**mydlink Cloud NVR**

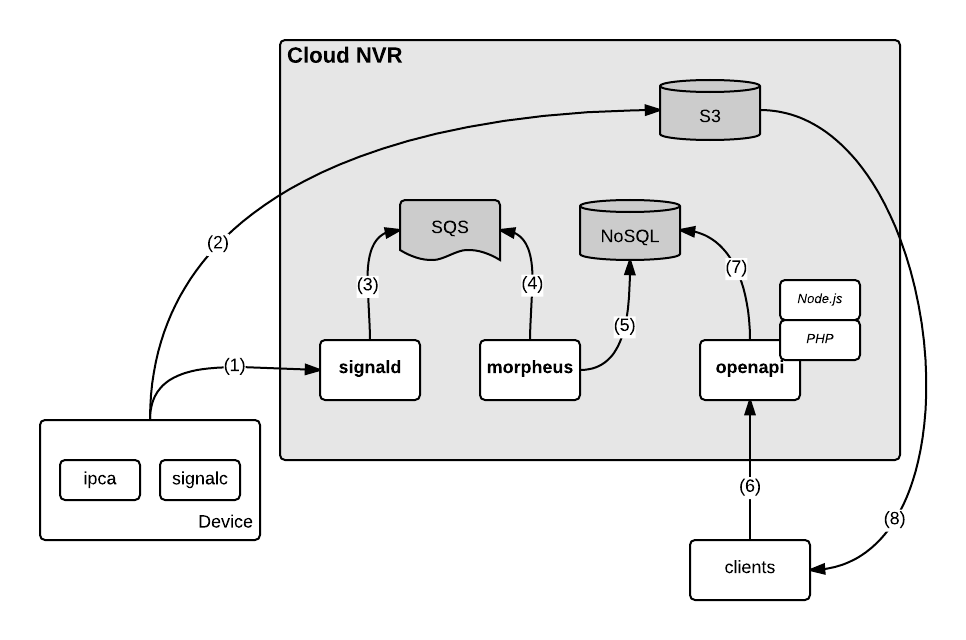
**Design Specification**

Version 1.1.1

**Revise History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Publish Date** | **Author** | **Descriptions** |
| 1.0.0 | Jan. 14, 2013 | Jason Syu | Cloud NVR for POC |
| 1.1.0 | Oct. 01, 2014 | Jason Syu | Revise architecture and validate designs |
| 1.1.1 | Oct. 29, 2014 | Jason Syu | - Revise IV.1.4  - Add ‘sort’ field to IV.3.1 & IV.3.2 |

**1. Architecture**



1. Depends on policy specified (event base or continuous), device agent shall generate clips and send record to platform.
   1. grant storage upload permission from signald (two -way arrow)
2. In the same time, agent shall upload the generated clip to S3 directly, with given authentication from backend.
3. Once signald receives clip event from device, it pushes to queue for later processing.
4. Morpheus retrieves events from MQ as usual way. (arrow direction is reversed)
5. After processed, morpheus retrieves clip relative info and store to NoSQL DB for later processing.
6. Client requests clip info via openapi(two-way arrow)
7. Openapi loads clips info via NoSQL DB and returns to client.
8. Client loads clips via S3 directly.

**Remarks:**

* All the requests and clips upload/download shall be performed via secure connections.

**1.1 Authentication**

1.1.1) Device agent (named as DA) uses MDMP to connect to the target signald service

1.1.2) Signald shall return parameters for the device, including:

* nvr function enable flag: On or Off
* recording mode: event base or continuous

1.1.3) Signald shall provide API for openapi to turn on/off NVR service and notifies DA. DA shall start/stop the nvr services once received the notification.

1.1.4) DA shall monitor the health of the recording client and auto restart it once it’s terminated.

1.1.5) Operation flows:

1. DA connects to signald
2. signald return service enable flag and the recording mode.
3. DA launches the recording agent, and upload clip by mode.

**1.2 Signald daemon (SLD)**

1.2.1) DA authentication

1. Signald shall be able to authenticate the DA by the MDMP.
2. Signald shall be able to assign the recording mode for the DA.

1.2.2) Upload info

1. Signald shall be able to assign correct upload permission by its mode.
2. Signald shall be able to generate correct prefix of upload location.
   * refer Appendix II for location definition.

1.2.3) Event and clip storing

1. Signald shall handle the received clip ready events and pass to event handler.
2. Each event entry shall contain the following information

* from which device
* triggered time
* event type (clip ready event)
* S3 location of the clip
* start timestamp of the clip
* duration of the clip

**1.3 Recording client (RCA)**

1.3.1) The DA shall launch the recording agent (named as RCA) if the function is on.

1.3.2) RCA shall be able to access the audio & video stream of the device.

1.3.3) RCA shall parse and pack the audio/video frames to .ts (transport stream) file.

1.3.4) RCA shall be able to get the upload permission of cloud storage for clip uploading.

1.3.5) RCA shall be able to grant the upload permission from platform once it’s expired.

1.3.6) RCA shall be able to generate random name for each clip.

1.3.7) RCA shall able to handle:

* disconnection from audio/video provider (device web service).
* failed on clip upload.

1.3.8) RCA shall notify signalc agent once clip uploaded successfully with information:

* location of the clip upload to
* start timestamp of the clip
* end timestamp of the clip

1.3.9) Operation flows:

1. RCA is launched by DA.
2. RCA gets the video feed from system.
3. RCA uploads clips to cloud storage by the given mode.

**1.4 Event Handler (EH)**

The event handler shall handle event queuing and storing.

1.4.1) EH shall handle events from signald and store events to queue.

1.4.2) EH shall handle events from queue.

1.4.3) EH shal handle ‘clip ready’ event and update corresponding fields of the system

1.4.4) EH shall put ‘preview job’ to TaskQueue once received ‘clip ready’ event.

**1.5 Post Action Handler (PAH)**

The post handler is to process content once it’s uploaded to cloud.

1.5.1) PAH shall be able to generate snapshot of clips for preview, and save to correct location.

* have to generate one in original size, and one scaling down to 160x120.

1.5.2) PAH shall update snapshot status once 1.5.1 finished.

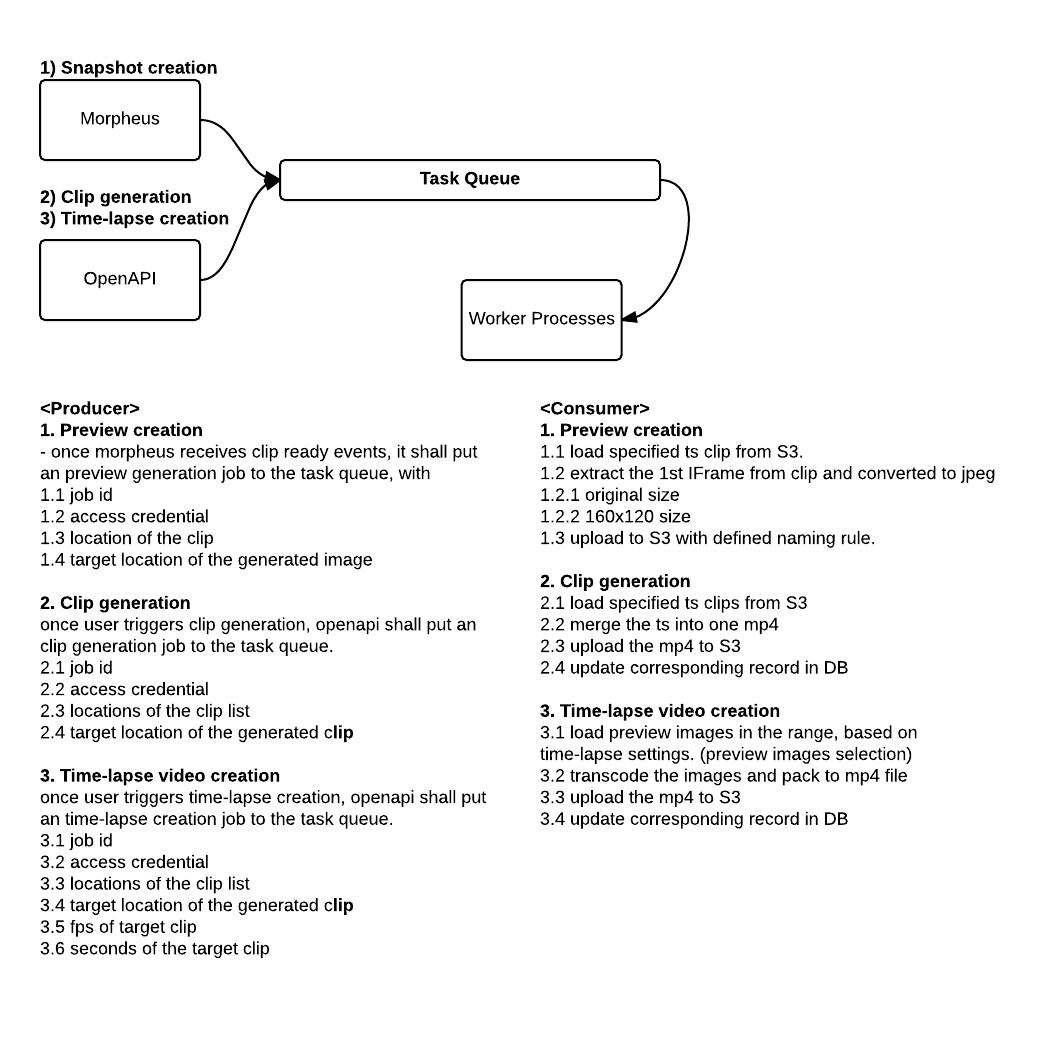
* refer Appendix II for location definition.

1.5.3) PAH shall be able to generate mp4 clip once user creates clip manually.

* once OpenAPI receives ‘make clip’ request from client, it shall add and entry to MQ to indicate clips in the specified duration shall be converted into one clip files.
* PAH shall running conversion function to make clips to one mp4 file.

1.5.4) PAH shall update clip conversion status once 1.5.3 finished.

* The converted info shall be stored back to DB to indicate the target clip is ready.



1.5.5) Task definitions:

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Task** |  |  |
| Preview generation |  |  |
| mp4 generation |  |  |

**1.6 Playlist Management API (mydlink OpenAPI)**

1.6.1) OpenAPI shall handle the requests from clients and HLS player. When user uses the cloud NVR functions, the client shall send video availability requests to OpenAPI for the timeline rendering, with session key carried. OpenAPI shall check the access key of the user to authorize the requests. Based on HLS, the video player will periodically fetch the up-to-date playlist. OpenAPI shall provide playlist based on the carried session key of the request.

1.6.2) OpenAPI shall provide API for query playlist

1. query HLS url
2. load HLS list (issue by HLS player)
3. query available clips in the given time range
4. query preview image of the given timestamp

Refer to Appendix IV for details

1.6.3) OpenAPI shall provide API for clip management

1. query stored clip list
2. query used time quota of the clips
3. create clip with the given time range
4. rename the specified clip
5. update read status of the specified clip
6. delete clip with the given time range
7. get playlist of the specified clip

1.6.4) OpenAPI shall provide API for query events

1. query event list by user and device

**1.7 S3 management**

1.7.1) Bucket selection

* The target bucket shall be created in the nearest region to get the best file transmission performance.

1.7.2) Expiration rules

* The auto expiration rule shall be set to clip files, depending on the given expiration time.
* The process shall be prepared for the modification of clip expiration date, in order to handle the changes of data plan by user.

1.7.3) Access account protection

* Should create access accounts by region of the bucket.

**1.8) Monitor tool**

1.8.1) TBD

**2. Billing System**

**2.1 Subscription flows**

TBD

**3. Security**

**3.1 Device authentication**

3.1.1) Only mydlink device can connect to mydlink platform

3.1.2) Only device which user subscribed cloud NVR service can get the clip upload permission.

* For event based plan, the agent shall query the upload permission every time.
* For continuous plan, the agent shall be assigned the upload permission in day basis.

**3.2 Data transmission**

3.2.1) Control message

* All over SSL connection.

3.2.2) Video clip upload

* All over HTTPS to cloud storage.

3.2.3) Video clip download

* All over HTTPS from cloud storage.

**3.3 Metadata access**

3.3.1) Only authorized users can get the metadata.

3.3.2) All over HTTPS of OpenAPI interfaces.

**3.4 Playlist access**

3.4.1) Only authorized users can get the playlist.

3.4.1) All over HTTPS of OpenAPI interfaces.

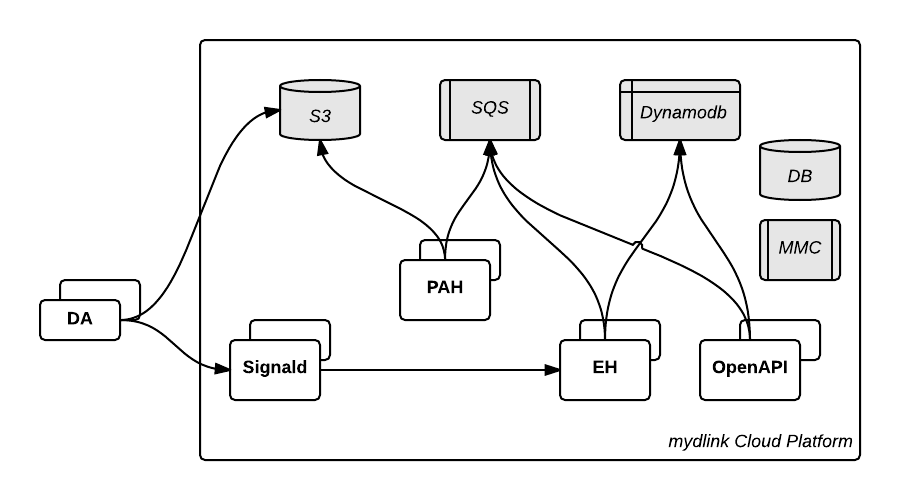
**3.5 Clip access**

3.5.1) the record clips shall be able to read publicly.

3.5.2) the clip name shall be random and long enough, and not be guessed easily.

3.5.3) should apply ‘encryption’ on the clips in S3.

**4. Scalability**



**4.1 AWS Services**

Assumed the AWS services is good enough for mydlink Cloud NVR service.

* S3 performance
* Dynamodb performance
* SQS performance

**4.2 Signald**

* Follow signald scaling plan.

**4.3 Event Handler**

* Follow EH scaling plan.

**4.4 Post Action Handler**

* Stateless, follow same scaling plan as EH.

**4.5 OpenAPI**

* Follow OpenAPI scaling plan.

**4.6 DB**

* Scale up DB instance to gain performance boost.

**4.7 MMC**

* Apply multiple memory cache instances.

**5. Service Availability**

**5.1 Fail over**

* Same as original platform failover plan

**5.2 Duplications**

* TBD

**6. Performance Index**

6.1 Max concurrent clients one signald server can handle

6.2 Max concurrent requests one OpenAPI server can handle

* Normal requests
* Playlist session request
* Playlist fetch request

6.3 Max throughput Post Action Handler can handle

* Preview generation
* Clip conversion
* Time-lapse clip generation

6.4

**Appendix I - Supported Matrix**

**1. Supported Devices**

* All H.264 enabled cameras.

**2. Supported Environments**

|  |  |  |
| --- | --- | --- |
| **Target** | **HTML5** | **HLS** |
| Internet Explorer | Yes (IE9+) | Yes (JW Player) |
| Safari | Yes | Yes |
| Chrome | Yes | Yes (JW Player) |
| Firefox | Yes | Yes (JW Player) |
| iOS | - | Yes |
| Android | - | Yes (3.0+) |
| Windows Phone | - | Yes (3ivx HLS Adapter) |

**3. Codec Requirement**

* H.264 / AAC

**Appendix II - Storage & NoSQL Table Management**

1. **S3 path definition**

Each file shall be in form of:

/<bucket-name>/<recycle-prefix>/<file-name>

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Permission** | **Expiration** | **Path** |
| Bucket | - | - | ead98f12 |
| Prefix of 7 days | - | - | dfa9 |
| Prefix of 30 days | - | - | 8e15 |
| Prefix of forever | - | - | 487f |
| Clip | Public | 7 or 30 days | <hash-id>.ts |
| Snapshot (original size) | Public | 7 or 30 days | <hash-id>\_s.jpg |
| Snapshot (160x120) | Public | 7 or 30 days | <hash-id>\_ss.jpg |
| Storyboard | Public | 7 or 30 days | <hash-id>\_sb.jpg |
| Saved Clip | Public | Forever | <hash-id>.mp4 |

Notes:

* hash-id:
  + The value shall be a **80** characters string (or longer)
  + The value shall contain random chars in set [A-Z][a-z][0-9][-\_.~]

1. **S3 objects management**

* S3 provides auto expiration functions by ‘prefix’
  + on service deployment, the settings must be configured properly.
* Signald shall well-manage the upload permission/path by each user subscription.
  + Direct files with 7/30 expiration days to proper path.
  + Saved clips shall be put to prefix w/o expiration.

1. **Dynamodb table management**

* Dynamodb service doesn’t provide auto expiration feature.
* Need to apply valid period as subscription plan, same as S3.
  + mydlink platform shall provide ability to create/delete dynamodb tables.
  + mydlink platform shall maintain a 30 days valid window. It shall move the window day-by-day. It shall create table if table in the windows doesn’t exist, and delete tables which are outside the window.

**Appendix III - Permissions Setup for AWS Services**

**Permission Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **SQS** | **S3** | **Dynamodb** |
| RCA | none | write only | none |
| Signald | none | write only | none |
| Event Handler | read/write | none | read/write |
| Post Action Handler | read/write | read/write | read/write |
| OpenAPI | none | none | read/write |

Note:

* SQS: Message queue service
* S3: Storage service
* Dynamodb: NoSQL database service

**Appendix IV- Database Structure**

1. **TS file info table (NoSQL DB)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field*** | ***Data Type*** | ***Descriptions*** |
| *1* | *uid (primary)* | *STRING* | *md5 hash of (user\_id + mydlink\_no (8 digit))* |
| *2* | *timestamp (range)* | *INTEGER* | *starting time of the video clip, in milliseconds. (since 1970-01-01 00:00:00 UTC)* |
| *3* | *path* | *STRING* | *download URL of the ts file* |
| *4* | *duration* | *INTEGER* | *time length of the ts file, in milliseconds.* |

**1.1. Table name rule**

TSFileInfo**YYYYMMDD** #e.g. TSFileInfo20141022

**1.2. Sample data (user\_id : 375330, device\_id: 7338993)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***uid*** | ***timestamp*** | ***path*** | ***duration*** |
| 5acd05443f36e34fbdd50fab9f281569 | 1413958396402 | https://s3-ap-southeast-1.amazonaws.com/rd-ee-project-bucket/e4537dd4a15c4eb5/00nk3BEEJWbKjNf31aWh2yrwjOFdISuR1Sb1rOCb/20141019.005458.mp4 | 6000 |

1. **Compose clip info table (SQL DB)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field*** | ***Data Type*** | ***Descriptions*** |
| *1* | *sid* | *INTEGER* | *sequential id of the record* |
| *2* | *uid (index)* | *STRING* | *unique id of the user* |
| *3* | *did (index)* | *STRING* | *unique id of the device* |
| *4* | *name* | *STRING* | *name of the clip* |
| *5* | *start\_time* | *INTEGER* | *start point of the clip to make, in millisecond* |
| *6* | *duration* | *INTEGER* | *millisecond of the clip to make* |
| *7* | *mode* | *INTEGER* | *0: standard, 1: timelapse* |
| *8* | *status* | *INTEGER* | *0: idle, 1: in progress, 2: done, -1: failed* |
| *9* | *path* | *STRING* | *URL of the converted video clip* |

1. **Event info table (NoSQL DB)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field*** | ***Data Type*** | ***Descriptions*** |
| *1* | *uid (primary)* | *STRING* | *md5 hash of (unique id of the user)* |
| *2* | *timestamp (range)* | *INTEGER* | *milliseconds starting time of the video clip. (since 1970-01-01 00:00:00 UTC)* |
| *3* | *did (index)* | *STRING* | *unique id of the device* |
| *4* | *eid* | *INTEGER* | *Event id* |
| *5* | *name* | *STRING* | *Name of which triggers the event* |
| *6* | *policy* | *STRING* | *Policy name* |
| *7* | *srcid* | *STRING* | *Id of which triggers the event* |

**3.1. Table name rule**

EventInfo**YYYYMMDD** #e.g. EventInfo20141022

1. **Recording history table (NoSQL DB)**

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field*** | ***Data Type*** | ***Descriptions*** |
| *1* | *uid (primary)* | *STRING* | *md5 hash of (user\_id + mydlink\_no (8 digit))* |
| *2* | *start\_ts(range)* | *INTEGER* | *milliseconds start time of the recording period. (since 1970-01-01 00:00:00 UTC)* |
| *3* | *end\_ts* | *INTEGER* | *milliseconds end time of the recording period. (since 1970-01-01 00:00:00 UTC)* |

**4.1. Table name rule**

RecordingHistory**YYYYMMDD** #e.g. RecordingHistory20141022

**Appendix IV- OpenAPI Extension for Cloud NVR**

1. Playlist management
   1. Grant playlist session
      1. live mode
      2. on demand mode
      3. clip tag mode
   2. Fetch playlist
   3. Query clips in the specified period
   4. Query preview image of the specified timestamp
2. Clip management
   1. Create clip tag
   2. Delete clip tag
   3. Query clip tag list
   4. Rename the clip tag
   5. Set the clip tag as read
3. Event query
   1. Query event of a given device
   2. Query event of a given user

**IV.1.1 Initiate playlist session**

To get HLS playlist, client shall invoke this API first to initiate the playback session.

|  |
| --- |
| POST /me/nvr/list/initiate?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123", # mydlink number of the target device  "start\_ts": 1412560393000,# interested time point, in milliseconds  "end\_ts": 1412560500000 # optional field.  # - end\_ts is set to start\_ts + 10min (at max) if not  # specified.  # - if current\_time - start\_ts < 10min, it becomes a  # liveview mode. (endless stream)  # - if there are clips not available during the 10min,  # the end\_ts will be set to the first discontinuous  # point.  }  } |

If the given information is correct and user also subscribes NVR service for this device, mydlink platform responses session key and actual start timestamp of the querying video stream.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "session": "e69d649bb49f4a598feceba63a5ccacb", # session for video playlist  "start\_ts": 1412560400000 # actual start time of video, in  # milliseconds  }  } |

**IV.1.2 Fetch playlist**

HLS player may periodically load the update playlist via the URL. The SESSION\_ID is the value returned in IV.1.1.

|  |
| --- |
| GET /me/nvr/list/video.m3u8?session=[SESSION\_ID] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json |

mydlink platform returns clips info if the given session is correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/vnd.apple.mpegurl  Cache-Control: no-store  #EXTM3U  #EXT-X-VERSION:1  #EXT-X-TARGETDURATION:6  #EXT-X-MEDIA-SEQUENCE:0  #EXTINF:6,  <URL of the clip>  #EXTINF:6,  <URL of the clip>  ... |

**IV.1.3 Query timeline availability in the specified period**

The API is for client to query the clip availability in the given time frames. Note that the range of value start\_ts and end\_ts shall not exceed 24 hours in a request.

|  |
| --- |
| POST /me/nvr/info/timeline?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  "data": {  "mydlink\_id": "44440123", # target device id  "start\_ts": 1412560400000, # start time of the interested range  "end\_ts": 1412561400000 # end time of the interested range  } |

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  "data": {  "info": [  [ 1412560413000, 1412560908000 ], # continuous range 1, in millisecond  [ 1412560980000, 1412561400000 ] # continuous range 2, in millisecond  ]  } |

If there are no records in the given time range, the info field in the response will be an empty array.

**IV.1.4 Query recording info**

The API provides functionality for client to get recording info of a device in the given timestamp. The time range of the info shall be provided, and most N records return per query. (N=30). Client shall adjust the given time range for the desired records.

|  |
| --- |
| POST /me/nvr/info/recording?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123", # mydlink number of the target device  "start\_ts": 1412560400000, # start time of the interested range  "end\_ts": 1412561400000 # end time of the interested range  "sort": "asc" # "asc" or "desc" for incremental or  # decremental sorting  }  } |

Once the request is validated, mydlink platform returns the actual time and the path of each recording.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "start\_ts": 1412560410000,  "end\_ts": 1412560495000,  "has\_more": false, # false: all records are returned  # true: client should adjust  # start/end time to get  # remaining records.  "list": [  {  "timestamp": 1412560410000, # actual start time of video (ms)  "path": "<URL of the recording>", # previs image of the given time  "duration": 6000 # milliseconds  },  {  "timestamp": 1412560495000, # actual start time of video (ms)  "path": "<URL of the recording>", # previs image of the given time  "duration": 6000 # milliseconds  }  } |

Client is able to derive the preview file location via the recording path. Check Appendix II for the details.

**IV.2.1 Create clip bookmark**

Client is able to create bookmark for any given time range. A bookmark is the metadata for users to seek to the specify position and range they interested.

|  |
| --- |
| POST /me/nvr/tag/create?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123", # mydlink number of the target device  "name": "funny doggy”, # name of the bookmark  "range": [ 1412560393000, 1412560500000 ] # time range of the bookmark (ms)  }  } |

Success returns if the given fields are correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark  "name": "funny doggy", # name of the bookmark  "range": [ 1412560393000, 1412560500000 ] # time range of the bookmark (ms)  }  } |

**IV.2.2 Delete clip bookmark**

Client can delete the bookmarks of the specified id

|  |
| --- |
| POST /me/nvr/tag/delete?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123", # mydlink number of the target device  "list": [  "abaac316a8ae46c69c1ba35af1b4fa61", # bookmark id  "1b123acd0703454c829941ee48adde82",  "9f7c8875bf544d2f85a2e2fefe507b11",  "a40b5d49be514b4fbb997b907b1390fd",  "139b1055221b4e1d8247ffbc3b346378"  ]  }  } |

Success returns if the given fields are correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "result": "success"  }  } |

**IV.2.3 Query clip bookmark list**

The API returns all created bookmarks of the given device id.

|  |
| --- |
| POST /me/nvr/tag/list?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123" # mydlink number of the target device  }  } |

Success returns if the request is correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "list": [  {  "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark  "range": [ 1412560393000, 1412560500000 ],# time range of the bookmark (ms)  "name": "funny doggy", # name of the bookmark  "watched": true # has watched the clip  },  {  "id": "1b123acd0703454c829941ee48adde82", # unique id of the bookmark  "range": [ 1412560000000, 1412560015000 ],# time range of the bookmark (ms)  "name": "baby smile", # name of the bookmark  "watched": false # not watch this clip yet  },  {  "id": "a40b5d49be514b4fbb997b907b1390fd", # unique id of the bookmark  "range": [ 1412560025000, 1412560035000 ],# time range of the bookmark (ms)  "name": "baby smile", # name of the bookmark  "watched": false # not watch this clip yet  }  ]  }  } |

**IV.2.4 Update clip bookmark**

The API returns all update bookmark attributes of the given device id.

|  |
| --- |
| POST /me/nvr/tag/update?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123" # mydlink number of the target device  "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark  "attributes": {  "name": "funny catty", # name to be changing  "watched": true # view status to be changed  }  }  } |

Success returns if the request is correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "result": "success"  }  } |

**IV.2.5 Generate clip file of a bookmark**

Client can make the video of a bookmark to one real clip file. It’s also able to choose the clip type to be generated. The supported types currently are ‘standard’ and ‘timelapse’.

|  |
| --- |
| POST /me/nvr/clip/create?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123" # mydlink number of the target device  "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # unique id of the bookmark  "type": 0, # 0: standard, 1: timelapse  "attributes": {  "duration": 60 # works for ‘timelapse’ type only,  # means the sec of the output video.  }  }  } |

Success returns if the request is correct. Note that the clip is not generated synchronously. Client shall query clip generation status via IV.2.6.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "result": "success"  }  } |

**IV.2.6 Query clip file status**

Client can query the clip list with the generating status.

|  |
| --- |
| POST /me/nvr/clip/list?access\_token=[ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "mydlink\_id": "44440123" # mydlink number of the target device  }  } |

Success returns if the request is correct.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data":{  "list": [  {  "id": "99ea1dc4f7824fe6924ccd04d08e82c0", # bookmark id  "type": 0, # type of the clip to be made,  # refer IV.2.5 of type definition  "status": 0, # clip generate status:  # 0: in progress  # 1: done  # -1: generate failed  "path": "" # URL of generated clip  },  {  "id": "a40b5d49be514b4fbb997b907b1390fd",  "type": 1,  "status": 1,  "path": "<URL of the clip file>"  }  ]  }  } |

**IV.3.1 Query events of a device**

Client can query events of a device via this API. For a query, the time range of the events shall be provided, and most N records will be returned per query. (N=30). Client shall adjust the given time range for the desired event records.

|  |
| --- |
| POST /me/device/events?access\_token=[APP\_ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "start\_ts": 1404050675000, # start time in milliseconds  "end\_ts": 1404057875000 # end time in milliseconds  "sort": "asc" # "asc" or "desc" for incremental or  # decremental sorting  }  } |

mydlink platform returns all events of the device within the specified time range.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data": {  "start\_time": 1404050675000,  "end\_time": 1404057875000,  "has\_more": false, # false: all records are returned  # true: client should adjust  # start/end time to get  # remaining records.  "list": [  {  "id": 100, # event id  "timestamp": 1404050685849, # event time  "source": "<UID of sensor>", # uid of event source (optional)  "name": "<name of sensor>", # name of event source (optional)  "policy": "<name of policy>" # name of policy (optional)  },  {  "id": 101, # event id  "timestamp": 1404050708578, # event time  "source": "<UID of sensor>", # uid of event source  "name": "<name of sensor>", # name of event source  "policy": "<name of policy>" # name of policy  }  ]  }  } |

**IV.3.2 Query events of an user**

Client can query all device events of the specified account via this API. For a query, the time range of the events shall be provided, and most N records will be returned per query. (N=30). Client shall adjust the given time range for the desired event records.

|  |
| --- |
| POST /me/user/events?access\_token=[APP\_ACCESS\_TOKEN] HTTP/1.1  Host: api.mydlink.com  Content-Type: application/json  {  "data": {  "start\_time": 1404050675000, # start time in milliseconds  "end\_time": 1404057875000 # end time in milliseconds  "sort": "asc" # "asc" or "desc" for incremental or  # decremental sorting  }  } |

mydlink platform returns all events of the user within the specified time range.

|  |
| --- |
| HTTP/1.1 200 OK  Content-Type: application/json  Cache-Control: no-store  {  "data": {  "start\_time": 1404050675000,  "end\_time": 1404057875000,  "has\_more": false, # false: all records are returned  # true: client should adjust  # start/end time to get  # remaining records.  "list": [  {  "mydlink\_id": "30038291", # mydlink number of the device  "id": 100, # event id  "timestamp": 1404050685849, # event time  "source": "<UID of sensor>", # uid of event source  "name": "<name of sensor>", # name of event source  "policy": "<name of policy>" # name of policy  },  {  "mydlink\_id": "30038291", # mydlink number of the device  "id": 101, # event id  "timestamp": 1404050708578, # event time  "source": "<UID of sensor>", # uid of event source  "name": "<name of sensor>", # name of event source  "policy": "<name of policy>" # name of policy  }  ]  }  } |

**Error Code Definitions**

|  |  |  |
| --- | --- | --- |
| **Code ID** | **Message** | **Description** |
| 30 | no such record | No record found for the given information. |
|  |  |  |
| 60 | subscription requires | need to subscribe service on the selected target for the requesting services. |
| 61 | subscription suspended currently | cannot access the specified service currently for some reasons. properly causes by the subscription issue. |
| 62 | invalid service session | the carried session is invalid to access the service. |