Remark				