

Ayla Service API Specification



Version 5

Date Released March 23, 2016

Document Part Number: AY006USA3-5



Table of Contents

1 INTRODUCTION	7
1.1 Audience and Access Requirements	7
1.2 Customer Support	7
1.3 Executing an API	7
2 USERS	8
2.1 Sign Up	8
2.2 Sign In	12
2.3 Provider Auth	14
2.4 Getting New Access Token from Refresh Token	16
2.5 Resend Confirmation Token	17
2.6 User Email Confirmation	19
2.7 Reset Password Instructions	19
2.7.1 Edit User Password	21
2.8 Reset Password with Token	22
2.9 Edit User Attributes	23
2.10 Update User Email	24
2.11 Delete User	24
2.12 Sign out user	25
2.13 Get User Attributes	25
2.14 Get User Attributes with associated Roles	26
3 USER METADATA	27
3.1 GET /api/v1/users/data[.format]	27
3.2 GET /api/v1/users/data/:key[.format]	28
3.3 GET /api/v1/users/:user_id/data[.format]	29
3.4 POST /api/v1/users/data[.format]	30
3.5 PUT /api/v1/users/data/:key[.format]	30
3.6 DELETE /api/v1/users/data/:key[.format]	31
4 DEVICES	32
4.1 GET apiv1/devices[.format]	32
4.2 GET apiv1/devices/<key>[.format]	35
4.3 GET apiv1/dsns/:dsn/nodes[.format]	41
4.4 GET apiv1/dsns/<dsn>[.format]	46
4.5 POST apiv1/devices/:device_id/registration_window[.format]	47
4.6 PUT apiv1/devices/:device_id/identify[.format]	48
4.7 GET apiv1/devices/register.xml using lp, dsn, regtype, time	49
4.8 GET apiv1/devices/find_by_user_email?email=<email_address>	51

4.9	POST apiv1/devices.xml	52
4.10	PUT apiv1/devices/<key>[.format]	54
4.11	PUT apiv1/dsns/<dsn>[.format]	55
4.12	PUT apiv1/devices/<device_dsn>/transfer[.format]	56
4.13	PUT apiv1/devices/<device_id>/locations[.format]	56
4.14	POST apiv1/devices/<device_id>/locations[.format]	57
4.15	PUT apiv1/devices/<device_id>/cmds/factory_reset[.format]	58
4.16	DELETE apiv1/devices/<key>[.format]	58
5	DEVICE METADATA.....	60
5.1	GET /apiv1/dsns/:dsn/data[.format]	60
5.2	GET /apiv1/dsns/:dsn/data/:key[.format]	61
5.3	POST /apiv1/dsns/:dsn/data[.format]	61
5.4	PUT /apiv1/dsns/:dsn/data/:key[.format]	62
5.5	DELETE /apiv1/dsns/:dsn/data/:key[.format]	63
6	PROPERTIES	64
6.1	GET apiv1/devices/<device_key>/properties.xml.....	64
6.2	GET apiv1/dsns/<device_dsn>/properties [.format]	67
6.3	GET /apiv1/properties/<key>[.format]	70
6.4	GET /properties/<property_key>/datapoints[.format]	71
6.4.1	POST apiv1/properties/<property_key>/datapoints[.format]	87
6.5	POST apiv1/dsns/:dsn/properties/:property_name/datapoints[.format]	90
6.6	GET /apiv1/properties/:property_id/trigger_apps.[format]	90
6.7	FILE PROPERTIES.....	91
6.7.1	Creating a Datapoint.....	92
6.7.2	Uploading Data to the Datapoint.....	93
6.7.3	Marking a Datapoint as Complete	94
6.7.4	Marking a Datapoint as Discarded.....	94
6.7.5	Downloading Data.....	95
6.7.6	Marking a Datapoint as Fetche	96
7	GROUPS	97
7.1	GET /apiv1/groups	97
7.2	POST /apiv1/groups	97
7.3	GET /apiv1/groups/:group_id	98
7.4	PUT /apiv1/groups/:group_id	99
7.5	DELETE /apiv1/groups/:group_id	100
7.6	POST /apiv1/groups/:group_id/devices	101
7.7	DELETE /apiv1/groups/:group_id/devices/:device_id	102
7.8	POST /apiv1/groups/:group_id/batched_datapoints.....	103
7.9	POST /apiv1/groups/:group_id/datapoints.....	105
7.10	GET /apiv1/groups/:group_id/datapoints	106

SCHEDULES	108
7.11 POST apiv1/devices/<device_id>/schedules[.format]	108
7.12 GET apiv1/devices/<device_id>/schedules[.format]	114
7.13 GET apiv1/schedules/<id>[.format]	118
7.14 GET apiv1/devices/<device_id>/schedules/find_by_name.xml?name=<name>	124
7.15 PUT apiv1/devices/<device_id>/schedules/<schedule_id>[.format]	129
7.16 PUT /apiv1/schedules/<schedule_id>/clear[.format]	136
7.17 GET apiv1/devices/<device_id>/schedules/base64[.format]	140
7.18 GET apiv1/schedules/all/by_user.xml	141
7.19 SCHEDULE ACTION	145
7.19.1 GET apiv1/schedules/<schedule_id>/schedule_actions[.format]	145
7.19.2 GET apiv1/schedule_actions/<id>[.format]	146
7.19.3 GET Schedules by Name	148
7.19.4 POST apiv1/schedules/<schedule_id>/schedule_actions[.format]	149
7.19.5 PUT apiv1/schedule_actions/<id>[.format]	151
7.19.6 DELETE apiv1/schedule_actions/<id>[.format]	152
8 TRIGGERS	153
8.1 POST apiv1/properties/<property_key>/triggers[.format]	153
8.2 PUT apiv1/triggers/<trigger_id>[.format]	155
8.3 GET apiv1/properties/<property_key>/triggers[.format]	159
8.4 GET apiv1/properties/<property_key>/triggers/all[.format]	160
8.5 DELETE apiv1/triggers/<key>[.format]	160
8.6 GET apiv1/triggers/all/by_user.xml	160
8.7 TRIGGER APP	161
8.7.1 POST apiv1/triggers/<trigger_key>/trigger_apps[.format]	162
8.7.2 Data Forwarding Notes	171
8.7.3 Push Notification Notes	172
8.7.4 PUT apiv1/trigger_apps/<trigger_app_id>[.format]	172
8.7.5 GET apiv1/triggers/<trigger_key>/trigger_apps[.format]	173
8.7.6 GET apiv1/triggers/<trigger_key>/trigger_apps/all[.format]	175
8.7.7 DELETE /trigger_apps/<triggerApp_key>[.format]	175
9 DEVICE NOTIFICATIONS	176
9.1 POST apiv1/devices/<device_key>/notifications[.format]	176
9.2 GET apiv1/devices/<device_key>/notifications[.format]	177
9.3 GET apiv1/devices/<device_id>/notifications/all[.format]	177
9.4 PUT apiv1/notifications/<id>[.format]	177
9.5 DELETE apiv1/notifications/<key>[.format]	178
9.5.1 Device Notification Notes	179
9.6 NOTIFICATION APPLICATIONS	180
9.6.1 GET /notifications/:notification_id/notification_apps	180

9.6.2	GET /notifications/:notification_id/notification_apps/:id.....	180
9.6.3	POST /notifications/:notification_id/notification_apps	181
9.6.4	PUT /notifications/:notification_id/notification_apps/:id.....	183
9.6.5	DELETE /notifications/:notification_id/notification_apps/:id	185
10	SHARE APIs	186
10.1	GET /api/v1/users/shares/	186
10.2	GET /api/v1/users/shares/received	188
10.3	GET /api/v1/users/shares/:id.....	189
10.4	POST /api/v1/users/shares/	190
10.5	PUT /api/v1/users/shares/:id.....	196
10.6	DELETE /api/v1/users/shares/:id	197
10.7	DELETE with resource_name & resource_id	197
11	LINKED USER ACCOUNTS	198
11.1	POST /api/v1/users/link[.format]	198
11.2	DELETE /api/v1/user//link/:user_id[:format].....	199
12	TIMEZONES	200
12.1	GET apiv1/devices/<device_id>/time_zones[.format]	200
12.2	POST apiv1/devices/<device_id>/time_zones[.format]	202
12.3	PUT apiv1/devices/<device_id>/time_zones[.format]	203
12.4	GET apiv1/time_zones?tz_id=.....	204
12.5	List of all supported Timezones	205
13	ADDRESS	219
13.1	GET /apiv1/devices/:id/addr[.format]	219
13.2	PUT /apiv1/devices/:id/addr[.format]	220
13.3	DELETE /apiv1/devices/:id/addr[.format]	221
14	Admin APIs	222
14.1	GET /apiv1/devices.json?paginated=true.....	222
15	USER AND DEVICE SERVICE URLS.....	224

Revision History

Revision	Date	Author	Change Description
2.2	Sept 14	LBoling	Removed Time Range Filter API
2.3	Oct 15	LBoling	Added Get User Attributes
3	Jan 8 2016	LBoling	Added Admin API section & API for getting list of all devices
4	Jan 27, 2016	LBoling	Added device location api
5	Mar 23, 2016	LBoling	Add DELETE Device Fixed provider auth input parameter error Added Time zones Added details to GET User Attributes with associated Roles Deleted DELETE apiv1/schedules/<id>[.format] & added note that schedules cannot currently be deleted Fixed User Service URL

1 INTRODUCTION

This document describes the APIs that are used to access the Ayla Cloud Service (ACS). The ACS uses a RESTful API to access its resources. The ACS is service-oriented entity with each service handling a single responsibility. The Ayla Device and User Services are the core services that manage devices and users.

1.1 Audience and Access Requirements

This document is intended for API developers. We assume that the users of this document are familiar with software using RESTful services over HTTPS. Users should be also familiar with the Ayla Cloud Service interaction with the device.

Requests to any of the Ayla Services require an access token. To obtain an access token, follow the instructions in the [Sign in](#) section of this document.

All requests to one of Ayla Services require an access token. To obtain an access token the user needs to login to the User Service. The Ayla User Service is at <https://user.aylanetworks.com>.

1.2 Customer Support

Technical support is available through the Ayla Support website at: <https://support.aylanetworks.com>, or via email at support@aylanetworks.com.

1.3 Executing an API

1. Use document search to help you find an API you want to execute.
2. Enter mandatory parameters and any optional parameters.
3. Add your authorization token.
4. Enter User or Service URL.
5. Execute the API.

HTTP Authorization Header has to be included in string “auth_token <token>” where <token> is obtained from user object returned on a successful sign in from a developer site.

Note: We provide XML and JSON examples and assume that you are familiar with the basic syntax rules of XML and JSON formats.

2 USERS

User Service API section covers APIs specific to the user account, such as, sign-up, sign_in, account confirmation, get access_token, refresh_token, update user information, delete user, reset passwords, sign-out user, get user attributes, and so on.

Several of these APIs initiate an email to the signing in user. You can upload a customized email template to the Ayla Service through the OEM dashboard. Refer to the document Customizing Ayla Emails for details. Once the template is uploaded, the unique template id can be passed as a parameter to the API calls.

- The first login of the user is done via the sign_in API call.
- If the user is logging in using the email and password, this method returns the access token and refresh token, otherwise it returns the URL to retrieve the auth token.
- If using an OAuth provider, send a parameter indicating if the logging in is "google_auth" or "facebook_auth".
- The default authentication method is an email and password.
- The user must provide the email and password if the authentication method is email_and_password.
- The app_id and app_secret are used to generate the token.

HTTP Authorization Header has to be included in string "auth_token <token>" where <token> is obtained from user object returned on a successful sign in from a developer site.

2.1 Sign Up

Use this API to create an account on the User Service.

Input:

Mandatory:

email : Email address used to access the account

Type : String

Password : Valid password

Type: String

firstname: Account user first name

Type: String

lastname: Account user last name

Type: String

country: Country of residence

Type: String

city: City of Residence

Type: String

app_id: The Application Id for the model of the oem to which user will be associated.

Type: String

app_secret: The Application Secret of model of the oem to which user will be associated.

Type: String

Optional:

Street: Street name

Type: String

Zip: area zip code

Type : Integer

phone_country_code: Country phone code

phone_number: User phone number

Type: Integer

ayla_dev_kit_num: If ayla_dev_kit exists, enter the number for the device to be associated to the user.

email_template_id - custom email template to be used while sending the confirmation email.

Type: Integer

email_subject - custom subject string to be sent while sending confirmation email

Type: String

Template is uploaded via the dashboard and a unique id is associated with it at the time of uploading.

Validation:

Valid format for phone number: 9-digit number

Valid regexp for email: `/A[\w+\.]+@[a-z\d\-.]+\.[a-z]+\z/i`

Valid password: 6 characters long

XML Example with only mandatory attributes:

```
curl -X POST -H"Content-Type: application/xml"
-d"<user><email>testemail@testemail.com
</email><password>password</password><firstname>fn</firstname><las
tname>ln</lastname><country>acountry</country><application><app_id
>app-id</app_id><app_secret>app-
secret<app_secret></application></user>"https://<user service
url>/users.xml
```

Response:

```
<user>
```

```
<admin>:false</admin>,
<approved>>false</approved>
<country>a country</country>
<created_at>2014-11-11T07:41:28Z</created_at>
<email>testemail@testemail.com </email>
<firstname>fn</firstname>
<id>185</id>
<lastname>ln</lastname>
<oem_approved>>false</oem_approved>,
<updated_at>2014-11-11T07:41:28Z</updated_at>
<terms_accepted_at>2014-11-11T07:41:28Z</terms_accepted_at>
<terms_accepted>False</terms_accepted>
</user>
```

XML Example with all attributes:

```
curl -X POST -k -H"Content-Type: application/xml"
-d"<user><email>testemail@testemail.com
</email><password>testuser1</password><firstname>fn</firstname><la
stname>ln</lastname><company>testuser1co</company><city>a
city</city><street>a street</street><country>a
country</country><zip>XXXXX</zip><phone>4089991XXX</phone><ayla_de
v_kit_num>xxx</ayla_dev_kit_num><application><app_id>app_id</app_i
d><app_secret>app_secret</app_secret></application></user>"
https://<user service url>/users.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<user>
  <country>a country</country>
  <email>testuser1@testuser1.com</email>
  <firstname>fn</firstname>
  <lastname>ln</lastname>
</user>
```

JSON Example with only mandatory attributes:

```
curl -X POST -k -H"Content-Type: application/json"
-d"{ "user": { "email": "testemail@testemail.com", "password":
"aylauser3", "firstname": "fn", "lastname": "ln", "country": "a
country", "application": { "app_id": "app_id", "app_secret":
"app_secret"}}}"
https://<User Service URL>/users.json
```

Response:

```
{ "admin":false,"approved":false,"country":"a
country","created_at":"2014-11-
11T07:41:28Z","email":testemail@testemail.com","firstname":"fn","i
d":185,"lastname":"ln","oem_approved":false,"updated_at":"2014-11-
11T07:41:28Z","terms_accepted_at":"2014-11-
11T07:41:28Z","terms_accepted": False }
```

JSON Example with all attributes:

```
curl -X POST -k -H"Content-Type: application/json"
-d{"user":{"email":"testuser4@testuser4.com",
"password":"testuser4", "firstname":"fn", "lastname":"ln",
"company":"testuser4 co", "city":"a city", "street":"a street",
"country":"a country", "zip":"951XX", "phone":"4089991111",
"ayla_dev_kit_num":"xxx", "application":{"app_id":"app_id",
"app_secret":"app_secret"}}}
https://<user service url>/users.json
```

Response:

```
{"admin":false,"approved":false,"ayla_dev_kit_num":"XXX","city":"a
city","company":"testuser4 co","country":"a
country","created_at":"2012-11-
12T01:13:14Z","email":"testuser4@testuser4.com","firstname":"fn",
"id":186,"lastname":"ln","oem_approved":false,"phone":"XXXXXXXXXX",
"street":"a street","updated_at":"2015-11-
12T01:13:14Z","zip":XXXXX}
```

Curl JSON Example with all attributes and custom email attributes:

```
curl -X POST -k -H"Content-Type: application/json" -
d{"\ "user\ ":{\ "email\ ":\"example@aylanetworks.com\", \"password\ ":\"
testuser1\", \"firstname\ ":\"fn\", \"lastname\ ":\"ln\", \"company\ ":
\"testuser1co\", \"city\ ":\"acity\", \"street\ ":\"astreet\", \"countr
y\ ":\"acountry\", \"zip\ ":\"95134\", \"phone\ ":\"4081112222\", \"ayla_dev_kit
_num\ ":\"322\", \"application\ ":{\ "app_id\ ":\"app_id\", \"app_secret
\ ":\"app_secret\"}}}" https://<user service
url>/users.json?email\_template\_id=ayla\_confirmation\_template\_01&
email\_subject=Confirmation%20Instructions\&email\_body\_html=%
service
url>/users.json?email\_template\_id=ayla\_confirmation\_template\_01&
email\_subject=Confirmation%20Instructions\&email\_body\_html=%3Ctr%3E
%09%3Ctd%20style%3D%27color%3A%20%23153643%3B%20font-
family%3A%20Arial%2C%20sans-serif%3B%20font-
size%3A%2024px%3B%27%3E%09%09%3Ch3%3E%09%09%09%3Cp%3EWelcome%20%5B
```

```
%5Buser_name%5D%5D%21%20%3C%2Fp%3E%09%09%3C%2Fh3%3E%09%3C%2Ftd%3E%
3C%2Ftr%3E%3Ctr%3E%09%3Ctd%20style%3D%27padding%3A%200px%200%2030p
x%200%3B%20color%3A%20%23153643%3B%20font-
family%3A%20Arial%2C%20sans-serif%3B%20font-
size%3A%2016px%3B%20line-
height%3A%2020px%3B%27%3E%09%09%3Cp%3EThank%20you%20for%20signing%
20up%20with%20us%21%20%3C%2Fp%3E%09%09%3Cp%3EHere%20is%20your%20si
gnup%20confirmation%20token%3C%2Fp%3E%09%09%3Cp%3E%3Ch4%3E%5B%5Bus
er_confirmation_token%5D%5D%3C%2Fh4%3E%3C%2Fp%3E%09%09%3Cp%3EYou%2
0can%20confirm%20your%20account%20by%20copying%20this%20token%20in
to%20the%20mobile%20app%3C%2Fp%3E%09%09%3Cp%3EOR%20by%20opening%20
the%20mobile%20app%20using%20the%20button%20below%3C%2Fp%3E%09%3C%
2Ftd%3E%3C%2Ftr%3E
```

Output:

200 – Success

422 – Failures

401 – Unauthorized, app_id or app_secret are not valid

2.2 Sign In

A user may sign in using Google OAuth or Facebook OAuth. When a user chooses one of these methods, the server check if the email of that user already exists. If the user already exists, Ayla service associates the oauth token with this user; otherwise a new user gets created.

Input:**Mandatory:**

email : Email address used to access the account

Type : String

Password : Valid password

Type: String

Validation: 6 characters long

app_id: The Application Id for the model of the oem to which user will be associated.

Type: String

app_secret: The Application Secret of model of the oem to which user will be associated.

Type: String

XML Example:

```
$ curl -X POST -d "<user><email>
testemail@testemail.com</email><password>password</password><ap
plication><app_id>app_id</app_id><app_secret>app_secret</app_se
```

```
curl -X POST -d '<?xml version="1.0" encoding="UTF-8"?>
<authorization>
  <access-token>access-token</access-token>
  <refresh-token>refresh-token</refresh-token>
  <expires-in type="integer">xxxx</expires-in>
</authorization>' -H "Content-Type:application/xml"
https://<user service url>/users/sign_in.xml
```

return

```
<?xml version="1.0" encoding="UTF-8"?>
<authorization>
  <access-token>access-token</access-token>
  <refresh-token>refresh-token</refresh-token>
  <expires-in type="integer">xxxx</expires-in>
</authorization>
```

JSON Example:

```
$ curl -X POST -d '{"user": { "email":
"testemail@testemail.com", "password": "password",
"application": { "app_id": "app_id", "app_secret":
"app_secret"} } }' -H "Content-Type:application/json"
https://<user service url>/users/sign_in.json
```

return

```
{"access_token":"access_token","refresh_token":"refresh_token","expires_in":86400,"
role":"Ayla::Smith","role_tags":[],"id":258}
```

Google OAuth

Any user may login using Google OAuth. When a user chooses this login flow, the server verifies the email address if a user already exists or Ayla service creates a new user.

This API returns a redirect URL to be used to get the auth_token of the user. The redirect URL is a Google page that asks the user to grant the permission to view the basic information (email, name, and so on).

To continue with the login, the application **must** be redirected to the provider URL.

```
$ curl -X POST -d '{"user": { "auth_method": "google_provider",
"application": { "app_id": "app_id", "app_secret":
"app_secret"} } }' -H "Content-Type:application/json"
https://<user service url>/users/sign_in.json
```

return

```
{"url": "https://www.google.com/....."}
```

Facebook OAuth

Any user can login using the Facebook Oauth, when a user chooses this the server check if the email of that user already exists or Ayla service creates a new user.

This API returns a redirect URL to be used to get the auth_token of the user. The redirected URL is a Facebook login dialog that asks the user to grant the permission to view the basic information (email, name, and so on).

To continue with the login, the application must redirect to the provider URL.

```
$ curl -X POST -d '{"user': {'auth_method': 'facebook_provider',  
'application': { 'app_id': 'app_id', 'app_secret':  
'app_secret' } } }' -H "Content-Type:application/json"  
https://<user service url>/users/sign\_in.json
```

return

```
{"url": "https://www.facebook.com/....."}
```

2.3 Provider Auth

Use this API to complete the login of an oath provider.

When a user is logging in using an external provider, the first step is to get a redirect URL. This URL returns a code that is used to get the oauth token.

This API is exchanging the code for the oauth token and logs the user into the Ayla service.

The complete flow of login in as follows:

1. The user calls the login API sending the name of the external provider as a parameter ([example google auth](#), [example facebook auth](#)).
2. The user receives a code.
3. The user calls this API sending the code received.
4. The user is logged in (or gets the reason of why the login failed).

Input:

Mandatory:

code: The code received from the step 1.

Type: String
 app_id: The Application Id of the Ayla service.
 Type: String
 redirect_url: The previously used redirect_url
 Type: String
 provider: The provider that processes the oauth token (google_provider
 or facebook_provider)
 Type: String

The API returns the access token, refreshes the token, and the expiration time as in the standard login process. If any error occurs during the login, an error object is received.

XML Example (Google oauth)

```
$ curl -X POST -d
"code=4%2F1F2i_X7snYgFytEKuaBTsWISYSep.4txQt7a1SLQdshQV0ieZDAps
HwjoegI&app_id=app_id&redirect_url=app_secret/token&provider=go
ogle_provider" -H "Content-Type:application/xml" https://<user
service url>/users/provider\_auth.xml
```

return

```
<?xml version="1.0" encoding="UTF-8"?>
<authorization>
  <access-token>access-token</access-token>
  <refresh-token>refresh-token</refresh-token>
  <expires-in type="integer">xxxx</expires-in>
</authorization>
```

JSON Example (Google oauth)

```
$ curl -X POST -d
"code=4%2F1F2i_X7snYgFytEKuaBTsWISYSep.4txQt7a1SLQdshQV0ieZDAps
HwjoegI&app_id=app_id&redirect_url=app_secret/token&provider=go
gle_provider" -H "Content-Type:application/json" https://<User
Service Url>/users/provider\_auth.json
```

return

```
{"authorization":
{"access_token":"access_token","refresh_token":"refresh_token","expires_in":xxxx}}
```

XML Example (Facebook oauth)

```
$ curl -X POST -d
"code=4%2F1F2i_X7snYgFytEKuaBTsWISYSep.4txQt7a1SLQdshQV0ieZDAps
HwjoegI&app_id=app_id&redirect_url=app_secret/token&provider=fa
cebook_provider" -H "Content-Type:application/xml"
https://<User Service URL>/users/provider\_auth.xml
```

return

```
<?xml version="1.0" encoding="UTF-8"?>
<authorization>
  <access-token>access-token</access-token>
  <refresh-token>refresh-token</refresh-token>
  <expires-in type="integer">xxxx</expires-in>
</authorization>
```

JSON Example (Facebook oauth)

```
$ curl -X POST -d
"code=4%2F1F2i_X7snYgFytEKuaBTsWISYSep.4txQt7a1SLQdshQV0ieZDAps
HwjoegI&app_id=app_id&redirect_url=app_secret/token&provider=fa
cebook_provider" -H "Content-Type:application/json"
https://<User Service URL>/users/provider\_auth.json
```

return

```
{"authorization":
{"access_token":"access_token","refresh_token":"refresh_token","expires_in":xxxx}}
```

2.4 Getting New Access Token from Refresh Token

The refresh token is a long-lived token that can be used to renew a short-lived access token.

Note: The refresh token does not have an expiry. It gets invalidated if a user explicitly logs out. A new refresh token is issued when the user logs back in.

Input:

Mandatory:

```
refresh-token:Token received when a user signs in(along
with access token)
Type:String
```

XML Example:


```
curl -X POST -d "<user><refresh-token>refresh-token</refresh-token>
</user>" -H "Content-Type:application/xml" https://<User Service URL>/users/refresh\_token.xml
```

return

```
<authorization>
  <access-token>access-token</access-token>
  <refresh-token>refresh-token</refresh-token>
  <expires-in>300</expires-in>
</authorization>
```

JSON Example:

```
curl -X POST -d '{"user" : {"refresh_token" : "refresh-token"}}' -
H "Content-Type:application/json" https://<User Service URL>/users/refresh\_token.xml
```

Output:

```
{"authorization":
{"access_token":"access_token","refresh_token":"refresh_token","expires_in":"300"}}
```

Response:

- 200 – Success, new access token was successfully generated
- 401 – Unauthorized, refresh token does not match current user

2.5 Resend Confirmation Token

Resends confirmation instructions to an unconfirmed user.

Input:**Mandatory:**

email : Email id to which the confirmation tokens needs to be sent

Type: String

application:

app_id: The Application id of the OEM model to which the user is associated.(You can find it from dashboard)

Type: String

app_secret: Application secret of the OEM model to which the user is associated.(You can find it from dashboard)

Type : String

Optional:

email_template_id - custom email template to be used while re-sending the confirmation email. (Template is uploaded via the dashboard and a unique id is associated with it at the time of uploading.)

Type: String

email_subject - custom subject (string)

Type: String

email_body_html - custom html styling and text to be embedded in the confirmation email.(String)

Type: String

JSON Example with custom email attributes:

```
curl -k -v -H"Content-Type: application/json" -
d"{\"user\":{\"email\":\"email@yahoo.com\", \"application\":
{\"app_id\":\"app_id\", \"app_secret\":\"app_secret\" }}}\" https://<user
service url>
/users/confirmation.json?email_template_id=ayla_confirmation_templ
ate_01&email_subject=Confirmation%20Instructions&email_body_html
service url>
/users/confirmation.json?email_template_id=ayla_confirmation_templ
ate_01&email_subject=Confirmation%20Instructions&email_body_html
=%3Ctr%3E%09%3Ctd%20style%3D%27color%3A%20%23153643%3B%20font-
family%3A%20Arial%2C%20sans-serif%3B%20font-
size%3A%2024px%3B%27%3E%09%09%3Ch%3E%09%09%09%3Cp%3EWelcome%20%5B
%5Buser_name%5D%5D%21%20%3C%2Fp%3E%09%09%3C%2Fh%3E%09%3C%2Ftd%3E%
3C%2Ftr%3E%3Ctr%3E%09%3Ctd%20style%3D%27padding%3A%200px%20%2030p
x%20%3B%20color%3A%20%23153643%3B%20font-
family%3A%20Arial%2C%20sans-serif%3B%20font-
size%3A%2016px%3B%20line-
height%3A%2020px%3B%27%3E%09%09%3Cp%3EThank%20you%20for%20signing%
20up%20with%20us%21%20%3C%2Fp%3E%09%09%3Cp%3EHere%20is%20your%20si
gnup%20confirmation%20token%3C%2Fp%3E%09%09%3Cp%3E%3Ch%3E%5B%5Bus
er_confirmation_token%5D%5D%3C%2Fh%3E%3C%2Fp%3E%09%09%3Cp%3EYou%2
0can%20confirm%20your%20account%20by%20copying%20this%20token%20in
to%20the%20mobile%20app%3C%2Fp%3E%09%09%3Cp%3EOR%20by%20opening%20
the%20mobile%20app%20using%20the%20button%20below%3C%2Fp%3E%09%3C%
2Ftd%3E%3C%2Ftr%3E
```

XML Example:

```
curl -k -v -H"Content-Type: application/xml" -
d"<user><email>email@yahoo.com</email><application><app_id>app_id<
/app_id><app_secret>app_secret</app_secret></application></user>"
https://<User Service URL>/users/confirmation.xml
```

Output:

201 – Success
422 – Failure, an account was already confirmed
Other HTTP pre-defined error codes

2.6 User Email Confirmation

Confirms account ownership from the user's confirmation token as it was received in the email.

Input:

Mandatory:

Confirmation_token: Confirmation token that is received in the email.
Type: String

JSON Example:

```
curl -ki -X PUT -H"Content-Type: application/json" -d "{}"  
https://<User Service  
URL>/users/confirmation.json?confirmation\_token=<confirmation-  
token>
```

XML Example:

```
curl -ki -X PUT -H"Content-Type: application/xml" -d "{}"  
https://<User Service  
URL>/users/confirmation.xml?confirmation\_token=<confirmation-  
token>
```

Success:

```
{"admin":false,"approved":true,"ayla_dev_kit_num":"","city":"Sunnyvale","company":"","co  
untry":"United States","created_at":"2013-10-  
30T20:24:24Z","email":"something@gmail.com","firstname":"Someone","id":506,"lastnam  
e":"Sample","oem_approved":false,"phone":"4084084XXX","state":"California","street":"12  
3","terms_accepted":true,"terms_accepted_at":"2014-10-  
30T20:24:24Z","updated_at":"2014-10-30T20:27:14Z","zip":9XXXX}
```

Response:

201 – Success
422 – Failure, confirmation token is not valid
Other HTTP pre-defined error codes

2.7 Reset Password Instructions

Resets password for a confirmed and approved user.

Input:

Mandatory:

Email: Email id of the user for which the password needs to be reset

Type: String

app_id : The Application id of the OEM model to which the user is associated.(You can find it from dashboard)

Type: String

app_secret: Application secret of the OEM model to which the user is associated.(You can find it from dashboard)

Type: String

Optional:

email_template_id - custom email template to be used while sending the reset password email. (Template is uploaded via the dashboard and a unique id is associated with it at the time of uploading.)

Type: String

email_subject - custom subject string

Type: String

email_body_html - custom html styling and text to be embedded in the password reset email.

Type: String

JSON Example (with custom email parameters):

```
curl -k -v -H"Content-Type: application/json" -
d{"\user\":{\ "email\":"email@yahoo.com\","application":
{"app_id":"app_id", "app_secret":"app_secret" } } }"© 2016 Ayla Networks
```

```
size%3A%2016px%3B%20line-
height%3A%2020px%3B%27%3E%09%09%3Cp%3ESomeone%20has%20requested%20
a%20link%20to%20change%20your%20password.%3C%2Fp%3E%09%09%3Cp%3EHe
re%20is%20your%20password%20reset%20token%3C%2Fp%3E%09%09%3Cp%3E%3
Ch4%3E%5B%5Buser_password_reset_token%5D%5D%3C%2Fh4%3E%3C%2Fp%3E%0
9%09%3Cp%3EYou%20can%20can%20paste%20the%20below%20token%20in%20yo
ur%20mobile%20app%3C%2Fp%3E%09%09%3Cp%3EOR%2C%20you%20can%20open%2
0the%20mobile%20app%20link%20below%3C%2Fp%3E%09%3C%2Ftd%3E%3C%2Ftr
%3E%3Ctr%3E%09%3Ctd%20style%3D%27color%3A%20%23153643%3B%20font-
family%3A%20Arial%2C%20sans-serif%3B%20font-
size%3A%2016px%3B%20line-
height%3A%2020px%3B%27%3E%09%09%3Chr%2F%3E%09%09%3Cp%3EIf%20you%20
didn%27t%20request%20this%2C%20please%20ignore%20this%20email.%3C%
2Fp%3E%09%09%3Cp%3EYour%20password%20won%27t%20change%20until%20yo
u%20access%20the%20link%20above%20and%20create%20a%20new%20one.%3C
%2Fp%3E%09%3C%2Ftd%3E%3C%2Ftr%3E
```

XML Example:

```
curl -k -v -H"Content-Type: application/xml" -
d"<user><email>email@yahoo.com</email><application><app_id>app_id<
/app_id><app_secret>app_secret</app_secret></application></user>"
https://<User Service URL>/users/password.xml
```

Response:

- 201 – Success
- 422 – Failure, user has not confirmed yet
- Other HTTP pre-defined error codes

2.7.1 Edit User Password

The currently logged in user is allowed to edit the password.

Input:

Mandatory:

current password: Unique current password that is used so far to login

Type: String

new password: Unique new password that will be used to login from here
on

Type: String

XML Example:

```
curl -X PUT -k -v -H"Authorization: auth_token <auth_token>
-d"<user><current_password>current
password/current_password><password>new password/password></user>"
```

```
-H"Content-Type: application/xml" https://<User Service URL>/users.xml
```

Curl JSON Example:

```
curl -X PUT -k -v -H"Authorization: auth_token <auth_token"&  
-d'{"user":{"current_password":"current password", "password":"new  
password"}}"  
-H"Content-Type: application/json" https://<User Service URL>/users.json
```

Output:

- 200 – Success, password was successfully changed
- 422 – Unprocessable entity, the current password is invalid
- 401 – Unauthorized, auth token does not match current user

2.8 Reset Password with Token

Verifies the password reset token provided by the user and resets the user's password. The token is sent to the user in an email.

Input:**Mandatory:**

- password_reset_token: password reset token provided by the user that is sent in the email to resets the user's password
Type: String
- password: New password that will be used from here on
Type: String
- password_confirmation: Re-type the above 'password' for confirmation
Type: String

XML Example:

```
curl -X PUT -k -v -H"Authorization: auth_token <auth_token>"  
-d"<user><reset_password_token  
>abcdef</reset_password_token  
><password>newpass</password><password_confirmation  
>newpass</password_confirmation  
></user>"  
-H"Content-Type: application/xml" https://<User Service URL>/users/password.xml
```

JSON Example:

```
curl -X PUT -k -v -H"Authorization: auth_token <auth_token>"  
-d{"user":{"reset_password_token":"abcdef","password":"newpass",  
"password_confirmation":"newpass"}}"  
-H"Content-Type: application/json" https://<User Service  
URL>/users/password.json
```

Output:

200 – Success

422 – Failure, if the token is invalid, or has expired, or if the password validation fails

2.9 Edit User Attributes

The currently logged in user is allowed to edit only its own attributes. The Ayla admin can also edit user attributes. The email address cannot be modified with this call. If the email address is passed as an attribute, the corresponding user is updated rather than the current user.

Input:**Optional:**

Firstname: Account user first name

Type: String

Lastname: Account user last name

Type: String

Country: Country of residence

Type: String

XML Example:

```
curl -k -X PUT -H"Authorization: auth_token <auth_token>" -  
H"Content-Type: application/json"  
-d"<user><firstname>fn1</firstname></user>"  
https://<user service url>/users.xml
```

JSON Example:

```
curl -k -X PUT -H"Authorization: auth_token <auth_token>" -  
H"Content-Type: application/json"  
-d{"user":{"firstname":"fn1", "lastname":"ln1", "country": "a  
country"}}"  
https://<user service url>/users.json
```

Response:

200 – Success, user was successfully edited

401 – Failure, unauthorized, auth token does not match current user

2.10 Update User Email

Updates the email of the current user. A new login is required for the user to continue to operate in the system. The caller/user must ensure the correctness of the new email address. Any errors, such as, inadvertent typos could result in the user not being able to log in. Ayla recommends that the UI confirms the new email address before this API is called.

Input:

Mandatory:

email: the new email address of the user

Type: String

XML Example:

```
curl -v -X PUT -H"Authorization: auth_token <auth_token>"  
https://<user service  
url>/users/update\_email.xml?email=newemail@gmail.com
```

JSON Example:

```
curl -v -X PUT -H"Authorization: auth_token <auth_token>"  
https://<user service  
url>/users/update\_email.json?email=newemail@gmail.com
```

Output:

200 – Success, user was successfully edited

401 – Unauthorized, auth token does not match current user

2.11 Delete User

The currently logged in user is allowed to delete an email account.

XML Example:

```
curl -k -v -X DELETE -H"Authorization: auth_token <auth_token>"  
https://<user service url>/users.xml
```

JSON Example:

```
curl -k -v -X DELETE -H"Authorization: auth_token <auth_token>"  
https://<user service url>/users.json
```


Response:

- 204 – Success, user was successfully edited
- 401 – Unauthorized, auth token does not match current user

2.12 Sign out user

Removes the user token from the user service database.

XML Example:

```
$ curl -X POST -d "<user><access-token>access -token </access-token></user>" -H "Content-Type:application/xml" https://<user service url>/users/sign\_out.xml
```

return

```
<?xml version="1.0" encoding="UTF-8"?>
<authorization>
  <logout>true</logout>
</authorization>
```

JSON Example:

```
$ curl -X POST -d '{"user": {"access_token": "access_token"}}' -H
"Content-Type:application/json" https://<user service url>/users/sign\_out.json
```

return

```
{"logout":"true"}
```

Response:

- 200 – Success, a user was successfully logged out
- 401 – Unauthorized, auth token does not match current user

2.13 Get User Attributes

Gets user attributes with the user's access token.

JSON example:

```
curl -X GET -H"Authorization: auth_token <auth_token>"
https://User Service URL/users/get\_user\_profile.json
```

Response:

```
{
  "approved": true,
  "city": "Sunnyvale",
  "company": "Ayla",
  "confirmed_at": "2014-06-11T17:44:38Z",
  "country": "USA",
  "created_at": "2012-06-11T17:42:11Z",
  "email": "testemail@testemail.com",
  "firstname": "fn",
  "lastname": "ln",
  "phone": "1111111",
  "state": "",
  "street": "",
  "updated_at": "2014-04-19T21:10:46Z"
}
```

Output:

- 200 – Success, lists all the user attributes
- 401 – Unauthorized, auth token does not match current user

2.14 Get User Attributes with associated Roles

API to get the attributes of a user with associated Roles.

GET /api/v1/users/:user_id[.format]

Input:

Mandatory:

dashboard>

user_id: User's ID that is generated when the user is created. <Find it from

Type: Integer

Please confirm with Engineering on the inputs and outputs and Response Codes.

2.14 Get User Attributes with associated Roles API to get the attributes of a user with associated Roles.

Curl JSON example:

```
curl -X GET -H "Authorization: auth_token 9c6fdea0ea214640a6eb27e4b65c140a"
https://staging-user.aylanetworks.com/users/444.json
{"admin": true, "approved": true, "ayla_dev_kit_num": "", "city": "BA", "company": "aylanetworks",
"country": "Argentina", "created_at": "2012-12-27T17:38:58Z", "email": "pablo@aylanetworks.com", "firstname": "Pablo", "id": 444, "lastname": "Rivera", "oem_approved": false, "oem_id": 3, "origin_oem_id": 3, "phone": "55553333", "state": "", "street": "", "terms_accepted": true, "terms_accepted_at": "2013-10-18T14:25:44Z", "terms_email_sent_at": "2015-02-05T14:03:47Z", "terms_token": "4dafcbaeaf3248ed991fddb33e765dc8", "updated_at": "2015-02-05T14:12:27Z", "zip": "4444", "roles": [{"name": "Ayla::Admin"}, {"name": "Ayla::Staff"}]}
```

Curl JSON example:

```
curl -X GET -H "Authorization: auth_token 9c6fdea0ea214640a6eb27e4b65c140a"
https://user.aylanetworks.com/users/444.json
```

```
{"admin":true,"approved":true,"ayla_dev_kit_num":"","city":"BA","company":"aylanetworks",
"country":"Argentina","created_at":"2012-12-27T17:38:58Z","email":"pablo@aylanetworks.com",
"firstname":"Pablo","id":444,"lastname":"Rivera","oem_approved":false,"oem_id":3,"origin_oem_id":3,
"phone":"55553333","state":"","street":"","terms_accepted":true,"terms_accepted_at":"2013-10-18T14:25:44Z",
"terms_email_sent_at":"2015-02-05T14:03:47Z","terms_token":"4dafcbaeaf3248ed991fddb33e765dc8",
"updated_at":"2015-02-05T14:12:27Z","zip":"4444","roles":[{"name":""}, {"name":"Ayla::Admin"}, {"name":"OEM::Staff"}]}
```

3 USER METADATA

3.1 GET /api/v1/users/data[.format]

Returns a list of user data keys.

Input:

Optional:

key: user datum key

Type: String

Validation:(Limit: 255 bytes)

user_email: User metadata to be updated

Type: String

{"keys": ["contact1", "contact2"]} –

Returns all the datum objects for the keys in the array.

A pattern string like "contact%" -

The format of the pattern is as follows:

The "%25" sign is used to define both before and after the pattern.

To select all data ending with the letter "s": %s25

To select all data beginning with the letter "s": s%25

To select all data containing the pattern "input":
%25input%25

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/data.json
```

XML Examples:

```
$ curl -g -H "Authorization: auth_token ,<auth_token>"  
"https://<user service  
url>/api/v1/users/data.xml?keys\[]=contact1&keys\[]=contact2"
```

Or:

```
curl -g -H "Authorization: auth_token <auth_token>" https://<user service url>/api/v1/users/data.xml?keys=contact%
```

Response:

```
[{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"contact1","updated_at":"2014-02-19T23:42:46Z","value":"caontact 2 value","dsn":"123"}}, {"datum":{"created_at":"2014-02-19T23:42:46Z","key":"contact2","updated_at":"2014-02-19T23:42:46Z","value":"contact 2 value","dsn":"123"}}]
```

Output:

200 – Success

3.2 GET /api/v1/users/data/:key[.format]

Returns a datum value for a key.

Input

Mandatory:

key: a datum key

Type: String
Validation:(Limit: 255 bytes)

Optional:

user_email: Update user metadata
Type: String

XML Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<User Service URL>/api/v1/users/data/<key>.xml
```

JSON Example

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/data/<key>.json
```

Response

```
{ "datum": { "created_at": "2014-02-19T23:42:46Z", "key": "aKey", "updated_at": "2014-02-19T23:42:46Z", "value": "street value", "dsn": "123" } }
```

Response:

200 – Success
404 – Not found, the datum doesn't exist

3.3 GET /api/v1/users/:user_id/data[.format]

Returns a list of user data.

Input:**Mandatory:**

user_id: User's ID that is generated when the user is created. <Find it from dashboard>

Type: Integer

XML Example:

```
https://<User service url>/api/v1/users/<user-id>/data.xml
```

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/<user-id>/data.json
```

Output:

```
[ { "datum": { "created_at": "2015-05-18T17:53:59Z", "key": "away", "updated_at": "2015-05-18T17:53:59Z", "user_id": 10815, "value": "something" } } ]
```

Response

200 – Success

404 – Not Found, the user ID doesn't exist

3.4 POST /api/v1/users/data[.format]

Creates a new datum with the provided parameters for a user.

Input:**Mandatory:**

datum: A hash with the datum to be created:

key: datum key

Type: String

Validation:(Limit: 255 bytes)value: datum

value

Type: String

Validation:(Limit: 65536 bytes)

XML Example:

```
$ curl -X POST -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" -d
"<datum><key>away</key><value>something</value></datum>"
https://<User Service URL>/api/v1/users/data.xml
```

JSON Example:

```
curl -v -X POST -H "Content-Type: application/json" -H
"Authorization: auth_token <auth_token>" -d
'{"datum":{"key":"away","value":"something"}}' https://<User Service URL>/api/v1/users/data.json
```

Response**Success: 201 OK**

```
{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"data","updated_at":"2014-
02-19T23:42:46Z","value":"street value","dsn":"123"}}
```

3.5 PUT /api/v1/users/data/:key[.format]

Updates a datum with the provided parameters.

Input:**Mandatory:**

datum: A hash with the datum to be updated:

key: a datum key

Type: String

value: datum's value

Type: String

JSON Example:

```
$ curl -X PUT -H "Content-Type: application/json" -H  
"Authorization: auth_token <auth_token>" -d:  
'{"datum":{"value":"something"}}' https://<User Service  
URL>/api/v1/users/data/<Key>.json
```

Response

```
{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"aKey","updated_at":"2014-02-  
19T23:42:46Z","value":"something","dsn":"123"}}
```

Error Codes:

- 200 – Success
- 422 – Unprocessable entity
- 404 – Not Found in case the datum doesn't exist

3.6 DELETE /api/v1/users/data/:key[.format]

Destroys a datum.

Input:**Mandatory:**

- datum: A hash with the datum to be updated:
- key: a datum key
- Type: String

JSON Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth_token>"  
https://<User Service URL>/api/v1/users/data/aKey.json
```

Response:

- 200 – Success
- 404 – Not found, the datum doesn't exist

4 DEVICES

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

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

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

4.1 *GET apiv1/devices[.format]*

Obtains the list of devices for a particular user.

Input: HTTP Authorization Header has to include string "auth_token <token>" where <token> is obtained from user object returned on a successful sign in from a developer site.

XML Example:

```
$ curl -H "Authorization: auth_token <auth-token>" https://<Device
service url>/apiv1/devices.xml
[{"trigger_app":{"name":"email","nickname":null,"key":34837,"trigger_key":36939,"userna
me":"alankritha","repeat_freq":30,"param1":"aylasharing@gmail.com","param2":null,"para
m3":"test
msg","param4":null,"param5":null,"email_template_id":"customizable_body","email_body_
html":"\u012erenginio numeris gavo ... \u012esp\u0117jimas ma\u017eai
druskos","email_subject":"\u012erenginio","contact_id":null}}]
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<devices type="array">
  <device>
    <product-name>demo2</product-name>
    <model>model_demo2</model>
    <dsn>DSN2</dsn>
    <oem-model></oem-model>
    <template-id type="integer" nil="true"></template-id>
```



```

    <mac nil="true"></mac>
    <connected-at type="datetime">2014-07-
03T05:12:10Z</connected-at>
    <key type="integer">2</key>
    <has-properties type="boolean">false</has-properties>
    <product-class></product-class>
    <connection-status>"Online"</connection_status>
  </device>
  <device>
    <product-name>demo1</product-name>
    <model>m1</model>
    <dsn>DSN1</dsn>
    <oem-model nil="true"></oem-model>
    <template-id type="integer" nil="true"></template-id>
    <mac nil="true"></mac>
    <connected-at type="datetime">2014-06-
30T05:30:57Z</connected-at>
    <key type="integer">1</key>
    <has-properties type="boolean">true</has-properties>
    <product-class></product-class>
    <connection-status>"Offline"</connection_status>
  </device>
</devices>

```

If a device is shared with the current user, it contains a "grant" section as shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<devices type="array">
  <device>
    <product-name>demo2</product-name>
    <model>model_demo2</model>
    <dsn>DSN2</dsn>
    <oem-model></oem-model>
    <template-id type="integer" nil="true"></template-id>
    <mac nil="true"></mac>
    <connected-at type="datetime">2014-07-
03T05:12:10Z</connected-at>
    <key type="integer">2</key>
    <has-properties type="boolean">false</has-properties>
    <product-class></product-class>
    <connection-status>"Online"</connection_status>
    <grant>
      <user-id type="integer">1</user-id>
    </grant>
  </device>
</devices>

```

```

        <start-date-at type="datetime">2014-06-
17T23:14:33Z</start-date-at>
        <end-date-at nil="true"></end-date-at>
        <operation>write</operation>
    </grant>

```

If a device is a gateway, it contains an additional section as shown below:

```

<?xml version="1.0" encoding="UTF-8"?>
<devices type="array">
  <device>
    <product-name>Ayla EVB</product-name>
    <model>model number </model>
    <dsn>ACXXXXXXXXXXXX</dsn>
    <oem-model><oem-model></oem-model>
    <template-id type="integer">1077</template-id>
    <mac><mac Id></mac>
    <lan-ip>172.17.0.xxx</lan-ip>
    <connected-at type="datetime">2014-12-
21T04:51:26Z</connected-at>
    <key type="integer">XXX</key>
    <lan-enabled type="boolean">true</lan-enabled>
    <has-properties type="boolean">true</has-properties>
    <product-class>abcd</product-class>
    <connection-status>Online</connection-status>
    <lat type="decimal">30.7749</lat>
    <lng type="decimal">-133.419</lng>
    <device-type>Gateway</device-type>
    <gateway-type>gateway type</gateway-type>
  </device>
</devices>

```

JSON Example:

```
$ curl -H "Authorization: auth_token <auth-token>"https://<Device Service URL>/apiv1/devices.json
```

Response:

```

[
  {
    "device": {
      "dsn": "DSN2",

```

```

        "mac": null,
        "model": "model_demo2",
        "oem_model": "",
        "product_name": "demo2",
        "template_id": null,
        "connected_at": "2014-07-03T05:12:10Z",
        "key": 2,
        "has_properties": false,
        "product_class": "",
        "connection_status": "Online",
        "grant": {"user_id": 1, "start_date_at": "2014-06-17T23:14:33Z", "end_date_at": null, "operation": "write"}
    }
}
]

```

4.2 GET *apiv1/devices/<key>[.format]*

Obtains details for a device id specified by the key.

Input:

Mandatory:

key: Id of the device as returned when retrieving devices for a user.

Type: String

XML Example:

```
$ curl -H "Authorization: auth_token <auth-token>" https://<Device Service URL>/apiv1/devices/<device-id>.xml
```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<device>
  <product-name>demo1</product-name>
  <model>m1</model>
  <dsn>DSN1</dsn>
  <oem-model nil="true"></oem-model>
  <user-id type="integer">1</user-id>
  <template-id type="integer" nil="true"></template-id>
  <mac nil="true"></mac>
  <connected-at type="datetime">2014-07-15T19:48:33Z</connected-at>
  <key type="integer">1</key>
  <product-class></product-class>

```

```

    <has-properties type="boolean">true</has-properties>
    <lat type="decimal">37.7749</lat>
    <lng type="decimal">-122.419</lng>
    <connection_status>"Online"</connection_status>
  </device>

```

Response for a Gateway device with nodes:

```

<?xml version="1.0" encoding="UTF-8"?>
<device>
  <id type="integer">18981</id>
  <product-name>Ayla Linux</product-name>
  <model>model </model>
  <dsn>ACXXXXXXXXXXXX</dsn>
  <oem>XXXX</oem>
  <oem-model>zigbee1</oem-model>
  <sw-version>devd 0.3-beta</sw-version>
  <user-id type="integer">user id</user-id>
  <template-id type="integer">984</template-id>
  <mac>00:15:8D:52:xx.xx</mac>
  <ip>xxx.xx.xxxx.xxx</ip>
  <lan-ip>192.xxx.x.xxx</lan-ip>
  <ssid nil="true"></ssid>
  <connected-at type="datetime">2015-02-25T20:12:26Z</connected-at>
  <key type="integer">18981</key>
  <product-class><product class></product-class>
  <has-properties type="boolean">true</has-properties>
  <lan-enabled type="boolean">true</lan-enabled>
  <enable-ssl type="boolean">false</enable-ssl>
  <enable-ip-loc type="boolean">true</enable-ip-loc>
  <enable-wifi-loc type="boolean">false</enable-wifi-loc>
  <ans-enabled type="boolean">true</ans-enabled>
  <ans-server>ans.aylanetworks.com</ans-server>
  <log-enabled type="boolean">false</log-enabled>
  <registered type="boolean">true</registered>
  <connection-status>Online</connection-status>
  <registration-type>Button-Push</registration-type>
  <lat type="decimal">37.8808</lat>
  <lng type="decimal">-122.176</lng>
  <homekit nil="true"></homekit>
  <enable-setup-loc type="boolean">true</enable-setup-loc>
  <force-location type="boolean">false</force-location>

```

```
<module-updated-at type="datetime">2014-08-04T21:24:23Z</module-
updated-at>
<registrable type="boolean">true</registrable>
<regtoken>regtoken</regtoken>
<device-type>Gateway</device-type>
<gateway-type><gateway-type></gateway-type>
<nodes type="array">
  <node>
    <id type="integer">xxxxx</id>
    <product-name>product name </product-name>
    <model>Smart_Bulb_Converter</model>
    <dsn>device dsn </dsn>
    <oem>oem</oem>
    <oem-model><oem-model></oem-model>
    <sw-version>48</sw-version>
    <user-id type="integer">2986</user-id>
    <template-id nil="true"></template-id>
    <mac>mac id</mac>
    <ip nil="true"></ip>
    <lan-ip nil="true"></lan-ip>
    <ssid nil="true"></ssid>
    <connected-at nil="true"></connected-at>
    <key type="integer">xxxx</key>
    <product-class></product-class>
    <has-properties type="boolean">true</has-properties>
    <lan-enabled type="boolean">false</lan-enabled>
    <enable-ssl nil="true"></enable-ssl>
    <enable-ip-loc type="boolean">true</enable-ip-loc>
    <enable-wifi-loc type="boolean">false</enable-wifi-loc>
    <ans-enabled type="boolean">true</ans-enabled>
    <ans-server>ans.aylanetworks.com</ans-server>
    <log-enabled type="boolean">false</log-enabled>
    <registered type="boolean">true</registered>
    <connection-status>Offline</connection-status>
    <registration-type>Node</registration-type>
    <lat nil="true"></lat>
    <lng nil="true"></lng>
    <homekit nil="true"></homekit>
    <enable-setup-loc type="boolean">true</enable-setup-loc>
    <force-location type="boolean">false</force-location>
    <module-updated-at nil="true"></module-updated-at>
    <registrable type="boolean">true</registrable>
    <regtoken>eac585</regtoken>
```

```

<device-type>Node</device-type>
<gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
<node-type>node-type</node-type>
  </node>
  <node>
    <id type="integer">52x2xxx</id>
    <product-name> product-name </product-name>
    <model>Door_Sensor</model>
    <dsn>device dsn</dsn>
    <oem>oem id</oem>
    <oem-model>oem-model</oem-model>
    <sw-version>39</sw-version>
    <user-id type="integer">2986</user-id>
    <template-id nil="true"></template-id>
    <mac>mac id</mac>
    <ip nil="true"></ip>
    <lan-ip nil="true"></lan-ip>
    <ssid nil="true"></ssid>
    <connected-at nil="true"></connected-at>
    <key type="integer">xxxxxx</key>
    <product-class></product-class>
    <has-properties type="boolean">true</has-properties>
    <lan-enabled type="boolean">false</lan-enabled>
    <enable-ssl nil="true"></enable-ssl>
    <enable-ip-loc type="boolean">true</enable-ip-loc>
    <enable-wifi-loc type="boolean">false</enable-wifi-loc>
    <ans-enabled type="boolean">true</ans-enabled>
    <ans-server>ans.aylanetworks.com</ans-server>
    <log-enabled type="boolean">false</log-enabled>
    <registered type="boolean">true</registered>
    <connection-status>Online</connection-status>
    <registration-type>Node</registration-type>
    <lat nil="true"></lat>
    <lng nil="true"></lng>
    <homekit nil="true"></homekit>
    <enable-setup-loc type="boolean">true</enable-setup-loc>
    <force-location type="boolean">false</force-location>
    <module-updated-at nil="true"></module-updated-at>
    <registrable type="boolean">true</registrable>
    <regtoken>xxxxxxx</regtoken>
    <device-type>Node</device-type>
    <gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
    <node-type>Zigbee</node-type>

```

```
</node>
<node>
<id type="integer">xxxxx</id>
<product-name> product-name </product-name>
<model>Motion_Sensor</model>
<dsn>device dsn</dsn>
<oem>oem </oem>
<oem-model>zigbee1</oem-model>
<sw-version>1</sw-version>
<user-id type="integer">2986</user-id>
<template-id nil="true"></template-id>
<mac>mac id</mac>
<ip nil="true"></ip>
<lan-ip nil="true"></lan-ip>
<ssid nil="true"></ssid>
<connected-at nil="true"></connected-at>
<key type="integer">xxxx</key>
<product-class></product-class>
<has-properties type="boolean">true</has-properties>
<lan-enabled type="boolean">false</lan-enabled>
<enable-ssl nil="true"></enable-ssl>
<enable-ip-loc type="boolean">true</enable-ip-loc>
<enable-wifi-loc type="boolean">false</enable-wifi-loc>
<ans-enabled type="boolean">true</ans-enabled>
<ans-server>ans.aylanetworks.com</ans-server>
<log-enabled type="boolean">false</log-enabled>
<registered type="boolean">true</registered>
<connection-status>Online</connection-status>
<registration-type>Node</registration-type>
<lat nil="true"></lat>
<lng nil="true"></lng>
<homekit nil="true"></homekit>
<enable-setup-loc type="boolean">true</enable-setup-loc>
<force-location type="boolean">false</force-location>
<module-updated-at nil="true"></module-updated-at>
<registrable type="boolean">true</registrable>
<regtoken>reg toen</regtoken>
<device-type>Node</device-type>
<gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
<node-type>Zigbee</node-type>
</node>
</nodes>
</device>
```

For node device:

```
<?xml version="1.0" encoding="UTF-8"?>
<device>
  <id type="integer">52210</id>
  <product-name> product-name </product-name>
  <model>Wireless_Switch</model>
  <dsn>device dsn</dsn>
  <oem>oem</oem>
  <oem-model>zigbee1</oem-model>
  <sw-version>15</sw-version>
  <user-id type="integer">xxxx</user-id>
  <template-id nil="true"></template-id>
  <mac>mac id</mac>
  <ip nil="true"></ip>
  <lan-ip nil="true"></lan-ip>
  <ssid nil="true"></ssid>
  <connected-at nil="true"></connected-at>
  <key type="integer">52210</key>
  <product-class></product-class>
  <has-properties type="boolean">true</has-properties>
  <lan-enabled type="boolean">false</lan-enabled>
  <enable-ssl nil="true"></enable-ssl>
  <enable-ip-loc type="boolean">true</enable-ip-loc>
  <enable-wifi-loc type="boolean">false</enable-wifi-loc>
  <ans-enabled type="boolean">true</ans-enabled>
  <ans-server>ans.aylanetworks.com</ans-server>
  <log-enabled type="boolean">false</log-enabled>
  <registered type="boolean">true</registered>
  <connection-status>Online</connection-status>
  <registration-type>Node</registration-type>
  <lat nil="true"></lat>
  <lng nil="true"></lng>
  <homekit nil="true"></homekit>
  <enable-setup-loc type="boolean">true</enable-setup-loc>
  <force-location type="boolean">false</force-location>
  <module-updated-at nil="true"></module-updated-at>
  <registrable type="boolean">true</registrable>
  <regtoken>reg token</regtoken>
  <device-type>Node</device-type>
  <gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
  <node-type>Zigbee</node-type>
</device>
```


JSON Example:

```
$ curl -H "Authorization: auth_token <auth-token> https://<Device Service URL>/apiv1/devices/2.json
```

Response:

```
{
  "device": {
    "dsn": "DSN2",
    "mac": null,
    "model": "model_demo2",
    "oem_model": null,
    "product_name": "demo2",
    "template_id": 5,
    "user_id": 1,
    "connected_at": "2014-07-15T20:04:11Z",
    "key": 2,
    "product_class": "demo",
    "has_properties": true,
    "lat": "37.7749",
    "lng": "-122.419",
    "connection_status": "Online"
  }
}
```

4.3 GET *apiv1/dsns/:dsn/nodes[.format]*

Obtains the list of node devices along with their respective properties for a particular user gateway. The HTTP Authorization header needs to include the auth_token parameter as returned in the user object after a successful login.

Input:**Mandatory:**

DSN: Dsn of the device as returned when retrieving devices for a user

Type: String

Output: An array of node devices with properties

XML Example:

```
$ curl -H "Authorization: auth_token <auth-token>" https://<Device Service URL>/apiv1/devices/<device-id>/nodes.xml
```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<devices type="array">
  <device>
    <product-name>product-name</product-name>
    <model>Smart_Plug</model>
    <dsn>dsn</dsn>
    <oem-model>oem-model</oem-model>
    <user-id type="integer">2986</user-id>
    <template-id nil="true"></template-id>
    <mac>mac id</mac>
    <lan-ip nil="true"></lan-ip>
    <connected-at nil="true"></connected-at>
    <key type="integer">xxxx</key>
    <registered type="boolean">true</registered>
    <lan-enabled type="boolean">false</lan-enabled>
    <has-properties type="boolean">true</has-properties>
    <product-class></product-class>
    <connection-status>Offline</connection-status>
    <lat nil="true"></lat>
    <lng nil="true"></lng>
    <power type="integer">1</power>
    <ntwkaddr>0xC72E</ntwkaddr>
    <device-type>Node</device-type>
    <gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
    <node-type>Zigbee</node-type>
    <properties type="array">
      <property>
        <name>1_dev_id</name>
        <base-type>string</base-type>
        <read-only type="boolean">true</read-only>
        <direction>output</direction>
        <scope>oem</scope>
        <data-updated-at type="datetime">2015-02-
24T20:36:48Z</data-updated-at>
        <key type="integer">key</key>
        <device-key type="integer">52206</device-key>
        <product-name nil="true"></product-name>
        <track-only-changes type="boolean">false</track-only-
changes>
        <display-name>1_dev_id</display-name>
        <host-sw-version type="boolean">false</host-sw-version>
        <time-series type="boolean">false</time-series>

```

```

    <derived type="boolean">false</derived>
    <app-type nil="true"></app-type>
    <recipe nil="true"></recipe>
    <value>0x0051</value>
    </property>
    <property>
    <name>1_in_0x0006_0x0000</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">false</read-only>
    <direction>input</direction>
    <scope>user</scope>
    <data-updated-at type="datetime">2015-02-
26T03:11:07Z</data-updated-at>
    <key type="integer">xxxx</key>
    <device-key type="integer">52206</device-key>
    <product-name nil="true"></product-name>
    <track-only-changes type="boolean">false</track-only-
changes>
    <display-name>display-name</display-name>
    <host-sw-version type="boolean">false</host-sw-version>
    <time-series type="boolean">false</time-series>
    <derived type="boolean">false</derived>
    <app-type nil="true"></app-type>
    <recipe nil="true"></recipe>
    <value type="integer">1</value>
    <metadata>
    <endpt type="integer">1</endpt>
    <clusterId>0x0006</clusterId>
    <attribId>0x0000</attribId>
    <attribType>16</attribType>
    </metadata>
    </property>
  </properties>
</device>
<device>
  <product-name>product-name</product-name>
  <model>Smart_Bulb_Converter</model>
  <dsn>dsn</dsn>
  <oem-model>zigbeel</oem-model>
  <user-id type="integer">2986</user-id>
  <template-id nil="true"></template-id>
  <mac>mac id</mac>
  <lan-ip nil="true"></lan-ip>

```

```

<connected-at nil="true"></connected-at>
<key type="integer">52207</key>
<registered type="boolean">true</registered>
<lan-enabled type="boolean">false</lan-enabled>
<has-properties type="boolean">true</has-properties>
<product-class></product-class>
<connection-status>Offline</connection-status>
<lat nil="true"></lat>
<lng nil="true"></lng>
<power type="integer">1</power>
<ntwkaddr>0x46EC</ntwkaddr>
<device-type>Node</device-type>
<gateway-dsn>ACXXXXXXXXXXXX</gateway-dsn>
<node-type>Zigbee</node-type>
  <properties type="array">
    <property>
      <name>1_dev_id</name>
      <base-type>string</base-type>
      <read-only type="boolean">true</read-only>
      <direction>output</direction>
      <scope>oem</scope>
      <data-updated-at type="datetime">2015-02-
24T20:37:27Z</data-updated-at>
      <key type="integer">364286</key>
      <device-key type="integer">52207</device-key>
      <product-name nil="true"></product-name>
      <track-only-changes type="boolean">false</track-only-
changes>
      <display-name>1_dev_id</display-name>
      <host-sw-version type="boolean">false</host-sw-version>
      <time-series type="boolean">false</time-series>
      <derived type="boolean">false</derived>
      <app-type nil="true"></app-type>
      <recipe nil="true"></recipe>
      <value>0x0100</value>
    </property>
    <property>
      <name>1_in_0x0006_0x0000</name>
      <base-type>boolean</base-type>
      <read-only type="boolean">false</read-only>
      <direction>input</direction>
      <scope>user</scope>

```

```

    <data-updated-at type="datetime">2015-02-
24T21:29:54Z</data-updated-at>
    <key type="integer">364289</key>
    <device-key type="integer">52207</device-key>
    <product-name nil="true"></product-name>
    <track-only-changes type="boolean">>false</track-only-
changes>
    <display-name>display-name</display-name>
    <host-sw-version type="boolean">>false</host-sw-version>
    <time-series type="boolean">>false</time-series>
    <derived type="boolean">>false</derived>
    <app-type nil="true"></app-type>
    <recipe nil="true"></recipe>
    <value type="integer">0</value>
    <metadata>
      <endpt type="integer">1</endpt>
      <clusterId>0x0006</clusterId>
      <attribId>0x0000</attribId>
      <attribType>16</attribType>
    </metadata>
  </property>
  <property>
    <name>l_in_clusters</name>
    <base-type>string</base-type>
    <read-only type="boolean">>true</read-only>
    <direction>output</direction>
    <scope>oem</scope>
    <data-updated-at type="datetime">2015-02-
24T20:37:27Z</data-updated-at>
    <key type="integer">364287</key>
    <device-key type="integer">52207</device-key>
    <product-name nil="true"></product-name>
    <track-only-changes type="boolean">>false</track-only-
changes>
    <display-name>l_in_clusters</display-name>
    <host-sw-version type="boolean">>false</host-sw-version>
    <time-series type="boolean">>false</time-series>
    <derived type="boolean">>false</derived>
    <app-type nil="true"></app-type>
    <recipe nil="true"></recipe>
    <value>["0x0000", "0x0004", "0x0003", "0x0006",
"0x0005", "0xFFFF"]</value>
  </property>
</properties>

```

```

    </device>
</devices>

```

4.4 GET apiv1/dsns/<dsn>[.format]

Obtains details for a device specified by <dsn>.

Input:

Mandatory:

dsn: Dsn for a device as returned when retrieving a device for a user.

Type : String

XML Example:

```
$ curl -H "Authorization: auth_token <auth-token>"
https://<device service url> /apiv1/dsns/ACXXXXXXXXXXXXX.xml
```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<device>
  <product-name>demo1</product-name>
  <model>m1</model>
  <dsn>AACXXXXXXXXXXXXX</dsn>
  <oem-model nil="true"></oem-model>
  <user-id type="integer">1</user-id>
  <template-id type="integer" nil="true"></template-id>
  <mac nil="true"></mac>
  <connected-at type="datetime">2014-07-
15T19:48:33Z</connected-at>
  <key type="integer">1</key>
  <product-class></product-class>
  <has-properties type="boolean">true</has-properties>
  <lanip>
    <lanip_key_id>1</lanip_key_id>
    <lanip_key>idPypxxxxxxV+Qobzngu9H2XXXX==</lanip_key>
    <keep-alive>3600</keep-alive>
  </lanip>
  <lat type="decimal">37.7749</lat>
  <lng type="decimal">-122.419</lng>
  <connection_status>"Online"</connection_status>
</device>

```

JSON Example:

```
$ curl -H "Authorization: <auth-token>auth-token" https://<device service url>/apiv1/dsns/ACXXXXXXXXXXXXX.json
```

Response:

```
{
  "device": {
    "dsn": "ACXXXXXXXXXXXXX",
    "mac": null,
    "model": "model_demo2",
    "oem_model": null,
    "product_name": "demo2",
    "template_id": 5,
    "user_id": 1,
    "connected_at": "2014-07-15T20:04:11Z",
    "key": 2,
    "product_class": "demo",
    "has_properties": true,
    "lat": "37.7749",
    "lng": "-122.419",
    "connection_status": "Online"
  }
}
```

4.5 **POST** [apiv1/devices/:device_id/registration_window\[.format\]](#)

Opens and polls on the registration window.

Input:**Mandatory:**

Device_id : Id of the device as returned when retrieving devices for a user

Type : Integer

duration: Time in seconds

Default: 200. Use '0' value to close the window immediately

Type: Integer

XML Example:

```
$ curl -X POST -H"Authorization: auth_token <auth-token> -d
"<duration>200</duration>" -H "Content -Type: application/xml"
https://<device service url>/apiv1/devices/<device-
id>/registration window.xml
```

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:

401 – Failure, property not found or method not allowed

4.6 PUT apiv1/devices/:device_id/identify[.format]

It provides ability for turning a device on or off. A device must be identify on a node.

Input:**Mandatory:**

Id: Device Id. Node must exists

Type: Integer

Optional:

Value: time(in seconds) on or off

Default : set to “On” and time interval 60 seconds

Type: String

XML Example:

```
$ curl -X PUT -H"Authorization: auth_token <auth-token>auth-token"
-d "<value>On</value><time>60</time>" -H "Content -Type:
application/xml" https://<device service
url>/apiv1/devices/<device-id>/identify.xml
```

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:

200 – Success -{"id": "12345"} (For “On” and “Off” option)

or

{“id”:“on_0x123456789abc”,“status”:“success”} (For “Result”

404 – Device/Node does not exist

405 – Method not allowed or not supported on this device

401 – Unauthorized

4.7 GET apiv1/devices/register.xml using ip, dsn, regtype, time

GET apiv1/devices/register.xml?ip=<ip address>&dsn=<serial number>®type=<registration_type>&time=<latest_gw_nodes>

Obtains a registration candidate that matches the passed public ip address. If the dsn is passed then the candidate has to match the specified dsn. Only the most recently connected candidate is returned if multiple devices are found.

Input:

Mandatory:

regtype: registration type for this device. This is required for “Node” registration type, but optional for “Button-Push” and “Same-LAN” registration types.

Type: String

Validation: “Same-LAN”, “Button-Push”, “AP-Mode”, “Display”, “Dsn”, “None”(OEM), “Node”(For Gateway)

time: Time in minutes.

Default: “5”

Type: Integer

Optional:

ip: public ip address of requesting client. If no ip address is passed, then the service matches the ip address of the client making the request. When the web site is implementing device registration it must pass the remote ip of the client making the request in order for the proper candidate to be returned. When the request is originating from a mobile application that is using the same network as the device then no ip parameter is required because both the mobile device and the Ayla module share the same public ip.

dsn: Unique dsn of the device as returned when retrieving devices for a user.

If specified, the service candidate matches the dsn.

Note 1: For “Same-LAN” - device should not be registered. Default, if no request paramsFor “Button-Push” - button on device should have been pressed within last two min. Requires at least regtype or dsn in request params.

Note 2: “Display”, “AP-Mode” and “None” regtype should not use this API. However, if they choose to use this API, no devices are returned.

Output: A device object or not_found error

XML Example:

```
$ curl -H "Authorization: auth_token <auth-token>"https://device.service.url/apiv1/devices/register.xml?ip=96.249.XXX.XX
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<device>
  <product-name>proto 1</product-name>
  <dsn>AACXXXXXXXXXXXX</dsn>
  <model>AY001MXXX</model>
  <oem-model nil="true"></oem-model>
  <connected-at type="datetime">2014-11-
14T00:16:44Z</connected-at>
  <lan-ip>10.10.1.xxx</lan-ip>
</device>
```

JSON Example:

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

Response:

```
{
  "device": {
    "dsn": "AC000W000XXXX",
    "lan_ip": "192.XXX.3.XXX",
    "model": "AY001XXX",
    "oem_model": "AY001XXX",
    "product_name": "carrier 54",
    "connected_at": "2014-07-14T16:23:28Z",
    "product_class": ""
  }
}
```

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device.service.url>/apiv1/devices/register.json?ip=96.249.XXX.XX&dsn=ACXXXXXXXXXXXXXX&regtype=Button-Push
```

Response:

```
{
  "device": {
    "dsn": "ACXXXXXXXXXXXX",
```

```
"lan_ip": "192.168.3.XXX",  
"model": "AY001MXXX",  
"oem_model": "AY001MXXX",  
"product_name": "carrier 54",  
"connected_at": "2014-07-14T16:23:28Z",  
"product_class": ""  
}  
}
```

Return:

200 – Success

404 – Unable to find registrable device

412 – Precondition failed, gateway registration window is not open

\$ curl -H"Authorization: auth_token 2f74dee982df4b0dfe22fc50e7de5a3"

\$ curl -H"Authorization: auth_token 2f74dee982df4b0dfe22fc50e7de5a3"

4.8 GET***apiv1/devices/find_by_user_email?email=<email_address>***

Retrieves devices registered to a user based on the user's email.

URI: GET https://<device service url>/apiv1/devices/find_by_user_email?email=<email>**Access Control:**

Can be accessed by:

OEM admin

Users who have permissions to access the devices that are returned. This includes registered or associated (through tags) users.

Input:**Mandatory:**

Email: email_address of the user

Type: String**JSON Example:**

```
curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/devices/find\_by\_user\_email?email=abc@aylanetworks.com  
[{"device":{"dsn":"ACXXXXXXXXXXXXX"}}]
```

Return:

200 – Success

422 – Email parameter absent or OEM of user cannot be determined. Invalid email or user

404 – Cannot retrieve information from the email parameter. User with the email not found

4.9 *POST apiv1/devices.xml*

Registers a new device for a particular user.

Input:

Mandatory:

dsn: unique serial number of the device as returned when retrieving devices for a user. Required for all types except “Display”

Type: String

Optional:

Regtoken: registration token obtained from a device. Required for “Same-LAN” and “Display” registration type, optional in other cases

Type : String

setup_token: setup token generated by app. Required only for AP-Mode registration type, optional in other cases

Type: String

ip: public ip address of the device to be registered. If not specified then server uses public ip address of this post, request should match the public ip address of device.

Type: String

Lat: device latitude

Type: Float

Lng: device longitude

Type: Float

Note: lat and lng used only for “Same-LAN”, “Button-Push”, “AP-Mode”, if registration is successful

Valid registration types: “Same-LAN”, “Button-Push”, “AP-Mode”, “Display”, “None”(OEM), “Dsn”

Response :

200 - Success

417 – User ID is nil or registration candidate is invalid

Output: Device object that was registered.

XML Example:

```
$ curl -X POST -H"Authorization: auth_token <auth-token>" -d
"<device><dsn>DSN1</dsn><regtoken>ebc907</regtoken></device>" -H
"Content -Type: application/xml" https://<device service url>/apiv1/devices.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<device>
  <product-name>demo1</product-name>
  <dsn>DSN1</dsn>
  <model>m1</model>
  <connected-at type="datetime">2012-07-
15T16:39:18Z</connected-at>
  <key type="integer">1</key>
</device>
```

AP-Mode Case

```
$ curl -v -X POST -H "Authorization:auth_token <auth_token>" -H
"Content-Type: application/xml" -d
"<device><dsn>ACXXXXXXXXXXXX</dsn><setup_token>gWYQ9fiZ</setup_tok
en></device>" https://<device service url>/apiv1/devices.xml
```

JSON Example:**# Same-LAN (Default) Case**

```
$ curl -X POST -H"Authorization: auth_token <auth-token>" -d
'{"device":{"dsn":"AC000WT00000XXX","regtoken":"4d5XXX"}}' -H
"Content-Type: application/json" https://<device service url>/apiv1/devices.json
```

Response:

```
{
  "device": {
    "dsn": "AC000WTXXXXXXXXX",
    "model": "AY001MTP1",
    "product_name": "carrier T999",
    "connected_at": "2014-07-14T16:23:28Z",
    "key": 99
  }
}
```

AP-Mode Case

```
$ curl -X POST -H"Authorization: auth_token <auth-token>" -d
'{"device":{"dsn":"AC000WTXXXXXXXXX"},"setup_token":"12345678"}}' -
H "Authorization: auth_token <auth_token>==" -H "Content-Type:
application/json"https://<device service url>/apiv1/devices.json
```

Response:

```
{
  "device": {
    "dsn": "AC000WT00000XXXX",
    "model": "AY001MTXX",
    "product_name": "carrier T999",
    "connected_at": "2014-07-14T16:23:28Z",
    "key": 99
  }
}
```

Button-Push Case

```
$ curl -X POST -H"Authorization: auth_token <auth-token>" -d
'{"device":{"dsn":"dsn","reg":"12345678"}}' -H "Authorization:
auth_token <auth_token>"-H "Content-Type:
application/json"https://<device service url>/apiv1/devices.json
```

Response:

```
{
  "device": {
    "dsn": "AC000WT00000XXX",
    "model": "AY001MTP1",
    "product_name": "carrier T999",
    "connected_at": "2014-07-14T16:23:28Z",
    "key": 99
  }
}
```

Return:

200 – Success

417 – User id is nil or registration candidate is invalid

4.10 PUT *apiv1/devices/<key>[.format]*

Updates the device service with any necessary device attributes.

Input:**Mandatory:**

Device id: Id for a device as returned when retrieving a device for a user.

Type: Integer

XML Example:

```
$ curl -X PUT -H"Authorization: auth_token <auth-token>" -d
"<device><product_name>test1</product_name></device>" -H "Content-
Type:application/xml" https://<device service
url>/apiv1/devices/<id>.xml
```

JSON Example:

```
curl -v -X PUT -H"Authorization:auth_token<auth_token>"
-d'{"device": {"product_name":"New Name"}}' -H
"Content-Type: application/json" https://<device service
url>/apiv1/devices/<id>.json
```

Return:

200 – Success

4.11 PUT *apiv1/dsns/<dsn>[.format]*

Updates the device service with device attributes. For example, updates an OEM model.

Input:**Mandatory:**

dsn: unique serial number for a device as returned when retrieving a device for a user.

Type: String

XML Example:

```
$ curl -X PUT -H"Authorization: auth_token <auth-token>" -d
"<device><oem_model>test1</oem_model></device>" -H "Content-
Type:application/xml" https://<device service
url>/apiv1/dsns/AACXXXXXXXXXXXXX.xml
```

JSON Example:

```
curl -v -X PUT -H"Authorization:auth_token
<auth_token>" -d'{"device": {"product_name":"New
Name"}}' -H "Content-Type:
application/json" https://<device service
url>/apiv1/dsns/AC000W0000XXXX.json
```

Response:

200 – Success

4.12 PUT *apiv1/devices/<device_dsn>/transfer[.format]*

Transfers ownership from a current registered user to a target user.

Input:

Mandatory:

Device_dsn: unique serial number for a device as returned when retrieving a device for a user

Type: String

target: target user's email or ID

Type: String

XML Example:

```
$ curl -k -X PUT -H"Authorization: auth_token
75a4fe124d6247efbecbc6107ae252ce" https://<device service
url>/apiv1/devices/<device-
dsn>/transfer.xml?target=customer@email.com
```

JSON Example:

```
$ curl -k -X PUT -H"Authorization: auth_token<auth_token>"
https://<device service url>/apiv1/devices/<device-
dsn>/transfer.json?target=customer@email.com
```

Response:

200 – Success, device transferred

404 – Device with a device_dsn was not found or user with email or id was not found

403 – User doesn't have write access to a target user

401 – User doesn't have write access to a device

4.13 PUT *apiv1/devices/<device_id>/locations[.format]*

Updates a device location allowing users to override the device location.

Input:

Mandatory:

device id: unique serial number for a device as returned when retrieving a device for a user.

Lat: device latitude

Lng: device longitude

Curl XML Example:

```
$ curl -k -X PUT -H"Authorization: auth_token <auth_token>" -d
'<location><lat>38</lat><long>121.4835</long></location>'
http://<device>serviceurl>/apiv1/devices/<device\_id>/locations.xml
```

Curl JSON Example:

```
$ curl -k -X PUT -H"Authorization: auth_token <auth_token>"
-d '{ "location" : { "lat" : 38, "long" : 121.4835 }
}' https://<device>serviceurl>/apiv1/devices/<device-id>/locations.json
```

Response:

- 200 – Success, updated the device location
- 404 – Device with a device id was not found or user with email or id was not found
- 422 – Location information is not complete or is not valid
- 401 – A user doesn't have write access to a device

4.14 POST *apiv1/devices/<device_id>/locations[.format]*

Update device location allowing user to override the device location.

Input:

target: target user's email or ID

HTTP Methods: PUT

Data Formats: XML, JSON

Response:

Success: OK (200) - Device transferred

Failure: Not Found (404) - Device with <device_id> was not found

Failure: Unauthorized (401) - User doesn't have write access to device

Failure: Forbidden (422) - Location information is not complete or is not valid

Curl XML Example:

```
$ curl -k -X POST -H"Authorization: auth_token
75a4fe124d6247efbecbc6107ae252ce" -H"Content-Type:application/json"
-d '<location><lat>38</lat><long>121.4835</long></location>' https://ads-dev.aylanetworks.com/apiv1/devices/27671/locations.xml
```

Curl JSON Example:

```
$ curl -k -X POST -H"Authorization: auth_token  
75a4fe124d6247efbecbc6107ae252ce" -H"Content-Type:application/json"
```

```
-d '{ "location" : { "lat" : 38, "long" : 121.4835 } }' https://ads-  
dev.aylanetworks.com/apiv1/devices/27671/locations.json
```

4.15 PUT apiv1/devices/<device_id>/cmds/factory_reset[.format]

Sends a factory reset cmd comand to the device (Wi-Fi or Gateway or Node) with a device id.

This is not supported for devices with software version lower than 1.6.

Input:**Mandatory:**

Device id: unique serial number for a device as returned when retrieving a device for a user.

JSON Example:

```
$ curl -k -X PUT -H"Content-Type: application/json" -  
H"Authorization: auth_token (auth_token>" -d ' ' https://device  
service url>/apiv1/devices/<device\_id>/cmds/factory\_reset.json
```

Response:

200 – Success, cmd command was sent to the device successfully

422 – Unprocessable entity, software_version is lower than 1.6

4.16 DELETE apiv1/devices/<key>[.format]

Used to unregister a device from the current user.

Input: HTTP Authorization Header: has to include string “auth_token <token>” where <token> is obtained from user object returned on a successful sign in from developer site. See User Service Specification.

Output: None

HTTP Methods: DELETE

Data Formats: XML, JSON

Curl XML Example:

```
$ curl -X DELETE -H "Authorization: auth_token 6edf3c9bcab0485a89c82090a6c61091" -  
H "Content-Type: application/xml" https://ads-dev.aylanetworks.com/apiv1/devices/1.xml
```

Curl JSON Example:

```
$ curl -X DELETE -H "Authorization: auth_token j+8pGO8l6pr171x1ww+qJQ==  
85veg/r1A0ODsdtQ9hwxQw==" -H "Content-Type: application/json" https://ads-dev.aylanetworks.com/apiv1/devices/2.json
```

5 DEVICE METADATA

Use this API to get metadata for a device.

5.1 GET /apiv1/dsns/:dsn/data[.format]

Returns a list of device data keys.

Input:

Mandatory:

Dsn: unique dsn number for a device as returned when retrieving a device or a user
 Type: String

Optional:

{“keys”: [“street”, “addr”]}– Returns the datum objects for the key(s) in the array.

Validation: A pattern string like “street%” – The format of the pattern is represented as:

The “%25” sign is used to define wildcards both before and after the pattern.

To select all data ending with the letter “s”: %s25

To select all data beginning with the letter “s”:

s%25

To select all data containing the pattern “input”:
 %input%

Type: String

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-id>/data.json
```

XML Examples:

```
$ curl -g -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-id>/data.xml?keys\[\\]=street&keys\[\\]=something
```

Or:

```
curl -g -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-id>/data.xml?keys=s%
```

Output:

```
[{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"street","updated_at":"2014-02-19T23:42:46Z","value":"street value","dsn":"123"}}, {"datum":{"created_at":"2014-02-19T23:42:46Z","key":"contact","updated_at":"2014-02-19T23:42:46Z","value":"contact value","dsn":"123"}}]
```

Response:**200** – Success**5.2 GET /apiv1/dsns/:dsn/data/:key[.format]**

Returns a data value for a key.

Input:**Mandatory:**

Dsn : Device dsn obtained from GET /apiv1/devices

Type: String

key: a datum key

Type: String

Validation: Limit: 255 bytes

XML Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-id>/data/street.xml
```

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-id>/data/street.json
```

Output:

```
{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"street","updated_at":"2014-02-19T23:42:46Z","value":"street value","dsn":"123"}}
```

Response:**200** – Success**404** – Not Found, the datum doesn't exist.**5.3 POST /apiv1/dsns/:dsn/data[.format]**

Creates a new datum with the provided parameters for a device.

Input:**Mandatory:****datum:** A hash with the datum to be created:**key:** datum key

Validation: Limit: 255 bytes
 Type: String
value: datum string value (blank is ok)
 Validation: (Text: max. 2 MB) (For binary file - base64 encode it to a string)
 Type: String

XML Example:

```
$ curl -X POST -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/dsns/<device-id>/data.xml -d
"<datum><key>street</key><value>street value</value></datum>"
```

JSON Example:

```
curl -X POST -H "Content-Type: application/json" -H
"Authorization: auth_token <auth_token>" -d
'{"datum":{"key":"street","value":"value" }}' https://<device service url>/apiv1/dsns/<device-id>/data.json
```

Output :

```
{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"street","updated_at":"2014-02-19T23:42:46Z","value":"street value","dsn":"123"}}
```

Response

200 – Success

5.4 PUT /apiv1/dsns/:dsn/data/:key[.format]

Updates a datum with the provided parameters.

Input:**Mandatory:**

Dsn: Device dsn

Type: String **data:** A hash with the datum to be updated:

key: a datum key

Type: String

value: datum value (blank is acceptable)

Type: String

Validation:Text max - 2 MB

For binary file - base64 encode is set to a string

XML Example:

```
$ curl -X PUT -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/dsns/<device-dsn>/data/street.xml -d
"<datum><key>street</key><value>street value
updated</value></datum>"
```

JSON Example:

```
CURL -X PUT -H"Authorization: auth_token <auth_token>"
-d '{"datum":{"key":"street","value":"street value updated"}}'
https://<device service url>/apiv1/dsns/<device-dsn>/data/street.json
```

Output:

```
{"datum":{"created_at":"2014-02-19T23:42:46Z","key":"street","updated_at":"2014-02-19T23:42:50Z","value":"street value updated","dsn":"123"}}
```

Response:

- 200 – Success
- 422 – Unprocessable entity
- 404 – Not found, the data doesn't ex

5.5 DELETE /apiv1/dsns/:dsn/data/:key[.format]

Destroys a datum.

Input:**Mandatory:**

- Dsn : Device dsn obtained from GET /apiv1/devices
 - Type: String
- key: a datum key
 - Type: String
 - Validation: Limit: 255 bytes

XML Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth_token>"
https://<Device service URL>/apiv1/dsns/<device-dsn>/data/street.xml
```

JSON Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/dsns/<device-dsn>/data/street.json
```

Response

- 204 – Success
- 404 – Not found, the datum doesn't exist

6 PROPERTIES

6.1 GET *apiv1/devices/<device_key>/properties.xml*

Retrieves all the properties for a particular device.

Input:

Mandatory:

device_key: the device id
Type: Integer

Optional:

names: Property name.
Type: String
Validation: A subset of properties
instead of a subset, a pattern like "file%".
The format of the pattern is as follows:

- The "%25" sign is used to define wildcards both before and after the pattern.
- To select all properties ending with the letter "s": %25s
- To select all properties beginning with the letter "s": s%
- To select all properties containing the pattern "input": %input%

Output: array of properties

XML Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device service url>/apiv1/devices/<device-id>/properties.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<properties type="array">
  <property>
    <name>Blue_LED</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">>false</read-only>
    <direction>input</direction>
    <data-updated-at type="datetime">2014-11-
14T22:35:36Z</data-updated-at>
    <key type="integer">92</key>
    <value type="integer">1</value>
    <device-key type="integer">11</device-key>
    <product-name>proto 1</product-name>
```



```

</property>
<property>
  <name>button</name>
  <base-type>boolean</base-type>
  <read-only type="boolean">true</read-only>
  <direction>output</direction>
  <data-updated-at type="datetime">2014-11-
14T22:23:49Z</data-updated-at>
  <key type="integer">57</key>
  <value type="integer">0</value>
  <device-key type="integer">11</device-key>
  <product-name>proto 1</product-name>
</property>
<property>
  <name>Green_LED</name>
  <base-type>boolean</base-type>
  <read-only type="boolean">>false</read-only>
  <direction>input</direction>
  <data-updated-at type="datetime">2014-11-
14T22:35:41Z</data-updated-at>
  <key type="integer">93</key>
  <value type="integer">1</value>
  <device-key type="integer">11</device-key>
  <product-name>proto 1</product-name>
</property>
</properties>

```

```

$ curl -g -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/devices/1/properties.xml?names\\[\\\]=green\_led&names\\[\\\]=blue\_led

```

Or:

```

curl -g -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/devices/1/properties.xml?names=%led

```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<properties type="array">
  <property>
    <name>Blue_LED</name>

```

```

    <base-type>boolean</base-type>
    <read-only type="boolean">true</read-only>
    <direction>output</direction>
    <scope>user</scope>
    <data-updated-at nil="true"></data-updated-at>
    <key type="integer">1</key>
    <device-key type="integer">1</device-key>
    <product-name nil="true"></product-name>
    <track-only-changes type="boolean">false</track-only-
changes>
    <display-name nil="true"></display-name>
    <value nil="true"></value>
  </property>
  <property>
    <name>Green_LED</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">true</read-only>
    <direction>input</direction>
    <scope>user</scope>
    <data-updated-at type="datetime">2013-09-
25T14:27:30Z</data-updated-at>
    <key type="integer">6</key>
    <device-key type="integer">1</device-key>
    <product-name nil="true"></product-name>
    <track-only-changes type="boolean">false</track-only-
changes>
    <display-name nil="true"></display-name>
    <value type="integer">0</value>
  </property>
</properties>

```

JSON Example:

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

Response:

```

[
  {
    "property" : {
      "base_type" : "boolean",
      "value" : 1,
      "data_updated_at" : "2011-11-14T22:35:36Z",

```

```

        "device_key" : 11,
        "name" : "Blue_LED",
        "key" : 92,
        "direction" : "input",
        "read_only" : false,
        "product_name" : "proto 1"
    }
},
{
    "property" : {
        "base_type" : "boolean",
        "value" : 0,
        "data_updated_at" : "2014-11-14T22:23:49Z",
        "device_key" : 11,
        "name" : "button",
        "key" : 57,
        "direction" : "output",
        "read_only" : true,
        "product_name" : "proto 1"
    }
},
{
    "property" : {
        "base_type" : "boolean",
        "value" : 1,
        "data_updated_at" : "201411-14T22:35:41Z",
        "device_key" : 11,
        "name" : "Green_LED",
        "key" : 93,
        "direction" : "input",
        "read_only" : false,
        "product_name" : "proto 1"
    }
}
]

```

6.2 GET *apiv1/dsns/<device_dsns>/properties [.format]*

Retrieves all the properties for a particular device with its dsns.

Input:

Mandatory:

device_dsns: the device dsns.

Type: String

Output: array of properties

XML Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/dsns/AACXXXXXXXXXXXXX/properties.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<properties type="array">
  <property>
    <name>Blue_LED</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">false</read-only>
    <direction>input</direction>
    <data-updated-at type="datetime">2014-11-
14T22:35:36Z</data-updated-at>
    <key type="integer">92</key>
    <value type="integer">1</value>
    <device-key type="integer">11</device-key>
    <product-name>proto 1</product-name>
  </property>
  <property>
    <name>button</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">true</read-only>
    <direction>output</direction>
    <data-updated-at type="datetime">2011-11-
14T22:23:49Z</data-updated-at>
    <key type="integer">57</key>
    <value type="integer">0</value>
    <device-key type="integer">11</device-key>
    <product-name>proto 1</product-name>
  </property>
  <property>
    <name>Green_LED</name>
    <base-type>boolean</base-type>
    <read-only type="boolean">false</read-only>
    <direction>input</direction>
    <data-updated-at type="datetime">2011-11-
14T22:35:41Z</data-updated-at>
    <key type="integer">93</key>
    <value type="integer">1</value>
    <device-key type="integer">11</device-key>
```

```
<product-name>proto 1</product-name>
</property>
</properties>
```

JSON Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device service url>/apiv1/dsns/AACXXXXXXXXXXXXX/properties.json
```

Response:

```
[
  {
    "property" : {
      "base_type" : "boolean",
      "value" : 1,
      "data_updated_at" : "2011-11-14T22:35:36Z",
      "device_key" : 11,
      "name" : "Blue_LED",
      "key" : 92,
      "direction" : "input",
      "read_only" : false,
      "product_name" : "proto 1"
    }
  },
  {
    "property" : {
      "base_type" : "boolean",
      "value" : 0,
      "data_updated_at" : "2011-11-14T22:23:49Z",
      "device_key" : 11,
      "name" : "button",
      "key" : 57,
      "direction" : "output",
      "read_only" : true,
      "product_name" : "proto 1"
    }
  },
  {
    "property" : {
      "base_type" : "boolean",
      "value" : 1,
      "data_updated_at" : "2011-11-14T22:35:41Z",
      "device_key" : 11,
      "name" : "Green_LED",
```

```

        "key" : 93,
        "direction" : "input",
        "read_only" : false,
        "product_name" : "proto 1"
    }
}
]

```

6.3 GET /apiv1/properties/<key>[.format]

Retrieves details for the property corresponding to the property key.

Other supported APIs:

GET /apiv1/devices/:device_id/properties/:property_name(.:format)

GET /apiv1/dsns/:dsn/properties/:property_name(.:format)

Input:

Mandatory:

Key: property id obtained via GET of all device properties

Type: Integer

Output: property object

XML Examples:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/properties/<property-id>.xml
```

or

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/devices/<device-id>/properties/blue_led.xml
```

or

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/dsns/ACXXXXXXXXXXXXX/properties/blue_led.xml
```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<property>
<name>Blue_LED</name>
<base-type>boolean</base-type>
<read-only type="boolean">false</read-only>
<direction>input</direction>
<key type="integer">92</key>
<device-key type="integer">11</device-key>
<value type="integer">1</value>

```

```
<product-name>proto 1</product-name>
</property>
```

JSON Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/properties/<prop-id>.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"
  }
}
```

6.4 GET /properties/<property_key>/datapoints[.format]

Retrieves the last count data points for the property specified by the property key. If limit is not specified then service returns last 100 data points.

Input:**Mandatory:**

property_key: property id obtained via GET of all device properties
Type: Integer

Optional:

per_page: the number of datapoints requested per page when not paginated.

Type: Integer

Validation: should be less than 100

Default: 100

next: The id of the last datapoint in current page, available in the <metadata> section of the previous response.

Type: String

previous: The id of the first datapoint in current page, available in the <metadata> section of the previous response.

Type: String

is_forward_page: true. For first page or next page request, set it to true, else set it to false.

Type: String

paginated: true. If request paginated response, set it to true and provide 'limit' as the number of datapoints per page.

Type: String

filter: params within filter can filter datapoints based on request:

created_at_since_date: set time restriction for datapoints created after

Type: time

created_at_end_date: set time restriction for datapoints created before

Type: Time

Output: An array of datapoint objects in time ascending order.

XML Example:

Paginated:

1. First page request and response

```
curl -v -H "Authorization: auth_token <auth_token>"
https://<device\_service\_url>/apiv1/properties/4/datapoints.xml?paginated=true&is\_forward\_page=true&per\_page=3
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <meta>
    <previous-page>20bc1470-a71b-11e4-b170-8153baf3c000</previous-page>
    <next-page>d64114d0-a59f-11e4-ab79-5bf859e6955d</next-page>
    <current-page-number type="integer">1</current-page-number>
  </meta>
  <datapoints type="array">
```



```

    <datapoint>
      <updated-at type="datetime">2015-01-
26T21:11:04Z</updated-at>
      <created-at type="datetime">2015-01-
26T21:11:04Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id>d64114d0-a59f-11e4-ab79-5bf859e6955d</id>
    </datapoint>
    <datapoint>
      <updated-at type="datetime">2015-01-
28T18:24:42Z</updated-at>
      <created-at type="datetime">2015-01-
28T18:24:42Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id>ed8e6d00-a71a-11e4-b170-8153baf3c000</id>
    </datapoint>
    <datapoint>
      <updated-at type="datetime">2015-01-
28T18:26:08Z</updated-at>
      <created-at type="datetime">2015-01-
28T18:26:08Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id>20bc1470-a71b-11e4-b170-8153baf3c000</id>
    </datapoint>
  </datapoints>
</result>

```

JSON Example:

```

curl -v -H "Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/properties/4/datapoints.json?paginated=true&is\_forward\_page=true&per\_page=3

```

```

{
  "meta": {
    "previous_page": "20bc1470-a71b-11e4-b170-XXXXXX",
    "next_page": "d64114d0-a59f-11e4-ab79-5bf8XXXXXX",
    "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-ab79-5bf859eXXXX"
        }
    },
    {
        "datapoint": {
            "updated_at": "2015-01-28T18:24:42Z",
            "created_at": "2015-01-28T18:24:42Z",
            "echo": false,
            "value": 65.353454,
            "id": "ed8e6d00-a71a-11e4-b170-8153baf3XXXX"
        }
    },
    {
        "datapoint": {
            "updated_at": "2015-01-28T18:26:08Z",
            "created_at": "2015-01-28T18:26:08Z",
            "echo": false,
            "value": 65.353454,
            "id": "20bc1470-a71b-11e4-b170-8153baf3c000"
        }
    }
]
}

```

2. Next page from current page:

XML

```
curl -v -H "Authorization: auth_token <auth_token> https://<device
service
url>/apiv1/properties/4/datapoints.xml?paginated=true&is_forward_p
age=true&per_page=3&next=<auth_token>
```

Note: next= "next_page" in "meta" section in current page response.

```

<result>
  <meta>
    <previous-page><xxxxx> </previous-page>
    <next-page><xxxxx></next-page>
    <current-page-number type="integer">1</current-page-number>

```

```

</meta>
<datapoints type="array">
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:03:35Z</updated-at>
    <created-at type="datetime">2015-01-26T21:03:35Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id><id></id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:09:25Z</updated-at>
    <created-at type="datetime">2015-01-26T21:09:25Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id><id></id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:10:15Z</updated-at>
    <created-at type="datetime">2015-01-26T21:10:15Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id>b91e6ec0-a59f-11e4-ab79-5bf859e6955d</id>
  </datapoint>
</datapoints>
</result>

```

JSON

```

curl -v -H "Authorization: auth_token <auth_token>"
"https://<device
service>/apiv1/properties/4/datapoints.json?paginated=true&is_forw
ard_page=true&per_page=3&next=xxxxxx"

```

Note: next= "next_page" in "meta" section in current page response.

```

{
  "meta": {
    "previous_page": "<xxxxx>",
    "next_page": "<xxxxx>",
    "current_page_number": 1
  },
  "datapoints": [
    {
      "datapoint": {
        "updated_at": "2015-01-26T21:03:35Z",

```

```

        "created_at": "2015-01-26T21:03:35Z",
        "echo": false,
        "value": 65.353454,
        "id": "<id>"
      }
    },
    {
      "datapoint": {
        "updated_at": "2015-01-26T21:09:25Z",
        "created_at": "2015-01-26T21:09:25Z",
        "echo": false,
        "value": 65.353454,
        "id": "<id>"
      }
    },
    {
      "datapoint": {
        "updated_at": "2015-01-26T21:10:15Z",
        "created_at": "2015-01-26T21:10:15Z",
        "echo": false,
        "value": 65.353454,
        "id": "<id>"
      }
    }
  ]
}

```

3. Previous page from current page

XML

```

curl -v -H "Authorization: auth_token <auth_token>"
"https://<device service
url>/apiv1/properties/4/datapoints.xml?paginated=true&is_forward_p
age=false&per_page=3&previous=xxxxxxx"

```

Note: next= "previous_page" in "meta" section in current page response.
is_forward_page need to be set to false.

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <meta>
    <previous-page><xxxxxx></previous-page>
    <next-page><xxxxxx></next-page>
    <current-page-number type="integer">1</current-page-number>
  
```

```

</meta>
<datapoints type="array">
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:11:04Z</updated-at>
    <created-at type="datetime">2015-01-26T21:11:04Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id><xxxxx></id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-28T18:24:42Z</updated-at>
    <created-at type="datetime">2015-01-28T18:24:42Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id><id></id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-28T18:26:08Z</updated-at>
    <created-at type="datetime">2015-01-28T18:26:08Z</created-at>
    <echo type="boolean">false</echo>
    <value type="float">65.353454</value>
    <id>2</id>
  </datapoint>
</datapoints>
</result>

```

JSON

```

curl -v -H "Authorization: auth_token <auth_token>"
"https://<device service
url>/apiv1/properties/4/datapoints.json?paginated=true&is_forward
page=false&per_page=3&previous=xxxxxxx"

```

Note: next= "previous_page" in "meta" section in current page response.
is_forward_page need to be set to false.

```

{
  "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": "<id>"
    }
},
{
    "datapoint": {
        "updated_at": "2015-01-28T18:24:42Z",
        "created_at": "2015-01-28T18:24:42Z",
        "echo": false,
        "value": 65.353454,
        "id": "<id>"
    }
},
{
    "datapoint": {
        "updated_at": "2015-01-28T18:26:08Z",
        "created_at": "2015-01-28T18:26:08Z",
        "echo": false,
        "value": 65.353454,
        "id": "<id>"
    }
}
]
}

```

Paginated with time range restriction query:

add filter params like below, everything else stays the same as above normal pagination.

1 . First page request and response

XML:

```

curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device
service
url>/apiv1/properties/4/datapoints.xml?paginated=true&is_forward_p
age=true&per_page=3"

```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <meta>

```

```

<previous-page>20bc1470-a71b-11e4-b170-8153baf3c000</previous-page>
<next-page>d64114d0-a59f-11e4-ab79-5bf859e6955d</next-page>
<current-page-number type="integer">1</current-page-number>
</meta>
<datapoints type="array">
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:11:04Z</updated-at>
    <created-at type="datetime">2015-01-26T21:11:04Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>d64114d0-a59f-11e4-ab79-5bf859e6955d</id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-28T18:24:42Z</updated-at>
    <created-at type="datetime">2015-01-28T18:24:42Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>ed8e6d00-a71a-11e4-b170-8153baf3c000</id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-28T18:26:08Z</updated-at>
    <created-at type="datetime">2015-01-28T18:26:08Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>20bc1470-a71b-11e4-b170-8153baf3c000</id>
  </datapoint>
</datapoints>
</result>

```

JSON

```

curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device
service
url>/apiv1/properties/4/datapoints.json?paginated=true&is_forward_
page=true&per_page=3"

```

```

{
  "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"
  }
},
{
  "datapoint": {
    "updated_at": "2015-01-28T18:24:42Z",
    "created_at": "2015-01-28T18:24:42Z",
    "echo": false,
    "value": 65.353454,
    "id": "ed8e6d00-a71a-11e4-b170-8153baf3c000"
  }
},
{
  "datapoint": {
    "updated_at": "2015-01-28T18:26:08Z",
    "created_at": "2015-01-28T18:26:08Z",
    "echo": false,
    "value": 65.353454,
    "id": "20bc1470-a71b-11e4-b170-8153baf3c000"
  }
}
]
```

2. Next page from current page:

XML

```
curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device
service
url>/apiv1/properties/4/datapoints.xml?paginated=true&is_forward_p
age=true&per_page=3&next=<auth_token>"
```

Note: next= “next_page” in “meta” section in current page response.

<result>

<meta>


```

<previous-page>b91e6ec0-a59f-11e4-ab79-5bf859e6955d</previous-page>
<next-page>cb036290-a59e-11e4-9cee-5bf859e6955d</next-page>
<current-page-number type="integer">1</current-page-number>
</meta>
<datapoints type="array">
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:03:35Z</updated-at>
    <created-at type="datetime">2015-01-26T21:03:35Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>cb036290-a59e-11e4-9cee-5bf859e6955d</id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:09:25Z</updated-at>
    <created-at type="datetime">2015-01-26T21:09:25Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>9b485730-a59f-11e4-a39e-5bf859e6955d</id>
  </datapoint>
  <datapoint>
    <updated-at type="datetime">2015-01-26T21:10:15Z</updated-at>
    <created-at type="datetime">2015-01-26T21:10:15Z</created-at>
    <echo type="boolean">>false</echo>
    <value type="float">65.353454</value>
    <id>b91e6ec0-a59f-11e4-ab79-5bf859e6955d</id>
  </datapoint>
</datapoints>
</result>

```

JSON

```

curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device
service
url>/apiv1/properties/4/datapoints.json?paginated=true&is_forward_
page=true&per_page=3&next=<auth_token>"

```

Note: next= "next_page" in "meta" section in current page response.

```

{
  "meta": {
    "previous_page": "b91e6ec0-a59f-11e4-ab79-5bf859e6955d",
    "next_page": "cb036290-a59e-11e4-9cee-5bf859e6955d",
    "current_page_number": 1
  }
}

```

```

},
"datapoints": [
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:03:35Z",
      "created_at": "2015-01-26T21:03:35Z",
      "echo": false,
      "value": 65.353454,
      "id": "cb036290-a59e-11e4-9cee-5bf859e6955d"
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:09:25Z",
      "created_at": "2015-01-26T21:09:25Z",
      "echo": false,
      "value": 65.353454,
      "id": "9b485730-a59f-11e4-a39e-5bf859e6955d"
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:10:15Z",
      "created_at": "2015-01-26T21:10:15Z",
      "echo": false,
      "value": 65.353454,
      "id": "b91e6ec0-a59f-11e4-ab79-5bf859e6955d"
    }
  }
]
}

```

3. Previous page from current page

XML

```

curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device service url>/apiv1/properties/4/datapoints.xml?paginated=true&is\_for\_ward\_page=false&per\_page=3&previous=b91e6ec0-a59f-11e4-ab79-5bf859e6955d"

```

Note: next= "previous_page" in "meta" section in current page response.
is_forward_page need to be set to false.

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <meta>
    <previous-page>20bc1470-a71b-11e4-b170-8153baf3c000</previous-page>
    <next-page>d64114d0-a59f-11e4-ab79-5bf859e6955d</next-page>
    <current-page-number type="integer">1</current-page-number>
  </meta>
  <datapoints type="array">
    <datapoint>
      <updated-at type="datetime">2015-01-26T21:11:04Z</updated-at>
      <created-at type="datetime">2015-01-26T21:11:04Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id>d64114d0-a59f-11e4-ab79-5bf859e6955d</id>
    </datapoint>
    <datapoint>
      <updated-at type="datetime">2015-01-28T18:24:42Z</updated-at>
      <created-at type="datetime">2015-01-28T18:24:42Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id>ed8e6d00-a71a-11e4-b170-8153baf3c000</id>
    </datapoint>
    <datapoint>
      <updated-at type="datetime">2015-01-28T18:26:08Z</updated-at>
      <created-at type="datetime">2015-01-28T18:26:08Z</created-at>
      <echo type="boolean">>false</echo>
      <value type="float">65.353454</value>
      <id><id></id>
    </datapoint>
  </datapoints>
</result>
```

JSON

```
curl -v -F filter\[created_at_start_date\]='2015-01-05 00:00:00'
-F filter\[created_at_end_date\]='2015-05-07 00:00:00' -X GET
-H "Authorization: auth_token <auth_token>" "https://<device
service
url>/apiv1/properties/4/datapoints.json?paginated=true&is_forward_
page=false&per_page=3&previous=<auth_token>"
```

Note: next= "previous_page" in "meta" section in current page response.
is_forward_page need to be set to false.

```
{
  "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"
      }
    },
    {
      "datapoint": {
        "updated_at": "2015-01-28T18:24:42Z",
        "created_at": "2015-01-28T18:24:42Z",
        "echo": false,
        "value": 65.353454,
        "id": "ed8e6d00-a71a-11e4-b170-8153baf3c000"
      }
    },
    {
      "datapoint": {
        "updated_at": "2015-01-28T18:26:08Z",
        "created_at": "2015-01-28T18:26:08Z",
        "echo": false,
        "value": 65.353454,
        "id": "20bc1470-a71b-11e4-b170-8153baf3c000"
      }
    }
  ]
}
```

Non-paginated:

XML

```
curl -v -H "Authorization: auth_token <auth_token>"  
"https://<device service  
url>/apiv1/properties/4/datapoints.xml?limit=5"
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<datapoints type="array">  
  <datapoint>  
    <updated-at type="datetime">2015-01-26T21:09:25Z</updated-at>  
    <created-at type="datetime">2015-01-26T21:09:25Z</created-at>  
    <echo type="boolean">false</echo>  
    <value type="float">65.353454</value>  
  </datapoint>  
  <datapoint>  
    <updated-at type="datetime">2015-01-26T21:10:15Z</updated-at>  
    <created-at type="datetime">2015-01-26T21:10:15Z</created-at>  
    <echo type="boolean">false</echo>  
    <value type="float">65.353454</value>  
  </datapoint>  
  <datapoint>  
    <updated-at type="datetime">2015-01-26T21:11:04Z</updated-at>  
    <created-at type="datetime">2015-01-26T21:11:04Z</created-at>  
    <echo type="boolean">false</echo>  
    <value type="float">65.353454</value>  
  </datapoint>  
  <datapoint>  
    <updated-at type="datetime">2015-01-28T18:24:42Z</updated-at>  
    <created-at type="datetime">2015-01-28T18:24:42Z</created-at>  
    <echo type="boolean">false</echo>  
    <value type="float">65.353454</value>  
  </datapoint>  
  <datapoint>  
    <updated-at type="datetime">2015-01-28T18:26:08Z</updated-at>  
    <created-at type="datetime">2015-01-28T18:26:08Z</created-at>  
    <echo type="boolean">false</echo>  
    <value type="float">65.353454</value>  
  </datapoint>  
</datapoints>
```

JSON

```
curl -v -H "Authorization: auth_token <auth_token>"<auth_token>  
"https://<device service  
url>/apiv1/properties/4/datapoints.json?limit=5"
```

```
[
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:09:25Z",
      "created_at": "2015-01-26T21:09:25Z",
      "echo": false,
      "value": 65.353454
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:10:15Z",
      "created_at": "2015-01-26T21:10:15Z",
      "echo": false,
      "value": 65.353454
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-26T21:11:04Z",
      "created_at": "2015-01-26T21:11:04Z",
      "echo": false,
      "value": 65.353454
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-28T18:24:42Z",
      "created_at": "2015-01-28T18:24:42Z",
      "echo": false,
      "value": 65.353454
    }
  },
  {
    "datapoint": {
      "updated_at": "2015-01-28T18:26:08Z",
      "created_at": "2015-01-28T18:26:08Z",
      "echo": false,
      "value": 65.353454
    }
  }
]
```

6.4.1 POST apiv1/properties/<property_key>/datapoints[.format]

Creates one datapoint for a property specified by the property key.

Input:

Mandatory:

Property_key: id of the property
datapoint value: datapoints object specifying new value

Optional:

Meta Data: Key, Value:

Each datapoint may also contain a "metadata" field containing key/value pairs. No more than 10 metadata entries per datapoint are supported.

Validations:

Boolean: 0 or 1

Integer value range: -2147483648 to 2147483647

String Value: Max length 1024 bytes

Decimal value range: -21474836.48 to 21474836.47. Up to 2 decimal places supported.

XML Example for a single datapoint:

```
$ curl -X POST -H"Authorization: auth_token
<auth_token>"<datapoint><value>65</value></datapoint>" -H
"Content-Type: application/xml" https://<device service
url>/apiv1/properties/<prop\_id>/datapoints.xml
```

Response:

in case the property has base type:

```
<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
<created-at type="datetime">2011-11-15T01:54:56Z</created-at>
<updated-at type="datetime">2011-11-15T01:54:56Z</updated-at>
<value type="integer">65</value>
</datapoint>
```

or

XML Example for Posting a single datapoint using property name:

```
$ curl -X POST -H"Authorization: auth_token
<auth_token>"<datapoint><value>65</value></datapoint>" -H
"Content-Type: application/xml" https://<device service
url>/apiv1/dsns/AC000W0001XXXX/properties/blueled/datapoints.xml
1
```

Response:

in case the property has base type:

```
<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
<created-at type="datetime">2014-11-15T01:54:56Z</created-at>
<updated-at type="datetime">2014-11-15T01:54:56Z</updated-at>
<value type="integer">65</value>
</datapoint>
```

JSON Example for a single datapoint:

```
$ curl -X POST -H"Authorization: auth_token <auth_token>" -
H"Content-Type: application/json" -d '{
"datapoint":{"value":0}}' https://<device service
url>/apiv1/properties/<property\_id>/datapoints.json
```

Response:

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

XML Example for creating datapoint with key value pair:

```
$ curl -X POST -H"Authorization: auth_token <auth_token>" -d
"<datapoint><value>65</value><metadata key1="value1"
key2="value2"></metadata></datapoint>" -H "Content-Type:
application/xml" https://<device service
url>/apiv1/dsns/<device-
dsn>/properties/blueled/datapoints.xml
```

Response

in case the property has base type:

```
<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
<created-at type="datetime">2011-11-15T01:54:56Z</created-at>
<updated-at type="datetime">2011-11-15T01:54:56Z</updated-at>
<value type="integer">65</value>
<metadata key1="value1" key2="value2"></metadata>
```



```
</datapoint>
```

```
$ curl -X POST -H"Authorization: auth_token <device service url>" -d "<datapoint><value>65</value><metadata key1='value1' key2='value2'></metadata></datapoint>" -H "Content-Type: application/xml" https://<device service url>/apiv1/properties/33/datapoints.xml
```

Response

in case the property has base type:

```
<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
<created-at type="datetime">2011-11-15T01:54:56Z</created-at>
<updated-at type="datetime">2011-11-15T01:54:56Z</updated-at>
<value type="integer">65</value>
<metadata key1="value1" key2="value2"></metadata>
</datapoint>
```

JSON Example for a single datapoint:

```
$ curl -X POST -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" -d '{
"datapoint":{"value":0, "metadata": {"key1": "value1",
"key2": "value2"}}}' https://<device service url>/apiv1/properties/<property-id>/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"
    }
  }
}
```

Response:

201 – Success, the created datapoint/s is returned

6.5 **POST** *apiv1/dsns/:dsn/properties/:property_name/datapoints[.format]*

6.6 **GET** */apiv1/properties/:property_id/trigger_apps.[format]*

Retrieves trigger apps for the property corresponding to id <key> that could be useful in application libraries.

Input:

Mandatory:

property_id: is property key obtained via GET of all device properties.

Type: Integer

Output: trigger app objects

XML Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device service url>/apiv1/properties/<property id>/trigger\_apps.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<trigger-apps type="array">
  <trigger-app>
    <name>email</name>
    <username>bob</username>
    <param1>bob@acme.com</param1>
    <param2 nil="true"></param2>
    <param3 nil="true"></param3>
    <param4 nil="true"></param4>
    <param5 nil="true"></param5>
    <key type="integer">27</key>
    <trigger-key type="integer">42</trigger-key>
  </trigger-app>
  <trigger-app>
    <name>sms</name>
    <username>Bob</username>
    <param1>1</param1>
    <param2>4085551111</param2>
    <param3>Button pressed</param3>
    <param4 nil="true"></param4>
    <param5 nil="true"></param5>
    <key type="integer">28</key>
```

```
<trigger-key type="integer">42</trigger-key>
</trigger-app>
</trigger-apps>
```

JSON Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device service url>/apiv1/properties/< property\_id >/trigger\_apps.json
```

Response:

```
[
  {
    "trigger_app": {
      "name": "sms",
      "param1": "1",
      "param2": "4085551111",
      "param3": "Hi. Pushbutton event",
      "param4": null,
      "param5": null,
      "username": null,
      "key": 263,
      "trigger_key": 294
    }
  },
  {
    "trigger_app": {
      "name": "email",
      "param1": "bob@acme.com",
      "param2": null,
      "param3": null,
      "param4": null,
      "param5": null,
      "username": "Bob",
      "key": 264,
      "trigger_key": 294
    }
  }
]
```

6.7 FILE PROPERTIES

This section covers resumable upload and download of file properties to the Ayla service from the service API.

6.7.1 Creating a Datapoint

For file properties, a datapoint is created with a POST command on the datapoints resource. This returns a file element that corresponds to a URL for the newly created datapoint where you can upload the file. The URL has an expiry time of five minutes, to upload the actual file.

XML Example:

```
$ curl -k -X POST -H "Authorization: auth_token <token>" -H
"Content-Type: application/xml" -d '<datapoint></datapoint>'
https://<device service url>/apiv1/properties/< property id
>/datapoints.xml

<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
  <updated-at type="datetime">2015-01-02T13:50:01Z</updated-at>
  <created-at type="datetime">2015-01-02T13:50:01Z</created-at>
  <echo type="boolean">false</echo>
  <fetch-at nil="true"></fetch-at>
  <closed nil="true"></closed>
  <location>https://<device service
url>/apiv1/devices/27686/properties/file_down/datapoints/1063380</location>
  <file>https://<device service url>/datapoint-
1063380?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1420210202&a
mp;Signature=jSU9fBH%2F07DJ%2Bj8rWA7PYbvvuFU%3D</file>
</datapoint>
```

JSON Example:

```
$ curl -k -X POST -H "Authorization: auth_token <token>" -H
"Content-Type: application/json" -d '{"datapoint":{}}'
https://<device service url>/apiv1/properties/< property id
>/datapoints.json
{
  "datapoint": {
    "updated_at": "2014-12-31T07:40:59Z",
    "created_at": "2014-12-31T07:40:59Z",
    "echo": false,
    "fetch_at": null,
    "closed": null,
    "location": "https://<device service
url>/apiv1/devices/27686/properties/file_down/datapoints/1063224",
```

```

    "file": "https://<device service url>/datapoint-
1063224?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1420015259&Signatu
re=5AbcDnRIMz3lFzwK0Pf87yJzHkE%3D"
  }
}

```

6.7.2 Uploading Data to the Datapoint

For uploading data to a datapoint, use the file URL returned in the file element when creating a datapoint.

```

$ curl -X PUT -v -T /home/Documents/sample-file.pdf -H "Content-
Type: application/octet-stream" "https://<device service
url>/datapoint-
1063224?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1420015259&Sig
nature=5AbcDnRIMz3lFzwK0Pf87yJzHkE%3D"

```

If the URL has expired, use a PUT to get a new URL, as it shown in the following example:

XML Example:

```

$ curl -k -X PUT -H "Authorization: auth_token <auth_token>" -H
"Content-Type: application/json" -d '<datapoint></datapoint>'
https://<device service url>/apiv1/datapoints/1063224.xml

<?xml version="1.0" encoding="UTF-8"?>
<datapoint>
  <updated-at type="datetime">2014-12-31T07:40:59Z</updated-at>
  <created-at type="datetime">2014-12-31T07:40:59Z</created-at>
  <echo type="boolean">>false</echo>
  <fetch-at nil="true"></fetch-at>
  <closed nil="true"></closed>
  <location>https://<device service
url>/apiv1/devices/27686/properties/file_down/datapoints/1063224</location>
  <file>https://<device service url>/datapoint-
1063224?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1420211300&a
mp;Signature=OHCTKzfELQgTj9Vh4bPqkaxd08w%3D</file>
</datapoint>

```

JSON Example:

```
$ curl -k -X PUT -H "Authorization: auth_token <token>" -H
"Content-Type: application/json" -d '{"datapoint":{}}'
https://<device service url>/apiv1/datapoints/1063224.json
{
  "datapoint": {
    "updated_at": "2014-12-31T07:40:59Z",
    "created_at": "2014-12-31T07:40:59Z",
    "echo": false,
    "fetched_at": null,
    "closed": null,
    "location": "https://<device service
url>/apiv1/devices/27686/properties/file_down/datapoints/1063224",
    "file": "https://<device service url>/datapoint-
1063224?AWSAccessKeyId=AKIAJAR3VCJIAFO6AUDA&Expires=1420015626&Signatu
re=RG7NiGsielLxoTsiwekTsfUIhmQ%3D"
  }
}
```

6.7.3 Marking a Datapoint as Complete

If data was uploaded using multiple PUT requests and the total number of bytes was not known, an explicit call should be made to mark a datapoint as being complete. This is done with 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.

URL: <location>
 Method: PUT
 Format: n/a

Response:

200 – Success:
 Failure: all other status codes

Curl Example:

```
curl -x PUT https://<device service
url>/apiv1/datapoints/<datapoint-id>
```

6.7.4 Marking a Datapoint as Discarded

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

URL: <location>

Method: PUT

Format: n/a

Request Payload: <datapoint><discard>true</discard></datapoint> or corresponding json

Response:

200 – Success

Failure: all other status codes

XML Example:

```
curl -v -k -X PUT -H"Content-Type: application/xml" -  
d'<datapoint><discard>true</discard></datapoint>' https://<device  
service url>/apiv1/datapoints/9396.xml
```

JSON Example:

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

6.7.5 Downloading Data

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

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:

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=142186314
8&Signature=drHshNV%2F2CVXbbhJenyONAp89dE%3D"}}}
```

6.7.6 Marking a Datapoint as Fetched

When the client is done downloading the whole data file, it can indicate that the datapoint is fetched with an explicit call. In the absence of this, the datapoint is treated as yet unfetched.

URL: <location>

Method: PUT

Format: n/a

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

XML Example:

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

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:

200 – Success

406 – if datapoint is incomplete

all other status codes

7 GROUPS

7.1 GET /apiv1/groups

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.

JSON Example:

```
curl -H "Authorization: auth_token <auth_token>" https://<device service url>/apiv1/groups.json
```

Response:

```
[
  {
    "group": {
      "name": "asdf",
      "key": 1,
      "device_count": 3,
    }
  },
  {
    "group": {
      "name": "test_group",
      "key": 3, "device_count": 2,
    }
  }
]
```

7.2 POST /apiv1/groups

Creates a new group for the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.

Input:

Mandatory:

Name: A name for the group

Type: string

Device Id: Device ids as an array

Type: Integer

JSON Example:

```
curl -k -X POST -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" -d{"group": {"name": "test_group123", "devices": {"device_id": ["35"]}}}" https://<device service url>/apiv1/groups.json
```

Response:

200 – Success

422 – Failure, unprocessable entity

```
:
{
  "group":{
    "name":"test_group123",
    "key":13,
    "device_count":2,
    "devices":[
      { "product_name":"V4RZXEX7OY36",
        "dsn":"AC000W000000XXX",
        "oem_model":null,
        "key":3,
        "connection_status":"Online"
      },
      { "product_name":"EJXFEBRSTG26",
        "dsn":"AC000W000000XXXX",
        "oem_model":null,
        "key":5,
        "connection_status":"Offline"
      }
    ]
  }
}
```

7.3 *GET /apiv1/groups/:group_id*

Retrieves a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.

Input:

Mandatory:

Group id : Get key (Id) from POST /apiv1/groups

Type: Integer

JSON Example:

```
curl -H"Authorization: auth_token <auth_token>" https://<device
service url>/apiv1/groups/<group-id>.json
```

Response:

```
{
  "group":{
    "name":"ctrl1",
    "key":12,
```

```

    "device_count":2,
    "devices":[
      {
        "product_name":"asdf",
        "dsn":"AC000W000000XXXX",
        "oem_model":null,
        "key":12
      },
      {
        "product_name":"asdf",
        "dsn":"AC000W000000XXXX",
        "oem_model":null,
        "key":17,
        "connection_status":"Online"
      }
    ]
  }
}

```

7.4 PUT /apiv1/groups/:group_id

Modifies a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.

Input:

Mandatory:

Group ID: Get key (Id) from POST /apiv1/groups
Type: Integer

Optional:

Group name: if modification is desired
Type: String

JSON Example:

```

curl -k -X PUT -H"Authorization: auth_token <auth_token>" -
H"Content-Type: application/json" -d{"group": { "name":
"new_group123" }} "https://<device service
url>/apiv1/groups/<group-id>.json"

```

Response:

```

{
  "group":{
    "name":"new_group123",
    "key":13,
    "device_count":1,
    "devices":[

```

```
{
  "product_name": "W5PSFZXZXXXX",
  "dsn": "AC000W000000XXXX",
  "oem_model": null,
  "key": 8,
  "connection_status": "Online"
}
]
```

Response:

200 – Success

422 – Failure

7.5 **DELETE** /apiv1/groups/:group_id

Destroys a group belonging to the given user. A user is indicated by the auth_token in the authorization header of the HTTP request.

Input:**Mandatory:**

Group id: Get key (Id) from POST /apiv1/groups

Type: Integer

JSON Example:

```
curl -i -X DELETE -H"Authorization: auth_token <auth_token>"
https://<device service url>/apiv1/groups/<group-id>.json
```

Response:

200 – Success, group successfully deleted

404 – Failure

7.6 *POST* /apiv1/groups/:group_id/devices

Adds a single device to the group. A user is indicated by the auth_token in the authorization header of the HTTP request.

Input:

Mandatory:

Group ID: Get key (Id) from POST /apiv1/groups

Type: Integer

Device ID: Device id to be added to the group

Type: Integer

Output: The modified group as a hash

JSON Example:

```
curl -i -X POST -H"Authorization: auth_token <auth_token>" -
H"Content-Type: application/json" -d" { "device_id":"25"}"
https://<device service url>/apiv1/groups/<group-id>/devices.json
```

Response:

```
{ "group": {
  "name": "ctrl1", "key": 12, "device_count": 2,
  "devices": [
    {
      "product_name": "asdf", "dsn": "AC000W000000XXXX",
      "oem_model": null,
      "key": 12, "connection_status": "Online"
    },
    {
      "product_name": "PZIGQUFHXXXX",
      "dsn": "AC000W000000XXXX", "oem_model": null, "key":
      25, "connection_status": "Offline"
    }
  ]
}
```

Response:

201 – Success

422 – Failure

404 – Failure

7.7 DELETE /apiv1/groups/:group_id/devices/:device_id

Removes a single device from the group. A user is indicated by the auth_token in the authorization header of the HTTP request.

Note: A device is only removed from the group's list, but it is not destroyed.

Input:

Mandatory:

Group ID: Get key (Id) from POST /apiv1/groups

Type: Integer

Device ID: to be removed from the group

Type: Integer

Output: The modified group as a hash

JSON Example:

```
curl -i -X DELETE -H"Authorization: auth_token <auth_token>" -  
H"Content-Type: application/json" https://<device service  
url>/apiv1/groups/<grp-id>/devices/<dev-id>.json
```

Response:

```
{ "group": {  
  "name": "ctrl1", "key": 12, "device_count": 1,  
  "devices": [  
    { "product_name": "asdf", "dsn": "AC000W000000XXXX", "  
      oem_model": null, "key": 12, "connection_status": "Onl  
ine"  
    }  
  ]  
}
```

Response:

204 – Success

422 – Failure

404 – Failure

7.8 **POST /apiv1/groups/:group_id/batched_datapoints**

Changes values of one or more properties on one or more devices belonging to the group. A user is indicated by the auth_token in the authorization header of the HTTP request.

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

IMPOERTANT! This API should not be used for large scale updates to all your devices or any other large scale updates!

Input:

Mandatory:

Group ID: group id whose member properties are to be changed. Get from POST /apiv1/groups

Type: Integer

Parameter hash should look like this:

```
{
  "devices":{
    "device":[
      {
        "device_id":"206",
        "properties":{
          "property":[
            {
              "prop_id":"17",
              "prop_value":"fa3"
            },
            {
              "prop_id":"18",
              "prop_value":"tes11"
            }
          ]
        }
      }
    ]
  },
  {
    "device_id":"195",
    "properties":{
      "property":[
        {
          "prop_id":"110",
          "prop_value":"3334"
        },
        {
          "prop_id":"111",
          "prop_value":"1"
        }
      ]
    }
  }
}
```

```

    }
  }
]
}
}

```

Output: An array of hashes indicating the changes made.

JSON Example:

```

curl -i -X POST -H"Authorization: auth_token <auth_token>" -
H"Content-Type: application/json" -d"{
"devices":{"device":[{"device_id":"194",
"properties":{"property":[{"prop_id":"105", "prop_value":"0"},
{"prop_id":"106", "prop_value":"tester123"}]}},
{"device_id":"195", "properties":{"property":[{"prop_id":"111",
"prop_value":"0"}, {"prop_id":"112", "prop_value":"test456"}]}},
{"device_id":"187", "properties":{"property":[{"prop_id":"2",
"prop_value":"0"}]}]} }" https://<device service
url>/apiv1/groups/43/batched_datapoints

```

Response:

```

[
{
  "device_id":194,
  "prop_id":105,
  "prop_value":0
},
{
  "device_id":194,
  "prop_id":106,
  "prop_value":"tester123"
},
{
  "device_id":195,
  "prop_id":111,
  "prop_value":0
},
{
  "device_id":195,
  "prop_id":112,
  "prop_value":"test456"
},
{
  "device_id":187,
  "prop_id":2,
  "prop_value":0
}
]

```



```
    }
  ]
}
```

Response:

201 – Success
 422 – Failure
 404 – Failure

7.9 POST /apiv1/groups/:group_id/datapoints

Changes values of a single property on all devices belonging to the group. A user is indicated by the auth_token in the authorization header of the HTTP request.

Note: API performs all value changes that are valid and possible. Successful updates are returned in a hash in the response.

Input:**Mandatory:**

Group ID of the group whose member properties are to be changed.

Type: Integer

Parameter hash should look like this:

```
{
  "datapoints": {
    "datapoint": [{
      "property_name": "test_property1",
      "value": 234
    },
    {
      "property_name": "test_property2",
      "value": "somestring"
    }
  ]
}
```

Output: An array of hashes indicating the changes made.

JSON Example:

```
curl -k -X POST -H"Authorization: auth_token <auth_token>" -
H"Content-Type: application/json" -d{"datapoints":
{"datapoint":[{"property_name": "test1","value":
234}, {"property_name": "prop2","value": "somestring"}]}}
```

<https://<device service url>/apiv1/groups/16/datapoints.json>

Response:

```
[{"device_id":12,"prop_name":"test1","prop_value":234},  
 {"device_id":1,"prop_name":"prop2","prop_value":"somestring"}]
```

Response:

201 – Success
422 – Failure
404 – Failure

7.10 GET /apiv1/groups/:group_id/datapoints

Gets values for a list of properties (by name), from all devices belonging to the group. A user is indicated by the auth_token in the authorization header of the HTTP request.

Note: API performs all value retrievals that are valid and possible.

Input:**Mandatory:**

Group ID : Id of the group whose member properties are to be changed
Type: Integer
Property_names: Array of property names for which values are to be retrieved.
Type: Arrays of strings

Output: A datapoints hash organized by device, properties, and values.

JSON Example:

```
curl -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" -d{"property_names": ["test1", "prop2", "prop3", "prop4"]} "https://<device service url>/apiv1/groups/<group id>/datapoints.json"
```

Response:

```
{  
  "datapoints":{  
    "devices":{  
      "device":[  
        {"id":12,  
          "properties":{  
            "property":[  
              {"id":14,"name":"test1","value":1111}  
            ]  
          }  
        },  
      ],  
    },  
  },  
}
```

```
{
  "id": 17,
  "properties": {
    "property": [
      {
        "id": 12,
        "name": "test1",
        "value": 1111
      }
    ]
  }
},
{
  "id": 24,
  "properties": {
    "property": [
      {
        "id": 13,
        "name": "test1",
        "value": 1111
      }
    ]
  }
}
]
```

Response:

201 – Success

404 – Failure

SCHEDULES

7.11 *POST* `apiv1/devices/<device_id>/schedules[.format]`

Creates a schedule for a device with the device id. Note: Deleting a Schedule is not currently available.

Input:

Mandatory:

device id: Id of the device obtained via GET/apiv1/devices

start-time-each-day: start time in local time, following 24 hour format (HH:mm:ss)

Type: time

direction: indicates whether the schedule is “to device” or “from device” (“input” or “output”)

Type: String

display_name: User friendly schedule name that would be displayed on UIs.

Type: String

Default value: ‘name’ when not part of input.

name: Schedule name as an identifier.

Type: String

Optional:

active: used for pausing and resuming the schedule. If this field is not set the default is set to true.

Type: String

end-time-each-day: end time in local time, following 24 hour format (HH:mm:ss)

Type: String

time-before-end: time before end in local time, following 24 hour format (HH:mm:ss)

Type: String

utc: indicates the schedule to be set with UTC time zone. If this field is not set the default is set to false.

Type: String

start_date: date when the schedule starts running (“yyyy-mm-dd”)

Type: String

end_date: date when the schedule stops running (“yyyy-mm-dd”).

days_of_week: array of day#: [1,2,3,4,5,6,7], with 1 referring to Sunday and so on. If this field is not set, the default is set to every day.

days_of_month: array of day#: [1,2,3,4,5,6,7, 8 9..32], with 1 referring to 1st day of 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 28th, 30th, or 31st

months_of_year: array of month#: [1,2,3,4,...12] with 1 referring to Jan. If this field is not set, the default is set to every month.

default : set to every occurrence.

Type: Integer

day_occur_of_month: Array of Occurrence #'s: [1,2,3,4,5,6]. Day occurrence of month. For example, 1st Sunday or 2nd Sunday of month. As a special case, 6 means "last occurrence of the day in the month".

Duration: duration in seconds.

Type : Integer

interval: interval in seconds

Type : Integer

schedule_actions: a list of schedule actions.

Type : String

NOTE: Deleting a Schedule is not currently available.

XML Example:

```
$ curl -k -X POST -H"Content-Type: application/xml" -
d"<schedule><name>night schedule</name><start-date>2013-06-
01</start-date><end-date>2013-07-01</end-date><start-time-each-
day>20:15:00</start-time-each-day><end-time-each-
day>22:00:00</end-time-each-
day><direction>input</direction><schedule-actions
type='array'><schedule-action><name>Blue_LED</name><base-
type>boolean</base-type><value>0</value><in-range>false</in-
range><at-start>false</at-start><at-end>true</at-
end><type>SchedulePropertyAction</type></schedule-
action><schedule-action><name>Blue_LED</name><base-
type>boolean</base-type><value>1</value><in-range>false</in-
range><at-start>true</at-start><at-end>false</at-
end><type>SchedulePropertyAction</type></schedule-
actions></schedule-actions></schedule>" -H"Authorization:
auth_token <auth_token>"
https://<device service url>/apiv1/devices/3/schedules.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule>
  <version>1</version>
  <name>night schedule</name>
  <display_name>night schedule</display_name>
```

```
<start-date>2013-06-01</start-date>
<end-date>2013-07-01</end-date>
<days-of-week type="array">
  <days-of-week type="integer">1</days-of-week>
  <days-of-week type="integer">2</days-of-week>
  <days-of-week type="integer">3</days-of-week>
  <days-of-week type="integer">4</days-of-week>
  <days-of-week type="integer">5</days-of-week>
  <days-of-week type="integer">6</days-of-week>
  <days-of-week type="integer">7</days-of-week>
</days-of-week>
<days-of-month type="array">
  <days-of-month type="integer">1</days-of-month>
  <days-of-month type="integer">2</days-of-month>
  <days-of-month type="integer">3</days-of-month>
  <days-of-month type="integer">4</days-of-month>
  <days-of-month type="integer">5</days-of-month>
  <days-of-month type="integer">6</days-of-month>
  <days-of-month type="integer">7</days-of-month>
  <days-of-month type="integer">8</days-of-month>
  <days-of-month type="integer">9</days-of-month>
  <days-of-month type="integer">10</days-of-month>
  <days-of-month type="integer">11</days-of-month>
  <days-of-month type="integer">12</days-of-month>
  <days-of-month type="integer">13</days-of-month>
  <days-of-month type="integer">14</days-of-month>
  <days-of-month type="integer">15</days-of-month>
  <days-of-month type="integer">16</days-of-month>
  <days-of-month type="integer">17</days-of-month>
  <days-of-month type="integer">18</days-of-month>
  <days-of-month type="integer">19</days-of-month>
  <days-of-month type="integer">20</days-of-month>
  <days-of-month type="integer">21</days-of-month>
  <days-of-month type="integer">22</days-of-month>
  <days-of-month type="integer">23</days-of-month>
  <days-of-month type="integer">24</days-of-month>
  <days-of-month type="integer">25</days-of-month>
  <days-of-month type="integer">26</days-of-month>
  <days-of-month type="integer">27</days-of-month>
  <days-of-month type="integer">28</days-of-month>
  <days-of-month type="integer">29</days-of-month>
  <days-of-month type="integer">30</days-of-month>
```

```
<days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
  <months-of-year type="integer">9</months-of-year>
  <months-of-year type="integer">10</months-of-year>
  <months-of-year type="integer">11</months-of-year>
  <months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>20:15:00</start-time-each-day>
<end-time-each-day>22:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">34</key>
<schedule-actions type="array">
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">false</at-start>
    <at-end type="boolean">true</at-end>
    <active type="boolean">true</active>
    <key type="integer">28</key>
    <value type="integer">0</value>
  </schedule-action>
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
```

```
<in-range type="boolean">false</in-range>
<at-start type="boolean">true</at-start>
<at-end type="boolean">false</at-end>
<active type="boolean">true</active>
<key type="integer">29</key>
<value type="integer">1</value>
</schedule-action>
</schedule-actions>
</schedule>
```

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:

200 – Success, created schedule
 400 – Failure, bad request
 422 – Unprocessable Entity

7.12 GET *apiv1/devices/<device_id>/schedules[.format]*

Lists all the schedules for device with <device_id> id.

Input:**Mandatory:**

device_id : device-id that can be obtained from GET /devices
 Type: Integer

Output: array of schedules

XML Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/devices/3/schedules.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedules type="array">
<schedule>
  <version>1</version>
  <name>night schedule</name>
  <display_name>night schedule</display_name>
  <start-date>2013-06-01</start-date>
  <end-date>2013-07-01</end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>
    <days-of-week type="integer">2</days-of-week>
    <days-of-week type="integer">3</days-of-week>
    <days-of-week type="integer">4</days-of-week>
    <days-of-week type="integer">5</days-of-week>
    <days-of-week type="integer">6</days-of-week>
    <days-of-week type="integer">7</days-of-week>
  </days-of-week>
  <days-of-month type="array">
    <days-of-month type="integer">1</days-of-month>
    <days-of-month type="integer">2</days-of-month>
    <days-of-month type="integer">3</days-of-month>
    <days-of-month type="integer">4</days-of-month>
    <days-of-month type="integer">5</days-of-month>
    <days-of-month type="integer">6</days-of-month>
    <days-of-month type="integer">7</days-of-month>
    <days-of-month type="integer">8</days-of-month>
    <days-of-month type="integer">9</days-of-month>
    <days-of-month type="integer">10</days-of-month>
    <days-of-month type="integer">11</days-of-month>
    <days-of-month type="integer">12</days-of-month>
    <days-of-month type="integer">13</days-of-month>
    <days-of-month type="integer">14</days-of-month>
    <days-of-month type="integer">15</days-of-month>
    <days-of-month type="integer">16</days-of-month>
    <days-of-month type="integer">17</days-of-month>
    <days-of-month type="integer">18</days-of-month>
    <days-of-month type="integer">19</days-of-month>
    <days-of-month type="integer">20</days-of-month>
    <days-of-month type="integer">21</days-of-month>
```

```
<days-of-month type="integer">22</days-of-month>
<days-of-month type="integer">23</days-of-month>
<days-of-month type="integer">24</days-of-month>
<days-of-month type="integer">25</days-of-month>
<days-of-month type="integer">26</days-of-month>
<days-of-month type="integer">27</days-of-month>
<days-of-month type="integer">28</days-of-month>
<days-of-month type="integer">29</days-of-month>
<days-of-month type="integer">30</days-of-month>
<days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
  <months-of-year type="integer">9</months-of-year>
  <months-of-year type="integer">10</months-of-year>
  <months-of-year type="integer">11</months-of-year>
  <months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>20:15:00</start-time-each-day>
<end-time-each-day>22:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">49</key>
</schedule>
</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
    }
}
]

```

7.13 GET *apiv1/schedules/<id>[.format]*

Retrieves details for the schedule corresponding to the schedule id.

Input:**Mandatory:**

id: schedule id obtained via GET on all device schedules.

Type: Integer

XML Example:

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

Reponse:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule>
  <version>1</version>
  <name>night schedule</name>
  <display_name>my night schedule</display_name>
  <start-date>2013-06-01</start-date>
  <end-date>2013-07-01</end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>
    <days-of-week type="integer">2</days-of-week>
    <days-of-week type="integer">3</days-of-week>
    <days-of-week type="integer">4</days-of-week>
    <days-of-week type="integer">5</days-of-week>
    <days-of-week type="integer">6</days-of-week>
    <days-of-week type="integer">7</days-of-week>
  </days-of-week>
  <days-of-month type="array">
    <days-of-month type="integer">1</days-of-month>
    <days-of-month type="integer">2</days-of-month>
    <days-of-month type="integer">3</days-of-month>
    <days-of-month type="integer">4</days-of-month>
    <days-of-month type="integer">5</days-of-month>
    <days-of-month type="integer">6</days-of-month>
    <days-of-month type="integer">7</days-of-month>
    <days-of-month type="integer">8</days-of-month>
    <days-of-month type="integer">9</days-of-month>
    <days-of-month type="integer">10</days-of-month>
    <days-of-month type="integer">11</days-of-month>
    <days-of-month type="integer">12</days-of-month>
    <days-of-month type="integer">13</days-of-month>
    <days-of-month type="integer">14</days-of-month>
```

```
<days-of-month type="integer">15</days-of-month>
<days-of-month type="integer">16</days-of-month>
<days-of-month type="integer">17</days-of-month>
<days-of-month type="integer">18</days-of-month>
<days-of-month type="integer">19</days-of-month>
<days-of-month type="integer">20</days-of-month>
<days-of-month type="integer">21</days-of-month>
<days-of-month type="integer">22</days-of-month>
<days-of-month type="integer">23</days-of-month>
<days-of-month type="integer">24</days-of-month>
<days-of-month type="integer">25</days-of-month>
<days-of-month type="integer">26</days-of-month>
<days-of-month type="integer">27</days-of-month>
<days-of-month type="integer">28</days-of-month>
<days-of-month type="integer">29</days-of-month>
<days-of-month type="integer">30</days-of-month>
<days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
  <months-of-year type="integer">9</months-of-year>
  <months-of-year type="integer">10</months-of-year>
  <months-of-year type="integer">11</months-of-year>
  <months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>20:15:00</start-time-each-day>
<end-time-each-day>22:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
```



```
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">34</key>
<schedule-actions type="array">
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">false</at-start>
    <at-end type="boolean">true</at-end>
    <active type="boolean">true</active>
    <key type="integer">28</key>
    <value type="integer">0</value>
  </schedule-action>
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">true</at-start>
    <at-end type="boolean">false</at-end>
    <active type="boolean">true</active>
    <key type="integer">29</key>
    <value type="integer">1</value>
  </schedule-action>
</schedule-actions>
</schedule>
```

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
  }
]
```

```

    }
  ]
}
}

```

7.14 GET

apiv1/devices/<device_id>/schedules/find_by_name.xml?name=<name>

Retrieves details for the schedule corresponding to the name <find_by_name> and the device_id.

Input:

Mandatory:

name: schedule name obtained via GET on all device schedules.

Type: String

device_id: device id obtained via GET on all devices.

Type: Integer

Output: schedule object

XML Example:

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

Reponse:

```

<?xml version="1.0" encoding="UTF-8"?>
<schedule>
  <version>1</version>
  <name>night schedule</name>
  <name>my night schedule</name>
  <start-date>2013-06-01</start-date>
  <end-date>2013-07-01</end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>
    <days-of-week type="integer">2</days-of-week>
    <days-of-week type="integer">3</days-of-week>
    <days-of-week type="integer">4</days-of-week>
    <days-of-week type="integer">5</days-of-week>
    <days-of-week type="integer">6</days-of-week>
    <days-of-week type="integer">7</days-of-week>
  </days-of-week>

```

```
<days-of-month type="array">
  <days-of-month type="integer">1</days-of-month>
  <days-of-month type="integer">2</days-of-month>
  <days-of-month type="integer">3</days-of-month>
  <days-of-month type="integer">4</days-of-month>
  <days-of-month type="integer">5</days-of-month>
  <days-of-month type="integer">6</days-of-month>
  <days-of-month type="integer">7</days-of-month>
  <days-of-month type="integer">8</days-of-month>
  <days-of-month type="integer">9</days-of-month>
  <days-of-month type="integer">10</days-of-month>
  <days-of-month type="integer">11</days-of-month>
  <days-of-month type="integer">12</days-of-month>
  <days-of-month type="integer">13</days-of-month>
  <days-of-month type="integer">14</days-of-month>
  <days-of-month type="integer">15</days-of-month>
  <days-of-month type="integer">16</days-of-month>
  <days-of-month type="integer">17</days-of-month>
  <days-of-month type="integer">18</days-of-month>
  <days-of-month type="integer">19</days-of-month>
  <days-of-month type="integer">20</days-of-month>
  <days-of-month type="integer">21</days-of-month>
  <days-of-month type="integer">22</days-of-month>
  <days-of-month type="integer">23</days-of-month>
  <days-of-month type="integer">24</days-of-month>
  <days-of-month type="integer">25</days-of-month>
  <days-of-month type="integer">26</days-of-month>
  <days-of-month type="integer">27</days-of-month>
  <days-of-month type="integer">28</days-of-month>
  <days-of-month type="integer">29</days-of-month>
  <days-of-month type="integer">30</days-of-month>
  <days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
```

```
<months-of-year type="integer">9</months-of-year>
<months-of-year type="integer">10</months-of-year>
<months-of-year type="integer">11</months-of-year>
<months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>20:15:00</start-time-each-day>
<end-time-each-day>22:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">34</key>
<schedule-actions type="array">
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">false</at-start>
    <at-end type="boolean">true</at-end>
    <active type="boolean">true</active>
    <key type="integer">28</key>
    <value type="integer">0</value>
  </schedule-action>
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">true</at-start>
    <at-end type="boolean">false</at-end>
    <active type="boolean">true</active>
    <key type="integer">29</key>
    <value type="integer">1</value>
  </schedule-action>
</schedule-actions>
</schedule>
```

JSON Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/api/v1/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
    }
]
}

```

7.15 PUT

apiv1/devices/<device_id>/schedules/<schedule_id>[.format]

Updates the schedule specified by the schedule id.

Input:

Mandatory: schedule object

start_time: start time in local time

Validation: 24 hour format (HH:mm:ss)

Type: String

direction: indicates whether the schedule is “to device” or “from device” (“input” or “output”)

Type: String

active: used for pausing and resuming the schedule. If this field is not set,`
Default: true

Type: String

name: name used by the service to identify the schedule

Type: String

Optional:

end_time: end time in local time,

Validation: 24 hour format (HH:mm:ss)

Type: String

time-before-end: time before end in local time,

Validation: 24 hour format (HH:mm:ss)

Type: String **display_name**: A user friendly name for the schedule used on UIs. Default: Defaults to 'name' when not part of input

Type: String

utc: Indicates the schedule to be set with UTC timezone. If this field is not set the default is false.

Type: String

start_date: Date when the schedule starts running ("yyyy-mm-dd").

Type: String

end_date: Date when the schedule stops running ("yyyy-mm-dd").

Type: String

days_of_week: Array of Day#: [1,2,3,4,5,6,7], with 1 referring to Sunday and so on. If this field is not set the default is every day.

Type: Integer

days_of_month: Array of Day#: [1,2,3,4,5,6,7, 8 9..32], with 1 referring to 1st day of month, and so on. If this field is not set the default is every day. As a special case, 32 represents the last day of the month, whether it's the 28th, 30th, or 31st

Type: Integer

months_of_year: Array of Month#: [1,2,3,4,...12] with 1 referring to Jan. If this field is not set, the default is every month. If this field is not set the default is every occurrence.

Type: Integer

day_occur_of_month: Array of occurrence #'s: [1,2,3,4,5,6]. Day occurrence of month. For example, 1st Sunday or 2nd Sunday of month.

Type: Integer

duration: in seconds.

Type: Integer

interval: interval in seconds.

Type: Integer

schedule_actions: a list of schedule_actions.

Type: String

Response:

200 – Success, updated schedule

400 – Failure, bad request

422 – Unprocessable entity

XML Example:

```
$ curl -k -X PUT -H"Content-Type: application/xml" -d"<schedule><name>morning schedule</name><start-date>2013-06-
```

```
15</start-date><end-date>2013-08-01</end-date><start-
time>10:00:00</start-time><end-time>11:00:00</end-
time><direction>input</direction><schedule-actions
type='array'><schedule-
action><name>Blue_LED</name><key>28</key><base-type>boolean</base-
type><value>1</value><in-range>>false</in-range><at-start>true</at-
start><at-end>>false</at-end></schedule-action><schedule-
action><name>Blue_LED</name><key>29</key><base-type>boolean</base-
type><value>0</value><in-range>>false</in-range><at-
start>>false</at-start><at-end>true</at-end></schedule-
action></schedule-actions></schedule>" -H"Authorization:
auth_token <auth_token>"
https://<device service url>/apiv1/devices/3/schedules/7.xmlhttps://<device service url>/apiv1/schedules.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule>
  <version>1</version>
  <name>morning schedule</name>
  <display_name>morning schedule</display_name>
  <start-date>2013-06-15</start-date>
  <end-date>2013-08-01</end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>
    <days-of-week type="integer">2</days-of-week>
    <days-of-week type="integer">3</days-of-week>
    <days-of-week type="integer">4</days-of-week>
    <days-of-week type="integer">5</days-of-week>
    <days-of-week type="integer">6</days-of-week>
    <days-of-week type="integer">7</days-of-week>
  </days-of-week>
  <days-of-month type="array">
    <days-of-month type="integer">1</days-of-month>
    <days-of-month type="integer">2</days-of-month>
    <days-of-month type="integer">3</days-of-month>
    <days-of-month type="integer">4</days-of-month>
    <days-of-month type="integer">5</days-of-month>
    <days-of-month type="integer">6</days-of-month>
    <days-of-month type="integer">7</days-of-month>
    <days-of-month type="integer">8</days-of-month>
    <days-of-month type="integer">9</days-of-month>
    <days-of-month type="integer">10</days-of-month>
```

```
<days-of-month type="integer">11</days-of-month>
<days-of-month type="integer">12</days-of-month>
<days-of-month type="integer">13</days-of-month>
<days-of-month type="integer">14</days-of-month>
<days-of-month type="integer">15</days-of-month>
<days-of-month type="integer">16</days-of-month>
<days-of-month type="integer">17</days-of-month>
<days-of-month type="integer">18</days-of-month>
<days-of-month type="integer">19</days-of-month>
<days-of-month type="integer">20</days-of-month>
<days-of-month type="integer">21</days-of-month>
<days-of-month type="integer">22</days-of-month>
<days-of-month type="integer">23</days-of-month>
<days-of-month type="integer">24</days-of-month>
<days-of-month type="integer">25</days-of-month>
<days-of-month type="integer">26</days-of-month>
<days-of-month type="integer">27</days-of-month>
<days-of-month type="integer">28</days-of-month>
<days-of-month type="integer">29</days-of-month>
<days-of-month type="integer">30</days-of-month>
<days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
  <months-of-year type="integer">9</months-of-year>
  <months-of-year type="integer">10</months-of-year>
  <months-of-year type="integer">11</months-of-year>
  <months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>10:00:00</start-time-each-day>
<end-time-each-day>11:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
```

```

<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">34</key>
<schedule-actions type="array">
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">true</at-start>
    <at-end type="boolean">false</at-end>
    <active type="boolean">true</active>
    <key type="integer">28</key>
    <value type="integer">1</value>
  </schedule-action>
  <schedule-action type="SchedulePropertyAction">
    <name type="SchedulePropertyAction">Blue_LED</name>
    <base-type type="SchedulePropertyAction">boolean</base-type>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">false</at-start>
    <at-end type="boolean">true</at-end>
    <active type="boolean">true</active>
    <key type="integer">29</key>
    <value type="integer">0</value>
  </schedule-action>
</schedule-actions>
</schedule>

```

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
      }
    ]
  }
}

```

7.16 PUT /apiv1/schedules/<schedule_id>/clear[.format]

Clears the schedule specified by the schedule id. It marks the schedule as cleared, inactive and destroys all of its schedule actions.

Input:

Mandatory:

Schedule_id: Schedule_id returned when schedule is created

Type: String

XML Example:

```

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

```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<schedule>
  <version>1</version>
  <name>morning</name>
  <display_name>my morning schedule</display_name>
  <start-date>2013-11-11</start-date>
  <end-date></end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>

```



```
<days-of-week type="integer">2</days-of-week>
<days-of-week type="integer">3</days-of-week>
<days-of-week type="integer">4</days-of-week>
<days-of-week type="integer">5</days-of-week>
<days-of-week type="integer">6</days-of-week>
</days-of-week>
<days-of-month type="array">
  <days-of-month type="integer">1</days-of-month>
  <days-of-month type="integer">2</days-of-month>
  <days-of-month type="integer">3</days-of-month>
  <days-of-month type="integer">4</days-of-month>
  <days-of-month type="integer">5</days-of-month>
  <days-of-month type="integer">6</days-of-month>
  <days-of-month type="integer">7</days-of-month>
  <days-of-month type="integer">8</days-of-month>
  <days-of-month type="integer">9</days-of-month>
  <days-of-month type="integer">10</days-of-month>
  <days-of-month type="integer">11</days-of-month>
  <days-of-month type="integer">12</days-of-month>
  <days-of-month type="integer">13</days-of-month>
  <days-of-month type="integer">14</days-of-month>
  <days-of-month type="integer">15</days-of-month>
  <days-of-month type="integer">16</days-of-month>
  <days-of-month type="integer">17</days-of-month>
  <days-of-month type="integer">18</days-of-month>
  <days-of-month type="integer">19</days-of-month>
  <days-of-month type="integer">20</days-of-month>
  <days-of-month type="integer">21</days-of-month>
  <days-of-month type="integer">22</days-of-month>
  <days-of-month type="integer">23</days-of-month>
  <days-of-month type="integer">24</days-of-month>
  <days-of-month type="integer">25</days-of-month>
  <days-of-month type="integer">26</days-of-month>
  <days-of-month type="integer">27</days-of-month>
  <days-of-month type="integer">28</days-of-month>
  <days-of-month type="integer">29</days-of-month>
  <days-of-month type="integer">30</days-of-month>
  <days-of-month type="integer">32</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
```

```

<months-of-year type="integer">3</months-of-year>
<months-of-year type="integer">4</months-of-year>
<months-of-year type="integer">5</months-of-year>
<months-of-year type="integer">6</months-of-year>
<months-of-year type="integer">7</months-of-year>
<months-of-year type="integer">8</months-of-year>
<months-of-year type="integer">9</months-of-year>
<months-of-year type="integer">10</months-of-year>
<months-of-year type="integer">11</months-of-year>
</months-of-year>
<day-occur-of-month type="array">
  <day-occur-of-month type="integer">1</day-occur-of-month>
  <day-occur-of-month type="integer">2</day-occur-of-month>
  <day-occur-of-month type="integer">3</day-occur-of-month>
  <day-occur-of-month type="integer">4</day-occur-of-month>
  <day-occur-of-month type="integer">7</day-occur-of-month>
</day-occur-of-month>
<start-time-each-day>10:00:00</start-time-each-day>
<end-time-each-day></end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer">4</duration>
<interval type="integer">1</interval>
<direction>input</direction>
<active type="boolean">false</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">12</key>
<schedule-actions type="array"/>
</schedule>

```

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:

200 – Success
 400 – Failure, bad request
 422 – Unprocessable entity

7.17 GET `apiv1/devices/<device_id>/schedules/base64[.format]`

Gets the corresponding base64 that represents the sent schedule.

Input: schedule object

Mandatory:

Direction: To or from device

Type: String

start_time: Date when the schedule starts running ("yyyy-mm-dd").

Type: String

name: Display name that can be used for reference

Type: String

device-id: Device identification number obtained via GET /devices

Type: Integer

XML Example:

```
$ curl -X GET -H"Content-Type: application/xml" -d"<schedule><name>test</name><direction>input</direction><start_time>20:15:00</start_time></schedule>" -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/devices/3/schedules/base64.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
  <base64>MwEBLQQAARzE</base64>
</hash>
```

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:

200 – Success
422 – Failure, unprocessable entity

7.18 GET *apiv1/schedules/all/by_user.xml*

Lists all the schedules of devices the current user has access to. It also gets schedules from the linked user accounts.

Input: none

Output: array of schedules

XML Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apv1/schedules/all/by user.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedules type="array">
<schedule>
  <version>1</version>
  <name>night schedule</name>
  <display_name>night schedule</display_name>
  <start-date>2013-06-01</start-date>
  <end-date>2013-07-01</end-date>
  <days-of-week type="array">
    <days-of-week type="integer">1</days-of-week>
    <days-of-week type="integer">2</days-of-week>
    <days-of-week type="integer">3</days-of-week>
    <days-of-week type="integer">4</days-of-week>
    <days-of-week type="integer">5</days-of-week>
    <days-of-week type="integer">6</days-of-week>
    <days-of-week type="integer">7</days-of-week>
  </days-of-week>
  <days-of-month type="array">
    <days-of-month type="integer">1</days-of-month>
    <days-of-month type="integer">2</days-of-month>
    <days-of-month type="integer">3</days-of-month>
    <days-of-month type="integer">4</days-of-month>
    <days-of-month type="integer">5</days-of-month>
    <days-of-month type="integer">6</days-of-month>
    <days-of-month type="integer">7</days-of-month>
    <days-of-month type="integer">8</days-of-month>
    <days-of-month type="integer">9</days-of-month>
    <days-of-month type="integer">10</days-of-month>
    <days-of-month type="integer">11</days-of-month>
    <days-of-month type="integer">12</days-of-month>
    <days-of-month type="integer">13</days-of-month>
    <days-of-month type="integer">14</days-of-month>
    <days-of-month type="integer">15</days-of-month>
    <days-of-month type="integer">16</days-of-month>
    <days-of-month type="integer">17</days-of-month>
    <days-of-month type="integer">18</days-of-month>
    <days-of-month type="integer">19</days-of-month>
    <days-of-month type="integer">20</days-of-month>
    <days-of-month type="integer">21</days-of-month>
    <days-of-month type="integer">22</days-of-month>
    <days-of-month type="integer">23</days-of-month>
    <days-of-month type="integer">24</days-of-month>
```

```

<days-of-month type="integer">25</days-of-month>
<days-of-month type="integer">26</days-of-month>
<days-of-month type="integer">27</days-of-month>
<days-of-month type="integer">28</days-of-month>
<days-of-month type="integer">29</days-of-month>
<days-of-month type="integer">30</days-of-month>
<days-of-month type="integer">31</days-of-month>
</days-of-month>
<months-of-year type="array">
  <months-of-year type="integer">1</months-of-year>
  <months-of-year type="integer">2</months-of-year>
  <months-of-year type="integer">3</months-of-year>
  <months-of-year type="integer">4</months-of-year>
  <months-of-year type="integer">5</months-of-year>
  <months-of-year type="integer">6</months-of-year>
  <months-of-year type="integer">7</months-of-year>
  <months-of-year type="integer">8</months-of-year>
  <months-of-year type="integer">9</months-of-year>
  <months-of-year type="integer">10</months-of-year>
  <months-of-year type="integer">11</months-of-year>
  <months-of-year type="integer">12</months-of-year>
</months-of-year>
<day-occur-of-month type="yaml" nil="true"></day-occur-of-month>
<start-time-each-day>20:15:00</start-time-each-day>
<end-time-each-day>22:00:00</end-time-each-day>
<time-before-end></time-before-end>
<duration type="integer" nil="true"></duration>
<interval type="integer" nil="true"></interval>
<direction>input</direction>
<active type="boolean">true</active>
<device-id type="integer">3</device-id>
<utc type="boolean">false</utc>
<fixed-actions type="boolean">false</fixed-actions>
<key type="integer">49</key>
</schedule>
</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
}
}
]
```

7.19 SCHEDULE ACTION

Ayla supports only `SchedulePropertyAction` if you don't send the param "type," it uses `SchedulePropertyAction` as default.

7.19.1 GET

apiv1/schedules/<schedule_id>/schedule_actions[.format]

Lists all the schedule actions for schedule with the schedule id.

Input:

Mandatory:

schedule_id: **Schedule_id returned when schedule is created**

Type: String

type: The type of schedule action.

Currently Ayla supports `SchedulePropertyAction`

Type: String

XML Example:

```
$ curl -X GET -H"Content-Type: application/xml" -
d"<type>SchedulePropertyAction</type>" -H"Authorization:
```

```
auth_token <auth_token>" https://<device service url>/apiv1/schedules/1/schedule\_actions.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule_action type="array">
  <schedule_action>
    <name>Blue_LED</name>
    <base-type>boolean</base-type>
    <schedule-id type="integer">1</schedule-id>
    <active type="boolean">true</active>
    <key type="integer">31</key>
    <value type="integer">1</value>
  </schedule_action>
</schedule_action>
```

Curl 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
    }
  }
]
```

7.19.2 GET apiv1/schedule_actions/<id>[.format]

Retrieves details for the schedule action corresponding to the schedule action id.

Input:**Mandatory:**

id: schedule_action id obtained via GET of all schedules
type: the type of schedule action. Currently Ayla supports
SchedulePropertyAction

Output: schedule_action object

XML Example:

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

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule_action>
  <name>Blue_LED</name>
  <base-type>boolean</base-type>
  <schedule-id type="integer">36</schedule-id>
  <active type="boolean">true</active>
  <key type="integer">31</key>
  <value type="integer">1</value>
</schedule_action>
```

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
  }
}
```

7.19.3 GET Schedules by Name

GET

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

Retrieves details for the schedule actions corresponding to the name and the schedule id.

Input:

Mandatory:

name: schedule_action name obtained via GET of all schedule actions
Type: String
schedule_id: schedule id obtained via GET of all schedules
Type: Integer
type: type of schedule action('sms','email')
Type: String

Output: array of schedule_action objects

XML Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device
service
url>/apiv1/schedules/61/schedule_actions/find_by_name.xml?name=Blue_LED
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule-action type="array">
  <schedule-action>
    <name>Blue_LED</name>
    <base-type>boolean</base-type>
    <schedule-id type="integer">61</schedule-id>
    <in-range type="boolean">false</in-range>
    <at-start type="boolean">true</at-start>
    <at-end type="boolean">false</at-end>
    <active type="boolean">true</active>
    <key type="integer">64</key>
    <value type="integer">1</value>
  </schedule-action>
</schedule-action>
```

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
    }
  }
]
```

7.19.4 POST `apiv1/schedules/<schedule_id>/schedule_actions[.format]`

Creates a schedule action for schedule with the schedule id.

Input:

Mandatory: schedule action object

name: the name of the property this schedule_action is related with. .

type: the type of schedule action. Currently Ayla supports SchedulePropertyAction

base_type: currently Ayla only supports "string", "integer", "boolean", "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: used for pausing and resuming the schedule_action. If this field is not set the default is set to true.

Optional:

value: value for schedule_action base_type

(*1): only one parameter from in_range, at_start and at_end must be set.

XML Example:

```
$ curl -k -X POST -H"Content-Type: application/xml" -  
d"<schedule_action><name>Blue_LED</name><base_type>boolean</base_t  
ype><value>1</value><in_range>true</in_range><type>SchedulePropert  
yAction</type></schedule_action>" -H"Authorization: auth_token  
<auth_token>" https://<device service  
url>/apiv1/schedules/<schedule-id>/schedule\_actions.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>  
<schedule-action>  
  <name>Blue_LED</name>  
  <base-type>boolean</base-type>  
  <schedule-id type="integer">37</schedule-id>  
  <active type="boolean">true</active>  
  <key type="integer">32</key>  
  <value type="integer">1</value>  
</schedule-action>
```

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:

200 – Success:
Failure: all other status codes

7.19.5 PUT apiv1/schedule_actions/<id>[.format]

Updates the schedule specified by an id.

Input:

Mandatory: schedule_action object

Id: schedule id generated when a schedule is created

Type: Integer

Optional:

value: value for schedule_action base_type

Type: Integer

name: the name of the property this schedule_action is related with.

Type: String

type: the type of schedule action.

Type: String

base_type: currently Ayla 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: used for pausing and resuming the schedule_action. If this field is not set, the default is true. (*1): Set only one parameter from in_range, at_start and at_end.

XML Example:

```
$ curl -X PUT -H"Content-Type: application/xml" -
d"<schedule_action><name>Blue_LED</name><base_type>boolean</base_t
ype><value>1</value><in_range>true</in_range><type>SchedulePropert
yAction</type></schedule_action>" -H"Authorization: auth_token
<auth-token>" https://<device service
url>/apiv1/schedule\_actions/1.xml
```

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.xml
```

Response:

200 – Success

Failure, all other status codes

7.19.6 DELETE apiv1/schedule_actions/<id>[.format]

Deletes the schedule action specified by an id.

Input:

Mandatory:

Id: schedule id.

Type: Integer

XML Example:

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

Reponse:

```
<?xml version="1.0" encoding="UTF-8"?>  
<schedule-action>  
  <name>Blue_LED</name>  
  <key type="integer">1</key>  
</schedule-action>
```

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
```

Reponse:

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

Response:

200 – Success

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

8 TRIGGERS

Triggers are basic conditions that are checked every time a new datapoint for a given property is created. When the conditions match, one or more applications send notifications.

8.1 **POST** *apiv1/properties/<property_key>/triggers[.format]*

Creates a trigger for the property specified by the `property_key`.

Input: trigger object

Mandatory:

trigger-type: Select one of the following strings:

“always” trigger fires on every datapoint

“compare_absolute”: compares to a value specified in the trigger

“on_change”: trigger fires on any change to property value

compare-type: specified only for compare-absolute or compare-service trigger types. Select one of the following strings:

“==”: equals

“>”: greater than

“<”: less than

“>=”: greater than or equal

“<=”: less than or equal

value: value used in the *compare-absolute* trigger type

trigger_apps: a list containing trigger apps with the format specified in the create trigger app api

Optional:

active: false in case the trigger is deactivated

Default: true

device-nickname: A user-friendly name for the device. The device’s product-name is used if not specified.

property-nickname: A user-friendly name for the property. The property name is used if not specified.

asset: false in case of a device level trigger, not as user’s asset.

Default: true

Output: returns trigger object created on success, else errors

XML Example:

```
$ curl -X POST -H"Authorization: auth_token <auth_token>" -H
"Content-Type: application/xml" -d "<trigger><trigger-
```

```
type>compare_absolute</trigger-type><compare-type>>=</compare-
type><value>50</value><trigger_apps><trigger_app><name>custom</nam
e><username>test</username><param1>to@example.org</param1></trigge
r_app></trigger_apps></trigger>" https://<device service
url>/apiv1/properties/9/triggers.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<trigger>
  <device-nickname>Device</device-nickname>
  <property-nickname>a</property-nickname>
  <compare-type>>=</compare-type>
  <period>always</period>
  <trigger-type>compare_absolute</trigger-type>
  <base-type>integer</base-type>
  <value>50</value>
  <triggered-at nil="true"></triggered-at>
  <user-id type="integer">1000003</user-id>
  <key type="integer">1000029</key>
  <property-key type="integer">1056</property-key>
  <asset type="boolean">true</asset>
  <active type="boolean">true</active>
  <trigger_apps type="array">
    <trigger-app>
      <name>custom</name>
      <nickname nil="true"></nickname>
      <key type="integer">122</key>
      <trigger-key type="integer">1000029</trigger-key>
      <username>test</username>
      <param1>to@example.org</param1>
      <param2 nil="true"></param2>
      <param3 nil="true"></param3>
      <param4 nil="true"></param4>
      <param5 nil="true"></param5>
    </trigger-app>
  </trigger_apps>
</trigger>
```

JSON Example:

```
$ curl -X POST -H"Authorization: auth_token <auth_token>" -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"}]
} }' https://<device service
url>/apiv1/properties/<key>/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,
    "key": 1000028,
    "property_key": 1056,
    "asset": true,
    "active": true,
    "trigger_apps": [
      {
        "name": "custom",
        "nickname": null,
        "key": 121,
        "trigger_key": 1000XXX,
        "username": "test",
        "param1": "to@example.org",
        "param2": null,
        "param3": null,
        "param4": null,
        "param5": null
      }
    ]
  }
}
```

8.2 PUT apiv1/triggers/<trigger_id>[.format]

Updates a trigger for the property specified by the property key.

Input: trigger object

Mandatory:

trigger-type: Select one of the following strings:

“always” trigger fires on every datapoint

“compare_absolute”: compares to a value specified in the trigger

“on_change”: trigger fires on any change to property value

compare-type: specified only for compare-absolute or compare-service trigger types.

Select one of the following strings:

“==”: equals

“>”: greater than

“<”: less than

“>=”: greater than or equal

“<=”: less than or equal

value: value used in the *compare-absolute* trigger type.

active (default: true): false in case the trigger is deactivated.

trigger_apps: a list containing trigger apps with the format specified in the create trigger app api

Optional:

device-nickname: A user-friendly name for the device. The device’s product-name is used if not specified.

property-nickname: A user-friendly name for the property. The property name is used if not specified.

asset: false in case is a device level trigger, not a user’s asset.
Default value: true

Output: returns trigger object updated on success, else errors

XML Example:

```
$ curl -X PUT -H"Authorization: auth_token <auth_token>" -H
"Content-Type: application/xml" -d "<trigger><trigger-
type>compare_absolute</trigger-type><compare-type>=</compare-
type><value>55</value><trigger_apps><trigger_app><name>custom</nam
e><username>modified</username><param1>to@example.org</param1></tr
igger_app></trigger_apps></trigger>" https://<device service
url>/apiv1/triggers/<trigger-id>.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<trigger>
  <device-nickname>Device</device-nickname>
  <property-nickname>a</property-nickname>
  <compare-type>&gt;=</compare-type>
```

```

<period>always</period>
<trigger-type>compare_absolute</trigger-type>
<base-type>integer</base-type>
<value>55</value>
<triggered-at nil="true"></triggered-at>
<user-id type="integer">10000XX</user-id>
<key type="integer">10000XX</key>
<property-key type="integer">1056</property-key>
<asset type="boolean">true</asset>
<active type="boolean">true</active>
<trigger-apps type="array">
  <trigger-app>
    <name>custom</name>
    <nickname nil="true"></nickname>
    <key type="integer">121</key>
    <trigger-key type="integer">10000XX</trigger-key>
    <username>test</username>
    <param1>to@example.org</param1>
    <param2 nil="true"></param2>
    <param3 nil="true"></param3>
    <param4 nil="true"></param4>
    <param5 nil="true"></param5>
  </trigger-app>
  <trigger-app>
    <name>custom</name>
    <nickname nil="true"></nickname>
    <key type="integer">123</key>
    <trigger-key type="integer">10000XX</trigger-key>
    <username>modified</username>
    <param1>to@example.org</param1>
    <param2 nil="true"></param2>
    <param3 nil="true"></param3>
    <param4 nil="true"></param4>
    <param5 nil="true"></param5>
  </trigger-app>
</trigger-apps>
</trigger>

```

JSON Example:

```

$ curl -X PUT -H"Authorization: auth_token <auth_token>" -H
"Content-Type: application/json" -d '{"trigger":{"name":"test",
"trigger_type":"compare_absolute", "compare_type":">="},

```

```
"value":55, "trigger_apps": [{"name":"custom", "username":  
"modified", "param1":"to@example.org"}] } }'
```

<https://<device service url>/apiv1/triggers/<trig-id>.json>

Response:

```
{  
  "trigger": {  
    "device_nickname": "Device",  
    "property_nickname": "a",  
    "compare_type": ">=",  
    "period": "always",  
    "trigger_type": "compare_absolute",  
    "base_type": "integer",  
    "value": 55,  
    "triggered_at": null,  
    "user_id": 10000XX,  
    "key": 10000XX,  
    "property_key": 1056,  
    "asset": true,  
    "active": true,  
    "trigger_apps": [  
      {  
        "name": "custom",  
        "nickname": null,  
        "key": 121,  
        "trigger_key": 10000XX,  
        "username": "test",  
        "param1": "to@example.org",  
        "param2": null,  
        "param3": null,  
        "param4": null,  
        "param5": null  
      },  
      {  
        "name": "custom",  
        "nickname": null,  
        "key": 123,  
        "trigger_key": 10000XX,  
        "username": "modified",  
        "param1": "to@example.org",  
        "param2": null,  
        "param3": null,  
        "param4": null,  
        "param5": null  
      }  
    ]  
  }  
}
```

```

        "param5": null
      }
    ]
  }
}

```

8.3 GET *apiv1/properties/<property_key>/triggers[.format]*

Lists all the triggers of the current user for a property with the property key.

XML Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/properties/<property_key>/triggers.xml
```

Response:

```

<?xml version="1.0" encoding="UTF-8"?>
<triggers type="array">
<trigger>
  <device-nickname>MyThermostat</device-nickname>
  <property-nickname>SmallInt</property-nickname>
  <compare-type>>=</compare-type>
  <period>always</period>
  <trigger-type>compare_absolute</trigger-type>
  <base-type>integer</base-type>
  <value>60</value>
  <triggered-at type="datetime" nil="true"></triggered-at>
  <key type="integer">11</key>
  <property-key type="integer">9</property-key>
</trigger>
</triggers>

```

JSON Example:

```
$ curl -s https://<device service
url>/apiv1/properties/<property_key>/triggers.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
}
}
]
```

8.4 GET *apiv1/properties/<property_key>/triggers/all[.format]*

Similar to the API **GET *apiv1/properties/<property_key>/triggers.xml***, but also includes device level triggers.

8.5 DELETE *apiv1/triggers/<key>[.format]*

Deletes the trigger specified by the key.

Input: none

Output: none

XML Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" https://<device service url>/apiv1/triggers/12.xml
```

JSON Example:

```
$ curl -X DELETE -H "Content-Type: application/json"
https://<device service url>/apiv1/triggers/404.json
```

8.6 GET *apiv1/triggers/all/by_user.xml*

Lists all the triggers of the current user for all accessible devices. It also gets triggers from the linked accounts.

Input: none

Output: array of triggers

XML Example:

```
$ curl -H"Authorization: auth_token<auth_token>" https://<device
service url>/apiv1/triggers/all/by_user.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<triggers type="array">
<trigger>
  <device-nickname>MyThermostat</device-nickname>
  <property-nickname>SmallInt</property-nickname>
  <compare-type>&gt;=</compare-type>
  <period>always</period>
  <trigger-type>compare_absolute</trigger-type>
  <base-type>integer</base-type>
  <value>60</value>
  <triggered-at type="datetime" nil="true"></triggered-at>
  <key type="integer">11</key>
  <property-key type="integer">9</property-key>
</trigger>
</triggers>
```

JSON Example:

```
$ curl -s https://<device service
url>/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
  }
}
]
```

8.7 TRIGGER APP

Applications are launched by triggers. Applications provide notification mechanisms.

You can customize emails that are sent from the Ayla Service with OEM logos and text, refer to the *Customizing Ayla Emails* document.

8.7.1 POST apiv1/triggers/<trigger_key>/trigger_apps[.format]

Creates an application action for the trigger key.

Input:

Mandatory:

name: supported applications: "email", "sms", "forward", "push_ios", "push_android".

Optional:

nickname: user friendly name

Type: String

username: Provided to an application. For email it is used as a greeting, but it is not used for sms

Type: String

repeat_freq: Used to define a repeat time for each trigger_app.

Default: 30 seconds.

Type: String

param1: application dependent.

For email, it is the email address of user to send the notification to.

For sms, it is the country code of destination phone number.

For "forward", it is the service name to which the property value must be forwarded.

For "push_ios", it is the registration id provided by Apple, without dashes and without spaces.

For "push_android," is the registration id provided by Google.

param2: application dependent

For email, it is unused

For sms, it is phone number

For "forward", it is the complete URL to which the property value must be forwarded.

For "push_ios," it is the key assigned to the PEM.

For "push_android," it is unused

param3: application dependent

For email, it is the additional text to be sent with the email message.

For sms, it is text to send

For "forward", it is the username for basic auth required for the service.

For "push_ios," it is an optional message that can be send to the user when this trigger is fired. (max length 100 chars)

For "push_android," it is an optional message that can be send to the user when this trigger is fired. (max length 100 chars)

param4: application dependent

For email, it is unused

For sms when `requires_acceptance` is true, it has to contain acceptance message with `[[accept_code]]` and `[[reject_code]]` tags. It is unused, when `requires_acceptance` is false (by default).

For “forward”, it is the password for basic auth required for the service.

For “push_ios,” it is unused.

For “push_android,” it is unused.

param5: application dependent

For email, it is unused

For sms ,it is unused

For “forward,” it is the context data that the user may wish to provide along with the forwarded datapoint.

For “push_ios,” it is unused.

For “push_android,” it is unused.

push_sound: Used for iOS and Android push apps (max length 50 chars). Contains the sound file name present on mobile app, for example, “default”

Type: String

push_mdata: Used for iOS and Android push apps (max length 100 chars)

A string, for example, “additional data” or “key1:value1, key2:value2”

Type: String

email_template_id: id given to the custom email template uploaded via the dashboard

Type: String

email_subject: Custom email subject to go into the custom email template

Validation: UTF-8 and none UTF-8

Type: String

email_body_html: Relevant email body with email template tags to go into the custom email template

Validation: UTF-8 and none UTF-8

Type: String

asset:: false in case it is a device level trigger, not a user’s asset.

default: true

Type:String

requires_acceptance: (optional, default: false): true to enable acceptance flow for SMS applications.

contact_id: ID of one of the current_user contacts. It is used for email and sms applications.

Type: Integer

Output: returns application created on success, else returns errors

XML Example email app:

```
curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" -d "<trigger-
app><name>email</name><username>Bob</username><repeat-
freq>50</repeat-freq><param1>bob@acme.com</param1><param2>This is
an email message</param2></trigger-app>" https://<device service
url>/apiv1/triggers/<trigger-key>/trigger\_apps.xml
```

Response:

```
<trigger_app>
  <name>email</name>
  <username>Bob</username>
  <nickname></nickname>
  <repeat-freq>50</repeat-freq>
  <param1>sudha@aylanetworks.com</param1>
  <param2>This is an email message</param2>
  <param3 nil="true"></param3>
  <param4 nil="true"></param4>
  <param5 nil="true"></param5>
  <key type="integer">677</key>
</trigger-app>
```

JSON Example (email app with custom email template):

```
curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/json" -d '{"trigger-app":{"name":
"email","username": "Bob","param1": "bob@acme.com","param2": "This
is an email message","email_template_id":
"ayla_trigger_app_template_01","email_subject": "Custom
Trigger","email_body_html":"<h2>Hello [[user_name]]!
[[property_name]] is now [[property_value]]!!</h2>"}'
https://<device service
url>/apiv1/triggers/<trigger key>/trigger\_apps.json
```

Response:

```
{
  "trigger_app": {
    "name": "email",
    "nickname": "",
    "username": "Bob",
    "param1": "bob@acme.com",
    "param2": "This is an email message",
```

```

        "email_template_id": "ayla_trigger_app_template_01",
        "email_subject": "Custom Trigger",
        "email_body_html": "<h2>Hello [[user_name]]!
[[property_name]] is now [[property_value]]!!</h2>"
    }
}

```

JSON Example (email app with contact):

```

curl -X POST -H"Authorization: auth_token <auth_token>" -H
"Content-Type: application/json" -d'{"trigger_app":{"name":
"email","username": "Bob", "param3": "This is an email
message","contact_id":10}}' https://<device service
url>/apiv1/triggers/<trigger key>/trigger apps.json

```

Response:

```

{
  "trigger_app": {
    "name": "email",
    "nickname": null,
    "key": 154,
    "trigger_key": 11,
    "username": "Bob",
    "param1": null,
    "param2": null,
    "param3": "This is an email message",
    "param4": null,
    "param5": null,
    "email_template_id": null,
    "email_body_html": null,
    "email_subject": null,
    "contact_id": 10
  }
}

```

XML SMS Example:

param 1 : country code
 param 2 : phone number
 param 3 : message

```

curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" -d

```

```
"<trigger-app><name>sms</name><nickname>sms-app1</nickname><param1>1</param1><param2>4085551111</param2><param3>Button pressed 2</param3></trigger-app>" https://<device service url>/apiv1/triggers/<trigger-key>/trigger\_apps.xml
```

Response:

```
<trigger_app>
  <name>sms</name>
  <nickname>sms-app1</nickname>
  <username>Bob</username>
  <param1>1</param1>
  <param2>4085551111</param2>
  <param3>Button pressed 2</param3>
  <param4 nil="true"></param4>
  <param5 nil="true"></param5>
  <key type="integer">55</key>
</trigger_app>
```

JSON SMS Example:

```
$ curl -s -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/json" -d '{"trigger_app":
{"name": "sms", "nickname": "sms-app1", "param1": "1",
"param2": "4085551111", "param3": "Hi. Pushbutton event"}}'
https://<device service url>/apiv1/triggers/<trigger key>/trigger\_apps.json
```

Response:

```
{
  "trigger_app": {
    "name": "sms",
    "nickname": "sms-app1",
    "param1": "1",
    "param2": "4085551111",
    "param3": "Hi. Pushbutton event",
    "param4": null,
    "param5": null,
    "username": null,
    "key": 263,
  }
}
```

JSON SMS Example (require acceptance flow):

```
$ curl -X POST -H "Authorization: auth_token <auth_token>" -H
"Content-Type: application/json" -d '{"trigger_app":{"name":"sms",
"param1":"1", "param2":"55555555", "param3":"Notification
message", "requires_acceptance":true, "param4":"Reply with
[[accept_code]] if you want to receive notifications or with
[[reject_code]] if you want to reject them"}}' https://<device
service url>/apiv1/triggers/<trigger\_key>/trigger\_apps.json
```

Response:

```
{"trigger_app":{"name":"sms","nickname":null,"key":25746,"trigger_key":19996,"us
ername":null,"repeat_freq":null,"param1":"1","param2":"55555555","param3":"Notif
ication message","param4":"Reply with [[accept_code]] if you want to receive
notifications or with [[reject_code]] if you want to reject
them","param5":null,"contact_id":null,"status":"pending"}}
```

XML “forward” Example:

```
param1 - service name
param2 - url
param3 - username
param4 - password
param5 - data
```

```
curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" -d
"<trigger-
app><name>forward</name><param1>postbin</param1><param2>http://htt
pbin.org/post</param2><param3>user1</param3><param4>passwd</param4
><param5>Forwarded by Ayla</param5></trigger-app>"
https://<device service url>/apiv1/triggers/<trigger-
key>/trigger\_apps.xml
```

Response:

```
<trigger-app>
  <name>forward</name>
  <nickname></nickname>
  <username nil="true"></username>
  <param1>postbin</param1>
  <param2>http://httpbin.org/post</param2>
  <param3>user1</param3>
  <param4>passwd</param4>
  <param5>Forwarded by Ayla</param5>
  <key type="integer">678</key>
```

</trigger-app>

JSON “forward” Example:

param1 - service name
param2 - url
param3 - username
param4 - password
param5 - data

```
curl -X POST -H "Content-Type: application/json" -d
'{"trigger_app":{"name":"forward","param1":"postbin","param2":"http://httpbin.org/post","param3":"user1","param4":"passwd","param5":"Forwarded by Ayla as json"}}' https://<device service url>/apiv1/triggers/<trigger key>/trigger apps.xml
```

Response:

```
{
  "trigger_app": {
    "name": "forward",
    "nickname": "",
    "param1": "postbin",
    "param2": "http://httpbin.org/post",
    "param3": "user1",
    "param4": "passwd",
    "param5": "Forwarded by Ayla as json",
    "username": null,
    "key": 680,
    "status": "accepted"
  }
}
```

JSON “push_ios” Example:

param1 - registration_id
param2 - application_id
param3 - custom message
param4 - unused
param5 - unused

```
curl -X POST -H "Content-Type: application/json" -d
'{"trigger_app":{"name":"push_ios","param1":"aEDjjdujek1233-i.....","param2":"my_key_id","param3":"custom message"}}'
```


<https://<device service url>/apiv1/triggers/<trigger key>/trigger apps.json>

Response:

```
{
  "trigger_app": {
    "name": "push",
    "nickname": "",
    "param1": "aEDjjdujek1233-i.....",
    "param2": "my_key_id",
    "param3": "custom_message",
    "status": "accepted"
  }
}
```

XML Example:

```
curl -X POST -H "Content-Type: application/json" -d
"<trigger_app><name>push_ios</name><param1>aEDjjdujek1233-i.....</param1><param2>my_key_id</param2><param3>custom message</param3></trigger_app>"
https://<device service url>/apiv1/triggers/<trigger key>/trigger apps.xml
```

Response:

```
<trigger_app>
  <name>push_ios</name>
  <nickname>push_ios</nickname>
  <param1>aEDjjdujek1233-i.....</param1>
  <param2>my_key_id</param2>
  <param3>custom message</param3>
</trigger_app>
```

JSON “push_android” Example:

```
param1 - registration_id
param2 - unused
param3 - custom message
param4 - unused
param5 - unused
```

```
curl -X POST -H "Content-Type: application/json" -d
'{"trigger_app":{"name":"push_android","param1":"aEDjjdujek1233-i.....","param3":"custom message"}}' https://<device service url>/apiv1/triggers/<trigger key>/trigger apps.json
```

Response:

```
{
  "trigger_app":
  {
    "name": "push",
    "nickname": "",
    "param1": "aEDjjdujek1233-i.....",
    "param3": "custom_message"
  }
}
```

XML Example:

```
curl -X POST -H "Content-Type: application/json" -d
"<trigger_app><name>push_android</name><param1>aEDjjdujek1233-
i.....</param1><param3>custom message</param3></trigger_app>"
https://<device service url>/apiv1/triggers/<trigger-
key>/trigger\_apps.xml
```

Response:

```
<trigger_app>
  <name>push_android</name>
  <nickname></nickname>
  <param1>aEDjjdujek1233-i.....</param1>
  <param3>custom message</param3>
</trigger_app>
```

XML SMS App Example with acceptance flow enabled:

param 1 : country code
 param 2 : phone number
 param 3 : message
 param4: acceptance message

```
curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" -d
'<trigger-app><name>sms</name><nickname>sms-
app1</nickname><param1>1</param1><param2>4085551111</param2><param
3>Button pressed 2</param3><param4>SEND [[accept_code]] to accept
this application or [[reject_code]] to reject
it</param4><requires-acceptance>1</requires-acceptance></trigger-
app>' https://<device service url>/apiv1/triggers/<trigger-
key>/trigger\_apps.xml
```

Response:

```
<trigger_app>
  <name>sms</name>
  <nickname>sms-app1</nickname>
  <username>Bob</username>
```

```

<param1>1</param1>
<param2>4085551111</param2>
<param3>Button pressed 2</param3>
<param4 nil="true"></param4>
<param5 nil="true"></param5>
<key type="integer">55</key>
<status>pending</status>
</trigger_app>

```

NOTE: The application would not be enabled until the target mobile number owner has not replied with the unique five digits number. The application status changes to accepted or rejected depending on the reply.

8.7.2 Data Forwarding Notes

When selecting “forward”, in addition to the context data (param5) passed in as an argument, the service adds device and property values to the data forwarded, relating to the device and the property for which the trigger was generated.

Device: dsn, product_name

Property: property name, latest updated value, base type, data_updated_at (UTC timestamp in iso8601 format)

The data is formatted as JSON.

Any external service expecting data forwarding from the Ayla service must implement the following:

The URL specified in param2 must accept an HTTP POST with

- Basic Auth using the username(param3) and password(param 4) input while setting up the trigger
- Ability to handle json in the following format:

Example for file property type:

```

{
  "device":{"dsn":"AC003222XXXX","product_name":"test_device_1000"},
  "property":{"name":"p4","base_type":"file","value":"https://<device service url>/devices/85/properties/377/datapoints/9420","data_updated_at":"2012-09-20T19:02:04Z"},
  "data":"Long Property Forwarded by Ayla"
}

```

Example for integer property type:

```

{
  "device":{"dsn":"ACW000001","product_name":"test_developer_staging_001"},

```

```
"property":{"name":"p1","base_type":"integer","value":1,"data_updated_at":"2012-09-11T06:08:22Z"},
"data":"Forwarded by Ayla as json"
}
```

8.7.3 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 very 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.

Use case 1 (the user knows the registration id):

- The user creates a new trigger app with param1 as nil.
- When the mobile is registered in Google or Apple, send a PUT method to update the trigger app with the registration id.
- When the mobile is registered in Apple, the param2 must be filled with the key assigned to the PEM when uploaded to User service.
- If you want to display a custom message, fill 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 User service.
- If you want to display a custom message, fill in the param3 with a custom message string.

8.7.4 PUT apiv1/trigger_apps/<trigger_app_id>[.format]

Updates an existing trigger application for a trigger with the trigger key.

Input:

Mandatory:

trigger_app_id: The Id of the trigger_app to modify.
 trigger_app: A hash with the parameters to modify.

XML Example:

```
curl -X PUT -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" -d
```

```
"<trigger-app><param3>new custom message</param3></trigger-app>"
https://<device service url>/apiv1/trigger\_apps/<triggerApp-id>.xml
```

JSON Example:

```
curl -X PUT -H "Content-Type: application/json" -d
"{\"trigger_app\":{\"param3\":\"new custom message\"}}\" https://<device service url>/apiv1/trigger\_apps/<triggerApp-id>.xml
```

8.7.5 GET apiv1/triggers/<trigger_key>/trigger_apps[.format]

Lists all the applications of the current user for a trigger with the trigger_key.

Input:

Mandatory:

trigger_key:

XML Example with Email and SMS:

```
$ curl https://<device service url>/apiv1/triggers/<trigger-key>/trigger\_apps.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<trigger-apps type="array">
  <trigger-app>
    <name>email</name>
    <nickname>email-for-bob</nickname>
    <username>bob</username>
    <param1>bob@acme.com</param1>
    <param2 nil="true"></param2>
    <param3 nil="true"></param3>
    <param4 nil="true"></param4>
    <param5 nil="true"></param5>
    <key type="integer">27</key>
    <trigger-key type="integer">42</trigger-key>
  </trigger-app>
  <trigger-app>
    <name>sms</name>
    <nickname></nickname>
```

```
<username>Bob</username>
<param1>1</param1>
<param2>4085551111</param2>
<param3>Button pressed</param3>
<param4 nil="true"></param4>
<param5 nil="true"></param5>
<key type="integer">28</key>
<trigger-key type="integer">42</trigger-key>
</trigger-app>
</trigger-apps>
```

JSON Example:

```
$ curl -s https://<device service
url>/apiv1/triggers/<trigger_key>/trigger_apps.json
```

Response:

```
[
  {
    "trigger_app": {
      "name": "sms",
      "nickname": "sms-for-button",
      "param1": "1",
      "param2": "4085551111",
      "param3": "Hi. Pushbutton event",
      "param4": null,
      "param5": null,
      "username": null,
      "key": 263,
      "trigger_key": 294
    }
  },
  {
    "trigger_app": {
      "name": "email",
      "nickname": "",
      "param1": "bob@acme.com",
      "param2": null,
      "param3": null,
      "param4": null,
      "param5": null,
      "username": "Bob",
      "key": 264,

```

```
    "trigger_key": 294
  }
}
```

8.7.6 GET apiv1/triggers/<trigger_key>/trigger_apps/all[.format]

Similar to the API GET apiv1/triggers/<trigger_key>/trigger_apps.xml, but also includes device level applications.

8.7.7 DELETE /trigger_apps/<triggerApp_key>[.format]

Deletes the application specified by the application key.

Input:

Mandatory:

triggerApp_key: corresponds to key of application to delete.

Output: none

XML Example:

```
$curl -X DELETE -H "Content-Type: application/xml" https://<device service url>/apiv1/trigger\_apps/<trigger-key>.xml
```

JSON Example:

```
$ curl -X DELETE -H "Content-Type: application/json"
https://<device service url>/apiv1/trigger_apps/<trigger_key>.json
```

9 DEVICE NOTIFICATIONS

Device specific events, such as, `On_Connect` or `IP_Change` can trigger device notification APIs. Notifications are sent to service URL with or without username or password similar to the `trigger_app` forwarding.

Note: These notifications can be used per device basis or can be set using templates from UI.

9.1 *POST* `apiv1/devices/<device_key>/notifications[.format]`

Creates a notification for the device specified by the device key.

Input:

Mandatory:

url (mandatory only for `on_connect` and `ip_change` types): Provide full URL address to which the property value must be forwarded.

notification_type: Select one of the following strings:

`"on_connect"`, `"ip_change"`, `"on_connection_lost"`,
`"on_connection_restore"`

Threshold: (mandatory for `on_connection_lost` and `on_connection_restore` types):

number of seconds for which the condition must be active before notification is sent. Minimum is 300 seconds.

Optional:

username: Username for basic auth if required for the service.

Type : String

password: Password for basic auth required for the service.

Type : String

message: Custom message for this notification type along with default message.

Type : String

device_nickname: A nickname for the associated device.

Type : String

Output: returns notification object created on success, else errors

JSON Example:

```
$ curl -X POST -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/json" -d
'{"notification":{"url":"http://www.aylanetworks.com",
"username":"aya123", "password":"testing123",
```



```
"notification_type":"ip_change", "message":"IP address
changed", "device_nickname" : "carrier 15"} }' https://<device
service url>/apiv1/devices/7/notifications.json
```

Response:

```
{"notification":{"id":15000005,"url":"http://www.aylanetworks.com","username":"aya
123","password":"testing123","notification_type":"ip_change","message":"IP
address changed","threshold":0,"device_key":7,"device_nickname":"carrier 15"}}
```

9.2 GET apiv1/devices/<device_key>/notifications[.format]

Lists all the current user's notifications for device with the device key.

Input:**Mandatory:**

device_key:

XML Example:

```
$ curl -H"Authorization: auth_token <auth-token>" https://<device
service url>/apiv1/devices/<device-key>/notifications.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<notifications type="array">
  <notification>
    <id type="integer">15000005</id>
    <url>http://www.aylanetworks.com</url>
    <username>aya123</username>
    <password>testing123</password>
    <notification-type>ip_change</notification-type>
    <message>IP address changed</message>
    <threshold type="integer">0</threshold>
    <device-key type="integer">7</device-key>
    <device-nickname>carrier 15</device-nickname>
  </notification>
</notifications>
```

9.3 GET apiv1/devices/<device_id>/notifications/all[.format]

Similar to the API **GET apiv1/devices/<device_id>/notification.xml**, but also includes device level notifications.

9.4 PUT apiv1/notifications/<id>[.format]

Updates a notification by the id.

Input:**Mandatory:**

url (mandatory only for on_connect and ip_change types): Enter a full URL address to which the property value must be forwarded.

notification-type: Select one of the following strings:

“on_connect”, “ip_change”, “on_connection_lost”,
“on_connection_restore”

Threshold (mandatory for on_connection_lost and on_connection_restore types):

number of seconds for which the condition must be active before notification is sent. Minimum is 300 seconds.

Optional:

username: Username for basic auth if required for the service.

password: Password for basic auth required for the service.
Message: Custom message for this notification type along with default message.

device_nickname: A nickname for the associated device.

Output: returns notification object updated on success, else errors

JSON Example:

```
$ curl -X PUT -H"Authorization: auth_token <auth_token>" -H
"Content-Type: application/json" -d
'{"notification":{"url":"http://www.aylanetworks.test.com",
"username":"aya123", "password":"testing123",
"notification_type":"on_connect",
"message":"Connected", "device_nickname":"carrier 7"}}'
https://<device service url>/apiv1/notifications/<id>.json
```

Response:

```
{"notification":{"id":15000005,"url":"http://www.aylanetworks.test.com","username":
"aya123","password":"testing123","notification_type":"on_connect","message":"Co
nnected","threshold":0,"device_key":7,"device_nickname":"carrier 7"}}
```

9.5 DELETE apiv1/notifications/<key>[.format]

Deletes the notification specified by the key.

Input:**Mandatory:**

key: notification key object via POST apiv1/devices/<device_key>/

Type : String

XML Example:

```
$curl -X DELETE -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/xml" https://<device service
url>/apiv1/notifications/12.xml
```

Curl JSON Example:

```
$ curl -X DELETE -H"Authorization: auth_token <auth-token>" -H
"Content-Type: application/json" https://<device service
url>/apiv1/notifications/12.json
```

Response:

200 – Success

9.5.1 Device Notification Notes

Along with the message for device notification, the service adds the following attributes to the data forward, relating to the device and the property for which the trigger was generated.

- Device: dsn, product_name, ip
- Property: property name ("device_notification")

The data is formatted as json.

Any external service expecting device notification from the Ayla service must implement the following:

The URL specified in 'url' must accept an HTTP POST with

- Basic Auth using the 'username' and 'password' input while setting up the notification

Example for on_connect notification type:

```
{
  "device":{"dsn":"AC0032222XXXX","product_name":"test_device_1",
    "ip":"108.233.XXX.XXX"},
  "property":{"name":"device_notification"},
  "data":["[Device Notification] Device Connected to IP[108.233.125.XXX]"]
}
```

Example for ip_change notification type:

```
{
```

```
"device":{"dsn":"AC0032222XXXX","product_name":"test_device_1",
"ip":"107.23.12.XX"},
"property":{"name":"device_notification"},
"data":["[Device Notification] IP address changed from [108.233.125.XXX] to
[107.23.12.XX]"
}
```

9.6 NOTIFICATION APPLICATIONS

The notification application stores actions to be executed when the notification is triggered.

9.6.1 GET /notifications/:notification_id/notification_apps

Gets a list of notification applications associated with the given notification.

Input:

Mandatory:

notification_id: the id of the notification

JSON Example:

```
curl -H"Authorization: auth_token <auth_token>" https://<device
service url>/apiv1/notifications/< notification_id
>/notification_apps.json
```

Response:

```
[{"id":27,"notification_id":461,"app_type":"email","nickname":"test","notification_app_parameters":{"username":"abc","email":"abc@aylanetworks.com","message":"Connection Lost"}}]
```

Response:

200 – Success, gets a list of notification applications associated with the notification.

404 – if the notification is not found.

9.6.2 GET /notifications/:notification_id/notification_apps/:id

Returns a specific notification application.

Input:

Mandatory:

notification_id: the id of the notification returned when a notification is created

Type: Integer

id: the id of the notification application generated when a notification app is created

Type: Integer

JSON Example:

```
curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/notifications/<notification_id>/notification_apps/<notification_app_id>.json
{"id":27,"notification_id":461,"app_type":"email","nickname":"test",
"notification_app_parameters":{"username":"abc","email":"<a href='\"mailto:abc@aylanetworks.com\"'>abc@aylanetworks.com>","message":"Connection Lost"}}
```

Response:

200 – Success, gets the description of the notification application

404 – Failure, if the notification or the notification_app is not found

9.6.3 POST /notifications/:notification_id/notification_apps

Creates a new notification application associated with the notification.

This API supports custom email templates, contact Ayla to get custom email templates. To send customized emails, add the custom email parameters to the notification_app_parameter structure as it shown in the example below.

Input:

Mandatory:

notification_id: The Id of the notification returned when a notification is created

Type: Integer

type: The type of the notification application. Can be: 'sms', 'email', 'push_ios', 'push_android'

Type: String

nickname: A nickname for the notification

Type: String

notification_app_parameters: A hash with the parameters associated with this notification application.

Email Params:

username: A username used as a greeting.

Type: String

email: The email address of user to send the notification to.

Type: String

message: Additional text to be sent.

Type: String

email_template_id: ID given to the custom email template uploaded via the dashboard

Type: Integer

email_subject: Custom email subject to go into the custom email template

Type: String

email_body_html: Relevant email body with email template tags to go into the custom email template

Type: String

Sms Params:

username: A username used as a greeting

Type: String

country_code: The country code of user to send the notification to.

Type: String

phone_number: The phone number of user to send the notification to.

Type: Integer

message: Additional text to be sent

Type: String

Push iOS Params:

registration_id: An Id linking a phone/tablet to the Apple/Google notification service

Type: Integer

application_id:

message: Additional text to be sent.

Type: String

push_sound: Used for iOS push apps

Validation: max length 50 chars

Contains the sound file name present on mobile app, for example: *"default"*

push_mdata: Used for iOS push apps (max length 100 chars)

Type: string, for example: "additional data" or

"key1:value1,key2:value2"

Push Android Params:

registration_id: An id linking a phone/tablet to the Apple/Google notification service

Type: Integer

message: Additional text to be sent.

Type: String

push_sound: Used for Android push apps
 Validation: max length 50 chars
 Contains the sound file name present on mobile app, for example:
"default"
 Type: String
 push_mdata: Used for Android push apps (max length 100 chars)
 A string, for example, "additional data" or *"key1:value1,key2:value2"*

Response:

200 – Success, it creates the new notification application
 404 – Failure, if the notification is not found
 422 – Failure, if there is a problem with parameters

JSON Example:

```
curl -X POST -H"Authorization: auth_token <auth_token>" -
d{"notification_app":{"app_type":"email","nickname":"test","notif
ication_app_parameters":{"username":"abc","email":"abc@aylanetwork
s.com"}}} -H"Content-Type: application/json" https://<device
service
url>/apiv1/notifications/<notification_id>/notification_apps.json
```

Response:

```
{"notification_app":{"id":7,"notification_id":15000005,"app_type":"email","nickname":"test",
"notification_app_parameters":{"username":"abc","email":"abc@aylanetworks.com"}}}
```

JSON Example (with custom email params):

```
curl -X POST -H"Authorization: auth_token <auth_token>" -
d{"notification_app":{"app_type":"email","notification_app_parame
ters":{"username":"abc","email":"abc@aylanetworks.com","message":"
Connection Lost","email_template_id":
"notification_app_template_01","email_subject": "[[product_value]]
notification","email_body_html":"<h2>Hello [[user_name]]!
[[device_product_name]] is now [[property_value]]!!</h2>"}}}
" -H"Content-Type: application/json" https://<device service
url>/apiv1/notifications/<notification_id>/notification_apps.json
```

9.6.4 PUT /notifications/:notification_id/notification_apps/:id

Updates a specific notification application.

Input:

Mandatory:

notification_id: notification application id via POST
 /notifications/:notification_id/notification_apps
 Type: Integer

Other parameters:

type: new notification type ('sms', 'email', 'push_ios', 'push_android')
 Type: String
 nickname: notification nickname
 Type: String
 params: a hash with the associated parameters

JSON Example:

```
curl -X PUT -H"Authorization: auth_token <auth_token>" -
d{"notification_app":{"app_type":"email","nickname":"test","notif
ication_app_parameters":{"username":"abc","email":"abc@aylanetwork
s.com"}}} -H"Content-Type: application/json" https://<device
service url>/apiv1/notifications/<notification_id>
/notification_apps/<id>.json
```

Response:

```
{"notification_app":{"id":7,"notification_id":15000005,"app_type":"email","nickname":"test",
"notification_app_parameters":{"username":"abc","email":"abc@aylanetworks.com"}}}
```

JSON Example (with custom email params):

```
curl -X PUT -H"Authorization: auth_token <auth_token>" -
d{"notification_app":{"app_type":"email","notification_app_parame
ters":{"username":"abc","email":"def@aylanetworks.com","message":"
No Connection","email_template_id":
"notification_app_template_02","email_subject": "[[product_value]]
notification email","email_body_html":"<h2>Hello [[user_name]]!
[[device_product_name]] is now [[property_value]]!!</h2>"}}}
" -H"Content-Type: application/json" https://<device service
url>/apiv1/notifications/<notification_id>
/notification_apps/<id>.json
```

Response:

200 – Success, updates application id
 404 – Failure, the notification or the notification_app is not found
 422 – Problem with the parameters

9.6.5 DELETE /notifications/:notification_id/notification_apps/:id

Deletes a specific notification application.

Input:

Mandatory:

notification_id: notification id returned when a notification is created

Type: Integer

id: notification application id returned when a notification app is created

Type: Integer

JSON Example:

```
curl -X DELETE -H"Authorization: auth_token <auth_token>"  
https://<device service url>/apiv1/notifications/<notification_id>  
>/notification_apps/27.json
```

Response:

200 – Success, it deletes a specific notification application

404 – Failure, the notification or the notification_app is not found

10 SHARE APIs

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

- 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 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 can't have more than one share for the same `resource_name` and `resource_id`.

10.1 GET `/api/v1/users/shares/`

Returns a list of not expired owned shares.

Input:

Mandatory:

`resource_name`: Name of the resource class, mandatory if `resource_id` parameter included in the query, eg : device)

Type: String

Optional:

`owner_id`: id of the share's owner.

Default: current user

Type: Integer

`user_id`: id of the share's target user

Type: Integer

`resource_id`: id of the resource

`expired`: if it set to true it brings expired shares.

Default: false

Type: String

`accepted`: if it set to false it only brings acceptance pending shares, true to bring only accepted shares and both to bring pending and accepted.

Default: true

Type: String

XML Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<user service url>/api/v1/users/shares.xml
```

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<User Service URL>/api/v1/users/shares.json
```

JSON Example with resource_id and resource_name:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"
https://<user service url>/api/v1/users/shares.json?resource_name=Device&resource_id=DSN0011
```

Response:

```
[{"share": {"conditions": null, "created_at": "2013-09-26T19:57:27Z", "end_date_at": "2013-09-29T00:01:00Z", "grant_id": 19, "id": 2, "operation": "read", "owner_id": 1, "resource_id": "AC000W000000XXXX", "resource_name": "device", "start_date_at": "2013-09-26T19:57:27Z", "status": "pending", "updated_at": "2013-09-26T19:57:27Z", "user_id": 1}}]
```

JSON Example with owner_id:

OEM admin can get shares of other users using user_id

```
curl -X GET -H "Authorization: auth_token <auth_token>"
https://<user service url>/api/v1/users/shares.json?owner_id=<user_id>
```

Response:

```
[{"share": {"accepted": true, "accepted_at": null, "created_at": "2014-09-09T00:30:20Z", "end_date_at": null, "grant_id": 77, "id": 509, "operation": "write", "owner_id": 36, "resource_id": "", "resource_name": "device", "start_date_at": null, "updated_at": "2014-09-09T00:30:20Z", "user_id": 83, "owner_profile": {"firstname": "", "lastname": "", "email": ""}, "user_email": "rgrg@test.com", "user_profile": {"firstname": "fl", "lastname": "ln", "email": "dfgh@gmail.com"}}}]
```

JSON invalid query (resource_name missing):

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service  
url>/api/v1/users/shares.json?resource\_id=DSN0011
```

Response:

200 – Success
422 – Failure, Unprocessable entity

10.2 GET /api/v1/users/shares/received

Returns a list of not expired received shares.

Input:**Mandatory:**

resource_name: Name of the resource class, mandatory if resource_id
parameter included in the query, (Example,: device) Otherwise, optional.
Type: String

Optional:

user_id: User id of the share's target user
default: current_user
Type: Integer
owner_id: User id of the share's owner
Type: String resource_id: Id of the resource. If resource is device
then resource_id is device dsn
Type: String
expired: true to also bring expired shares,
default: false
Type: String
accepted: false to only bring acceptance pending shares, true to bring only
accepted shares and both to bring pending and accepted,
default: true
Type: String

XML Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/received.xml
```

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/received.json
```

JSON Example with resource_id and resource_name:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/received.json?resource\_name=Device&resource\_id=DSN0011
```

Response:

```
[{"share": {"conditions": null, "created_at": "2013-09-26T19:57:27Z", "end_date_at": "2013-09-29T00:01:00Z", "grant_id": 19, "id": 2, "operation": "read", "owner_id": 1, "resource_id": "AC000W000000XXXX", "resource_name": "device", "start_date_at": "2013-09-26T19:57:27Z", "status": "pending", "updated_at": "2013-09-26T19:57:27Z", "user_id": 1}}]
```

JSON Invalid query (resource_name missing):

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/received.json?resource\_id=DSN0011
```

Response:

200 – Success
422 – Failure, Unprocessable entity

10.3 GET /api/v1/users/shares/:id

Returns a share id.

Input:**Mandatory:**

id: Share id object returned when a share is created
Type: Integer

Optional:

owner_id: id of the share's owner,
default: current user
Type: Integer

XML Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/<id>.xml
```

JSON Example:

```
$ curl -X GET -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/<id>.json
```

Response:

```
{
  "share": {
    "conditions": null,
    "created_at": "2013-09-26T19:57:27Z",
    "end_date_at": "2014-01-26T12:00:00Z",
    "grant_id": 19,
    "id": 2,
    "operation": "read",
    "owner_id": 1,
    "resource_id": "AC000W000000XXXX",
    "resource_name": "device",
    "start_date_at": null,
    "status": "pending",
    "updated_at": "2013-09-26T19:57:27Z",
    "user_id": 1
  }
}
```

Response:

200 – Success

404 – Failure, not found, the share doesn't exist

10.4 POST /api/v1/users/shares/

Creates a new share or a list of new shares with the provided parameters. A user can't have more than one share for the same resource_name and resource_id.

Input:**Mandatory:**

share: A hash with the share to be created

- resource_name: Resource class name eg: Device
Type: String
- resource_id: Resource id. If resource is a device, then resource id will be device dsn
Type: String
- role_name: Role's name with which you want to create a share
Type: String
- user_email: email of the user to which you want to grant permission
Type: String

Optional:

- owner_id: id of the share's owner
default: current user
Type: String
- operation: read or write , not required when role is present
Type: String
Default: read
- start_date_at: UTC DateTime at which the share begins in the format YYYY-MM-DDTHH:MM:SSZ.
Default : takes immediate effect
Type: String
- end_date_at: UTC DateTime at which the share ends in the format YYYY-MM-DDTHH:MM:SSZ.
Default value: persists until the share or named resource is deleted
Type: String

email_template_id: URL parameter - custom email template to be used while sending the email to the target user.

Type: String

One Share

JSON Examples (with email_template_id):

```
$ curl -X POST -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" -d "<share>
<resource_name>Device</resource_name>
<resource_id>AC000W00000000XXXX</resource_id>
<user_email>a@b.com</user_email><start_date_at>2104-03-17
12:00:00</start_date_at><end_date_at>2014-03-17
12:30:00</end_date_at></share>" https://<user service
url>/api/v1/users/shares.json?email\_template\_id=oem1\_template3
```

Response:

```
{ "share": { "conditions": null, "created_at": "2013-09-
26T19:57:27Z", "end_date_at": null, "grant_id": 19, "id": 2, "operation": "read", "owner_id": 1, "res
ource_id": "AC000W00000000XXXX", "resource_name": "device", "start_date_at": null, "status":
"pending", "updated_at": "2013-09-26T19:57:27Z", "user_email": "a@b.com" } }
```

Response:

200 – Success

422 – Failure, user_id is already granted for that resource

Multiple Shares

XML Examples (with email_template_id):

```
$ curl -X POST -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" -d '<shares
type="array"><share> <resource_name>Device</resource_name>
<resource_id>AC000W00000000XX</resource_id>
<user_email><testemail@testemail.com></user_email><start_date_at>2
104-03-17 12:00:00</start_date_at><end_date_at>2014-03-17
12:30:00</end_date_at></share><share>
<resource_name>Device</resource_name>
<resource_id>AC000W00000000XX</resource_id>
<user_email><testemail@testemail.com></user_email><start_date_at>2
104-03-17 12:00:00</start_date_at><end_date_at>2014-03-17
12:30:00</end_date_at></share></shares>' https://<user service
url>/api/v1/users/shares.xml?email\_template\_id=oem1\_template3
```

Response

<?xml version="1.0" encoding="UTF-8"?>

```
<shares type="array">
  <share>
    <accepted type="boolean">true</accepted>
    <accepted-at type="datetime" nil="true"/>
    <created-at type="datetime">2014-12-10T19:31:04Z</created-at>
    <end-date-at type="datetime">2014-03-17T12:30:00Z</end-date-at>
    <grant-id type="integer">3</grant-id>
    <id type="integer">3</id>
    <operation>read</operation>
    <owner-id type="integer">1000003</owner-id>
    <resource-id>AC000W00000000XX</resource-id>
    <resource-name>device</resource-name>
    <start-date-at type="datetime">2104-03-17T12:00:00Z</start-date-at>
    <updated-at type="datetime">2014-12-10T19:31:04Z</updated-at>
    <user-id type="integer">1000006</user-id>
    <owner-profile>
      <firstname>fn</firstname>
      <lastname>ln</lastname>
      <email><testemail@testemail.com></email>
    </owner-profile>
    <user-email><testemail@testemail.com></user-email>
    <user-profile>
      <firstname>fn</firstname>
      <lastname>ln</lastname>
      <email>testemail@testemail.com</email>
    </user-profile>
  </share>
  <share>
    <accepted type="boolean">true</accepted>
    <accepted-at type="datetime" nil="true"/>
    <created-at type="datetime">2014-12-10T19:31:05Z</created-at>
    <end-date-at type="datetime">2014-03-17T12:30:00Z</end-date-at>
    <grant-id type="integer">4</grant-id>
    <id type="integer">4</id>
    <operation>read</operation>
    <owner-id type="integer">10000XX</owner-id>
    <resource-id>AC000W00000000XX</resource-id>
    <resource-name>device</resource-name>
    <start-date-at type="datetime">2104-03-17T12:00:00Z</start-date-at>
    <updated-at type="datetime">2014-12-10T19:31:05Z</updated-at>
    <user-id type="integer">1000002</user-id>
    <owner-profile>
```



```

    <firstname><fn></firstname>
    <lastname><ln></lastname>
    <email><testemail@testemail.com></email>
  </owner-profile>
  <user-email><testemail@testemail.com></user-email>
  <user-profile>
    <firstname>Kid</firstname>
    <lastname>ayla</lastname>
    <email>testemail@testemail.com</email>
  </user-profile>
</share>
</shares>

```

Response:

201 – Success

422 – Failure, user_id is already granted for that resource

When at least one share fails, the response show the errors list for each share sent in the request. When errors tag is empty, the request was successful for that share.

```

<?xml version="1.0" encoding="UTF-8"?>
<objects type="array">
  <object>
    <resource-id>AC000WTXXXXXXXXXX</resource-id>
    <errors/>
  </object>
  <object>
    <resource-id>AC000W00000000XX</resource-id>
    <errors>
      <error>User can't be blank</error>
      <error>User doesn't exist</error>
    </errors>
  </object>
</objects>

```

JSON Example:

```

$ curl -X POST -H "Content-Type: application/json" -H
"Authorization: auth_token <auth_token>" -d '{"shares":
[{"resource_name": "Device", "resource_id": "AC000W00000000XX",
"user_email": "testemail@testemail.com", "start_date_at": "2104-
03-17T12: 00: 00", "end_date_at": "2014-03-17T12: 30: 00"}],

```

```
{"resource_name": "Device", "resource_id": "AC000WTXXXXXXXXX",  
"user_email": "testemail@testemail.com", "start_date_at": "2104-03-  
1712: 00: 00", "end_date_at": "2014-03-1712: 30: 00"}]}
```

https://<user_service_url>/api/v1/users/shares.json?email_template_id=oem1_template3

Response

```
[  
  {  
    "share": {  
      "accepted": true,  
      "accepted_at": null,  
      "created_at": "2014-12-11T20:01:34Z",  
      "end_date_at": null,  
      "grant_id": 15,  
      "id": 15,  
      "operation": "read",  
      "owner_id": 1000003,  
      "resource_id": "AC000WTXXXXXXXXX",  
      "resource_name": "device",  
      "start_date_at": null,  
      "updated_at": "2014-12-11T20:01:34Z",  
      "user_id": 1,  
      "owner_profile": {  
        "firstname": "fn",  
        "lastname": "ln",  
        "email": "testemail@testemail.com "  
      },  
      "user_email": "adminemail@adminemail.com",  
      "user_profile": {  
        "firstname": "Doctor",  
        "lastname": "Doom",  
        "email": "testemail@testemail.com"  
      }  
    },  
  },  
  {  
    "share": {  
      "accepted": true,  
      "accepted_at": null,  
      "created_at": "2014-12-11T20:01:35Z",  
      "end_date_at": null,
```

```

    "grant_id": 16,
    "id": 16,
    "operation": "read",
    "owner_id": 1000003,
    "resource_id": "AC000WTXXXXXXXXX",
    "resource_name": "device",
    "start_date_at": null,
    "updated_at": "2014-12-11T20:01:35Z",
    "user_id": 1000002,
    "owner_profile": {
      "firstname": "fn",
      "lastname": "ln",
      "email": "testemail@testemail.com"
    },
    "user_email": "testemail@testemail.com",
    "user_profile": {
      "firstname": "Kid",
      "lastname": "ayla",
      "email": "testemail@testemail.com"
    }
  }
}
]

```

Response:

201 – Success

422 – Failure, user_id is already granted for that resource

When at least one share fails the response show the errors list for each share sent in the request. When errors tag is empty the request is considered successful for that share.

```

[
  {
    "resource_id": "AC000WTXXXXXXXXX",
    "errors": {
      "user_id": [
        "User ID is already granted for that resource"
      ]
    }
  },
  {
    "resource_id": "AC000W00000XXXXX",
    "errors": {}
  }
]

```

```
}
]
```

10.5 PUT /api/v1/users/shares/:id

Updates a share with the provided parameters.

Input:

Mandatory:

id: share id generated when a share is created

Type: Integer

Optional:

owner_id: id of the share's owner,

default: current user

Type: Integer

share: A hash with the share to be updated:

role_name: role's name should be present in the 'role' matrix

Type: String

operation: read or write

Type: String

start_date_at: date when share starts. Example: "2015-05-17T12:00:00Z"

Type: String

end_date_at: date when share expires. Example: "2115-05-17T12:00:00Z"

Type: String

email_template_id: email eemplate's id. Example:

"ayla_confirmation_template_05"

Type: String

user_email: email to identify a target user. Requires user_oem_id param if present. Example: "a_registered_ayla_user@gmail.com"

Type: String

user_oem_id: id of the oem user

Type: Integer

XML Example:

```
$ curl -X PUT -H "Content-Type: application/xml" -H
"Authorization: auth_token <auth_token>" https://<user service url>/api/v1/users/shares/1.xml -d
"<share><operation>write</operation></share>"
```

Response

```
[{"share": {"conditions": null, "created_at": "2013-09-
26T19:57:27Z", "end_date_at": null, "grant_id": 19, "id": 2, "operation": "write", "owner_id": 1, "re
```

```
source_id":"AC000W000000XXXX","resource_name":"device","start_date_at":null,"status":"pending","updated_at":"2013-09-26T19:57:27Z","user_id":1}}]
```

Response:

- 200 – Success
- 422 – Failure, operation is not valid
- 404 – Not found, the share does not exist

10.6 DELETE /api/v1/users/shares/:id

Deletes a share.

Input:**Mandatory:**

id: a share id

Optional:

owner_id: id of the share's owner, default, current user

JSON Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/1.json
```

Response :

- 204 – Success
- 404 – Failure, not found, the share does not exist
- 422 – Failure, something went wrong destroying the mirror Grant in the target service

10.7 DELETE with resource_name & resource_id**DELETE**

```
/api/v1/users/shares?resource_name=<resource_name>&resource_id=<resource_id>
```

Deletes shares associated with the resource name and the resource id.

Input**Mandatory:**

owner_id: User id of the share's owner

user_id: User id of the share's target user

resource_name: name of the resource class, for example, device

resource_id: id of the resource (Device DSN for device)

JSON Example:

```
$ curl -X DELETE -H "Authorization: auth_token <auth_token>"  
https://<user service url>/api/v1/users/shares/<share id>.json?resource name=device&resource id=DSN011&user id=642
```

XML Example:

```
$ curl -X DELETE -H "Authorization: auth_token  
<auth_token>"https://<user service url>/api/v1/users/shares/<share id>.xml?resource name=device&resource id=DSN011&user id=642
```

Response:

- 204 – Success
- 404 – Failure, not found, the share does not exist with provided parameters
- 422 – Failure, resource_id and resource_name parameters are not provided

11 LINKED USER ACCOUNTS

Users are able to link two or more user accounts across different OEMs. For each OEM, the user must authenticate using the credentials for that OEM. Users who want to link accounts, they need to initiate the linked accounts process. OEM users also have an ability to select OEMs they want to link or delink accounts.

The benefits of linking user account for users as follows:

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

11.1 *POST /api/v1/users/link[.format]*

Creates a link to a user account. The user should be signed in with the super app account.

Input**Mandatory:**

user:

email: email address of the account to be linked with
password: password of the account
origin_oem_id: origin_oem_id of the account

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:

- 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

11.2 DELETE /api/v1/user//link/:user_id[:format]

Deletes link to a user account. The user should be signed in with the super app account.

Input**Mandatory:**

user_id : user_id of the account to delete the link with

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, deleted a link to a super app
- 401 – Not Authorized
- 404 – Failure, invalid user id
- 422 – Failure, no link found with the provided user id

12 TIMEZONES

All supported timezones are listed at the ten of this section.

You can use a specific timeone for the following calls:

1. GET `apiv1/time_zones.json?tz_id=` : To get the details on the Timezone
2. POST `apiv1/devices/<device_id>/time_zones.xml` or PUT `apiv1/devices/<device_id>/time_zones[.format]` : To set timezone for the device.

All these API details can be found in our Device Service Specification doc. It has elaborate examples, so please refer for further details

12.1 GET `apiv1/devices/<device_id>/time_zones[.format]`

Gets the time zone of a device with the device id. If the time zone is not set, it returns a default time zone with nil key and the utc offset.

Input:

Mandatory:

device id: Device identifier. Obtain from GET `/devices`

Type: Integer

XML Example:

```
$ curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/devices/<device-id>/time\_zones.xml
```

Response when device time_zone is set:

```
<?xml version="1.0" encoding="UTF-8"?><time-zone>
  <utc-offset>-03:00</utc-offset>
  <dst type="boolean">true</dst>
  <dst-active type="boolean">true</dst-active>
  <dst-next-change-date>2014-01-01</dst-next-change-date>
  <dst-next-change-time>05:30</dst-next-change-time>
  <key type="integer">1</key>
</time-zone>
```


Response when device time_zone is not set:

```
<?xml version="1.0" encoding="UTF-8"?>
<time-zone>
  <utc-offset nil="true"></utc-offset>
  <dst type="boolean">false</dst>
  <dst-active type="boolean">false</dst-active>
  <dst-next-change-date nil="true"></dst-next-change-date>
  <dst-next-change-time nil="true"></dst-next-change-time>
  <key type="yaml" nil="true"></key>
</time-zone>
```

JSON Example:

```
$ curl -H"Authorization: auth_token <auth-token">
https://<device_service
url>/apiv1/devices/<device_id>/time_zones.json
```

Response when device time_zone is set:

```
{
  "time_zone": {
    "dst": true,
    "dst_active": true,
    "dst_next_change_date": "2014-01-01",
    "dst_next_change_time": "05:30",
    "utc_offset": "-03:00",
    "key": 1
  }
}
```

Response when device time_zone is not set:

```
{
  "time_zone": {
    "dst": false,
    "dst_active": false,
    "dst_next_change_date": null,
    "dst_next_change_time": null,
    "utc_offset": null,
    "key": null
  }
}
```

Response:

200 – Success

404 – Failure, device not found

12.2 *POST* `apiv1/devices/<device_id>/time_zones[.format]`

Creates (device has no time_zone set) or updates (device already has time_zone set) the time_zone of a device with the device id.

Input:

Mandatory:

device_id: The device identifier

Type: Integer

utc_offset: string which specifies utc offset.

Type: String

Validation: +HH:MM or -HH:MM. For example +05:00 or -03:00.

Optional:

dst: specifies if the location follows DST.

Type: Boolean

Default: false.

dst_active: specifies if DST is currently active.

Type: Boolean

Default: false.

dst_next_change_date: specifies next DST state change from active/inactive OR from inactive/active.

Type: Boolean

Validation: yyyy-mm-dd

tz_id :identifier for the timezone, for example, "America/New_York"

Type: String

Note: Although tz_id is optional, if it is not blank, it needs to correlate with the utc_offset. If not, the POST is rejected. It is acceptable for tz_id to be blank.

XML Example:

```
$ curl -k -X POST -H"Content-Type: application/xml" -d"<time-
zone><utc-offset>-03:00</utc-offset><dst>true</dst><dst-
active>true</dst-active><dst_next_change_date>2014-01-
01</dst_next_change_date></time-zone>" -H"Authorization:
auth_token <auth_token>" https://<device service
url>/apiv1/devices/<device-id>/time\_zones.xml
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<time-zone>
```

```
<utc-offset>-03:00</utc-offset>
```

```

<dst type="boolean">true</dst>
<dst-active type="boolean">true</dst-active>
<dst-next-change-date>2014-01-01</dst-next-change-date>
<dst-next-change-time nil="true"></dst-next-change-time>
<key type="integer">1</key>
</time-zone>

```

JSON Example:

```

$ curl -k -X POST -H"Content-type: application/json" -
d'{"time_zone": {"utc_offset": "-03:00", "dst": true,
"dst_active": true, "dst_next_change_date": "2014-01-01"}}' -
H"Authorization: auth_token <auth_token>"https://<device service
url>/apiv1/devices/<device_id>/time_zones.json

```

Response:

```

{
  "time_zone": {
    "dst": true,
    "dst_active": true,
    "dst_next_change_date": "2014-01-01",
    "dst_next_change_time": null,
    "utc_offset": "-03:00",
    "key": 1
  }
}

```

Response:

200 – Success, created or updated time_zone
 422 – Failure, unprocessable entity

12.3 PUT apiv1/devices/<device_id>/time_zones[.format]

Updates when the device has time_zone set ,or creates when device already has time_zone set, the time_zone of a device with the device id.

Input:

Mandatory:

device_id: The device identifier Returned from GET devices
 Type: Integer
 tz_id: Standard time zone
 Type: string.

For example, "America/Los_Angeles"

Note: DST attributes are updated, based on whether the timezone has DST or not..

XML Example:

```
curl -X PUT -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/xml" -d'<tz_id>Asia/Kolkata</tz_id>'
https://<device service url>/apiv1/devices/<device-id>/time\_zones.xml
```

Response:

```
<time-zone>
  <utc-offset>+05:30</utc-offset>
  <dst type="boolean">false</dst>
  <dst-active type="boolean">false</dst-active>
  <dst-next-change-date nil="true"></dst-next-change-date>
  <dst-next-change-time nil="true"></dst-next-change-time>
  <tz_id>Asia/Kolkata</tz_id>
  <key type="integer">1</key>
</time-zone>
```

JSON Example:

```
curl -X PUT -H"Authorization: auth_token <auth_token>" -H"Content-Type: application/json" -d'{"tz_id":"America/Los_Angeles"}'
https://<device service url>/apiv1/devices/<device id>/time\_zones.json
```

Response:

```
{"time_zone":{"dst":
true,"dst_active":true,"dst_next_change_date":"2013-11-
03","dst_next_change_time":"02:00","utc_offset":"-
08:00","tz_id":"America/Los_Angeles","key":1}}
```

Response:

200 – Success, created or updated time_zone
422 – Failure, unprocessable entity

12.4 GET apiv1/time_zones?tz_id=

Fetches a hash of timezone information given a valid timezone id. It is primarily intended for UI usage.

Input:**Mandatory:**

tz_id: Standard time zone identifier string.

For example, "America/Los_Angeles"

Type: String**XML Example:**

```
curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/time\_zones.xml?tz\_id="America/New\_York"
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<hash>
  <Base UTC Offset>-05:00</Base UTC Offset>
  <DST type="boolean">true</DST>
  <DST Active type="boolean">true</DST Active>
  <DST Next Change Date>2013-11-03</DST Next Change Date>
  <DST Next Change Time>02:00</DST Next Change Time>
</hash>
```

JSON Example:

```
curl -H"Authorization: auth_token <auth_token>" https://<device service url>/apiv1/time\_zones.json?tz\_id="Asia/Tokyo"
```

Response:

```
{"Base UTC Offset":"+09:00","DST":false,"DST Active":false,"DST Next Change Date":null,"DST Next Change Time":null}
```

Response:

200 – Success, found timezone and returning info

404 – Failure, not found

12.5 List of all supported Timezones

Africa/Abidjan

Africa/Accra

Africa/Addis_Ababa

Africa/Algiers

Africa/Asmara

Africa/Asmera

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

Africa/Porto-Novo
Africa/Sao_Tome
Africa/Timbuktu
Africa/Tripoli
Africa/Tunis
Africa/Windhoek
America/Adak
America/Anchorage
America/Anguilla
America/Antigua
America/Araguaina
America/Argentina/Buenos_Aires
America/Argentina/Catamarca
America/Argentina/ComodRivadavia
America/Argentina/Cordoba
America/Argentina/Jujuy
America/Argentina/La_Rioja
America/Argentina/Mendoza
America/Argentina/Rio_Gallegos
America/Argentina/Salta
America/Argentina/San_Juan
America/Argentina/San_Luis
America/Argentina/Tucuman
America/Argentina/Ushuaia
America/Aruba
America/Asuncion
America/Atikokan
America/Atka
America/Bahia
America/Bahia_Banderas
America/Barbados
America/Belem
America/Belize
America/Blanc-Sablon
America/Boa_Vista
America/Bogota
America/Boise
America/Buenos_Aires
America/Cambridge_Bay
America/Campo_Grande
America/Cancun
America/Caracas

America/Catamarca
America/Cayenne
America/Cayman
America/Chicago
America/Chihuahua
America/Coral_Harbour
America/Cordoba
America/Costa_Rica
America/Creston
America/Cuiaba
America/Curacao
America/Danmarkshavn
America/Dawson
America/Dawson_Creek
America/Denver
America/Detroit
America/Dominica
America/Edmonton
America/Eirunepe
America/El_Salvador
America/Ensenada
America/Fort_Wayne
America/Fortaleza
America/Glace_Bay
America/Godthab
America/Goose_Bay
America/Grand_Turk
America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Hermosillo
America/Indiana/Indianapolis
America/Indiana/Knox
America/Indiana/Marengo
America/Indiana/Petersburg
America/Indiana/Tell_City
America/Indiana/Vevay
America/Indiana/Vincennes

America/Indiana/Winamac
America/Indianapolis
America/Inuvik
America/Iqaluit
America/Jamaica
America/Jujuy
America/Juneau
America/Kentucky/Louisville
America/Kentucky/Monticello
America/Knox_IN
America/Kralendijk
America/La_Paz
America/Lima
America/Los_Angeles
America/Louisville
America/Lower_Princes
America/Maceio
America/Managua
America/Manaus
America/Marigot
America/Martinique
America/Matamoros
America/Mazatlan
America/Mendoza
America/Menominee
America/Merida
America/Metlakatla
America/Mexico_City
America/Miquelon
America/Moncton
America/Monterrey
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
America/Nipigon
America/Nome
America/Noronha
America/North_Dakota/Beulah
America/North_Dakota/Center
America/North_Dakota/New_Salem

America/Ojinaga
America/Panama
America/Pangnirtung
America/Paramaribo
America/Phoenix
America/Port-au-Prince
America/Port_of_Spain
America/Porto_Acre
America/Porto_Velho
America/Puerto_Rico
America/Rainy_River
America/Rankin_Inlet
America/Recife
America/Regina
America/Resolute
America/Rio_Branco
America/Rosario
America/Santa_Isabel
America/Santarem
America/Santiago
America/Santo_Domingo
America/Sao_Paulo
America/Scoresbysund
America/Shiprock
America/Sitka
America/St_Barthelemy
America/St_Johns
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Swift_Current
America/Tegucigalpa
America/Thule
America/Thunder_Bay
America/Tijuana
America/Toronto
America/Tortola
America/Vancouver
America/Virgin
America/Whitehorse
America/Winnipeg

America/Yakutat
America/Yellowknife
Antarctica/Casey
Antarctica/Davis
Antarctica/DumontDUrville
Antarctica/Macquarie
Antarctica/Mawson
Antarctica/McMurdo
Antarctica/Palmer
Antarctica/Rothera
Antarctica/South_Pole
Antarctica/Syowa
Antarctica/Vostok
Arctic/Longyearbyen
Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
Asia/Calcutta
Asia/Choibalsan
Asia/Chongqing
Asia/Chungking
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dili
Asia/Dubai
Asia/Dushanbe
Asia/Gaza
Asia/Harbin

Asia/Hebron
Asia/Ho_Chi_Minh
Asia/Hong_Kong
Asia/Hovd
Asia/Irkutsk
Asia/Istanbul
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul
Asia/Kamchatka
Asia/Karachi
Asia/Kashgar
Asia/Kathmandu
Asia/Katmandu
Asia/Khandyga
Asia/Kolkata
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuching
Asia/Kuwait
Asia/Macao
Asia/Macau
Asia/Magadan
Asia/Makassar
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novokuznetsk
Asia/Novosibirsk
Asia/Omsk
Asia/Oral
Asia/Phnom_Penh
Asia/Pontianak
Asia/Pyongyang
Asia/Qatar
Asia/Qyzylorda
Asia/Rangoon
Asia/Riyadh
Asia/Riyadh87
Asia/Riyadh88
Asia/Riyadh89

Asia/Saigon
Asia/Sakhalin
Asia/Samarkand
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Tel_Aviv
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Asia/Ust-Nera
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan
Atlantic/Azores
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde
Atlantic/Faeroe
Atlantic/Faroe
Atlantic/Jan_Mayen
Atlantic/Madeira
Atlantic/Reykjavik
Atlantic/South_Georgia
Atlantic/St_Helena
Atlantic/Stanley
Australia/ACT
Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Canberra
Australia/Currie

Australia/Darwin
Australia/Eucla
Australia/Hobart
Australia/LHI
Australia/Lindeman
Australia/Lord_Howe
Australia/Melbourne
Australia/NSW
Australia/North
Australia/Perth
Australia/Queensland
Australia/South
Australia/Sydney
Australia/Tasmania
Australia/Victoria
Australia/West
Australia/Yancowinna
Brazil/Acre
Brazil/DeNoronha
Brazil/East
Brazil/West
CET
CST6CDT
Canada/Atlantic
Canada/Central
Canada/East-Saskatchewan
Canada/Eastern
Canada/Mountain
Canada/Newfoundland
Canada/Pacific
Canada/Saskatchewan
Canada/Yukon
Chile/Continental
Chile/EasterIsland
Cuba
EET
EST
EST5EDT
Egypt
Eire
Etc/GMT
Etc/GMT+0

Etc/GMT+1
Etc/GMT+10
Etc/GMT+11
Etc/GMT+12
Etc/GMT+2
Etc/GMT+3
Etc/GMT+4
Etc/GMT+5
Etc/GMT+6
Etc/GMT+7
Etc/GMT+8
Etc/GMT+9
Etc/GMT-0
Etc/GMT-1
Etc/GMT-10
Etc/GMT-11
Etc/GMT-12
Etc/GMT-13
Etc/GMT-14
Etc/GMT-2
Etc/GMT-3
Etc/GMT-4
Etc/GMT-5
Etc/GMT-6
Etc/GMT-7
Etc/GMT-8
Etc/GMT-9
Etc/GMT0
Etc/Greenwich
Etc/UCT
Etc/UTC
Etc/Universal
Etc/Zulu
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belfast
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
Europe/Brussels
Europe/Bucharest

Europe/Budapest
Europe/Busingen
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Guernsey
Europe/Helsinki
Europe/Isle_of_Man
Europe/Istanbul
Europe/Jersey
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/Ljubljana
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Mariehamn
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Nicosia
Europe/Oslo
Europe/Paris
Europe/Podgorica
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/San_Marino
Europe/Sarajevo
Europe/Simferopol
Europe/Skopje
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Tiraspol
Europe/Uzhgorod
Europe/Vaduz

Europe/Vatican
Europe/Vienna
Europe/Vilnius
Europe/Volgograd
Europe/Warsaw
Europe/Zagreb
Europe/Zaporozhye
Europe/Zurich
GB
GB-Eire
GMT
GMT+0
GMT-0
GMT0
Greenwich
HST
Hongkong
Iceland
Indian/Antananarivo
Indian/Chagos
Indian/Christmas
Indian/Cocos
Indian/Comoro
Indian/Kerguelen
Indian/Mahe
Indian/Maldives
Indian/Mauritius
Indian/Mayotte
Indian/Reunion
Iran
Israel
Jamaica
Japan
Kwajalein
Libya
MET
MST
MST7MDT
Mexico/BajaNorte
Mexico/BajaSur
Mexico/General
Mideast/Riyadh87

Mideast/Riyadh88
Mideast/Riyadh89
NZ
NZ-CHAT
Navajo
PRC
PST8PDT
Pacific/Apia
Pacific/Auckland
Pacific/Chatham
Pacific/Chuuk
Pacific/Easter
Pacific/Efate
Pacific/Enderbury
Pacific/Fakaofo
Pacific/Fiji
Pacific/Funafuti
Pacific/Galapagos
Pacific/Gambier
Pacific/Guadalcanal
Pacific/Guam
Pacific/Honolulu
Pacific/Johnston
Pacific/Kiritimati
Pacific/Kosrae
Pacific/Kwajalein
Pacific/Majuro
Pacific/Marquesas
Pacific/Midway
Pacific/Nauru
Pacific/Niue
Pacific/Norfolk
Pacific/Noumea
Pacific/Pago_Pago
Pacific/Palau
Pacific/Pitcairn
Pacific/Pohnpei
Pacific/Ponape
Pacific/Port_Moresby
Pacific/Rarotonga
Pacific/Saipan
Pacific/Samoa

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

13 ADDRESS

Gets an address for a device. A device has only one address.

13.1 GET /apiv1/devices/:id/addr[.format]

Returns an address of the device.

Input:

Mandatory:

Device_id: Device Id that is returned from GET devices

Type: Integer

JSON Example:

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

Response

200 – Success

```
{"addr":{"city":"Sunnyvale","country":"USA","state":"CA","street":"6 W.
California Ave","zip":94086}}
```

(Initial case)

```
{"addr":{"city":null,"country":null,"state":null,"street":null,"zip":null}}
```

404 – Failure, not found the device doesn't exist

13.2 PUT /apiv1/devices/:id/addr[.format]

Updates an address with the provided parameters.

Input:

Mandatory:

Device_id: Device Id that is returned from GET devices

Type: Integer

Optional:

addr: A hash with the address to be created

Type: String

street: street address

Type: String

Validation: Limit: 255 bytes (Only single sentence)

city: city name

Type: String

Validation: Limit: 255 bytes (only string words)

state: state name

Type: String

Validation: (Limit: 255 bytes) (only string words)

country: country name

Type: String

Validation: (Limit: 255 bytes) (only string words)

zip: zip code

Type: String

Validation: (Limit: up to 9 digits)

XML Example:

```
curl -X PUT -H "Content-Type: application/xml" -
H"Authorization:auth_token <auth_token>" -
```

```
d'<addr><street>new_street</street><city>some
city</city><state>some
state</state><country>usa</country></addr>'
https://<device service
url>/apiv1/devices/<id>/addr.xml
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<addr>
  <street>new_street</street>
  <city>some city</city>
  <state>some state</state>
  <country>usa</country>
  <zip type="integer" nil="true"></zip>
</addr>
```

* Connection #0 to host <device service url> left intact

JSON Example:

```
curl -X PUT -H "Content-Type: application/json" -d
'{"addr":{"street":"new street", "city":"new city",
"state":"new state", "country": "new country"} }' -H
"Authorization: auth_token <auth_token>" https://<device
service url>/apiv1/devices/50220/addr
```

output :

```
{"addr":{"city":"new city","country":"new country","state":"new state","street":"new
street","zip":null}}
```

Response

- 200 – Success
- 422 – Failure,unprocessable entity
- 404 – Failure not found, the device doesn't exist

13.3 **DELETE** /apiv1/devices/:id/addr[.format]

Clears an address of the device.

Input:

Mandatory:

Device_id: Device Id that is returned from GET devices
Type: Integer

JSON Example:

```
curl -X DELETE -H"Authorization:auth_token
<auth_token>" https://<device service
url>/apiv1/devices/<device_id>/addr.json
```

Response:

204 – Success

404 – Failure, not found, the device doesn't exist

14 Admin APIs

14.1 **GET /apiv1/devices.json?paginated=true**

Used to obtain a paginated list of devices belonging to the OEM. The HTTP Authorization header needs to include the admin auth_token parameter as returned in the user object after a successful login.

Inputs: HTTP Authorization Header: has to include string “auth_token <token>” where <token> is obtained from user object returned on a successful sign in from developer site. See Section 4.1

paginated: “true” indicates that the response should be paginated

per_page: number of elements per page

page: the page number to fetch

Output: A paginated list of devices

HTTP Methods: GET

Data Formats: XML, JSON

Required Roles: OEM admin or OEM user who has access to view OEM devices

Curl XML Example:

```
curl -X GET -d"paginated=true&page=1&per_page=2" -H"Authorization: auth_token
76f1a104193a4414862a7705e288cfbb" https://ads-
dev.aylanetworks.com/apiv1/devices.xml
<?xml version="1.0" encoding="UTF-8"?>
<devices>
  <next-page type="integer">2</next-page>
  <current-page-number type="integer">1</current-page-number>
  <start-count-on-page type="integer">1</start-count-on-page>
```

```
<end-count-on-page type="integer">2</end-count-on-page>
<total type="integer">45166</total>
<devices type="array">
  <device>
    <product-name nil="true"></product-name>
    <model>AY001MTP2</model>
    <dsn>AC000W0000000003</dsn>
    <oem-model>model</oem-model>
    <sw-version>bc 0.5 09/18/11 11:20:56 ID jre/not/eb2617e+</sw-version>
    <mac nil="true"></mac>
    <lan-ip>192.168.2.101</lan-ip>
    <connected-at type="datetime">2014-04-02T18:26:20Z</connected-at>
    <key type="integer">3</key>
    <lan-enabled type="boolean">true</lan-enabled>
    <registered type="boolean">false</registered>
    <user-id nil="true"></user-id>
    <template-id nil="true"></template-id>
    <connection-status nil="true"></connection-status>
    <lat nil="true"></lat>
    <lng nil="true"></lng>
  </device>
  <device>
    <product-name nil="true"></product-name>
    <model>AY001MTP1</model>
    <dsn>AC000W0000000007</dsn>
    <oem-model nil="true"></oem-model>
    <sw-version>bc 0.7 10/05/11 17:58:18 ID jre/canary/52689fa+</sw-version>
    <mac nil="true"></mac>
    <lan-ip>172.31.0.137</lan-ip>
    <connected-at type="datetime">2011-10-07T18:18:11Z</connected-at>
    <key type="integer">8</key>
    <lan-enabled type="boolean">false</lan-enabled>
    <registered type="boolean">false</registered>
    <user-id nil="true"></user-id>
    <template-id nil="true"></template-id>
    <connection-status nil="true"></connection-status>
    <lat nil="true"></lat>
    <lng nil="true"></lng>
  </device>
</devices>
```

Curl JSON Example:

```
curl -X GET -d"paginated=true&page=1&per_page=2" -H"Authorization: auth_token
76f1a104193a4414862a7705e288cfbb" https://ads-
```

```
dev.aylanetworks.com/apiv1/devices.json
```

```
{"next_page":2,"current_page_number":1,"start_count_on_page":1,"end_count_on_page":2,"total":45164,"devices":[{"device":{"product_name":null,"model":"AY001MTP2","dsn":"AC000W0000000003","oem_model":"model","sw_version":"bc 0.5 09/18/11 11:20:56 ID jre/not/eb2617e+","mac":null,"lan_ip":"192.168.2.101","connected_at":"2014-04-02T18:26:20Z","key":3,"lan_enabled":true,"registered":false,"user_id":null,"template_id":null,"connection_status":null,"lat":null,"lng":null}},{"device":{"product_name":null,"model":"AY001MTP1","dsn":"AC000W0000000007","oem_model":null,"sw_version":"bc 0.7 10/05/11 17:58:18 ID jre/canary/52689fa+","mac":null,"lan_ip":"172.31.0.137","connected_at":"2011-10-07T18:18:11Z","key":8,"lan_enabled":false,"registered":false,"user_id":null,"template_id":null,"connection_status":null,"lat":null,"lng":null}}]}
```

15 USER AND DEVICE SERVICE URLS

Device Service:

To get App-id:

Go to dashboard

Click on OEM Profile

Under Apps tab – create a new app-id or choose existing app-id

USA: <https://<app-id>-<oem-id>-device.aylanetworks.com>

China: <https://<app-id>-<oem-id>-device.ayla.com.cn>

User Service:

USA: <https://user.aylanetworks.com>

China: <https://user.ayla.com.cn>