CTOUCH®
The power to engage

Product & Innovation

Specification Remote Management API

Authors: CTOUCH

CTOUCH®
The power to engage

Contents

History

Version	Date	Changes		Author(s)
1.1	16-10-2020	First release	P&I	CTOUCH/HC

Introduction

This document describes the CTOUCH 'Remote Management API' and it's details. This document primarily focusses on the Riva screen, but is written with applicability to future products in mind.

API Overview

The API consists of a HTTP server running on the screen which reacts to JSON-requests. The JSON requests contain a hash, timestamp and command, the response is JSON as well and consist of version, type and result field. In case the server cannot respond in the right way, it will send an error message. All these are described in more detail in the next few chapters.

Interface

- Request should be sent via a HTTP POST

- Request URL is: <ip-address>:<port>/managementapi

- With port being 8110

- Timestamp: current time, in ISO 8601 format

- Hash: SHA-256 of timestamp + token

- Type api_request commands: get / set / execute

- Normally the response should be within 500ms.

Response with HTTP Status Code:

HTTP Status Code	Situation
400	If not able to process request at all (malformed request)
401	If token is not valid
200	All other cases

The response should always contain a body text. See below for the different type of responses.

Location of IP address and token

For the CTOUCH Riva screen, the IP address used in the requests is the screen's mainboard IP address. This can be found in the Settings menu in COS. In case the client application is running on the screen's mainboard itself (= Android), requests can also be sent to 'localhost or '127.0.0.1'.

Token is only available visually on the screen, not in digital form on an API! In order to be able to see and use the API, your dealer needs to enable the 'API token' in the Control settings submenu of the 'Dealer menu'.

When the ‘API token’ is enabled by the dealer, the 8-character/digit token is found as the ‘Token number’ in ‘Display Info’ submenu of the ‘Settings menu’. For Settings menu: Push the light/CTOUCH-button centrally located at the bottom of the screen, then select the gear/settings button (see also Riva manual).

Security

The API requests will originate from the local network and run over http with the transmitted data not secret nor sensitive. An token-based authentication mechanism is used to make sure the request comes from an authorized source.

The API provides a basic authorization/authentication mechanism using a authentication token based hash, generated at random by the screen and consisting of 8 characters. The token is be generated by the screen and stored on the mainboard. All API requests need to contain a hash based on this token and data from within the request.

Resetting of the token is not possible from a remote location: The user has to be physically present at the screen. When Remote Management API is disabled (in the dealer menu) and re-enabled later, a new token will be generated (this is also the way to “reset” the token)

Read / Get

Example of reading single value

Read requests (for status or configuration information):

```
{"api_request": {
        "hash": "ba059253dd5b2878f4dca2427af4edd20b373accf33afa43b68d2f46c0044c20",
        "timestamp": "2019-08-14T13:56:32.427Z",
        "command": {
                "type":"get",
                "value_of": "Source"
        }
}}

# In these examples 6wfx9j1t is used as token to calculate the hash value
# above hash is SHA256(“2019-08-14T13:56:32.427Z6wfx9j1t”)
```

Response (in body):

```
{"api_response": {
        "version" : "1.0",
        "type" : "get",
        "result": {
                "Source" : "HDMI1"
        }
}}
```

Example of reading object

An example with an object :


```
{"api_request": {
        "hash": "ba059253dd5b2878f4dca2427af4edd20b373accf33afa43b68d2f46c0044c20",
        "timestamp": "2019-08-14T13:56:32.427Z",
         "command": {
                "type":"get",
                "value_of": "MainboardCon"
        }
}}
```


Response (in body):


```
{"api_response": {
        "version" : "1.0",
        "type" : "get",
        "result": {
                "MainboardCon" : {
                        "Ethernet": "On","Wifi": "Off"
                }
        }
}}
```


Write / Set

Example of setting single value

Write requests for configuration settings:


```
{"api_request": {
        "hash": "cf1dafdc67bcb4904be45f020b059b17977177cdcae24bfa90e00c25ba184675",
        "timestamp": "2019-08-14T14:16:09.835Z",
        "command": {
                "type":"set",
                "Source": "HDMI1"
        }
}}
```

Response: (in body)

```
{"api_response": {
        "version" : "1.0",
        "type":"set",
        "result": {
                "Source": "HDMI1" – echo request
        }
}}
```

Example of setting multiple values in one request

```
# Option: "multi_set": [{"key1":”value1"} , {"key2":"value2"} , {"key3":"value3"}]
# no ordering
# first check all key/values, only if all are valid then set keys/values. Otherwise no set and respond with error
# 30 or 40 and mention the invalid key/values in the message.

{"api_request": {
        "hash": "cf1dafdc67bcb4904be45f020b059b17977177cdcae24bfa90e00c25ba184675",
        "timestamp": "2019-08-14T14:16:09.835Z",
        "command": {
                "type":"set",
                "multi_set": [
                        { "Source": "HDMI1"},
                        { "Brightness": 80}
                ]
        }
}}
```

Response: (in body)

```
{"api_response": {
        "version" : "1.0",
        "type":"set",
        "result":{"multi_set": [
                {"Source": "HDMI1"},
                { "Brightness": 80}
                ]
        }
}}
```

Execute / Run

Request to execute a command:

```
{"api_request": {
        "hash": "cf1dafdc67bcb4904be45f020b059b17977177cdcae24bfa90e00c25ba184675",
        "timestamp": "2019-08-14T14:16:09.835Z",
        "command": {
                "type":"execute",
                "action": "RC_VOL_UP"
        }
}}
```

Response (in body):

```
{"api_response": {
        "version" : "1.0",
        "type":"exec",
        "result": {
                "action": "RC_VOL_UP" – echo request
        }
}}
```

Error response

Above examples show the response in case of a valid request. If the request cannot be carried out an error message will be replied: (in body)

```
{"api_response": {
        "version" : "1.0",
        "type":"error",
        "result": {
                "message": "Unknown key",
                "error-code": 30
        }
}}
```

Error-codes:

10 – Not authorized - also return HTTP-Status Code 401 instead of 200
11 – Request outdated - also return HTTP-Status Code 401 instead of 200
20 – Invalid document - also return HTTP-Status Code 400 instead of 200
30 – Unknown key
40 – Invalid key value combination
50 – Invalid key value combination for the current configuration

Command overview for CTOUCH Riva

Table 1: Supported commands in Riva Remote Management API version 1.0

	Key	Get	Set	Exec	Value
Picture	PictureMode	X	X	-	Dynamic/Standard/Soft/User
	Contrast	X	X	-	0-100
	Brightness	X	X	-	0-100
	Sharpness	X	X	-	0-100
	Saturation	X	X	-	0-100
	Backlight	X	X	-	0-100
	Energy Mode	X	X	-	Auto/Normal/Balanced/Dynamic/Ecofriendly
	BlueLightFilter	X	X	-	On/Off
	EyeProtect	X	X	-	On/Off
	PCMode	X	X	-	On/Off
	HDR	X	X	-	On/Off
	VGAAutoAdjust	X	X	-	On/Off
	ColorTemp	X	X	-	Cool/Normal/Warm/User
	Red	X	X	-	0-100
	Green	X	X	-	0-100
	Blue	X	X	-	0-100
	ZoomMode	X	X	-	P2P/1609
	OverScan	X	X	-	On/Off
Sound	SoundMode	X	X	-	Standard/Music/Movie/Sports/User
	MediaOut	X	X	-	Speaker/SPDIF/ARC/Bluetooth
	Balance	X	X	-	[-50,+50]
	EQ120	X	X	-	0-100
	EQ500	X	X	-	0-100
	EQ1.5k	X	X	-	0-100
	EQ5k	X	X	-	0-100
	EQ10k	X	X	-	0-100
	AVC	X	X	-	On/Off
	Surround	X	X	-	On/Off
	SPDIFMode	X	X	-	RAW/PCM
	SPDIFVolume	X	X	-	0-100
	MaxVolume	X	X	-	0-100
OSD	QuickControlButtons.Source	X	X	-	On/Off
	QuickControlButtons.PowerOn	X	X	-	On/Off
	QuickControlButtons.SettingsMenu	X	X	-	On/Off
	QuickControlButtons.BacklightMute	X	X	-	On/Off
	QuickControlButtons.Manual	X	X	-	On/Off

	Key	Get	Set	Exec	Value
	QuickControlButtons.Freeze	X	X	-	On/Off
	QuickControlButtons.LockTouch	X	X	-	On/Off
	QuickControlButtons.Mute	X	X	-	On/Off
	QuickControlButtons.EnergyMode	X	X	-	On/Off
	QuickControlButtons.Screenshot	X	X	-	On/Off
	QuickControlButtons.Back	X	X	-	On/Off
	QuickControlButtons.Home	X	X	-	On/Off
	QuickControlButtons.AppHistory	X	X	-	On/Off
	QuickControlButtons.Volume	X	X	-	On/Off
	LockFavoriteApps	X	X	-	On/Off
	HomeOption_RC	X	X	-	HDMI1/HDMI2/HDMI3/… etc, only enabled INPUT SOURCES (On) can be set as Home
Input	HDMI1	X	X	-	On/Off
	HDMI2	X	X	-	On/Off
	HDMI3	X	X	-	On/Off
	USB-C	X	X	-	On/Off
	DP	X	X	-	On/Off
	VGA	X	X	-	On/Off
	OPS	X	X	-	On/Off
	SlideIn	X	X	-	On/Off (only available when SlideIn present)
	Audio	X	X	-	On/Off
	SourceLabel	X	X	-	Object {"Source": "HDMI2", "Label":"Mac"} Source = HDMI1/HDMI2/.. Etc Label = name which is displayed max 10 characters (For get/read request only Source need to be defined)
	HDMIEDID	X	X	-	1.4/2.0
	HDMICEC	X	X	-	On/Off
	HDMIARC	X	X	-	On/Off
Touch	Touch	X	X	-	On/Off
Power	NoSignalPowerOff	X	X	-	Off/1m/3m/5m/10m/15m/30m/45m/60m
	StartUpByExternal	X	X	-	On/Off
	OnTimerPeriod	X	X	-	Off/Once/MoTeWeThFrSaSo (if you use weekdays then just mention the applicable days, so when only Wednesday and Thursday the value is WeTh)
	OnTimerTime	X	X	-	00:00 – 23:59

	Key	Get	Set	Exec	Value
	OffTimerPeriod	X	X	-	Off/Once/MoTeWeThFrSaSo (if you use weekdays then just mention the applicable days, so when only Wednesday and Thursday the value is WeTh)
	OffTimerTime	X	X	-	00:00 – 23:59
	SleepTime	X	X	-	Off/1m/10m/20m/30m/60m/90m/120m/180m
	PowerOnMode	X	X	-	Direct/Stand-by/Memory
	PowerOnSource	X	X	-	Memory/HDMI1/HDMI2/HDMI3/USB-c/DP/VGA/PC/COS/Audio
	PowerOnVolume	X	X	-	Memory/0-100 (combination with Preset value)
	PowerOnOPSDirect	X	X	-	On/Off
	CTOUCHButton	X	X	-	On/Off
Admin	COS	X	X	-	On/Off
	OTA	X	X	-	On/Off/Auto
	AutoDateTime	X	X	-	On/Off
	24HourFormat	X	X	-	On/Off
	Language	X	X	-	UK/ES/F/D/NL
	AppInstallLock	X	X	-	On/Off
	KeypadLock	X	X	-	On/Off
	IRLock	X	X	-	On/Off
	NetworkLock	X	X	-	On/Off
	TempProtection	X	X	-	On/Off
	AlarmTemp	X	X	-	50-100
Connectivity	UARTBaudRate	X	X	-	4800/9600/19200/38400/57600/115200
	UARTID	X	X	-	0-255
	MainboardCon				*-- Object -- (Ethernet,Wifi)*
	MainboardCon.Ethernet	X	X	-	On/Off
	MainboardCon.Wifi	X	X	-	On/Off
	SlideInConnectivity				*-- Object -- (Ethernet,Wifi,Bluetooth)*
	SlideInConnectivity.Wifi	X	X	-	On/Off
	SlideInConnectivity.Bluetooth	X	X	-	On/Off
	SlideInConnectivity.Ethernet	X	X	-	On/Off
	OPSEthernet	X	X	-	On/Off
	DockEthernet	X	X	-	On/Off
	WakeOnLan	X	X	-	On/Off
	HDMIOut	X	X	-	1080p60Hz/2160p30Hz/2160p60Hz
	USB	X	X	-	On/Off/TouchOnly
Control	Source	X	X	-	HDMI1/HDMI2/HDMI3/USB-c/DP/VGA/ PC/COS/Audio
	Freeze	X	X	-	On/Off

	Key	Get	Set	Exec	Value
	Volume	X	X	-	0-100
	Volume_Mute	X	X	-	On/Off
	Backlight_Mute	X	X	-	On/Off
	Power	-	X	-	Off
Info	DisplayInfo	X	-	-	*-- Object --* CTOUCH Model Number, Serial Number, MAC, IP-addres, Model, Panel, TouchFrame, OS Build Time, FW NR, FW Version, STM32 Version
	Temperature	X	-	-	0-99
Run	Action	-	-	X	Possible values are the keys of the executable commands. See table 2.

Table 2: Executable commands (possible values for Action in type Exec request)

Actions	Get	Set	Exec	Comment
RC_CURSOR_UP	-	-	X	
RC_CURSOR_LEFT	-	-	X	
RC_CURSOR_DOWN	-	-	X	
RC_CURSOR_RIGHT	-	-	X	
RC_ENTER	-	-	X	
RC_MENU	-	-	X	Like RC menu button
RC_HOME	-	-	X	Like RC home button
RC_EXIT	-	-	X	Like RC exit button
RC_VOL_UP	-	-	X	New Volume = Volume + 1
RC_VOL_DOWN	-	-	X	New Volume = Volume - 1
RC_SCREENSHOT	-	-	X	Take screenshot