

Qucam Device Web Server API

Version: 2.3



0. Revision History

Version	Date	Author	Description of Change
V1.0	2021/04/12	Gene He	Initial
V1.1	2021/04/12	Gene He	1. Add api to obtain the video file information from the device 2. Add api to delete file 3. Add api to obtain device information
V1.2	2021/04/12	Gene He	1. Add api to format SD card 2. Add api to get video thumbnail
V2.0	2021/09/15	Gene He	1. Update document
V2.1	2023/02/17	Daniel Yang	1. Add parameters 'count' and 'offset' in <i>Get Quecam Device Video Files Info</i> .
V2.2	2023/07/17	Daniel Yang	1. Add api to calibrate DMS module (Only valid for CV200)
V2.3	2023/11/22	Daniel Yang	1. Add api to get driver Assistance Calibration Params. 2. Extended <Driver Assistance Calibration>, add ADAS.

0. REVISION HISTORY	2
1. OVERVIEW	4
1.1. SCOPE	4
1.2. HOW TO CONNECT QUCAM DEVICE	4
2. MESSAGE DESCRIPTION.....	5
2.1. RESPONSE FORMAT	5
2.2. REQUEST METHOD SUPPORTED.....	5
2.3. REQUEST HEADER.....	5
2.4. EVENT TYPE.....	5
2.5. CAMERA TYPE	5
3. API LIST	6
3.1. GET QUCAM DEVICE VIDEO FILES INFO	6
3.2. DELETE QUCAM DEVICE VIDEO FILES.....	7
3.3. GET QUCAM DEVICE INFO.....	8
3.4. FORMAT SD CARD	8
3.5. GET QUCAM DEVICE VIDEO THUMBNAIL.....	9
3.6. DOWNLOAD QUCAM DEVICE VIDEO FILE.....	9
3.7. VIDEO-ON-DEMAND	10
3.8. UPDATE QUCAM DEVICE RTMP STATE (START OR STOP RTMP PUSH)	10
3.9. DRIVER ASSISTANCE CALIBRATION	11
3.10. GET DRIVER ASSISTANCE CALIBRATION PARAMS	11

1. Overview

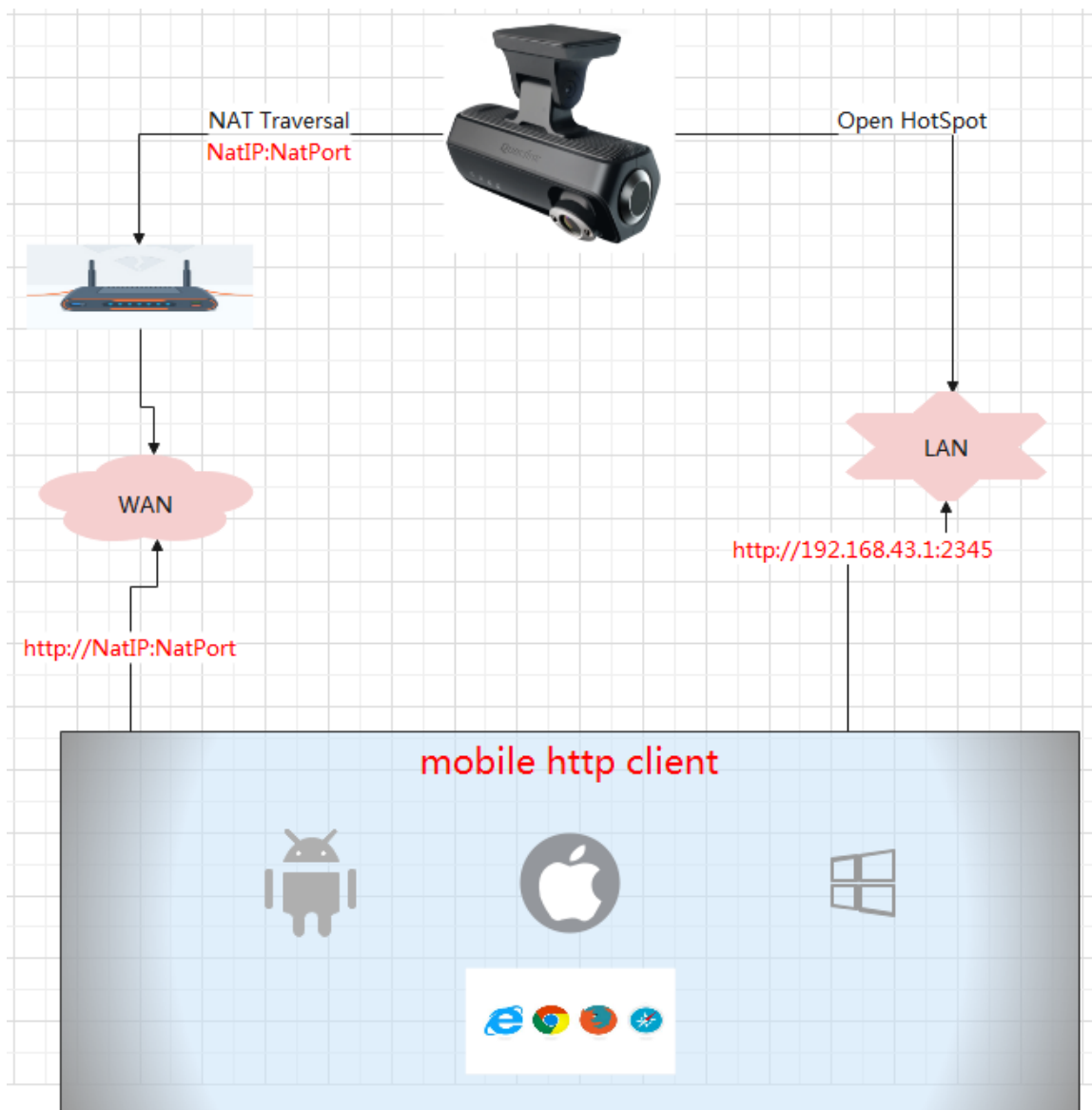
1.1. Scope

The @Qucam Device WebServer API, is used for communication between the Qucam device (as Server) and the client device (such as Android, IOS, WP and browser) via HTTP.

For detail functions, see the API list Below;

The purpose of this document is to describe how to build up the Client APP (Native App or Web App) based on the @Qucam Device Web API.

1.2. How to Connect Qucam Device



LAN mode: the device server URL is fixed <http://192.168.43.1:2345>

WAN mode: the device server URL is <http://NatIP:NatPort>

2. Message Description

2.1. Response Format

Common Format	<pre>{ "data": "[]", "errorCode": 200, "isSuccess": true }</pre>
data	A Json Array, for Detail, See API List
errorCode	200 means OK 301 means Camera Busy others code see API List
isSuccess	True means Response OK

2.2. Request Method Supported

Method	Description
GET	Requests the server to return the specified resource
POST	Request the server to perform a specific operation

2.3. Request Header

headerName	type	position	Required	description
clientId	String	header	yes	unique id of client identity

2.4. Event Type

Event Params Code	Description
01	Ignition ON
02	Ignition OFF
03	Power Disconnected
04	Crash Detection
05	Harsh Acceleration
06	Harsh Braking
07	Harsh Turning
08	Over Speed Alarm
09	Panic Event
0A	SoS Alarm
0E	Geo_Peo_Fence
0F	Parking Safeguard

2.5. Camera type

Camera Params	value	Description
Front	"_1."	Filter Video Front
Interior	"_2."	Filter Video Interior

Both Front and Interior	"1^2" or "" empty String	Filter video both Front and Interior
-------------------------	--------------------------	--------------------------------------

3. API List

3.1. Get Qucam Device Video Files Info

Method	POST			
Request URL	http://192.168.43.1:2345/device/get_video_files_info (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	time_type	String	no	1. Specify a time period, return the video files during this time period; 2. "" empty String means for all time period; 3. Params format: yyyyMMddHHmmss: yyyyMMddHHmmss Example: 20210502151741:20210506151741
	event_type	String	no	1. Filter by the event type, See 2.3 Event Type ; 2. "" empty String means for all event; 3. Combines Event filter, split by ":" ; event_type=_OA_:_OE_:_OF_: means filter by SoS Alarm, Geo_Peo_Fence, Parking Safeguard event;
	camera_type	String	no	See 2.4 Camera Type
	count	int	no	Number of video files requested this time
	offset	int	no	Offset requested this time
Response Body	<pre>{ "data":[{ "date":1619495554000, "duration":"01m:00s", "duration_long":60000, "format_size":"55.112MB", "header_title":"2021-04-27", "icon":null, "isDirectory":false, "name":"20210427_115134_00_1.mp4", "path":"/storage/sdcard1/queclink/video/20210427/11/20210427_115134_00_1.mp4", "permission":"rw", "select_mode":false, "size":57789466 }, { "date":1619495554000, "duration":"01m:00s",</pre>			

Qucam Device Web Server API

	<pre> "duration_long":60000, "format_size":"55.112MB", "header_title":"2021-04-27", "icon":null, "isDirectory":false, "name":"20210427_115134_00_1.mp4", "path":"/storage/sdcard1/queclink/video/20210427/11/20210427_115134_00_1.mp4", "permission":"rw", "select_mode":false, "size":57789466 }], "errorCode":200, "isSuccess":true } Or When device is busy { "data":"Camera Busy", "errorCode":200, "isSuccess":true } Or When response Failed { "data": "", "errorCode":"errorCode", "isSuccess":false } </pre>			

3.2. Delete Qucam Device Video Files

Method	POST			
Request URL	http://192.168.43.1:2345/device/delete_file (host see here)			
Request Header	clientId : unique id of the client, when no account, please generate a uuid to replace it			
	Param Name	Type	Required	Decription

Qucam Device Web Server API

Request Body	<pre>[{ "path":"/storage/sdcard1/queclink/video/20210916/02/20210916_025008_00_1.ts" }, { "path":"/storage/sdcard1/queclink/video/20210916/02/20210916_025007_00_2.ts" }]</pre>	Json Body, Path: the file to be delete
Response Body	<pre>{ "data":"success", "errorCode":200, "isSuccess":true }</pre>	

3.3. Get Qucam Device Info

Method	GET			
Request URL	http://192.168.43.1:2345/device/get_device_info (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	None			
Response Body	<pre>{ "data":{"application_version":"V06", "cameraPreviewState":{"isLving":false,"needUpdateUrl":true,"previewCameraId":-1,"rtmpUrl":"","device_name":"CV100","imei":"867867432522518","local_storage":"52.391GB/59.463GB","password":"12345678","sn":null,"sd_card_format":"FAT32","sd_card_model_no":"SC64G","sd_card_size":"59.463GB","sd_card_type":"SD","sd_card_used_space":"7.072GB","signal":"none","ssid":"queclinkgh"}}, "errorCode":200, "isSuccess":true }</pre>			

3.4. Format SD Card

Method	GET
--------	-----

Qucam Device Web Server API

Request URL	http://192.168.43.1:2345/device/format_sdcard (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	None			
Response Body	<pre>{ "data": "NO SD Card", "errorCode": 200, "isSuccess": true }</pre> <pre>{ "data": "Format Failed,Retry Again", "errorCode": 200, "isSuccess": true }</pre> <pre>{ "data": "Format Success", "errorCode": 200, "isSuccess": true }</pre>			

3.5. Get Qucam Device Video Thumbnail

Method	GET			
Request URL	http://192.168.43.1:2345/device/get_video_thumbnail_info (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	None			
Response Body	Stream of the thumbnail file			

3.6. Download Qucam Device Video File

Method	GET
--------	-----

Qucam Device Web Server API

Request URL	http://192.168.43.1:2345/filePath (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	None			
Response	Stream of the video file			

3.7. Video-on-Demand

Method	GET			
Request URL	http://192.168.43.1:2345/filePath (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	None			Just give the url to the Player or Broswer
Response	Stream of the video file			

3.8. Update Qucam Device Rtmp State (Start or Stop Rtmp Push)

Method	POST			
Request URL	http://192.168.43.1:2345/device/update_rtmp_state (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	rtmpUrl	String	No	The rtmp push url
	cameralId	String	No	The Push video stream from camera, 0 : measn Front Camera 1 : means Interior Camera
	start	String	No	Start or stop rtmp push
Response Body	<pre>{ "data": "success", "errorCode": 200, "isSuccess": true }</pre>			

3.9. Driver Assistance Calibration

Method	POST			
Request URL	http://192.168.43.1:2345/device/driver_assistance_calibration (host see here)			
Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	type	int	yes	1: DMS 2: ADAS
	calibration	int	yes	0: Failed 1: Successful
	adasExtrinsic Params	String	Yes for type 2	The ADAS extrinsic parameters are consists of multiple parameters and separated by vertical bars (" "), like this "1 2 3 4 5 6 7": ✧ Horizon, pixel coordinates of the horizon in images in the real world. Unit: pixel. ✧ Car Middle. Unit: pixel, range: -10 ~ 10. ✧ Car Width. Unit: cm. ✧ Camera Height. Unit: cm. ✧ Camera To Axle. Unit: cm. ✧ Camera To Bumper. Unit: cm. ✧ Camera To Left Wheel. Unit: cm.
Response Body	<pre>{ "data":{"calibration":1,"type":"1"}, "errorCode":200, "isSuccess":true }</pre> calibration: 0: Failed 1: Successful 2: Not Ready type: 1: DMS 2: ADAS			

3.10. Get Driver Assistance Calibration Params

Method	GET
Request URL	http://192.168.43.1:2345/device/get_driver_assistance_calibration_params (host see here)

Qucam Device Web Server API

Request Header	clientId : unique id of the client, when no account ,please generate a uuid to replace it			
	Param Name	Type	Required	Decription
Request Params	params	int	yes	0: ADAS extrinsic params
Response Body	<pre>{ "data": "1 2 3 4 5 6 7", "errorCode": 200, "isSuccess": true }</pre>			