

# Ayla Service API Specification

---



Version: 8.1

Date Released: February 20, 2020

Document Number: AY006USA3-8.1

---

## Copyright Statement

© 2020 Ayla Networks, Inc. All rights reserved. Do not make printed or electronic copies of this document, or parts of it, without written authority from Ayla Networks.

The information contained in this document is for the sole use of Ayla Networks personnel, authorized users of the equipment, and licensees of Ayla Networks and for no other purpose. The information contained herein is subject to change without notice.

---

## Trademarks Statement

Ayla™ and the Ayla Networks logo are registered trademarks and service marks of Ayla Networks. Other product, brand, or service names are trademarks or service marks of their respective holders. Do not make copies, show, or use trademarks or service marks without written authority from Ayla Networks.

---

## Referenced Documents

Ayla Networks does not supply all documents that are referenced in this document with the equipment. Ayla Networks reserves the right to decide which documents are supplied with products and services.

---

## Contact Information

### Ayla Networks TECHNICAL SUPPORT and SALES

Contact Technical Support: <https://support.aylanetworks.com>  
or via email at [support@aylanetworks.com](mailto:support@aylanetworks.com)

Contact Sales: <https://www.aylanetworks.com/company/contact-us>

### Ayla Networks REGIONAL OFFICES

For a complete contact list of our offices in the US, China, Europe, Taiwan, and Japan, click:

<https://www.aylanetworks.com/company/contact-us>

# Table of Contents

1	Introduction .....	1
1.1	About this Document .....	1
1.2	Intended Audience .....	1
1.3	Related Documentation .....	1
1.4	API Browser .....	1
1.5	Document Conventions .....	2
1.6	Abbreviations and Acronyms .....	2
1.7	Glossary .....	3
1.8	Revision History .....	3
2.	Pre-Requisites .....	4
3	User Service API .....	4
3.1	Provider Auth .....	6
4.	User Metadata .....	7
5.	Device Service APIs .....	7
5.1	GET apiv1/dsns/:dsn/ registered_nodes .....	8
5.2	POST apiv1/devices/:device_id/registration_window .....	9
5.3	PUT apiv1/devices/:device_id/identify .....	10
6.	Device Metadata APIs .....	11
7.	Properties APIs .....	11
7.1	Downloading Data .....	12
7.2	Marking a Datapoint as Fetched .....	13
7.3	GET /apiv1/dsns/:dsn/properties/:property_name(:format) .....	13
7.4	GET /apiv1/dsns/<dsn>/properties/<property_name>/datapoints .....	14
7.5	POST apiv1/dsns/:dsn/properties/:property_name/datapoints .....	16
7.6	GET apiv1/dsns/<dsn>/properties/<property_name>/datapoints /time_range_filter .....	17
8.	Groups .....	19
9.	Schedules .....	21
9.1	POST apiv1/devices/<device_id>/schedules .....	21
9.2	GET apiv1/devices/<device_id>/schedules .....	24
9.3	GET apiv1/schedules/<id> .....	26

9.4	GET apiv1/devices/<device_id>/schedules/find_by_name .....	29
9.5	PUT apiv1/devices/<device_id>/schedules/<schedule_id> .....	31
9.6	PUT /apiv1/schedules/<schedule_id>/clear .....	35
9.7	GET apiv1/devices/<device_id>/schedules/base64 .....	37
9.8	GET apiv1/schedules/all/by_user .....	38
9.9	Schedule Actions.....	39
9.9.1	GET apiv1/schedules/<schedule_id>/schedule_actions.....	40
9.9.2	GET apiv1/schedule_actions/<id> .....	41
9.9.3	GET Schedules by Name.....	42
9.9.4	POST apiv1/schedules/<schedule_id>/schedule_actions .....	43
9.9.5	PUT apiv1/schedule_actions/<id> .....	44
9.9.6	DELETE apiv1/schedule_actions/<id> .....	45
10.	Triggers.....	46
10.1	Push Notification Notes .....	47
10.2	GET /apiv1/dsns/{dsn}/properties/{property_name}/triggers/all .....	48
10.3	POST apiv1/dsns/:dsn/properties/:property_name/triggers .....	49
11.	Device Notifications .....	51
12	Share APIs .....	52
13	Linked User Accounts .....	53
13.1	POST /api/v1/users/link.....	53
13.2	DELETE /api/v1/user//link/:user_id. ....	54
14	Time Zones.....	55
14.1	List of the Supported Time Zones .....	55
15.	Address.....	61
16	User and Device Service URLs .....	62

# 1 Introduction

The Ayla Cloud Service (ACS) is a secure, distributed, scalable IoT platform, exposing a RESTful API for clients. The Ayla Device and Ayla User are the core REST resources that can be managed through the API.

## 1.1 About this Document

This document describes the APIs that are used to access ACS, most of which are available in the [Ayla API Browser](#). For APIs that are not in the browser yet, the document includes the following for the API:

- Description of mandatory and optional input
- JSON examples along with their responses
- List of status codes that the API may return

## 1.2 Intended Audience

This document is intended for developers of web portals or mobile applications familiar with software using RESTful services over HTTPS. Users should be also familiar with the Ayla Cloud Service interaction with devices and users.

Requests to any of the Ayla Services (including ACS) require an access token. To obtain an access token, users need to log in to the Ayla User Service using the [Sign-In API](#).

## 1.3 Related Documentation

You may want to refer to the following documents available on [Ayla Connection](#):

- *Ayla Dashboard Help*
- *Customizing Notification Messages*, AY006USE3
- *Ayla API Browser Quick Reference Guide*, AY006FAP2

## 1.4 API Browser

The Ayla API Browser provides developers with an interactive environment to learn and test Ayla APIs. The browser currently contains a subset of Ayla APIs, including those indicated throughout this document. In the browser, you can view all pertinent details for the API and exercise the API to communicate directly with the Ayla system as you would from your own applications and devices. Each API request results in a Request URL,

Response Body, Response Code, and Response Headers. The pertinent details provided in the browser include:

- Description of mandatory and optional input
- JSON examples along with their responses
- List of status codes that the API may return

To access the API Browser, log in to the Ayla Developer's Portal on <https://developer.aylanetworks.com/>, and click either the link or icon for the API Browser. If you have not created an account on the portal, refer to the *Ayla Developer Portal User's Guide*, AY006UDP3, on [Ayla Connection](#).

## 1.5 Document Conventions

This document uses the following Ayla documentation conventions:

- File names, scripts, names of commands, properties in a file, code, and the like are in `Courier New` font, for example: Use the `psm-dump` command to ...
- Ancillary information that is important to emphasize is shown as:

**NOTE** The commands provided in the example assume your evaluation board is `mw300_rd` and your chip is `mw300`. If otherwise, make the appropriate substitutions.

- Information describing system failures or hazards that could cause data loss or damage a product is shown as:



Make sure that the appropriate data buffering is accounted for in deployed devices, especially where the loss of data is critical to the core functionality or the services provided by the systems.

## 1.6 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document.

ACS	Ayla Cloud Service
DSN	Device Serial Number
OEM	Original Equipment Manufacturer
ADS	Ayla Device Service

## 1.7 Glossary

<b>Caller</b>	This is a programmatic instance performing incoming calls to an API where the <a href="#">access token</a> (that was given to the user) facilitates the identity of the caller.
<b>Refresh Token</b>	This token is an alphanumeric random string generated by the Ayla User Service and provided with the <a href="#">access token</a> to authenticate the user's communication with the Ayla Cloud Services (ACS). The refresh token replaces the access token after a new user is logged in for 24 hours, continuing the user's communications with the ACS.
<b>Access Token</b>	This token is used to authenticate the user's communication with the Ayla Cloud Services. The user is authenticated based on a combination of the user's login, password, and APP ID/APP Secret. This user information is provided to the Ayla User API Authentication service upon the original request for an access token. Once assigned, this same access token is used for subsequent interactions with various Ayla APIs. However, the access token expires after 24 hours; at which time, the user may obtain a refresh token to continue communications with the Ayla Cloud Services.

## 1.8 Revision History

The following table provides a summary of the most recent updates to the content of this document.

Revision #	Date (yyyy-mm)	Change
6.0	2016-12	Complete revision - copy edited and formatted in the new Ayla template. Removed XML examples/responses and updated JSON examples and other content.
7.0	2017-02	Tested and updated several JSON examples, corrected results for JSON examples and input parameters, and added information on the new Ayla API Browser.
7.1	2017-08	Removed all APIs that are available in the API Browser.
7.2	2018-05	Updated copyright dates, and made other minor edits.
8.0	2019-10	Removed End-of-Life APIs, and replaced with Alternative APIs.
8.1	2020-02	Updated Device Service URLs

## 2. Pre-Requisites

Prior to using the APIs, developers should obtain an application ID and application secret from the Ayla Customer Dashboard as follows:

1. Log in to the Ayla Customer Dashboard.
2. In the Navigation Panel (left side), click **OEM Profile**.
3. Under the **Apps** tab, create a new app-id, or choose an existing app-id.

Developers should also know the [Device Service and User Service URLs](#) for their region. See also [Section 14](#) on Time Zones for this information.

All APIs require an HTTP Authorization Header with the following the format:

```
Authorization: auth_token <token>
```

where `<token>` is obtained from the user object returned upon a successful sign-in from a developer site.

All URL parameters should be URL encoded.

## 3 User Service API

User Service APIs are specific to the user account management. All of the following User Service APIs are available in the [Ayla API Browser](#) ; under Users, click Users: APIs specific to the user account.

<b>Sign-Up</b>	This API is used to create an account on the Ayla User Service.
<b>Sign In</b>	<p>This API is used to log a user into the Ayla Cloud Service (ACS). The API returns an access token and refresh token. The access token can be used to access other ACS APIs by passing the token along in the Authorization header. The refresh token can be used to obtain a new access token.</p> <p><b>NOTE:</b> After five consecutive failed sign-in attempts, the user account is locked for 30 minutes.</p> <p>The access token expires in one day (24 hours). The refresh token does not expire until the user explicitly logs out using the access token corresponding to the refresh token.</p>
<b>Refresh Token</b>	This API is used to obtain a new access token from the refresh token. The refresh token does not have an expiration. The token is only invalidated if a user explicitly logs out. In that case, a new refresh token is issued when the user logs back in.



<b>Resend Confirmation Token</b>	This API resends the confirmation instructions to an unconfirmed user.
<b>User Email Confirmation</b>	This API confirms account ownership based on the user's confirmation token received in the email.
<b>Reset Password Instructions</b>	This is the procedure to reset the password for a confirmed and approved user. The API initiates an email to be sent to the user with an embedded password token that is valid for 6 hours.
<b>Edit User Password</b>	This API enables the user currently logged in to edit the password.
<b>Reset Password with Token</b>	This API is used to verify the password reset token provided by the user, and resets the user's password. The password reset token is sent to the user in an email with reset password instructions; this token is valid for 6 hours after it is generated.
<b>Edit User Attributes</b>	The user logged in is allowed to edit only the attributes for the user. The OEM admin can also edit any user attributes. The email address cannot be modified with this call. If the email address is passed as an attribute, the user who corresponds with that email is updated rather than the current user.
<b>Update User Email</b>	This API is used to update the email of the current user. The user is logged out upon successful completion of this call. Hence, the caller/user must ensure that the new email address is accurate. Any errors, such as inadvertent typos, could prevent the user from being able to log in. Ayla recommends that the UI confirms the new email address before this API is called.
<b>Delete User</b>	The user currently logged in is allowed to delete the user account.
<b>Sign Out User</b>	This API is used to log the user out from the ACS. This invalidates their access token and the corresponding refresh token.
<b>Get User Attributes</b>	This API obtains the user attributes from the user's access token

Several of these APIs initiate an email to the user signed in to the Ayla User Service account. You can upload a customized email template to the Ayla User Service through the Ayla Customer Dashboard. For details on how to create, update, and upload a template, refer to *Customizing Notification Messages* on [Ayla Connection](#). The only User Service API that is not in the Ayla API Browser is Provider Auth. The remainder of this section provides information on the Provider Auth API.

## 3.1 Provider Auth

The Provider Auth API is used to complete the login with an OAuth2 provider. When logging in using an external provider, (such as, Google or Facebook), you receive a redirect URL first. This URL then returns code that the Provider Auth API exchanges for the oauth token and logs you in to the Ayla User Service.

The complete log-in process is as follows:

1. The user calls the log-in API, sending the name of the external provider as a parameter (for example google auth or facebook auth).
2. The user receives a code.
3. The user calls the Provider Auth API sending the code received in Step 2. During this step, the Provider Auth API returns the access token and then refreshes the token and the expiration time (as in the standard log-in process).
4. The user is either logged in or receives a reason for the log-in failure. (This error object is for any error that occurs during the log-in process.)

Table 1 provides the mandatory input required to log in an OAuth provider successfully.

**Table 1: Mandatory Input for Logging In an OAuth Provider**

Mandatory Input:		
Parameter	Description	Type
code	This is the code returned when the user calls the log-in API (Step 1 above.)	string
app_id	The application ID of the Ayla User Service.	string
redirect_url	The URL returned when logging in using an external provider, (such as, google auth or facebook auth).	string
provider	The provider that processes the oauth token (google_provider or facebook_provider).	string

### JSON Example

```
$ curl -X POST -d
"code=4%2F1F2i_X7snYgFytEKuaBTsWISYSep.4txQt7a1SLQdshQV0ieZDApsHwjoegI
&app_id=app_id&redirect_url=app_secret/token&provider=google_provider"
-H "Content-Type:application/json" https://<User Service
Url>/users/provider_auth.json
```

### Response

```
{ "authorization":
{ "access_token": "access_token", "refresh_token": "refresh_token", "expire
s_in": xxxx } }
```

## 4. User Metadata

All of the APIs for user metadata (described below) are available in the [Ayla API Browser](#) under Users, click UserMetaData: User MetaData.

<b>GET /api/v1/users/data</b>	This API retrieves a list of user data keys.
<b>GET /api/v1/users/data/:key</b>	This API retrieves a datum value for a key.
<b>POST /api/v1/users/data</b>	This API creates a new datum using the provided parameters for a user.
<b>Delete /api/v1/users/data/:key</b>	This API completely removes a datum by destroying it.

## 5. Device Service APIs

Device Service APIs are specific to the device resource and its associations, such as properties, schedules, and groups.

The Ayla Device Service (ADS) is the main service that communicates with the device. The service is responsible for managing all of the data to and from the device. The device service provides the following functionality:

- Viewing data to and from the device.
- Notifies the device using a notification service, when there is new data or tasks for the device to perform.
- Manages the triggers for data updates. When a trigger fires, it requires an application that notifies the application service.
- Manages groups and schedules for the device.

The following Device Service APIs are available in the [Ayla API Browser](#); under Devices, click Devices:Devices.

<b>GET apiv1/devices</b>	This API retrieves a list of user data keys.
<b>GET apiv1/devices/&lt;devID&gt;</b>	This API retrieves details for a device ID specified by the key.
<b>GET apiv1/dsns/&lt;dsn&gt;</b>	This API retrieves the details for a device identified by its Ayla Device Serial Number (DSN).
<b>GET apiv1/devices/register using IP, DSN, Registration</b>	This API retrieves a registration candidate that matches the input parameters. If the DSN is passed, the candidate has to match the specific DSN. Only the

Type	most recently connected candidate is returned if multiple devices are found.
POST apiv1/devices	This API creates the registration of a new device for a particular user.
PUT apiv1/devices/<device ID>	This API updates the device with the user-friendly display name of the device.
PUT apiv1/dsns/<dsn>	This API updates the device with the user-friendly display name of the device.
PUT apiv1/devices/<device_dsn>/transfer	This API updates the transfer of ownership from a current registered user to a target user.
POST apiv1/devices/<device_id>/locations	This API creates an update to a device location, allowing users to override the device location.
PUT apiv1/devices/<device_id>/cmds/factory_reset	This API updates a factory reset command, which is sent to the device Wi-Fi, Gateway, or Node with a device ID.
DELETE apiv1/devices/<devID>	This API is used to unregister a device from the current user.

The remainder of Section 5 provides information on the Device Service APIs that are not in the Ayla API Browser.

## 5.1 GET apiv1/dsns/:dsn/ registered\_nodes

This API applies to gateways. The API retrieves the list of node devices along with their respective properties for a particular user gateway. The HTTP Authorization header must include the auth\_token parameter as returned in the user object after a successful login. Table 2 provides the mandatory input and the output for this API.

**Table 2:** Mandatory Input to Obtain a List of Node Devices and Their Properties for a User Gateway

Mandatory Input:		
Parameter	Description	Type
dsn	The device serial number (DSN) of the device as returned when retrieving devices for a user.	string
Output:		
An array of registered nodes.		

## JSON Example

```
curl -H"Authorization: auth_token <"auth_token>
https://<device_service_url>/apiv1/dsns/AC000W0000000000/registered_nod
es.json
```

## Response

```
[{"device":{"product_name":"VR00ZN0000000000","model":"GenericNode","d
sn":"VR00ZN0000000000","oem_model":"nodemodel","user_id":0,"user_uuid":
"<user_uuid>","template_id":null,"mac":null,"lan_ip":null,"connected_a
t":"2016-12-
10T04:24:19Z","key":<key>,"registered":true,"lan_enabled":false,"has_p
roperties":true,"product_class":"","connection_status":"Online","power
":0,"ntwkaddr":null,"address":"address_01","device_type":"Node","gatew
ay_dsn":"AC000W0000000000","node_type":"Generic"}}]
```

## 5.2 POST apiv1/devices/:device\_id/registration\_window

This API opens and polls on the registration window, and only applies to gateways. Table 3 provides the mandatory input for this API.

**Table 3:** Mandatory Input to Open and Poll on the Registration Window

Mandatory Input:		
Parameter	Description	Type
device_id	The ID of the device as returned when retrieving devices for a user.	integer
duration	The time in seconds to complete this API (the duration for which the registration window is open). The default is 200 seconds. Use 0 value to close the registration immediately.	integer

## JSON Example

```
curl -X POST -H"Content-Type:application/json" -
H"Authorization:auth_token $auth_token" -d'{"duration":"200"}'
https://<device_service
url>/apiv1/devices/18981/registration_window.json
```

## Response Status Code

401 – Failure; the property is not found; the method is not allowed.

## 5.3 PUT apiv1/devices/:device\_id/identify

This API only applies to Zigbee nodes. This API sends a command to the Zigbee node so that it can identify itself among many nodes by turning itself on for a duration of time. This is useful when the user wants to identify one node among many nodes in a Zigbee gateway. Table 4 provides the mandatory and optional input for this API.

**Table 4:** Mandatory and Optional Input for the Ability to Turn a Device On or Off (Zigbee Node only)

Mandatory Input:		
Parameter	Description	Type
id	The device ID. The device must be identified on a node.	integer
Optional Input:		
Parameter	Description	Type
value	The amount of time (in seconds) before turning the device On or Off . The default setting is "On" for a time interval of 60 seconds.	string

### JSON Example

```
curl -X PUT -H"Authorization: auth_token $auth_token" -d
'{"value":"On", "time":"60"}' -H "Content -Type:application/json"
https://<device service url>/apiv1/devices/<device -id>/identify.json
{"id":"on 0x00158D0000626CAE"}
```

### Response Status Code

200 – Successful

401 – Unauthorized

404 – Device/Node does not exist

405 – Method not allowed or not supported on this device

## 6. Device Metadata APIs

These APIs are used to get metadata for a device. All of the Device Metadata APIs (described below) are available in the [Ayla API Browser](#); under Devices, click DeviceMetaData.

<b>GET</b> /apiv1/dsns/:dsn/data	This API retrieves a list of device data keys.
<b>GET</b> /apiv1/dsns/:dsn/data/:key	This API retrieves a data value (device datum) for a specific key.
<b>POST</b> /apiv1/dsns/:dsn/data	This API creates a new datum with the provided parameters for a device.
<b>PUT</b> /apiv1/dsns/:dsn/data/:key	This API updates a datum with the provided parameters.
<b>DELETE</b> /apiv1/dsns/:dsn/data/:key	This API completely removes a datum by destroying it.

## 7. Properties APIs

These APIs are used to obtain specific properties for a device and information for the device property. The following Properties APIs are available in the [Ayla API Browser](#). In the browser, click **Devices**, and then click **Properties**.

<b>GET</b> apiv1/dsns/<device_dsn>/properties	This API retrieves all of the properties for a particular device using the Device Serial Numbers (DSNs).
<b>POST</b> apiv1/dsns/:dsn/properties/:property_name/datapoints	This API creates one datapoint for a property specified by the property name.
<b>GET</b> /apiv1/properties/:property_id/trigger_apps	This API retrieves trigger apps for the property corresponding to id <key> that could be useful in application libraries.
<b>Uploading Data to the Datapoint</b>	For uploading data to a datapoint, use the file URL returned in the location element when creating a datapoint.

---

**Marking a Datapoint as Complete:**

**PUT /apiv1/datapoints/<datapoint-id>**

If the data is uploaded using multiple PUT requests and the total number of bytes is not known, an explicit call should be made to mark a datapoint as being complete. This is done using a PUT request with no body and no Content-Range header.

An attempt to upload more data to a completed datapoint results in a 406 (Unacceptable) status code.

---

**Marking a Datapoint as Discarded:**

**PUT /apiv1/datapoints/<datapoint-id>**

When the client wants to abandon a datapoint before completion, the client can indicate that the datapoint is discarded with an explicit call

The remainder of Section 7 provides information on the Properties APIs that are not in the Ayla API Browser.

## 7.1 Downloading Data

---

Download data as follows:

1. Call an HTTP GET on the location URL representing the datapoint to download the data from a file datapoint.
2. Call an HTTP GET on the file URL returned by the previous HTTP GET response.

---

**NOTE** The datapoint must be marked as completed before it can be fetched.

---

The client can download the whole file in one call or provide a beginning offset for the download.

URL: <location>

Method: GET

Format: n/a

## Response Status Code

Success: URL of the uploaded file. A GET needs to be done on that URL to fetch the actual file content.

406 – Failure, if datapoint is incomplete

All other status codes



## JSON Example

```
curl -k -X GET -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/devices/<device-
id>/properties/file_down/datapoints/1063224
{"datapoint":{"updated_at":"2015-01-21T16:40:53Z","created_at":"2015-
01-21T16:40:53Z","echo":false,"closed":true,"value":"https://<device
service url>apiv1/devices/<device-
id>/properties/test/datapoints/43c561e0-a18c-11e4-8c86-
b935dba94266.json","file":"https://<device service url>/datapoint-
43c561e0-a18c-11e4-8c86-
b935dba94266?AWSAccessKeyId=AKIAJAR3VCJIAFO6AXXX&Expires=1421863148&Si
gnature=drHshNV%2F2CVXbbhJenyONAp89dE%3D"}}}
```

## 7.2 Marking a Datapoint as Fetched

After downloading the entire data file, the client can indicate that the datapoint is fetched with an explicit call. In the absence of this, the datapoint is treated as unfetched.

URL: <location>

Method: PUT

Format: n/a

Request Payload: <datapoint><fetched>true</fetched></datapoint> or corresponding JSON

## JSON Example

```
curl -v -k -X PUT -H"Content-Type: application/json" -
d{"datapoint":{"fetched":"true"}} " https://<device service
url>/apiv1/datapoints/9396.json
```

## Response Status Code

200 – Success

406 – if datapoint is incomplete

## 7.3 GET /apiv1/dsns/:dsn/properties/:property\_name(.:format)

This API is used to retrieve details for the property corresponding property name.

**Table 5:** Mandatory and Optional Input for the Property Details Based on a Property Name

Mandatory Input:		
Parameter	Description	Type
<name>	The name assigned to the property (should match the property name of the device host application.	string

## JSON Example

```
$ curl -H"Authorization: auth_token 6edf3c9bcab0485a89c82090a6c61091" http://a
ds-dev.aylanetworks.com/apiv1/properties/test_property.json
```

## Response

```
{
  "property" : {
    "base_type" : "boolean",
    "value" : 1,
    "device_key" : 11,
    "name" : "Blue_LED",
    "key" : 92,
    "direction" : "input",
    "read_only" : false,
    "product_name" : "proto 1"
  }
}
```

## 7.4 GET /apiv1/dsns/<dsn>/properties/<property\_name>/datapoints

This API is used to retrieve the latest <count> datapoints for the property specified by its device serial number (DSN) and property name. If the limit is not specified for the datapoints, the Ayla Device Service returns the last 100 datapoints.

**Table 6:** Mandatory Input and the Output for Retrieving Datapoints Based on the DSN and Property Name

Mandatory Input:		
Parameter	Description	Type
<dsn>	The device serial number (DSN) for the device property.	string
<property name>	The name assigned to the property (should match the property name of the device host application.	string
limit	The number of datapoints requested for the page when not paginated, should be less than 100.	string
next	The ID of the last datapoint in the current page, which is available in the <metadata> section of the previous response.	string
previous	The ID of the first datapoint in the current page, which is available in the <metadata> section of the previous response.	string

is_forward_page	For the first or next page request, this is set to "true"; otherwise, this is set to "false."	boolean
paginated	If this is the request paginated response, this is set to "true"; otherwise, this is set to "false."	boolean
per_page	The number of datapoints per page, which should be less than 100.	string
filter	Parameters used to filter the datapoints based on the request.	string
created_at_since_date	This sets a time restriction for datapoints created after a specified date.	string
created_at_end_date	This sets a time restriction for datapoints created before a specified date	string
<b>Output</b>		
An array of datapoint objects in time ascending order.		

## JSON Example

```
curl -v -H
"Authorization: auth_token ba81ec8681f44c048a2df505ec4406ab"
"http://localhost:3000/apiv1/properties/4/datapoints.json?paginated=
true&is_forward_page=true&per_page=3"
```

## Response

```
{
  "meta": {
    "previous_page": "20bc1470-a71b-11e4-b170-8153baf3c000",
    "next_page": "d64114d0-a59f-11e4-ab79-5bf859e6955d",
    "current_page_number": 1
  },
  "datapoints": [
    {
      "datapoint": {
        "updated_at": "2015-01-26T21:11:04Z",
        "created_at": "2015-01-26T21:11:04Z",
        "echo": false,
        "value": 65.353454,
        "id": "d64114d0-a59f-11e4-ab79-5bf859e6955d",
        "metadata": {
          key1: "value1",
          key2: "value2"
        }
      }
    }
  ]
}
```

## 7.5 POST apiv1/dsns/:dsn/properties/:property\_name/datapoints

This API is used to create a datapoint for the property based on the property name and device serial number (DSN). You can also use:

POST apiv1/devices/<device\_id>/properties/<property\_name>/datapoints

This API creates a datapoint based on the property name and device ID (also called device key). All information provided below is the same for both APIs.

**Table 7:** Mandatory Input and the Output for Creating a Datapoint Based on the Property Name

Mandatory Input:		
Parameter	Description	Type
<datapoint><value>	The datapoint object that specifies a new value.	string
<metadata>	Each datapoint may contain a metadata field providing key/value pairs. For example:  <metadata key1="value1" key2="value2">  <b>NOTE:</b> No more than 10 metadata entries per datapoint are supported.	string
Output		
On success, the newly created datapoint is returned.		

- NOTE** The validations for this API are as follows:
- Boolean: 0 or 1
  - Integer value range: -2147483648 to 2147483647.
  - String value: Maximum length of 1024 Bytes
  - Decimal value range: -21474836.48 to 21474836.47. Maximum of two decimal places supported.

### JSON Example

```
$ curl -X POST -
H"Authorization: auth_token 6edf3c9bcab0485a89c82090a6c61091" -
H"Content-Type: application/json" -d '{ "datapoint":{"value":0,
"metadata": {"key1": "value1", "key2": "value2"}}}' http://ads-
dev.aylanetworks.com/apiv1/properties/test_property/datapoints.json
```

### Response

```
{
  "datapoint":{
    "created_at":"2011-11-15T06:22:44Z",
    "updated_at":"2011-11-15T06:22:44Z",
    "value":0,
    "metadata": {
```

```

        "key1": "value1",
        "key2": "value2"
    }
}

```

## 7.6 GET apiv1/dsns/<dsn>/properties/<property\_name>/datapoints/time\_range\_filter

This API is used to retrieve property datapoints specified by the device serial number (DSN) and property name. The API is based on multiple time ranges. For each time range, this API returns the first (exclusive) and last (inclusive) datapoint, such as:

```
datapoint.created_at > start_time && datapoint.created_at <= end_time
```

**Table 8:** Mandatory Input and the Output for Retrieving Datapoints Based on Time Ranges

Mandatory Input:		
Parameter	Description	Type
<dsn>	The unique device serial number for the device.	string
<property name>	The name assigned to the property (should match the property name of the device host application.	string
Output		
On success, this API returns filtered datapoints based on the multiple time ranges used.		

### JSON Example

```
$ curl -X GET -
H"Authorization: auth_token 8e17f384baf147a397937a7413149705" -H
"Content-Type: application/json" -d '{"time_range_filter":
[{"time_range_id": "Year 2014", "start_time": "2014/01/01 00:00:00",
"end_time": "2014/01/31 23:59:59"}]}' http://ads-
dev.aylanetworks.com/apiv1/dsns/AC0000W83748/properties/test_prop_name
/datapoints/time_range_filter.json
```

### Response

```

{
  "property": {
    "name": "test_prop_name",
    "base_type": "string",
    "read_only": false,
    "direction": "input",
    "scope": "user",
    "data_updated_at": "2014-01-10T16:54:20Z",
    "key": 387,
    "device_key": 3,
  }
}

```

```

        "product_name": "Device",
        "track_only_changes": false,
        "display_name": "prop1",
        "value": "hellooo world"
    },
    "time_ranges": [
        {
            "time_range_id": "Year 2014",
            "datapoints": [
                {
                    "updated_at": "2014-01-08T14:05:55Z",
                    "created_at": "2014-01-08T14:05:55Z",
                    "echo": false,
                    "value": null
                },
                {
                    "updated_at": "2014-01-10T16:54:20Z",
                    "created_at": "2014-01-10T16:54:20Z",
                    "echo": false,
                    "value": "hellooo world"
                }
            ]
        }
    ]
}

```

## 8. Groups

The Group APIs can be used to manage a logical group of devices, such as kitchen lights or all living room devices.

---

**NOTE** The maximum number of devices in a group cannot exceed 10.

---

The APIs for creating and modifying groups (described below) are available in the [Ayla API Browser](#). This section provides information and JSON examples (with responses and results) on how to do the following:

<b>GET /apiv1/groups</b>	This API retrieves a list of groups for the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>POST /apiv1/groups</b>	This API creates a new group for the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>GET /apiv1/groups/:group_id</b>	This API retrieves a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>PUT /apiv1/groups/:group_id</b>	This API updates a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>DELETE /apiv1/groups/:group_id</b>	This API destroys a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>POST /apiv1/groups/:group_id/devices</b>	This API adds a single device to the group. A user is indicated by the auth_token in the authorization header of the HTTP request.
<b>DELETE /apiv1/groups/:group_id/devices/:device_id</b>	<p>This API removes a single device from the group. A user is indicated by the auth_token in the authorization header of the HTTP request.</p> <p><b>NOTE:</b> The device is only removed from the group's list, not destroyed.</p>
<b>POST /apiv1/groups/:group_id/batched_datapoints</b>	This API changes the values of the properties on one or more devices belonging to a group. A user is indicated by the auth_token in the authorization header of the HTTP request.

---

**POST /apiv1/groups/:group\_id/  
datapoints**

This API changes values of a single property on all of the devices belonging to the group. A user is indicated by the auth\_token in the authorization header of the HTTP request.

**NOTE:** This API performs all value changes that are valid and possible and ignores invalid requests. Successful changes are indicated in the returned hash.

---

**GET /apiv1/groups/:group\_id/  
datapoints**

This API obtains values for a list of properties (by name) from all of the devices belonging to the group. A user is indicated by the auth\_token in the authorization header of the HTTP request.



## 9. Schedules

This section provides APIs that pertain to creating, listing, retrieving, updating, clearing/destroying schedules and details about the schedules. Schedules are delivered to the device and are fired from the device; they continue to work even without internet connectivity.

### 9.1 POST apiv1/devices/<device\_id>/schedules

This API creates a schedule for a device with the device ID. Table 9 provides the mandatory and optional input required for this API.

**NOTE** Deleting a Schedule is not currently available.

**Table 9:** Mandatory and Optional Input for Creating a Schedule with the Device ID

Mandatory Input:		
Parameter	Description	Type
device id	The ID of the device obtained using GET/apiv1/devices.	integer
start-time-each-day	The start time of the schedule in local time, using the 24-hour format (HH:mm:ss).	time
direction	This indicates whether the schedule is directed "to device" or "from device" ("input" or "output")	string
display_name	User-friendly schedule name that displays on the UIs. Default value: 'name' when not part of input.	string
name	The name of the schedule and is used as an identifier	string
Optional Input:		
Parameter	Description	Type
active	This is used for pausing and resuming the schedule. If this field is not set the default is set to "true."	string
end-time-each-day	The end time in local time, using the 24-hour format (HH:mm:ss).	string
time-before-end	The time before end in local time, using the 24-hour format (HH:mm:ss).	string
utc	This indicates the schedule to be set to the UTC time zone (Coordinated Universal Time). If this field is not set, the default is set to "false."	string
start_date	The date when the schedule starts (yyyy-mm-dd).	string

Optional Input:		
Parameter	Description	Type
end_date	The date when the schedule stops (yyyy-mm-dd).	string
days_of_week	An array of day#: [1,2,3,4,5,6,7], with 1 representing Sunday and so on. If this field is not set, the default is set to "every day."	integer
days_of_month	An array of day#: [1,2,3,4,5,6,7, 8 9..32], with 1 representing the first day of the month, and so on. If this field is not set, the default is set to "every day." As a special case, "32" represents the last day of the month, whether it's the 28, 30, or 31.	integer
months_of_year	An array of month#: [1,2,3,4,...12] with 1 representing January. If this field is not set, the default is set to "every month." The default is set to "every occurrence."	integer
day_occur_of_month	An array of occurrence #'s: [1,2,3,4,5,6]. This is the day of occurrence in the month. For example, the first Sunday or the second Sunday of month. As a special case, 6 means "last occurrence of the day in the month."	integer
Duration	The duration in seconds.	integer
interval	The interval in seconds.	integer
schedule_actions	A list of scheduled actions	string

## JSON Example

```
$ curl -k -X POST -H"Content-type: application/json" -d'{"schedule":
{"name":"night schedule", "display_name":"night schedule",
"start_date": "2013-06-01", "end_date": "2013-07-01", "direction":
"input", "start_time": "20:15:00", "end_time": "22:00:00",
"schedule_actions": [{"name": "Blue_LED", "base_type":"boolean",
"value": 1, "in_range": true, "at_start": false, "at_end": false,
"type": "SchedulePropertyAction"}]}' -H"Authorization: auth_token
<auth_token>"
https://<device service url>/apiv1/devices/3/schedules.json
```

## Response

```
{
  "schedule": {
    "active": true,
    "day_occur_of_month": null,
    "days_of_month": [
      1,
```

```
2,  
3,  
4,  
5,  
6,  
7,  
8,  
9,  
10,  
11,  
12,  
13,  
14,  
15,  
16,  
17,  
18,  
19,  
20,  
21,  
22,  
23,  
24,  
25,  
26,  
27,  
28,  
29,  
30,  
31  
],  
"days_of_week": [  
  1,  
  2,  
  3,  
  4,  
  5,  
  6,  
  7  
],  
"device_id": 3,  
"direction": "input",  
"duration": null,  
"end_date": "2013-07-01",  
"end_time": "22:00:00",  
"time_before_end": "",  
"interval": null,  
"months_of_year": [  
  1,  
  2,  
  3,  
  4,  
  5,  
  6,  
  7,  
  8,  
  9,
```

```

        10,
        11,
        12
    ],
    "name": "night schedule",
    "display_name": "night schedule",
    "start_date": "2013-06-01",
    "start_time": "20:15:00",
    "version": "1",
    "key": 36,
    "schedule_actions": [
        {
            "active": true,
            "at_end": false,
            "at_start": false,
            "base_type": "boolean",
            "in_range": true,
            "name": "Blue_LED",
            "key": 31,
            "value": 1
        }
    ]
}

```

## Response

201 – Success, created schedule

400 – Failure, bad request

422 – Unprocessable Entity

## 9.2 GET apiv1/devices/<device\_id>/schedules

This API lists all of the schedules for a device with <device\_id> id. Table 10 provides the mandatory input required for this API and the output.

**Table 10:** Mandatory Input and Output for Listing all of the Schedules for a Device

Mandatory Input:		
Parameter	Description	Type
device_id	The device ID that can be obtained from GET /devices.	integer
Output:		
An array of schedules.		

## JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/devices/<device-id>/schedules.json
```

```
[
  {
    "schedule": {
      "active": true,
      "day_occur_of_month": null,
      "days_of_month": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12,
        13,
        14,
        15,
        16,
        17,
        18,
        19,
        20,
        21,
        22,
        23,
        24,
        25,
        26,
        27,
        28,
        29,
        30,
        31
      ],
      "days_of_week": [
        1,
        2,
        3,
        4,
        5,
        6,
        7
      ],
      "device_id": 3,
      "direction": "input",
      "duration": 0,
      "end_date": "",
    }
  }
]
```

```

        "end_time_each_day": "",
        "fixed_actions": false,
        "interval": 0,
        "months_of_year": [
            1,
            2,
            3,
            4,
            5,
            6,
            7,
            8,
            9,
            10,
            11,
            12
        ],
        "name": "night schedule",
        "display_name": "night schedule",
        "start_date": "2013-06-11",
        "start_time_each_day": "13:30:00",
        "time_before_end": "",
        "utc": false,
        "version": "1",
        "key": 7
    }
}
]

```

## 9.3 GET apiv1/schedules/<id>

This API retrieves the details for the schedule corresponding to the schedule ID. Table 11 provides the mandatory input required for this API.

**Table 11:** Mandatory Input for Retrieving Details Per the Schedule ID

Mandatory Input:		
Parameter	Description	Type
id	This is the schedule ID obtained using a GET on all of the device schedules.	integer

### JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/schedules/<schedule-key>.json
```

### Response

```

{
    "schedule": {
        "active": true,
        "day_occur_of_month": null,

```

```
"days_of_month": [
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13,
  14,
  15,
  16,
  17,
  18,
  19,
  20,
  21,
  22,
  23,
  24,
  25,
  26,
  27,
  28,
  29,
  30,
  31
],
"days_of_week": [
  1,
  2,
  3,
  4,
  5,
  6,
  7
],
"device_id": 3,
"direction": "input",
"duration": null,
"end_date": "2013-07-01",
"end_time_each_day": "22:00:00",
"fixed_actions": false,
"time_before_end": "",
"interval": null,
"months_of_year": [
  1,
  2,
  3,
  4,
  5,
  6,
```

```

        7,
        8,
        9,
        10,
        11,
        12
    ],
    "name": "night schedule",
    "display_name": "my night schedule",
    "start_date": "2013-06-01",
    "start_time_each_day": "20:15:00",
    "version": "1",
    "key": 34,
    "schedule_actions": [
        {
            "active": true,
            "at_end": true,
            "at_start": false,
            "base_type": "boolean",
            "in_range": false,
            "name": "Blue_LED",
            "key": 28,
            "value": 0
        },
        {
            "active": true,
            "at_end": false,
            "at_start": true,
            "base_type": "boolean",
            "in_range": false,
            "name": "Blue_LED",
            "key": 29,
            "value": 1
        }
    ]
}

```



## 9.4 GET apiv1/devices/<device\_id>/schedules/find\_by\_name

This API retrieves details for the schedule corresponding to the name <find\_by\_name> and the device\_id. Table 12 provides the mandatory input required for this API and the output.

**Table 12:** Mandatory Input and Output for Retrieving Details for the Schedule <find\_by\_name>

Mandatory Input:		
Parameter	Description	Type
name	The schedule name obtained using a GET on all device schedules.	string
device_id	The device ID obtained using a GET on all devices.	integer
Output:		
schedule object		

### JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device
service url>/apiv1/devices/<device-
id>/schedules/find_by_name.json?name=night+schedule
```

### Response

```
{
  "schedule": {
    "active": true,
    "day_occur_of_month": null,
    "days_of_month": [
      1,
      2,
      3,
      4,
      5,
      6,
      7,
      8,
      9,
      10,
      11,
      12,
      13,
      14,
      15,
      16,
      17,
      18,
      19,
      20,
```

```

        21,
        22,
        23,
        24,
        25,
        26,
        27,
        28,
        29,
        30,
        31
    ],
    "days_of_week": [
        1,
        2,
        3,
        4,
        5,
        6,
        7
    ],
    "device_id": 3,
    "direction": "input",
    "duration": null,
    "end_date": "2013-07-01",
    "end_time_each_day": "22:00:00",
    "fixed_actions": false,
    "time_before_end": "",
    "interval": null,
    "months_of_year": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12
    ],
    "name": "night schedule",
    "display_name": "my night schedule",
    "start_date": "2013-06-01",
    "start_time_each_day": "20:15:00",
    "version": "1",
    "key": 34,
    "schedule_actions": [
        {
            "active": true,
            "at_end": true,
            "at_start": false,
            "base_type": "boolean",
            "in_range": false,
            "name": "Blue_LED",

```

```

        "key": 28,
        "value": 0
      },
      {
        "active": true,
        "at_end": false,
        "at_start": true,
        "base_type": "boolean",
        "in_range": false,
        "name": "Blue_LED",
        "key": 29,
        "value": 1
      }
    ]
  }
}

```

## 9.5 PUT apiv1/devices/<device\_id>/schedules/<schedule\_id>

This API updates the schedule specified by the schedule ID. Table 13 provides the mandatory and optional input required for this API.

**Table 13:** Mandatory and Optional Input for Updating the Schedule Per the Schedule ID

Mandatory Input:		
Parameter	Description	Type
start_time	The start time in local time. Validation: 24-hour format (HH:mm:ss).	string
direction	This indicates whether the schedule is directed "to device" or "from device" ("input" or "output")	string
active	This is used for pausing and resuming the schedule. If this field is not set, the default is set to "true."	string
name	This is the name used by the service to identify the schedule.	string
Optional Input:		
Parameter	Description	Type
end_time	The end time in local time. Validation: 24-hour format (HH:mm:ss).	string
time-before-end	The time before end in local time, using the 24-hour format (HH:mm:ss).	string
display_name	User-friendly schedule name that displays on the UIs. Default value: 'name' when not part of input.	string

utc	This indicates the schedule to be set to the UTC time zone (Coordinated Universal Time). If this field is not set, the default is set to "false."	string
start_date	The date when the schedule starts (yyyy-mm-dd).	string
end_date	The date when the schedule stops (yyyy-mm-dd).	string
days_of_week	An array of day#: [1,2,3,4,5,6,7], with 1 representing Sunday and so on. If this field is not set, the default is set to "every day."	integer
days_of_month	An array of day#: [1,2,3,4,5,6,7, 8 9..32], with 1 representing the first day of the month, and so on. If this field is not set, the default is set to "every day." As a special case, "32" represents the last day of the month, whether it's the 28, 30, or 31.	integer
months_of_year	An array of month#: [1,2,3,4,...12] with 1 representing January. If this field is not set, the default is set to "every month." The default is set to "every occurrence."	integer
day_occur_of_month	An array of occurrence #'s: [1,2,3,4,5,6]. This is the day of occurrence in the month. For example, the first Sunday or the second Sunday of month.	integer
Duration	The duration in seconds.	integer
interval	The interval in seconds.	integer
schedule_actions	A list of scheduled actions	string

## JSON Example

```
$ curl -k -X PUT -H"Content-type: application/json" -d'{"schedule":
{"name": "morning schedule", "display_name": "morning schedule",
"start_date": "2013-06-15", "end_date": "2013-08-01", "direction":
"input", "start_time": "10:00:00", "end_time": "11:00:00",
"schedule_actions": [{"name": "Blue_LED", "key":
28, "base_type": "boolean", "value": 1, "in_range": false, "at_start":
true, "at_end": false}, {"name": "Blue_LED", "key":
29, "base_type": "boolean", "value": 0, "in_range": false, "at_start":
false, "at_end": true}]}' -H"Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/devices/<device id>/schedules/<schedule id>.json
```

## Response

```
{
  "schedule": {
    "active": true,
    "day_occur_of_month": null,
    "days_of_month": [
```

```
1,  
2,  
3,  
4,  
5,  
6,  
7,  
8,  
9,  
10,  
11,  
12,  
13,  
14,  
15,  
16,  
17,  
18,  
19,  
20,  
21,  
22,  
23,  
24,  
25,  
26,  
27,  
28,  
29,  
30,  
31  
],  
"days_of_week": [  
  1,  
  2,  
  3,  
  4,  
  5,  
  6,  
  7  
],  
"device_id": 3,  
"direction": "input",  
"duration": null,  
"end_date": "2013-08-01",  
"end_time_each_day": "11:00:00",  
"fixed_actions": false,  
"time_before_end": "",  
"interval": null,  
"months_of_year": [  
  1,  
  2,  
  3,  
  4,  
  5,  
  6,  
  7,
```

```

        8,
        9,
        10,
        11,
        12
    ],
    "name": "morning schedule",
    "display_name": "morning schedule",
    "start_date": "2013-06-15",
    "start_time_each_day": "10:00:00",
    "version": "1",
    "key": 34,
    "schedule_actions": [
        {
            "active": true,
            "at_end": false,
            "at_start": true,
            "base_type": "boolean",
            "in_range": false,
            "name": "Blue_LED",
            "key": 28,
            "value": 1
        },
        {
            "active": true,
            "at_end": true,
            "at_start": false,
            "base_type": "boolean",
            "in_range": false,
            "name": "Blue_LED",
            "key": 29,
            "value": 0
        }
    ]
}

```

## Response Status Code

201 – Success, updated schedule

400 – Failure, bad request

422 – Unprocessable Entity

## 9.6 PUT /apiv1/schedules/<schedule\_id>/clear

This API clears the schedule specified by schedule ID. The API marks the schedule as cleared (inactive) and destroys all of the scheduled actions. Table 14 provides the mandatory input required for this API.

**Table 14:** Mandatory Input for Clearing Schedules Per the Schedule ID

Mandatory Input:		
Parameter	Description	Type
schedule_id	This is the schedule_id returned when the schedule is created.	integer

### JSON Example

```
$ curl -k -X PUT -H"Content-type: application/json" -H"Authorization:
auth_token <auth_token>"
https://<device service url>/apiv1/schedules/<key>/clear.json
```

### Response

```
{
  "schedule": {
    "active": false,
    "day_occur_of_month": [
      1,
      2,
      3,
      4,
      7
    ],
    "days_of_month": [
      1,
      2,
      3,
      4,
      5,
      6,
      7,
      8,
      9,
      10,
      11,
      12,
      13,
      14,
      15,
      16,
      17,
      18,
      19,
      20,
```

```

        21,
        22,
        23,
        24,
        25,
        26,
        27,
        28,
        29,
        30,
        32
    ],
    "days_of_week": [
        1,
        2,
        3,
        4,
        5,
        6
    ],
    "device_id": 3,
    "direction": "input",
    "duration": 4,
    "end_date": "",
    "end_time_each_day": "",
    "fixed_actions": false,
    "time_before_end": "",
    "interval": 1,
    "months_of_year": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11
    ],
    "name": "morning",
    "name": "my morning schedule",
    "start_date_each_day": "2013-11-11",
    "start_time_each_day": "10:00:00",
    "version": "1",
    "key": 12,
    "schedule_actions": []
}

```

## Response Status Code

201 – Success, updated schedule

400 – Failure, bad request

422 – Unprocessable Entity



## 9.7 GET apiv1/devices/<device\_id>/schedules/base64

This API obtains the corresponding base64 that represents the sent schedule. Table 15 provides the mandatory input required for this API.

**Table 15:** Mandatory Input for Obtaining base64 Representing the Schedule

Mandatory Input:		
Parameter	Description	Type
direction	This indicates whether the schedule is directed "to device" or "from device" ("input" or "output")	string
start_time	The start time in local time, 24-hour format (HH:mm:ss).	string
name	This is the name of the schedule displayed for reference.	string
device_id	The device ID obtained using a GET on all devices.	integer

### JSON Example

```
$ $ curl -X GET -H"Content-Type: application/json" -d'{"schedule":  
{"name": "test", "direction": "input", "start_time": "20:15:00"}}' -  
H"Authorization: auth_token <auth_token>https://<device\_service  
url>/apiv1/devices/<device-id>/schedules/base64.json
```

### Response

```
{  
  "base64": "MwEBLQQAARzE"  
}
```

### Response Status Code

200 – Success

422 – Failure, unprocessable Entity

## 9.8 GET apiv1/schedules/all/by\_user

This API lists all of the schedules for the devices to which the current user has access. The API also obtains schedules from linked user accounts. There is no mandatory input for this API, and the output is an array of schedules.

### JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device>service url>/apiv1/schedules/all/by\_user.json
```

### Response

```
[
  {
    "schedule": {
      "active": true,
      "day_occur_of_month": null,
      "days_of_month": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12,
        13,
        14,
        15,
        16,
        17,
        18,
        19,
        20,
        21,
        22,
        23,
        24,
        25,
        26,
        27,
        28,
        29,
        30,
        31
      ],
      "days_of_week": [
        1,
        2,
```

```

        3,
        4,
        5,
        6,
        7
    ],
    "device_id": 3,
    "direction": "input",
    "duration": 0,
    "end_date": "",
    "end_time_each_day": "",
    "fixed_actions": false,
    "interval": 0,
    "months_of_year": [
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12
    ],
    "name": "night schedule",
    "display_name": "night schedule",
    "start_date": "2013-06-11",
    "start_time_each_day": "13:30:00",
    "time_before_end": "",
    "utc": false,
    "version": "1",
    "key": 7
    }
}
]

```

## 9.9 Schedule Actions

Schedule actions specify the actions that need to be taken when the schedule fires.

## 9.9.1 GET apiv1/schedules/<schedule\_id>/schedule\_actions

This API retrieves a list of all the scheduled actions for a schedule based on the schedule ID. Table 16 provides the mandatory input required for this API.

**Table 16:** Mandatory Input for Listing Schedule Actions Based on the Schedule ID

Mandatory Input:		
Parameter	Description	Type
schedule_id	This is the schedule_id returned when the schedule is created.	string
type	This is the type of schedule action. Ayla Networks currently supports SchedulePropertyAction.	string

### JSON Example

```
$ curl -k -X GET -H"Content-type: application/json" -d'{"type":
"SchedulePropertyAction"}' -H"Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/schedules/<schedule\_id>/schedule\_actions.json
```

### Response

```
[
  {
    "schedule_action": {
      "active": true,
      "base_type": "boolean",
      "name": "Blue_LED",
      "schedule_id": 36,
      "key": 31,
      "value": 1
    }
  }
]
```

## 9.9.2 GET apiv1/schedule\_actions/<id>

This API retrieves details for the schedule action that corresponds to the schedule action ID. Table 17 provides the mandatory input required for this API and the output.

**Table 17:** Mandatory Input and Output for Retrieving Details for the Schedule Action

Mandatory Input:		
Parameter	Description	Type
id	The schedule_action ID is obtained using a GET of all schedules.	string
type	This is the type of schedule action. Ayla Networks currently supports SchedulePropertyAction.	string
Output:		
schedule_action object		

### JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>"  
https://<device service url>/apiv1/schedule\_actions/<key>.json
```

### Response

```
{  
  "schedule_action": {  
    "active": true,  
    "base_type": "boolean",  
    "name": "Blue_LED",  
    "schedule_id": 36,  
    "key": 31,  
    "value": 1  
  }  
}
```

## 9.9.3 GET Schedules by Name

The API is:

GET `apiv1/schedules/<schedule_id>/schedule_actions/find_by_name.xml?name=<name>`

This API retrieves details for the schedule actions that correspond to a specific name and schedule ID. Table 18 provides the mandatory input required for this API and the output.

**Table 18:** Mandatory Input and Output for Retrieving Details Corresponding to Schedule Name and ID

Mandatory Input:		
Parameter	Description	Type
name	This is the schedule_action name obtained using a GET of all schedule actions.	string
schedule_id	This is the schedule_id returned when the schedule is created.	integer
type	This is the type of schedule action. Ayla Networks currently supports SchedulePropertyAction.	string
Output:		
array of schedule_action objects		

### JSON Example

```
$ curl -H"Authorization: auth_token <auth_token>"
https://<device service
url>/apiv1/schedules/<schedule id>/schedule_actions/find_by_name.json?
name=Blue_LED
```

### Response

```
[
  {
    "schedule_action": {
      "active": true,
      "at_end": false,
      "at_start": true,
      "base_type": "boolean",
      "in_range": false,
      "name": "Blue_LED",
      "schedule_id": 61,
      "key": 64,
      "value": 1
    }
  }
]
```

## 9.9.4 POST apiv1/schedules/<schedule\_id>/schedule\_actions

This API creates a schedule action for a schedule with the schedule id. . Table 19 provides the mandatory and optional input required for this API.

**Table 19:** Mandatory and Optional Input for Creating a Schedule with the Schedule ID

Mandatory Input:		
Parameter	Description	Type
name	This is the name of the property related to this schedule_action.	string
type	This is the type of schedule action. Ayla Networks currently supports SchedulePropertyAction.	string
base_type	Ayla Networks currently only supports "string," "integer," "Boolean," and "decimal."	
in_range	(*1): sets the current time in the range of the specified period.	
at_start	(*1): sets the current time at the start of the specified period.	
at_end	(*1): sets the current time at the end of the specified period.	
active	This is used for pausing and resuming the schedule. If this field is not set the default is set to "true."	
Optional Input:		
Parameter	Description	Type
value	This is the value for schedule_action base_type. (*1): only one parameter from in_range, at_start, and at_end must be set.	

### JSON Example

```
$ curl -k -X POST -H"Content-type: application/json" -d'{"schedule_action": {"name": "Blue_LED", "base_type": "boolean", "value": 1, "in_range": true, "type": "SchedulePropertyAction"}}' -H"Authorization: auth_token <auth-token>"https://<device service url>/apiv1/schedules/<schedule_id>/schedule_actions.json
```

### Response

```
{
  "schedule_action": {
    "active": true,
    "base_type": "boolean",
    "name": "Blue_LED",
    "schedule_id": 37,
    "key": 34,
    "value": 1
  }
}
```

## Response Status Code

Success - 200

### 9.9.5 PUT apiv1/schedule\_actions/<id>

This API updates the schedule specified by an ID. Table 20 provides the mandatory and optional input required for this API.

**Table 20:** Mandatory and Optional Input for Updating a Schedule with the Schedule ID

Mandatory Input:		
Parameter	Description	Type
id	This is the schedule ID generated when a schedule is created.	integer
Optional Input:		
Parameter	Description	Type
value	This is the value for schedule_action base_type. (*1): only one parameter from in_range, at_start, and at_end must be set.	integer
name	This is the name of the property related to this schedule_action.	string
type	This is the type of schedule action. Ayla Networks currently supports SchedulePropertyAction.	string
base_type	Ayla Networks currently only supports "string," "integer," "Boolean," and "decimal."	base_type
in_range	(*1): sets the current time in the range of the specified period.	
at_start	(*1): sets the current time at the start of the specified period.	
at_end	(*1): sets the current time at the end of the specified period.	
active	This is used for pausing and resuming the schedule. If this field is not set the default is set to "true."	

## JSON Example

```
$ curl -k -X PUT -H"Content-type: application/json" -
d'{"schedule_action": {"name": "Blue_LED", "base_type": "boolean",
"value": 1, "in_range": true, "type": "SchedulePropertyAction"}}' -
H"Authorization: auth_token <auth-token>" https://<device service
url>/apiv1/schedule\_actions/1.json
```

## Response

Success - 200

Failure: all other status codes



## 9.9.6 DELETE apiv1/schedule\_actions/<id>

This API deletes the schedule action specified by an ID. Table 21 provides the mandatory input required for this API.

**Table 21:** Mandatory Input for Deleting a Schedule with the Schedule ID

Mandatory Input:		
Parameter	Description	Type
id	This is the schedule ID generated when a schedule is created.	integer

### JSON Example

```
$ curl -X DELETE -H"Content-Type: application/json" -H"Authorization:
auth_token <auth_token>" https://<device service
url>/apiv1/schedule\_actions/1.json
```

### Response

```
{
  "schedule_action": {
    "name": "Blue_LED",
    "key": 1
  }
}
```

### Response Status Code

200 – Success

422 – Failure: unprocessable entity, not allowed to delete schedule action that belongs to a schedule which has fixed actions enabled

## 10. Triggers

Triggers are basic conditions that are checked every time a new datapoint for a given property is created. When the conditions match during these checks, the trigger occurs and one or more applications are launched to provide notifications. You can customize emails that are sent from the Ayla Device Service with customer logos and text; for more information, refer *Customizing Notification Messages*, AY006USE0-2 (available [Ayla Connection](#).)

All of the APIs for triggers (described below) are available in the [Ayla API Browser](#).

<b>PUT</b> <code>apiv1/triggers/&lt;trigger_id&gt;</code>	This API updates a trigger for the property specified by the property key.
<b>DELETE</b> <code>apiv1/triggers/&lt;key&gt;</code>	This API deletes the trigger specified by the key.
<b>POST</b> <code>apiv1/triggers/&lt;trigger_key&gt;/trigger_apps</code>	This API creates an application action for the trigger key.
<b>PUT</b> <code>apiv1/trigger_apps/&lt;trigger_app_id&gt;</code>	This API updates an existing trigger application for a trigger with the <code>trigger_key</code> .
<b>GET</b> <code>apiv1/triggers/&lt;trigger_key&gt;/trigger_apps</code>	This API lists all of the applications of the current user for a trigger with the <code>trigger_key</code> .
<b>DELETE</b> <code>/trigger_apps/&lt;triggerApp_key&gt;</code>	This API deletes the application specified by the application key.

The remainder of Section 10 provides information on Push Notifications and the Device Service APIs that are not in the Ayla API Browser.

## 10.1 Push Notification Notes

---

When a user creates a push notification trigger\_app using the REST API, it is possible that the user does not know the registration ID, which is a long string. If this is the case, the param1 has been set to “nil” and must be updated using the PUT method to update a trigger app. Otherwise, param1 contains the registration ID. The use cases for this are outlined in this section.

### Use Case 1 (the user knows the registration ID):

- The user creates a new trigger app with param1 set to “nil.”
- When the mobile is registered in Google or Apple, the user updates the trigger app with the registration ID, using the PUT method.
- When the mobile is registered in Apple, the param2 must be filled with the key assigned to the PEM when uploaded to Ayla User Service.
- If the user wants to display a custom message, the user fills in the param3 with a custom message string.

### Use Case 2 (the user doesn't know the registration id):

- The user creates a new trigger app with param1 that is set to the registration ID provided by Google or Apple.
- When the mobile is registered in Apple, the param2 must be filled with the key assigned to the PEM when uploaded to the Ayla User Service.
- If the user wants to display a custom message, the user fills in the param3 with a custom message string.

## 10.2 GET /apiv1/dsns/{dsn}/properties/{property\_name}/triggers/all

This API is used to retrieve device-level triggers using the device serial number (DSN) and property name. Table 22 provides the mandatory input and the output for this API.

**Table 22:** Mandatory Input and the Output for Obtaining Device Level Triggers

Mandatory Input:		
Parameter	Description	Type
dsn	The device serial number (DSN) of the device as returned when retrieving devices for a user.	string
property_name	The name assigned to the property (should match the property name of the device host application)	string
Output:		
An array of triggers.		

### JSON Example

```
$ curl -s https://ads-dev.aylanetworks.com/apiv1/triggers/all/by_user.json
```

### Response

```
[
  {
    "trigger": {
      "base_type": "boolean",
      "compare_type": "==",
      "device_nickname": "carrier 73",
      "period": "always",
      "property_nickname": "Blue_button",
      "trigger_type": "compare_absolute",
      "triggered_at": "2012-07-12T16:56:07Z",
      "value": "1",
      "key": 404,
      "property_key": 401
    }
  }
]
```

## 10.3 POST apiv1/dsns/:dsn/properties/:property\_name/triggers

This API is used to create a trigger for the provided device serial number (DSN) and property name. Table 23 provides the mandatory input and the output for this API.

**Table 23:** Mandatory/Optional Input and the Output for Obtaining Device Level Triggers

Mandatory Input:		
Parameter	Description	Type
trigger-type	Can be one of the following: <ul style="list-style-type: none"> <li>always – fires a trigger on every datapoint</li> <li>compare_absolute – compares to a value specified in the trigger</li> <li>on_change – fires a trigger for any change to the property value</li> </ul>	string
value	Value used in the compare-absolute trigger type.	string
active	Set to "true" by default, and is "false" when the trigger is deactivated.	
trigger_apps	A list containing trigger apps with the format specified in the create trigger app setting	string
compare-type	This is specified only for compare-absolute or compare-service trigger types. Can be one of the following strings: <ul style="list-style-type: none"> <li>== equals</li> <li>&gt; greater than</li> <li>&lt; less than</li> <li>&gt;= greater than or equal</li> <li>&lt;= less than or equal</li> </ul>	string
Optional Input:		
device nickname	A user-friendly name for the device. If this does not exist, the product name for the device is used.	string
property nickname	A user-friendly name for the property. If this does not exist, the property name for the device is used.	string
asset	Set to "true" by default for user's assets, and is "false" when the asset is a device-level trigger, not a user's asset.	boolean
Output:		
Upon success, returns the trigger object created; otherwise, returns errors.		

## JSON Example

```
$ curl -X POST -H"Authorization:
auth_token d78a9a3d35ad45c8ad1c703530746d65" -H "Content-Type:
application/json" -d '{"trigger":{"trigger_type":"compare_absolute",
"compare_type":">=", "value":60, "trigger_apps": [{"name":"custom",
"username":"test", "param1":"to@example.org"}] } }' http://ads-dev.aylanetworks.com/apiv1/dsns/<dsn>/properties/<property\_name>/triggers.json
```

## Response

```
{
  "trigger": {
    "device_nickname": "Device",
    "property_nickname": "a",
    "compare_type": ">=",
    "period": "always",
    "trigger_type": "compare_absolute",
    "base_type": "integer",
    "value": 60,
    "triggered_at": null,
    "user_id": 1000003,
    "user_uuid": "e07d6aa2-79b1-11e5-94b1-0ea626ed4846",
    "key": 1000028,
    "property_key": 1056,
    "property_name": "prop_name",
    "asset": true,
    "active": true,
    "trigger_apps": [
      {
        "name": "custom",
        "nickname": null,
        "key": 121,
        "trigger_key": 1000028,
        "username": "test",
        "param1": "to@example.org",
        "param2": null,
        "param3": null,
        "param4": null,
        "param5": null
      }
    ]
  }
}
```

## Response Status Codes

- 404 – When the DSN or property\_name is invalid.
- 422 – For validation errors. The errors are in input body.
- 403 – Forbidden. When user does not have access to the DSN and property name.

## 11. Device Notifications

Device-specific events, such as connection lost or connection restored can trigger notifications to users. This is useful when users want to know when a device has lost connectivity.

**NOTE** You can use these notifications per device basis, or you can set them using the templates from the UI.

The APIs for device notifications (described below) are available in the [Ayla API Browser](#).

<b>POST</b> <code>apiv1/devices/&lt;device_key&gt;/notifications</code>	This API creates a notification for the device specified by the device key.
<b>GET</b> <code>apiv1/devices/&lt;device_key&gt;/notifications</code>	This API lists all of the current user's notifications for device with the device key.
<b>PUT</b> <code>apiv1/notifications/&lt;id&gt;</code>	This API updates a notification by the ID.
<b>DELETE</b> <code>apiv1/triggers/&lt;key&gt;</code>	This API deletes the trigger specified by the key.
<b>GET</b> <code>/notifications/:notification_id/notification_apps</code>	This API obtains a list of notification applications associated with a specific notification.
<b>GET</b> <code>/notifications/:notification_id/notification_apps/:id</code>	This API returns a specific notification application.
<b>POST</b> <code>/notifications/:notification_id/notification_apps</code>	This API creates a new notification application associated with the notification. This API supports custom email templates. Contact your Ayla Networks representative for custom email templates. To send customized emails, add the custom email parameters to the <code>notification_app_parameter</code> structure.
<b>PUT</b> <code>/notifications/:notification_id/notification_apps/:id</code>	This API updates a specific notification application.
<b>DELETE</b> <code>/notifications/:notification_id/notification_apps/:id</code>	This API deletes a specific notification application.

## 12 Share APIs

Share API is a resource-sharing application between registered users. By specifying a resource class and a unique resource identifier, such as, `Create`, `Read`, `Update` and `Delete` (CRUD), these Share APIs support sharing the resource. When the owner of a device shares a resource, the resource for the target user contains updated grant information, as follows:

- Currently, only devices may be shared.
- Only the owner to whom the device has been registered may share a device.
- A resource may be shared with one or more registered user.
- Share access controls the access rights: read and write are supported.
- Shares may include a start and end time-stamp.
- Sharing supports custom email templates for sharing notification on creation.
- A user cannot have more than one share for the same `resource_name` and `resource_id`.

The Share APIs (described below) are available in the [Ayla API Browser](#).

<b>GET</b> <code>/api/v1/users/shares/</code>	This API returns a list of owned shares that are not expired.
<b>GET</b> <code>/api/v1/users/shares/received</code>	This API returns a list of received shares that are not expired.
<b>GET</b> <code>/api/v1/users/shares/:id</code>	This API returns a share ID.
<b>POST</b> <code>/api/v1/users/shares/</code>	This API creates a new share or a list of new shares with the provided parameters. A user cannot have more than one share for the same <code>resource_name</code> and <code>resource_id</code> .
<b>PUT</b> <code>/api/v1/users/shares/:id</code>	This API updates a share with the provided parameters.
<b>DELETE</b> <code>/api/v1/users/shares/:id</code>	This API deletes a share.
<b>DELETE</b> <code>/api/v1/users/shares?resource_name=&lt;resource_name&gt;&amp;resource_id=&lt;resource_id&gt;</code>	This API deletes shares associated with the resource name and the resource ID.



## 13 Linked User Accounts

Users can link two or more user accounts across different customers. A prerequisite is that the customers should have a business relationship as captured in the Ayla system by an Ayla Admin. For each customer, the user must authenticate using the credentials for that customer. For users who want to link accounts, they need to initiate the linked accounts process. Users also have the ability to select the customers to whom they want to link or de-link accounts.

The benefits of linking the user account are as follows:

- Control devices from one single app
- Manage all accounts, triggers, notifications, and schedules from a single app
- Control all of the devices from the different customers

### 13.1 POST /api/v1/users/link

This API creates a link to a user account. The user should be signed in with the super app account. Table 24 provides the mandatory input required for this API.

**Table 24:** Mandatory Input for Creating a Link to a User Account

Mandatory Input:		
Parameter	Description	Type
email	The email address of the account to be linked with	string
password	The password for the user account.	string
origin_oem_id	The origin_oem_id of the account.	integer

#### JSON Example

```
curl -X POST -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" -d'{"user": {"email": "abc@abc.com", "password": "password", "origin_oem_id": 4}}' https://<user service url>/api/v1/users/link.json
```

#### Response Status Code

201 – Success, created a link to a super app

401 – Not authorized

404 – Failure, a user with provided credentials was not found

422 – Failure, invalid user password or a user already linked or users origin\_oem\_id cannot link to this client

## 13.2 DELETE /api/v1/user//link/:user\_id.

This API deletes the link to a user account. The user should be signed in with the super app account. Table 25 provides the mandatory input required for this API.

**Table 25:** Mandatory Input for Deleting the Link to a User Account

Mandatory Input:		
Parameter	Description	Type
user_id	The user_id of the account with which to delete the link.	integer

### JSON Example

```
curl -X DELETE -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" https://<user service url>/api/v1/users/link/5.json
```

### Response

- 201 – Success, created a link to a super app
- 401 – Not authorized
- 404 – Failure, invalid user id
- 422 – Failure, no link found with the provided user id

## 14 Time Zones

The most current APIs for time zones (described below) are available in the [Ayla API Browser](#).

**GET** `apiv1/devices/<device_id>/time_zones`

This API obtains the time zone of a device with the device ID. If the time zone is not set, the API returns a default time zone with nil key and the utc offset.

**PUT** `apiv1/devices/<device_id>/time_zones`

This API updates the time\_zone of a device with the device ID when the device has time\_zone set. Or, when the device already has time\_zone set, this API creates the time\_zone of a device with the device ID.

### 14.1 List of the Supported Time Zones

Table 26 is an alphabetized list of the supported time zones.

**Table 26:** Supported Time Zones

Time Zone/Country		
Africa/Abidjan	Africa/Ceuta	Africa/Libreville
Africa/Accra	Africa/Conakry	Africa/Lome
Africa/Addis_Ababa	Africa/Dakar	Africa/Luanda
Africa/Algiers	Africa/Dar_es_Salaam	Africa/Lubumbashi
Africa/Asmara	Africa/Djibouti	Africa/Lusaka
Africa/Asmera	Africa/Douala	Africa/Malabo
Africa/Bamako	Africa/El_Aaiun	Africa/Maputo
Africa/Bangui	Africa/Freetown	Africa/Maseru
Africa/Banjul	Africa/Gaborone	Africa/Mbabane
Africa/Bissau	Africa/Harare	Africa/Mogadishu
Africa/Blantyre	Africa/Johannesburg	Africa/Monrovia
Africa/Brazzaville	Africa/Juba	Africa/Nairobi Africa/Ndjamena
Africa/Bujumbura	Africa/Kampala	Africa/Niamey
Africa/Cairo	Africa/Khartoum	Africa/Nouakchott
Africa/Casablanca	Africa/Kigali	Africa/Ouagadougou
	Africa/Kinshasa	
	Africa/Lagos	

Time Zone/Country		
Africa/Porto-Novo	America/Bahia_Banderas	America/Eirunepe
Africa/Sao_Tome	America/Barbados	America/El_Salvador
Africa/Timbuktu	America/Belem	America/Ensenada
Africa/Tripoli	America/Belize	America/Fort_Wayne
Africa/Tunis	America/Blanc-Sablon	America/Fortaleza
Africa/Windhoek	America/Boa_Vista	America/Glace_Bay
America/Adak	America/Bogota	America/Godthab
America/Anchorage	America/Boise	America/Goose_Bay
America/Anguilla	America/Buenos_Aires	America/Grand_Turk
America/Antigua	America/Cambridge_Bay	America/Grenada
America/Araguaina	America/Campo_Grande	America/Guadeloupe
America/Argentina/Buenos_Aires	America/Cancun	America/Guatemala
America/Argentina/Catamarca	America/Caracas	America/Guayaquil
America/Argentina/ComodRivadavia	America/Catamarca	America/Guyana
America/Argentina/Cordoba	America/Cayenne	America/Halifax
America/Argentina/Jujuy	America/Cayman	America/Havana
America/Argentina/La_Rioja	America/Chicago	America/Hermosillo
America/Argentina/Mendoza	America/Chihuahua	America/Indiana/ Indianapolis
America/Argentina/Rio_Gallegos	America/Coral_Harbour	America/Indiana/Knox
America/Argentina/Salta	America/CordobaAmerica/	America/Indiana/Marengo
America/Argentina/San_Juan	Costa_Rica	America/Indiana/Petersburg
America/Argentina/San_Luis	America/Creston	America/Indiana/Tell_City
America/Argentina/Tucuman	America/Cuiaba	America/Indiana/Vevay
America/Argentina/Ushuaia	America/Curacao	America/Indiana/Vincennes
America/Aruba	America/Danmarkshavn	America/Indiana/Winamac
America/Asuncion	America/Dawson	America/Indianapolis
America/Atikokan	America/Dawson_Creek	America/Inuvik
America/Atka	America/Denver	America/Iqaluit
America/Bahia	America/Detroit	America/Jamaica
	America/Dominica	America/Jujuy
	America/Edmonton	

Time Zone/Country		
America/Juneau	America/Noronha	America/St_Vincent
America/Kentucky/Louisville	America/North_Dakota/Beulah	America/Swift_Current
America/Kentucky/Monticello	America/North_Dakota/Center	America/Tegucigalpa
America/Knox_IN	America/North_Dakota/New_Salem	America/Thule
America/Kralendijk	America/Ojinaga	America/Thunder_Bay
America/La_Paz	America/Panama	America/Tijuana
America/Lima	America/Pangnirtung	America/Toronto
America/Los_Angeles	America/Paramaribo	America/Tortola
America/Louisville	America/Phoenix	America/Vancouver
America/Lower_Princes	America/Port-au-Prince	America/Virgin
America/Maceio	America/Port_of_Spain	America/Whitehorse
America/Managua	America/Porto_Acre	America/Winnipeg
America/Manaus	America/Porto_Velho	America/Yakutat
America/Marigot	America/Puerto_Rico	America/Yellowknife
America/Martinique	America/Rainy_River	Antarctica/Casey
America/Matamoros	America/Rankin_Inlet	Antarctica/Davis
America/Mazatlan	America/Recife	Antarctica/DumontDUrville
America/Mendoza	America/Regina	Antarctica/Macquarie
America/Menominee	America/Resolute	Antarctica/Mawson
America/Merida	America/Santa_Isabel	Antarctica/McMurdo
America/Metlakatla	America/Santarem	Antarctica/Palmer
America/Mexico_City	America/Santiago	Antarctica/Rothera
America/Miquelon	America/Santo_Domingo	Antarctica/South_Pole
America/Moncton	America/Sao_Paulo	Antarctica/Syowa
America/Monterrey	America/Scoresbysund	Antarctica/Vostok
America/Montevideo	America/Shiprock	Arctic/Longyearbyen
America/Montreal	America/Sitka	Asia/Aden
America/Montserrat	America/St_Barthlemy	Asia/Almaty
America/Nassau	America/St_Johns	Asia/Amman
America/New_York	America/St_Kitts	Asia/Anadyr
America/Nipigon	America/St_Lucia	Asia/Aqtau
America/Nome	America/St_Thomas	

Time Zone/Country		
Asia/Aqtobe	Asia/Jayapura	Asia/Qatar
Asia/Ashgabat	Asia/Jerusalem	Asia/Qyzylorda
Asia/Ashkhabad	Asia/Kabul	Asia/Rangoon
Asia/Baghdad	Asia/Kamchatka	Asia/Riyadh
Asia/Bahrain	Asia/Karachi	Asia/Riyadh87
Asia/Baku	Asia/Kashgar	Asia/Riyadh88
Asia/Bangkok	Asia/Kathmandu	Asia/Riyadh89
Asia/Beirut	Asia/Katmandu	Asia/Saigon
Asia/Bishkek	Asia/Khandyga	Asia/Sakhalin
Asia/Brunei	Asia/Kolkata	Asia/Samarkand
Asia/Calcutta	Asia/Krasnoyarsk	Asia/Seoul
Asia/Choibalsan	Asia/Kuala_Lumpur	Asia/Shanghai
Asia/Chongqing	Asia/Kuching	Asia/Singapore
Asia/Chungking	Asia/Kuwait	Asia/Taipei
Asia/Colombo	Asia/Macao	Asia/Tashkent
Asia/Dacca	Asia/Macau	Asia/Tbilisi
Asia/Damascus	Asia/Magadan	Asia/Tehran
Asia/Dhaka	Asia/Makassar	Asia/Tel_Aviv
Asia/Dili	Asia/Manila	Asia/Thimbu
Asia/Dubai	Asia/Muscat	Asia/Thimphu
Asia/Dushanbe	Asia/Nicosia	Asia/Tokyo
Asia/Gaza	Asia/Novokuznetsk	Asia/Ujung_Pandang
Asia/Harbin	Asia/Novosibirsk	Asia/Ulaanbaatar
Asia/Hebron	Asia/Omsk	Asia/Ulan_Bator
Asia/Ho_Chi_Minh	Asia/Oral	Asia/Urumqi
Asia/Hong_Kong	America/Rio_Branco	Asia/Ust-Nera
Asia/Hovd	America/Rosario	Asia/Vientiane
Asia/Irkutsk	Asia/Phnom_Penh	Asia/Vladivostok
Asia/Istanbul	Asia/Pontianak	Asia/Yakutsk
Asia/Jakarta	Asia/Pyongyang	Asia/Yekaterinburg

Time Zone/Country		
Asia/Yerevan	Australia/West	Europe/Bratislava
Atlantic/Azores	Australia/Yancowinna	Europe/Brussels
Atlantic/Bermuda	Brazil/Acre	Europe/Bucharest
Atlantic/Canary	Brazil/DeNoronha	Europe/Budapest
Atlantic/Cape_Verde	Brazil/East	Europe/Busingen
Atlantic/Faeroe	Brazil/West	Europe/Chisinau
Atlantic/Faroe	CET	Europe/Copenhagen
Atlantic/Jan_Mayen	CST6CDT	Europe/Dublin
Atlantic/Madeira	Canada/Atlantic	Europe/Gibraltar
Atlantic/Reykjavik	Canada/Central	Europe/Guernsey
Atlantic/South_Georgia	Canada/East-Saskatchewan	Europe/Helsinki
Atlantic/St_Helena	Canada/Eastern	Europe/Isle_of_Man
Atlantic/Stanley	Canada/Mountain	Europe/Istanbul
Australia/ACT	Canada/Newfoundland	Europe/Jersey
Australia/Adelaide	Canada/Pacific	Europe/Kaliningrad
Australia/Brisbane	Canada/Saskatchewan	Europe/Kiev
Australia/Broken_Hill	Canada/Yukon	Europe/Lisbon
Australia/Canberra	Chile/Continental	Europe/Ljubljana
Australia/Currie	Chile/EasterIsland	Europe/London
Australia/Darwin	Cuba	Europe/Luxembourg
Australia/Eucla	EET	Europe/Madrid
Australia/Hobart	EST	Europe/Malta
Australia/LHI	EST5EDT	Europe/Mariehamn
Australia/Lindeman	Egypt	Europe/Minsk
Australia/Lord_Howe	Eire	Europe/Monaco
Australia/Melbourne	Etc/GMT	Europe/Moscow
Australia/NSW	Etc/GMT+0	Europe/Nicosia
Australia/North	Etc/GMT+1	Europe/Oslo
Australia/Perth	Etc/GMT+10	Europe/Paris
Australia/Queensland	Etc/GMT+11	Europe/Podgorica
Australia/South	Etc/GMT+12	Europe/Prague
Australia/Sydney	Etc/GMT+2	Europe/Riga
Australia/Tasmania	Etc/GMT+3	Europe/Rome

Time Zone/Country		
Europe/Samara	Indian/Cocos	Pacific/Efate
Europe/San_Marino	Indian/Comoro	Pacific/Enderbury
Europe/Sarajevo	Indian/Kerguelen	Pacific/Fakaofu
Europe/Simferopol	Indian/Mahe	Pacific/Fiji
Europe/Skopje	Indian/Maldives	Pacific/Funafuti
Europe/Sofia	Indian/Mauritius	Pacific/Galapagos
Europe/Stockholm	Indian/Mayotte	Pacific/Gambier
Europe/Tallinn	Indian/Reunion	Pacific/Guadalcanal
Europe/Tirane	Iran	Pacific/Guam
Europe/Tiraspol	Israel	Pacific/Honolulu
Europe/Uzhgorod	Jamaica	Pacific/Johnston
Europe/Vaduz	Japan	Pacific/Kiritimati
Europe/Vatican	Kwajalein	Pacific/Kosrae
Europe/Vienna	Libya	Pacific/Kwajalein
Europe/Vilnius	MET	Pacific/Majuro
Europe/Volgograd	MST	Pacific/Marquesas
Europe/Warsaw	MST7MDT	Pacific/Midway
Europe/Zagreb	Mexico/BajaNorte	Pacific/Nauru
Europe/Zaporozhye	Mexico/BajaSur	Pacific/Niue
Europe/Zurich	Mexico/General	Pacific/Norfolk
GB	Mideast/Riyadh87	Pacific/Noumea
GB-Eire	Mideast/Riyadh88	Pacific/Pago_Pago
GMT	Mideast/Riyadh89	Pacific/Palau
GMT+0	NZ	Pacific/Pitcairn
GMT-0	NZ-CHAT	Pacific/Pohnpei
GMT0	Navajo	Pacific/Ponape
Greenwich	PRC	Pacific/Port_Moresby
HST	PST8PDT	Pacific/Rarotonga
Hongkong	Pacific/Apia	Pacific/Saipan
Iceland	Pacific/Auckland	Pacific/Samoa
Indian/Antananarivo	Pacific/Chatham	Pacific/Tahiti
Indian/Chagos	Pacific/Chuuk	Pacific/Tarawa
Indian/Christmas	Pacific/Easter	Pacific/Tongatapu



Time Zone/Country		
Pacific/Truk	UCT	US/Mountain
Pacific/Wake	US/Alaska	US/Pacific
Pacific/Wallis	US/Aleutian	US/Pacific-New
Pacific/Yap	US/Arizona	US/Samoa
Poland	US/Central	UTC
Portugal	US/East-Indiana	Universal
ROC	US/Eastern	W-SU
ROK	US/Hawaii	WET
Singapore	US/Indiana-Starke	Zulu
Turkey	US/Michigan	

## 15. Address

The Address APIs manage the device's physical address. A device only has one address. This section provides APIs for the device address. The Address APIs (described below) are available in the [Ayla API Browser](#).

**GET /apiv1/devices/:id/addr** This API returns an address of the device.

**PUT /apiv1/devices/:id/addr** This API updates an address with the provided parameters.

## 16 User and Device Service URLs

Dashboard URLs for the different regions are as follows:

US Developer Service: <https://dashboard-dev.aylanetworks.com>

China Developer Service: <https://dashboard-dev.ayla.com.cn>

### Device Service

The Regional Device Service URLs are as follows:

US Developer Service: <https://ads-dev.aylanetworks.com>

US Field Service: <https://ads-field.aylanetworks.com>

EU Field Service: <https://ads-field-eu.aylanetworks.com>

CN Developer Service: <https://ads-dev.ayla.com.cn>

CN Field Service: <https://ads-field.ayla.com.cn>

### User Service

The Regional User Service URLs are as follows:

USA Developer Service: <https://user-dev.aylanetworks.com/>

US Field Service: <https://user-field.aylanetworks.com/>

EU Field Service: <https://user-field-eu.aylanetworks.com/>

CN Developer Service: <https://user-dev.ayla.com.cn/>

CN Field Service: <https://user-field.ayla.com.cn/>



680 N. McCarthy Blvd., Suite 100

Milpitas, CA 95054

Phone: +1 408 830 9844

Fax: +1 408 716 2621