Brickcom

BRICKCOM IPCAM HTTP API

1. Preface

This document details the Brickcom IPCAM HTTP API (Application Programming Interface) which enables applications to access and/or configure the IP Cameras manufactured by Brickcom over a TCP/IP capable network. Programmers who wish to write their own utility should follow the API specifications in this document.

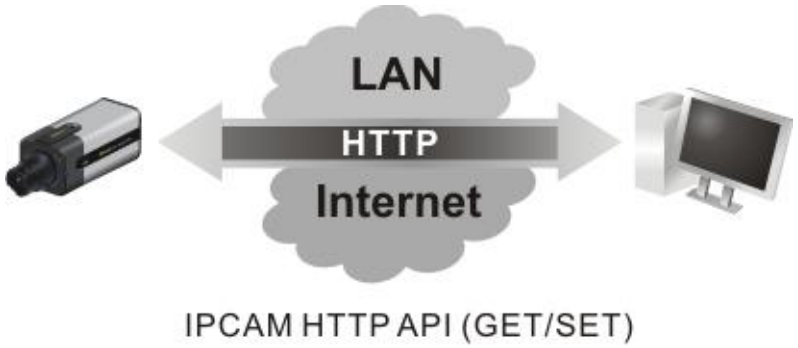
2. TOC

Overview

The IPCAM HTTP API is the proprietary network control protocol designed by Brickcom Technology to enable applications to access IP Cameras manufactured by Brickcom. The API allows for configuration of the settings and inquiry of current status on these IP Cameras. The API is structured and transmitted over HTTP protocols and hence it is given the name HTTP API.

The complete API is divided into several categories for ease of management. This manual contains one chapter for each API category to provide users with a full understanding of that API subset.

Figure 1, Illustration of API generic transactions



3. HTTP API Transaction

An HTTP API transaction begins with a request from a client application, usually a web browser. The request is received by the web server on the IP Camera device and processed by the IP Camera. Finally, it ends with a response which is sent back to the requesting client.

Note: In URL syntax and in descriptions of CGI arguments, text in italics within angle brackets denotes content that should be replaced with either a value or a string. When replacing the text string, the angle brackets must also be replaced. Text within square brackets denotes content that can be omitted.

The client HTTP request is taken in two forms:

- HTTP GET: Normally used to retrieve the settings or status of the IP Camera
- HTTP POST: Normally used to configure the settings of the IP Camera

If the request is successfully received by the IP Camera, the response will contain a HTTP header with a 200 OK response code and the HTTP body with the actual response data or other value when error occurs. An example is provided for each request type below:

Illustration 1: Getting the network setting from the IP Camera

Client request

```
GET http://<IP Camera address>/network.cgi HTTP/1.0
…
```

Server response

```
HTTP/1.0 200 OK
Content-Type: text/plain

IPAddress=192.168.1.1
SubnetMask=255.255.255.0
…
```

Illustration 2, Set the network setting from the IP Camera

Client request

POST http://<*IP Camera address*>/**network.cgi** HTTP/1.0

IPAddress=192.168.1.1
SubnetMask=255.255.255.0

Server response

HTTP/1.0 200 OK
…

Error Response

The IP Camera may be unable to handle the client HTTP API request due to certain conditions such as system busy, incorrect parameters, or any other reasons. An appropriate HTTP status code **400 Bad Request** will be returned, accompanied with an error code and an error string to explain the failure.

Client request

GET/POST …

Server response

HTTP/1.0 200 OK
…
ErrorCode=XXX
ErrorString=Invalid IP Address

4. API Categories

The API categories are listed in the table below.

Table 1, API Categories

API Category	Description
Streaming	Enables users to view/configure the setting of multimedia streaming
Camera	Enables users to view/configure the camera/lens settings
Audio	Enables user to view/configure the audio devices settings
Network	Enables users to view/configure the network settings
Event	Enables users to register to receive notifications from IPCAM
Storage	Enables users to configure storage device for storing media content
System	Enables users to view/configure miscellaneous system settings
Admin	Enables users to perform administrative tasks over the IP Camera.
Capability	Provides users with the list of available features supported by the IP Camera
Motion detection	Enables users to view/configure the motion detection settings and add/delete/update detection region
Event	Enables users to view/configure the event setting and the notification setting
I/O control	Enables users to control I/O status

Note: **Fields marked in gray are reserved.**

5. Streaming

Streaming API allows applications to
1) View/configure the IP Camera streaming settings
2) Help users view video streaming

5.1 GetChannels

ActionEvent: getChannels

URL Syntax	http://<*IP*>/cgi-bin/channels.cgi?action=get
Response	size = CH1.enabled= CH1.name= CH1.transportType= CH1.video.enabled= CH1.video.format.sourceDevice= CH1.video.format.codecType= CH1.video.format.codecSubType= CH1.video.format.constantBitrate= CH1.video.format.bitrateInKbps= CH1.video.format.resolutionWidth= CH1.video.format.resolutionHeight= CH1.video.format.frameRate= CH1.video.format.gop= CH1.video.format.quality= CH1.video.transport.multicastEnabled= CH1.video.transport.multicastAddress= CH1.video.transport.multicastPort= CH1.video.transport.ttl= CH1.audio.enabled= CH1.audio.format.codecType= CH1.audio.format.codecSubType= CH1.audio.transport.multicastEnabled= CH1.audio.transport.multicastAddress= CH1.audio.transport.multicastPort= CH1.audio.transport.ttl= CH1.meta.enabled= CH1.meta.format.mdAlarmEnabled= CH1.meta.transport.multicastEnabled= CH1.meta.transport.multicastAddress= CH1.meta.transport.multicastPort= CH1.meta.transport.ttl= ….
Comment	

HTTP Method	GET

5.2 UpdateChannels

ActionEvent: updateChannels

URL Syntax	http://<*IP*>/cgi-bin/channels.cgi
HTML Body	action=updateAll [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
c1Enable	[0,1]	0:disabled 1:enabled
c1Name	<*String*>	
c1TransportType	[0-4]	0:RTSP_RTP 1:RTP_ONLY 2:HTTP 3:MSN 4:ANALYTICS
c1VideoEnabled	[0,1]	0:disabled 1:enabled
c1VideoFormatCodecType	H264, MPEG4, MJPEG, Analytics	Depends on **Capability.**
c1VideoFormatConstantBitrate	[0,1]	When **constantBitrate** =0, only **quality** can be set; **constantBitrate** = 1, **bitrateInKbps** can be set. VBR=0 CBR=1
c1VideoFormatBitrateInKbps	[64,128,256,384,512, 768,1000,1500,2000, 4000,6000,8000,10000, 12000]	Kbps
c1VideoFormatResolutionWidth		1280x800,640x400,320x192 Based on **Capability.**
c1VideoFormatResolutionHeight		1280x800,640x400,320x192 Based on **Capability.**
c1VideoFormatFrameRate	[1-30]	HZ depends on the limit of hardware
c1VideoFormatGop	[0-N]	0(default) : 1 I-frame/second, N : 1 I-frame in N frames
c1VideoFormatQuality	[1-100]	
c1VideoTransportMulticast Enabled	[0,1]	
c1VideoTransportMulticast	232.0.1.0-232.255.255.255	RFC4607

Address		
c1VideoTransportMulticastPort	[1025-65534]	
c1VideoTransportTtl	[1-255]	
c1AudioEnabled	[0,1]	
c1AudioFormatCodecType	G711,AMR-NB	
c1AudioFormatCodecSubType	PCMU,AMR-MR122	G711=>PCMU AMR-NB =>AMR-MR122
c1AudioTransportMulticast Enabled	[0,1]	
c1AudioTransportMulticast Address	232.0.1.0-232.255.255.255	RFC4607
c1AudioTransportMulticastPort	[1025-65534]	
c1AudioTransportTtl	[1-255]	
c1MetaEnabled	[0,1]	
c1MetaFormatMdAlarm Enabled	[0,1]	
c1MetaTransportMulticast Enabled	[0,1]	
c1MetaTransportMulticast Address	232.0.1.0-232.255.255.255	RFC4607
c1MetaTransportMulticastPort	[1025-65534]	
c1MetaTransportTtl	[1-255]	
c2Enable	[0,1]	
c2Name	<*String*>	
c2TransportType	[0-4]	0:RTSP_RTP 1:RTP_ONLY 2:HTTP 3:MSN 4:ANALYTICS
c2VideoEnabled	[0,1]	
c2VideoFormatCodecType	H264, MPEG4, MJPEG,MIMIC, Analytics	Depends on Capability**.**
c2VideoFormatConstantBitrate	[0,1]	When **constantBitrate** =0, only **quality** can be set; **constantBitrate** = 1, **bitrateInKbps** can be set. VBR=0 CBR=1
c2VideoFormatBitrateInKbps	[64,128,256,384,512, 768,1000,1500,2000, 4000,6000,8000,10000, 12000]	Kbps
c2VideoFormatResolutionWidth		1280x800,640x400,320x192 Based on Capability.
c2VideoFormatResolution Height		1280x800,640x400,320x192 Based on Capability.
c2VideoFormatFrameRate	[1-30]	HZ depends on the limit of hardware

c2VideoFormatGop	[0-N]	0(default) : 1 I-frame/second, N : 1 I-frame in N frames
c2VideoFormatQuality	[1-100]	int
c2VideoTransportMulticast Enabled	[0, 1]	
c2VideoTransportMulticast Address	232.0.1.0-232.255.255.255	RFC4607
c2VideoTransportMulticastPort	[1025-65534]	
c2VideoTransportTtl	[1-255]	
c2AudioEnabled	[0,1]	
c2AudioFormatCodecType	G711,AMR-NB	
c2AudioFormatCodecSubType	PCMU,AMR-MR122	G711=>PCMU AMR-NB =>AMR-MR122
c2AudioTransportMulticast Enabled	[0,1]	
c2AudioTransportMulticast Address	232.0.1.0-232.255.255.255	RFC4607
c2AudioTransportMulticastPort	[1025-65534]	
c2AudioTransportTtl	[1-255]	
c2MetaEnabled	[0,1]	
c2MetaFormatMdAlarmEnabled	[0,1]	
c2MetaTransportMulticast Enabled	[0,1]	
c2MetaTransportMulticast Address	232.0.1.0-232.255.255.255	RFC4607
c2MetaTransportMulticastPort	[1025-65534]	
c2MetaTransportTtl	[1-255]	

Example:
Two profiles:

1. If you set stram1 and stream2 audio, you can to do:

   A. GetChannels:

   URL: http://192.168.1.1/cgi-bin/channels.cgi?action=get

   HTTP/1.1 200 Ok
   Server: mini_httpd
   Cache-Control: no-cache
   Pragma: no-cache
   Expires: 0
   Content-Type: text/html
   Connection: close

   size = 2
   CH1.enabled=1

CH1.name=MJPEG
CH1.transportType=0
CH1.video.enabled=1
CH1.video.format.sourceDevice=0
CH1.video.format.codecType=H264
CH1.video.format.constantBitrate=0
CH1.video.format.bitrateInKbps=1500
CH1.video.format.resolutionWidth=1280
CH1.video.format.resolutionHeight=800
CH1.video.format.frameRate=25
CH1.video.format.frameRate2=20
CH1.video.format.gop=3
CH1.video.format.quality=30
CH1.video.transport.multicastEnabled=0
CH1.video.transport.multicastAddress=234.1.2.3
CH1.video.transport.multicastPort=10004
CH1.video.transport.ttl=10
CH1.audio.enabled=1
CH1.audio.format.codecType=AMR
CH1.audio.format.codecSubType=AMR-MR122
CH1.audio.transport.multicastEnabled=0
CH1.audio.transport.multicastAddress=
CH1.audio.transport.multicastPort=10002
CH1.audio.transport.ttl=10
CH1.meta.enabled=1
CH1.meta.format.mdAlarmEnabled=0
CH1.meta.transport.multicastEnabled=0
CH1.meta.transport.multicastAddress=234.1.2.3
CH1.meta.transport.multicastPort=10004
CH1.meta.transport.ttl=10

CH2.enabled=1
CH2.name=MJPEG
CH2.transportType=2
CH2.video.enabled=1
CH2.video.format.sourceDevice=0
CH2.video.format.codecType=MJPEG

CH2.video.format.constantBitrate=0
CH2.video.format.bitrateInKbps=64
CH2.video.format.resolutionWidth=1280
CH2.video.format.resolutionHeight=800
CH2.video.format.frameRate=25
CH2.video.format.frameRate2=20
CH2.video.format.gop=0
CH2.video.format.quality=30
CH2.video.transport.multicastEnabled=0
CH2.video.transport.multicastAddress=234.1.2.3
CH2.video.transport.multicastPort=20000
CH2.video.transport.ttl=10
CH2.audio.enabled=1
CH2.audio.format.codecType=G711
CH2.audio.format.codecSubType=PCMU
CH2.audio.transport.multicastEnabled=0
CH2.audio.transport.multicastAddress=
CH2.audio.transport.multicastPort=20002
CH2.audio.transport.ttl=10
CH2.meta.enabled=1
CH2.meta.format.mdAlarmEnabled=0
CH2.meta.transport.multicastEnabled=0
CH2.meta.transport.multicastAddress=234.1.2.3
CH2.meta.transport.multicastPort=20004
CH2.meta.transport.ttl=10

B.You can set:
URL: http://192.168.1.1/cgi-bin/channels.cgi
HTML body: action=updateAll&
c1AudioFormatCodecType=AMR&c1AudioFormatCodecSubType=AMR-MR122&
c2AudioFormatCodecType=G711&c2AudioFormatCodecSubType=PCMU

5.3 GetStream

ActionEvent: getStream

URL Syntax	rtsp://<*IP*>/channel<*index*>
Response	
Comment	<*Index*> is the index number of the SChannelSetting. If user enables MJPG over http setting, user can set request http://<*IP*>/channel<*index*> For example:rtsp://192.168.1.100/channel1
HTTP Method	

5.4 SetRtsp

ActionEvent: setRtsp

URL Syntax	http://<*IP*>/cgi-bin/rtsp.cgi
HTML Body	action=set [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
port	554, [1025-65534]	0:disabled 1:enabled
authentication	[0,1,2]	0: NONE 1: BASIC 2: DIGEST

5.5 GetRtsp

ActionEvent: getRtsp

URL Syntax	http://<*IP*>/cgi-bin/rtsp.cgi?action=get
Response	rtsp.port=554 rtsp.authentication=
Comment	
HTTP Method	GET

5.6 GetinboundChannel

ActionEvent: getinboundChannel

URL Syntax	http://<*IP*>/cgi-bin/inboundChannel.cgi?action=get
Response	ch1.transportType=1 ch1.port=12345 ch1.codec=PCMA
Comment	
HTTP Method	GET

5.7 SetVideoRecord

ActionEvent: setVideoRecord

URL Syntax	http://<*IP*>/cgi-bin/videoRecord.cgi
HEML Body	action=set [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	Users can enable or disable the offline recording and configure the related parameters. Offline recording records the video feed to an SD or Micro SD card when the camera detects the disconnected physical or network layer for more than a specific duration of *physicalTime* or *networkTime*. If circulateEnabled is enabled, the camera will automatically overwrite the recorded files when a specific amount of free space is left on the SD or Micro SD card.
HTTP Method	POST

Argument	**Valid values**	**Description**
circulateEnabled	[0,1]	0:disabled 1:enabled
offlineRecordEnabled	[0,1]	0:disabled 1:enabled
physicalTime		Default: 10 seconds.
networkTime		Default: 30 seconds.

5.8 GetVideoRecord

ActionEvent: getVideoRecord

URL Syntax	http://<*IP*>/cgi-bin/videoRecord.cgi?action=get
Response	circulateEnabled= offlineRecordEnabled= physicalTime= networkTime=
Comment	
HTTP Method	GET

5.9 InboundAudio

ActionEvent: inboundAudio

URL Syntax	http://<*IP*>:<*port*>/cgi-bin/**instream.cgi** //// ---------- HTTP headers ------- Content-Type: audio/basic (indicate uLaw is used for audio compression) Cache-Control: no-cache (prevent caching at HTTP proxy server) Content-Length: 10000000 (indicate the length of the audio session) [Other headers...] //// ---------- HTTP body ---------- [uLaw audio frame] [uLaw audio frame] [uLaw audio frame] [uLaw audio frame] [uLaw audio frame] [uLaw audio frame] …
Response	N/A
Comment	Allow users to talk through an IP camera
HTTP Method	POST

5.10 GetSnapshot

ActionEvent: getSnapshot

URL Syntax	http://<*IP*>/cgi-bin/media.cgi?action=getSnapshot
Response	HTTP/1.1 200 OK .Content-Type: image/jpeg .Content-Length: 65542 .Content-Disposition: attachment; filename="snapshot20090101_003801.jpg" .Connection: close .....JFIF.....C. . . . . ……..
Comment	
HTTP Method	GET

6. Camera

Camera API allows applications to view/configure the Camera/lens setting.

6.1 SetCameraSetting

ActionEvent: setCameraSetting

URL Syntax	http://<*IP*>/cgi-bin/camera.cgi
HTML Body	action=**setCameraSetting** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
brightness.level	[-5 - 5]	Depends on the limit of hardware
colorSaturation.level	[-5 - 5]	Depends on the limit of hardware
sharpness.level	[-5 - 5]	Depends on the limit of hardware
contrast.level	[-5 - 5]	Depends on the limit of hardware
flipEnabled	[0,1]	0:disabled 1:enabled
mirrorEnabled	[0,1]	0:disabled 1:enabled
freq	[0,1]	0, FREQ_60HZ 1, FREQ_50HZ
effectMode	[0,1]	0: Auto 1: Manual
colorMode	[0,1]	0: Color 1: Black&White
envMode	[0,1]	0: OUTDOOR 1: INDOOR
IRCutFilter.mode	[0,1,2]	0: OFF 1: ON 2: AUTO
IRLED.mode	[0,1,2]	0: OFF 1: ON 2: AUTO
Exp.mode	[0,1,2,4]	0:Sport

		1: Normal 2: Night Vision 3: AE_LOCK, // Reserve 4: User Defined
Exp.AGC.enabled	[0,1]	Exp.mode is User Defined: When Exp.AELockEnabled =0, AGC 、shutterSpeed can be set; Exp.AELockEnabled =1, disabled AGC 、shutterSpeed function. 0:disabled 1:enabled
Exp.AGC.level	[0-4]	Exp.mode is User Defined: When Exp.AELockEnabled =0, AGC 、shutterSpeed can be set; Exp.AELockEnabled =1, disabled AGC 、shutterSpeed function. depends on the limit of hardware
Exp.shutterSpeed.enabled	[0,1]	Exp.mode is User Defined: When Exp.AELockEnabled =0, AGC 、shutterSpeed can be set; Exp.AELockEnabled =1, disabled AGC 、shutterSpeed function. 0:disabled 1:enabled
Exp.shutterSpeed.level	[-2 - 4]	Exp.mode is User Defined: When Exp.AELockEnabled =0, AGC 、shutterSpeed can be set; Exp.AELockEnabled =1, disabled AGC 、shutterSpeed function. depends on the limit of hardware
Exp.AELockEnabled	[0,1]	Exp.mode is User Defined: When Exp.AELockEnabled =0, AGC 、shutterSpeed can be set; Exp.AELockEnabled =1, disabled AGC 、shutterSpeed function. 0:disabled 1:enabled

autoIris.enabled	[0,1]	0:disabled 1:enabled
videoOverlay.useTimestamp	[0,1]	0:disabled 1:enabled
videoOverlay.useText	[0,1]	0:disabled 1:enabled
videoOverlay.displayString	<*string*> (25)	Allows [A-Z][0-9][=,-.:] and space
videoOverlay.osdWindow1.XPos	0	Text 0:Left
videoOverlay.osdWindow1.YPos	[0,1]	Text 0: Top 1: Bottom
videoOverlay.osdWindow1.transparent	[0,1,2,3]	0:0% 1:50% 2:75% 3:100%
videoOverlay.osdWindow2.XPos	0	Timestamp 0:Left
videoOverlay.osdWindow2.YPos	[0,1]	Timestamp 0: Top 1: Bottom
videoOverlay.osdWindow2.transparent	[0,3]	0:0% 1:50% 2:75% 3:100%
videoOverlay.osdPalette1.y	[0-255]	
videoOverlay.osdPalette1.Cb	[0-255]	
videoOverlay.osdPalette1.Cr	[0-255]	
videoOverlay.osdPalette2.y	[0-255]	
videoOverlay.osdPalette2.Cb	[0-255]	
videoOverlay.osdPalette2.Cr	[0-255]	
privacy.enabled	[0,1]	Depends on mode name
privacy.privacyButtonEnabled	[0,1]	Depends on mode name
lightSensor.thresholdLevel	[0-255]	Depends on model name
deNoise.level	0,1,2,3,4	0: auto de-noise level: 1: (default) 2: 3: 4: best

6.2 GetCameraSetting

ActionEvent: getCameraSetting

URL Syntax	http://<*IP*>/cgi-bin/camera.cgi?action=**getCameraSetting**
Response	brightness.level=0

	colorSaturation.level=0 flipEnabled=0 mirrorEnabled=0 sharpness.level=0 contrast.level=0 freq=0 envMode=1 effectMode=1 colorMode=0 lightSensor.thresholdLevel=11 IRCutFilter.mode=2 Exp.mode=1 Exp.AGC.enabled=0 Exp.AGC.level=0 Exp.shutterSpeed.enabled=0 Exp.shutterSpeed.level=0 Exp.AELockEnabled=0 autoIris.enabled=0 videoOverlay.useTimestamp=0 videoOverlay.displayString= videoOverlay.useText=0 videoOverlay.osdPalette1.y=255 videoOverlay.osdPalette1.Cb=128 videoOverlay.osdPalette1.Cr=128 videoOverlay.osdPalette2.y=16 videoOverlay.osdPalette2.Cb=128 videoOverlay.osdPalette2.Cr=128 videoOverlay.osdWindow1.XPos=0 videoOverlay.osdWindow1.YPos=0 videoOverlay.osdWindow1.transparent=0 videoOverlay.osdWindow2.XPos=0 videoOverlay.osdWindow2.YPos=0 videoOverlay.osdWindow2.transparent=0 privacy.enabled= privacy.PrivacyButtonEnabled=
Comment	
HTTP Method	GET

7. Audio

Audio API allows applications to
1) View/adjust the audio device settings
2) Adjust the volume of the audio device

7.1 SetAudioDevice

ActionEvent: setAudioDevice

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML Body	action=**setAudioDevice** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
muted	[0,1]	0: (un-muted) 1: (muted)
level	[1-100]	Speaker volume
mediaType	0	0,Full 1, Half duplex (Reserve)
voiceSource	[0,1]	0, MIC 1, Line in
echo	[0,1]	0:disabled 1:enabled

7.2 GetAudioDevice

ActionEvent: getAudioDevice

URL Syntax	http://<*IP*>/cgi-bin/ audio.cgi?action=**getAudioDevice**
Response	muted = level = mediaType= voiceSource = echo=
Comment	
HTTP Method	GET

7.3 Playaudio

ActionEvent: play

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML BODY	action=play [&<*argument*>=<*value*>]
Response	
Comment	The device will play the audio recording
HTTP Method	POST

Argument	Valid values	Description
name	<*String*>	

7.4 Stopaudio

ActionEvent: stopaudio

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML Body	action=**stopaudio** [&<*argument*>=<*value*>]
Response	
Comment	The device will stop playing the audio recording
HTTP Method	POST

Argument	Valid values	Description
name	<*String*>	

7.5 Recordaudio

ActionEvent: record

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML Body	action=**record** [<*argument*>=<*value*>]
Response	
Comment	The device will start recording audio
HTTP Method	POST

Argument	Valid values	Description
name	<*String*> (10)	

7.6 Stoprecordaudio

ActionEvent: stoprecord

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML Body	action=**stoprecord** [&<*argument*>=<*value*>]
Response	
Comment	The device will stop recording audio
HTTP Method	POST

7.7 GetFilestatus

ActionEvent: getFilestatus

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi?action=**getFilestatus**
Response	size= audiofile1.name= audiofile1.size= audiofile1.time= audiofile1.codecType= audiofile2.name= audiofile2.size= audiofile2.time= audiofile2.codecType= …….
Comment	
HTTP Method	GET

7.8 RemoveAudioFile

ActionEvent: remove

URL Syntax	http://<*IP*>/cgi-bin/audio.cgi
HTML Body	action=**remove** [&<*argument*>=<*value*>]
Response	
Comment	The device will remove file
HTTP Method	POST

Argument	**Valid values**	**Description**
name	<*String*>	

8. Network

Network API allows applications to view/adjust the network-related settings, including IP address, WIFI network, etc.

8.1 SetBasicNetwork

ActionEvent: setBasicNetwork

URL Syntax	http://<*IP*>/cgi-bin/basicNetwork.cgi
HTML Body	action= **set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
addressType	[0,1,2]	0: STATIC 1: DHCP 2: PPPOE
ipv4Address		addressType=0 1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
subnetMask		addressType=0 1~255 . 0~255 . 0~255 . 0~254
gatewayAddress		addressType=0 1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
dnsAddress1		addressType=0 1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
dnsAddress2		addressType=0 1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
pppoe.username	<*String*> (128)	addressType=2
pppoe.password	<*String*> (128)	addressType=2

8.2 GetBasicNetwork

ActionEvent: getBasicNetwork

URL Syntax	http://<*IP*>/cgi-bin/basicNetwork.cgi?action=**get**
Response	addressType= (0=Static,1=DHCP, 2=PPPoE) ipv4Address= subnetMask= gatewayAddress= dnsAddress1= dnsAddress2= pppoe.username= pppoe.password=
Comment	
HTTP Method	GET

8.3 SetUPnP

ActionEvent: setUPnP

URL Syntax	http://<*IP*>/cgi-bin/upnp.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
enabled	[0,1]	0:disabled 1:enabled
name	<*String*> (128)	

8.4 GetUPnP

ActionEvent: getUPnP

URL Syntax	http://<*IP*>/cgi-bin/upnp.cgi?action=**get**
Response	enabled= name=
Comment	
HTTP Method	GET

8.5 SetDDNS

ActionEvent: setDDNS

URL Syntax	http://<*IP*>/cgi-bin/ddns.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]

Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
dyndnsEnabled	[0,1]	0:disabled 1:enabled
dyndns.wildcardEnabled	[0,1]	0:disabled 1:enabled
dyndns.username	<*String*> (128)	
dyndns.password	<*String*> (128)	
dyndns.hostname	<*String*> (128)	
tzodnsEnabled	[0,1]	0:disabled 1:enabled
tzodns.wildcardEnabled	[0,1]	0:disabled 1:enabled
tzodns.username	<*String*> (128)	
tzodns.password	<*String*> (128)	
tzodns.hostname	<*String*> (128)	

8.6 GetDDNS

ActionEvent: getDDNS

URL Syntax	http://<*IP*>/cgi-bin/ddns.cgi? action=get
Response	dyndnsEnabled=0 dyndns.wildcardEnabled= dyndns.username= dyndns.password= dyndns.hostname= tzodnsEnabled= tzodns.wildcardEnabled= tzodns.username= tzodns.password= tzodns.hostname=
Comment	
HTTP Method	GET

8.7 SetEthernet

ActionEvent: setEthernet

Syntax	http://<*IP*>/cgi-bin/ethernet.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	

HTTP Method	POST

Argument	Valid values	Description
mediaType	[0,1,2,3,4]	0:AUTO 1: 10_HALF_DUPLEX 2: 10_FULL_DUPLEX 3: 100_HALF_DUPLEX 4: 100_FULL_DUPLEX

8.8 GetEthernet

ActionEvent: getEthernet

URL Syntax	http://<IP>/cgi-bin/ethernet.cgi?action=**get**
Response	mediaType=
Comment	
HTTP Method	GET

8.9 SetWIFI

ActionEvent: setWIFI

URL Syntax	http://<*IP*>/cgi-bin/wifi.cgi
HTML Body	action= **set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
enabled	[0,1]	0:disabled 1:enabled (Reserve)
securityMode	[0,1,2,3]	0:None 1:WEP 2: WPAPSK 3: WPA2PSK
WPA.algorithmType	[0,1,2]	securityMode=2(WPAPSK) 0: TKIP 1: AES 2: TKIP_AES securityMode=3(WPA2PSK) 1: AES
WPA.sharedKey	<*String*> (64)	securityMode=2,3
Channel	0	(0) Auto 1-13 for ADHOC(Reserve)
mode	[0,1]	0: INFRASTRUCTURE

		1: ADHOC (Reserve)
SSID	<*String*> (33)	
WEP.defaultTransmitKeyIndex	[1-4]	Available when securityMode=1 (WEP)
WEP.wepKeyLength		Available when securityMode=1 (WEP) 1: 40/64 bits (10 hex digits) 2: 104/128 bits (26 hex digits)
WEP.encryptionKeyList.keyEntry1.encryptionKey	<*String*> (64)	Available when securityMode=1 (WEP)
WEP.encryptionKeyList.keyEntry2.encryptionKey	<*String*> (64)	Available when securityMode=1 (WEP)
WEP.encryptionKeyList.keyEntry3.encryptionKey	<*String*> (64)	Available when securityMode=1 (WEP)
WEP.encryptionKeyList.keyEntry4.encryptionKey	<*String*> (64)	Available when securityMode=1 (WEP)
WEP.authenticationType	[0,1,2]	Available when securityMode=1 (WEP) 0:Open System 1:Shared Key 2:Auto
wmm	[0,1]	0:disabled 1:enabled **802.1e QoS**
operationMode	[0-10]	0: AUTO 1: 11G 2: 11B 3: 11N 4: 11BG 5: 11GN 6: 11BGN 7: 11A 8: 11AN 9: 11N_5G 10: 11ABGN depends on the limit of hardware
channelBandWidth	[0,1]	0:Auto (20/40MHZ) 1: 20 MHZ

8.10 GetWIFI

ActionEvent: getWIFI

URL Syntax	http://<*IP*>/cgi-bin/wifi.cgi?action=**get**
Response	enabled= channel= mode= SSID= wmm= operationMode= channelBandWidth= securityMode= countryregion= WPA.algorithmType= WPA.sharedKey= WEP. authenticationType= WEP. defaultTransmitKeyIndex = WEP. wepKeyLength= WEP. encryptionKeyList.Keyentry1.encryptionKey= WEP. encryptionKeyList.Keyentry2.encryptionKey= WEP. encryptionKeyList.Keyentry3.encryptionKey= WEP. encryptionKeyList.Keyentry4.encryptionKey= WPS.wpsbtn_enabled= WPS.PINCode=
Comment	
HTTP Method	GET

8.11 SetIPFilter

ActionEvent: setIPFilter

URL Syntax	http://<*IP*>/cgi-bin/IPFilter.cgi
HTML Body	action= **set** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
enabled	[0,1]	0:disabled 1:enabled
permissionType	[0,1]	0: Deny 1: Allow
allow.enabled1	[0,1]	0:disabled 1:enabled
allow.startIP1	<*String*> (16)	
allow.endIP1	<*String*> (16)	

allow.enabled2	[0,1]	0:disabled 1:enabled
allow.startIP2	<*String*> (16)	
allow.endIP2	<*String*> (16)	
allow.enabled**N**	[0,1]	0:disabled 1:enabled **N**=[1-10]
allow.startIP**N**	<*String*> (16)	**N**=[1-10]
allow.endIP**N**	<*String*> (16)	**N**=[1-10]
deny.enabled1	[0,1]	0:disabled 1:enabled
deny.startIP1	<*String*> (16)	
deny.endIP1	<*String*> (16)	
deny.enabled2	[0,1]	0:disabled 1:enabled
deny.startIP2	<*String*> (16)	
deny.endIP2	<*String*> (16)	
deny.enabled**N**	[0,1]	0:disabled 1:enabled **N**=[1-10]
deny.startIP**N**	<*String*> (16)	**N**=[1-10]
deny.endIP**N**	<*String*> (16)	**N**=[1-10]

8.12 GetIPFilter

ActionEvent: getIPFilter

URL Syntax	http://<*IP*>/cgi-bin/ IPFilter.cgi?action=**get**
Response	enabled= permissionType= allow.size= allow.enabled1= allow.startIP1= allow.endIP1= allow.enabled2= allow.startIP2= allow.endIP2= ……. deny.size= deny.enabled1= deny.startIP1= deny.endIP1= deny.enabled2= deny.startIP2= deny.endIP2=

Comment	
HTTP Method	GET

9. Storage

Storage API allows applications to configure the storage devices which could be reached by the IPCAM unit.

9.1 GetSDstatus

ActionEvent: getSDstatus

URL Syntax	http://<*IP*>/cgi-bin/sdcard.cgi?action=**getSDstatus**
Response	size= file1.name= file1.size= file1.time= file2.name= file2.size= file2.time= ……
Comment	
HTTP Method	GET

9.2 Mount

ActionEvent: mount

URL Syntax	http://<*IP*>/cgi-bin/sdcard.cgi?action=**mount**
Response	
Comment	
HTTP Method	GET

9.3 Umount

ActionEvent: umount

URL Syntax	http://<*IP*>/cgi-bin/sdcard.cgi?action=**umount**
Response	
Comment	
HTTP Method	GET

9.4 RemoveFile

ActionEvent: rm

URL Syntax	http://<*IP*>/cgi-bin/sdcard.cgi?action=rm&filename=<*value*>
Response	
Comment	
HTTP Method	GET

10. System

System API allows applications to configure miscellaneous system settings which are not covered by other categories. These settings include Time, Syslog, Storage, and Device Information.

NOTE: In the future, the API may switch to rsyslog instead of syslogd.

10.1 GetDeviceInfo

ActionEvent: getDeviceInfo

URL Syntax	http://<*IP*>/cgi-bin/system.cgi?action=**get**
Response	chipVersion= sensorID= macAddress= firmwareVersion= firmwareReleasedDate= InternalName= ProductName= ModelNumber= CompanyName= Comments=
Comment	
HTTP Method	GET

10.2 SetTimeSetting

ActionEvent: setTimeSetting

URL Syntax	http://<*IP*>/cgi-bin/ time.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
type	[0,1,2]	0:None 1:Manual 2:NTP
enableDST	[0,1]	0:disabled 1:enabled
timezoneID	[0-24]	Table
manual.year	[2009-2038]	

manual.month	[1-12]	
manual.day	[1-31]	
manual.hour	[0-23]	
manual.minute	[0-59]	
manual.second	[0-59]	
ntp.ntpServerLoc1	<*String*> (128)	
ntp.ntpServerLoc2	<*String*> (128)	

Table: Available **timezoneID**

timezoneID	Value
TIME_ZONE_MIN	**0**
TIME_ZONE_KWAJALEIN	**1**
TIME_ZONE_SAMOA	**2**
TIME_ZONE_HAWAII	**3**
TIME_ZONE_ALASKA	**4**
TIME_ZONE_LOS_ANGELES	**5**
TIME_ZONE_PHOENIX	**6**
TIME_ZONE_MEXICO_CITY	**7**
TIME_ZONE_NEW_YORK	**8**
TIME_ZONE_SANTIAGO	**9**
TIME_ZONE_SAO_PAULO,	**10**
TIME_ZONE_NORONHA_ISLAND,	**11**
TIME_ZONE_PRAIA,	**12**
TIME_ZONE_LONDON,	**13**
TIME_ZONE_PARIS,	**14**
TIME_ZONE_CAIRO,	**15**
TIME_ZONE_MOSCOW,	**16**
TIME_ZONE_DUBAI,	**17**
TIME_ZONE_KARACHI,	**18**
TIME_ZONE_DHAKA,	**19**
TIME_ZONE_JAKARTA,	**20**
TIME_ZONE_HONG_KONG,	**21**
TIME_ZONE_TOKYO,	**22**
TIME_ZONE_SYDNEY,	**23**
TIME_ZONE_NOUMEA,	**24**
TIME_ZONE_NewZealand,	**25**
TIME_ZONE_MAX	**26**

10.3 GetTimeSetting

ActionEvent: getTimeSetting

URL Syntax	http://<*IP*>/cgi-bin/time.cgi?action=**get**
Response	type= enableDST= timezoneID=

	manual.year= manual.month= manual.day= manual.hour= manual.minute= manual.second= enableDST= timezoneID= ntp.ntpServerLoc1= ntp.ntpServerLoc2=
Comment	
HTTP Method	GET

10.4 SetSyslogSetting

ActionEvent: setSyslogSetting

URL Syntax	http://<*IP*>/cgi-bin/ syslog.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
localLogLevel	[0-7]	table
useRemoteLog	[0,1]	0:disabled 1:enabled
addressingFormatType	[0,1]	0:IP 1:Hostname
remoteServerAddress	<*String*> (128)	
remoteServerPort	[514, 1025-65534]	

Table: Available localLogLevel

timezoneID	Value
SLOG_EMERG	0
SLOG_ALERT	1
SLOG_CRIT	2
SLOG_ERR	3
SLOG_WARNING	4
SLOG_NOTICE	5
SLOG_INFO	6
SLOG_DEBUG	7

10.5 GetSyslogSetting

ActionEvent: getSyslogSetting

URL Syntax	http://<*IP*>/cgi-bin/syslog.cgi ?action=**get**
Response	localLogLevel= useRemoteLog= addressingFormatType= remoteServerAddress= remoteServerPort=
Comment	
HTTP Method	GET

10.6 GetSyslogFile

ActionEvent: getSyslogFile

URL Syntax	http://<*IP*>/syslog.dump
Response	Content of syslog.
Comment	
HTTP Method	GET

10.7 SyslogClear

ActionEvent: syslogClear

URL Syntax	http://<*IP*>/cgi-bin/syslog.cgi?action=clear
Response	
Comment	Clear syslog.
HTTP Method	GET

10.8 SetOperationSetting

ActionEvent: set

URL Syntax	http://<*IP*>/cgi-bin/OperationSetting.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	Default=en-US
HTTP Method	POST

Argument	**Valid values**	**Description**
locale	<*String*> (65)	Default=en-US

10.9 GetOperationSetting

ActionEvent: get

URL Syntax	http://<*IP*>/cgi-bin/OperationSetting.cgi?action=**get**
Response	locale=
Comment	
HTTP Method	GET

11. Admin

Admin API enables applications to execute administrative tasks on the IPCAM unit. The tasks include add/delete users, upgrade firmware, rebooting the IP camera, and import/exporting configuration settings.

11.1 AddUser

ActionEvent: addUser

Syntax	http://<*IP*>/cgi-bin/users.cgi
HTML Body	action=**add** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
index	<*String*> (65)	depends on GUI index
username	<*String*> (30)	Unique key Length limited:[4,29]
password	<*String*> (30)	Length limited:[4,29]
privilege	[0,1,2]	0:Viewer 1:Admin 2:Remote viewer

11.2 DeleteUser

ActionEvent: deleteUser

URL Syntax	http://<*IP*>/cgi-bin/users.cgi?action=**delete** [<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
username	<*String*> (30)	

11.3 GetUsers

ActionEvent: getUsers

URL Syntax	http://<*IP*>/cgi-bin/users.cgi?action=**getUsers**

Response	Size= User1.index= User1.username= User1.password= User1.privilege= … User2.username= User2.password= User2.privilege=
Comment	
HTTP Method	GET

11.4 UpdateUser

ActionEvent: updateUser

Syntax	http://<*IP*>/cgi-bin/users.cgi
HTML Body	action=**update** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
index	<*String*> (65)	depends on GUI index
username	<*String*> (30)	Unique key Length limited:[4,29]
password	<*String*> (30)	Length limited:[4,29]
privilege	[0,1,2]	0:Viewer 1:Admin 2:Remote viewer

11.5 SetHTTP/HTTPS

ActionEvent: setHTTP/HTTPS

URL Syntax	http://<*IP*>/cgi-bin/http.cgi
HTML Body	action= **setAll** [&<*argument*>=<*value*>&<*argument*>=<*value*>...]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
enabled	[0,1]	0:disabled 1:enabled
port	[80, 1025-65534]	

httpsEnabled	[0,1]	0:disabled 1:enabled
httpsPort	[443, 1025-65534]	

11.6 GetHTTP

ActionEvent: getHTTP

URL Syntax	http://<*IP*>/cgi-bin/http.cgi?action=**get**
Response	enabled= port=
Comment	
HTTP Method	GET

11.7 GetHTTPS

ActionEvent: getHTTPS

URL Syntax	http://<*IP*>/cgi-bin/https.cgi?action=**get**
Response	enabled= port=
Comment	
HTTP Method	GET

11.8 ResetToDefault

ActionEvent: resetToDefault

URL Syntax	http://<*IP*>/cgi-bin/reset.cgi?action=**reset** **[&keepip=[0,1]&keepwifi=[0,1]]**
Response	
Comment	Reset all settings to factory default
HTTP Method	GET

Argument	**Valid values**	**Description**
keepip	[0,1]	0:disabled 1:enabled
keepwifi	[0,1]	0:disabled 1:enabled **depends on the limit of hardware**

11.9 UpgradeFirmware

ActionEvent: upgradeFirmware

URL Syntax	http://<*IP*>/cgi-bin/upgradeFirmware.cgi
HTML Body	action=**upgrade** ***< boundary>*** ***< firmware>*** ***< boundary>***
Response	
Comment	Upgrade the system firmware upon this request
HTTP Method	POST

11.10 Reboot

ActionEvent: reboot

URL Syntax	http://<*IP*>/cgi-bin/reboot.cgi?action=**reboot**
Response	
Comment	Reboot the system
HTTP Method	GET/POST

11.11 ImportConfigFile

ActionEvent: importConfigFile

Syntax	http://<*IP*>/cgi-bin/ConfigFile.cgi
HTML Body	<*boundary*> <*Configuration file*> <*boundary*>
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
filename	<*String*> (64)	

11.12 ExportConfigFile

ActionEvent: exportConfigFile

URL Syntax	http://<*IP*>/cgi-bin/ConfigFile.cgi?action=**get**
Response	
Comment	
HTTP Method	GET

12. Capability

ActionEvents

ActionEvent	Description
getCapability	Get camera's capability.
getVideoCodecs	Get video codecs
getResolutions	Get video resolutions
getAudioCodecs	Get audiocodecs

12.1 GetCapability

ActionEvent: getCapability

URL Syntax	http://<*IP*>/cgi-bin/capability.cgi?action=**get**
Response	[media] channels=2 videoCodecs=H264,MJPEG,MPEG4 audioCodesc=G711,AMR-NB resolutions=1280x800,640x400,320x192 H264frameRate=2,3,5,7,10,15,20,25,30 MJPEGframeRate=2,3,5,7,10,15,20,25,30 MPEG4frameRate=2,3,5,7,10,15 bitrate=64,128,256,384,512,768,1000,1500,2000,4000,6000,8000,10000,12000
Comment	
HTTP Method	GET

12.2 GetVideoCodecs

ActionEvent: getVideoCodecs

URL Syntax	http://<*IP*>/cgi-bin/capability.cgi?action=**getVideoCodecs**
Response	videoCodecs=H264,MJPEG,MPEG4
Comment	
HTTP Method	GET

12.3 GetResolutions

ActionEvent: getResolutions

URL Syntax	http://<*IP*>/cgi-bin/capability.cgi?action=**getResolutions**
Response	resolutions=1280x800,640x400,320x192
Comment	
HTTP Method	GET

12.4 GetAudioCodecs

ActionEvent: getAudioCodecs

URL Syntax	http://<*IP*>/cgi-bin/capability.cgi?action=**getAudioCodecs**
Response	audioCodesc=G711,AMR-NB
Comment	
HTTP Method	GET

13. Motion detection

Motion detection API allows applications to view/adjust the motion detection settings.

13.1 SetMotionDetection

ActionEvent: setMotionDetection

URL Syntax	http://<*IP*>/cgi-bin/motiondetection.cgi
HTML Body	action=**set** &channelIndex=1 [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
channelIndex	1	It match stream channel index, but it is always equal 1 now.
enabled	[0,1]	0:disabled 1:enabled
detectionInterval	>0	millisecond
region1.enabled	[0,1]	0:disabled 1:enabled
region1.sensitivity	[1-100]	
region1.threshold	[1-100]	
region1.x		Depends on resolution
region1.y		Depends on resolution
region1.x1		Depends on resolution
region1.y1		Depends on resolution
region2.enabled	[0,1]	0:disabled 1:enabled
region2.sensitivity	[1-100]	
region2.threshold	[1-100]	
region2.x		Depends on resolution
region2.y		Depends on resolution
region2.x1		Depends on resolution
region2.y1		Depends on resolution
region3.enabled	[0,1]	0:disabled 1:enabled
region3.sensitivity	[1-100]	
region3.threshold	[1-100]	
region3.x		Depends on resolution
region3.y		Depends on resolution
region3.x1		Depends on resolution

region3.y1		Depends on resolution

Note:
Sensitivity: When sensitivity is a high value (such as 90), the motion detection is easily triggered.
Threshold: When threshold is a low value (such as 10), the motion detection is easily triggered.

13.2 GetMotionDetections

ActionEvent: getMotionDetections

URL Syntax	http://<*IP*>/cgi-bin/ motiondetection.cgi?action=**get**
Response	size=1 MD1.enabled=0 MD1.channelIndex=1 MD1.detectionInterval=100 MD1.region.size=3 MD1.region1.enabled=0 MD1.region1.sensitivity=90 MD1.region1.threshold=10 MD1.region1.x=0 MD1.region1.y=0 MD1.region1.x1=0 MD1.region1.y1=0 MD1.region2.enabled=0 MD1.region2.sensitivity=90 MD1.region2.threshold=10 MD1.region2.x=0 MD1.region2.y=0 MD1.region2.x1=0 MD1.region2.y1=0 MD1.region3.enabled=0 MD1.region3.sensitivity=90 MD1.region3.threshold=10 MD1.region3.x=0 MD1.region3.y=0 MD1.region3.x1=0 MD1.region3.y1=0
Comment	
HTTP Method	GET

14. Event

Event API allows applications to
1) View/adjust the event settings
2) View/adjust the notification settings

14.1 AddEventSetting

ActionEvent: addEventSetting

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?
HTML Body	action=**addEventSetting** [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
name	<*String*> (128)	unique ID
enabled	[0,1]	0:disabled 1:enabled
eventID	<*String*> (128) MD, SCHEDULE, DI, AD, PIR, IVAPC, IVAMD, IVAFZ	1.MD : motion detection 2. SCHEDULE: schedule event, only support Save Stream to SD. 3.DI: 4.AD :audio detection 5.PIR: 6.IVAPC :IVA people count 7.IVAMD :IVA motion detection 8.IVAFZ :IVA Only support one eventID. Depends on capability. See example.
sched.type	[0,1,2]	0:Always 1:Weekly 2: Durative

sched.time	<*String*> (128)	
actions	<*String*> (128)	SAVESTREAM SNAPSHOT EMAIL LIGHTLED SPEAKER DO UDP HTTP MULTICAST UDP Depends on capability. See example.
speakerName	<*String*> (64)	Depends on capability. (SPEAKER)
receiverAddress1	<*String*> (128)	
receiverAddress2	<*String*> (128)	
senderAddress	<*String*> (128)	
senderName	<*String*> (64)	
subject	<*String*> (64)	
udp.ipAddress	<*String*> (32)	
udp.portNo	[1025-65534]	
multicast.ipAddress	<*String*> (32)	
multicast.portNo	[1025-65534]	

The syntax of actions is [<*ACTION*>:<*METHOD*>][,<*ACTION*>:<*METHOD*>]…
Example:

SAVESTREAM: [FTP, EMAIL, SAMBA, SD]
SNAPSHOT: [FTP, EMAIL, SAMBA, SD]
EMAIL: EMAIL
HTTP:NONE
UDP: NONE
UDP: NONE
MULTICAST: NONE
DO: NONE

NONE ➔read notifications.

Example:

Get capability:

Events=MD, SCHEDULE, DI, AD

Event.actions=SNAPSHOT, EMAIL, HTTP, UDP, MULTICAST, DO

Event.method=FTP, SMTP, SAMBA, SD

2. EventID:MD schedule type: Always
   Action:

☐ Take snapshot to FTP

☑ Activate Digital Output

☐ Send HTTP Notification

☐ Send to Email

☐ Send UDP Notification to IP address . . . .

Port

☐ Send Multicast Notification to IP address . . . .

Port

Take snapshot to ➔ use E-mail (need to set E-mail format)

Send to Email ➔use E-mail (need to set E-mail format)

Activate Digital Output ➔ read notifications.

Send HTTP Notification➔ read notifications.

Send UDP Notification to IP address ➔need to set ***udp.ipAddress*** 、 ***udp.portNo***

Send Multicast Notification to IP address➔need to set ***multicast.ipAddress*** 、 ***multicast.portNo***

URL: http://192.168.1.1/cgi-bin/event.cgi

HTML body:

action=addEventSetting&name=MD_Test&enabled=1&sched.type=0&eventID=MD&actions=SNAPSHOT:EMAIL,EMAIL:EMAIL,DO:NONE,HTTP:NONE,UDP:NONE,MULTICAST:NONE&senderAddress=aa@brickcom.com&receiverAddress1=bb@brickcom.com&receiverAddress2=cc@brickcom.com&senderName=Brickcom&subject=Test&udp.ipAddress=1.2.3.4&udp.portNo=3333&multicast.ipAddress=2.3.4.5&multicast.portNo=2234

3. EventID: SCHEDULE ; schedule type: Durative
   Action: save stream to SD

Time: Sun ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri ☑ Sat ☐

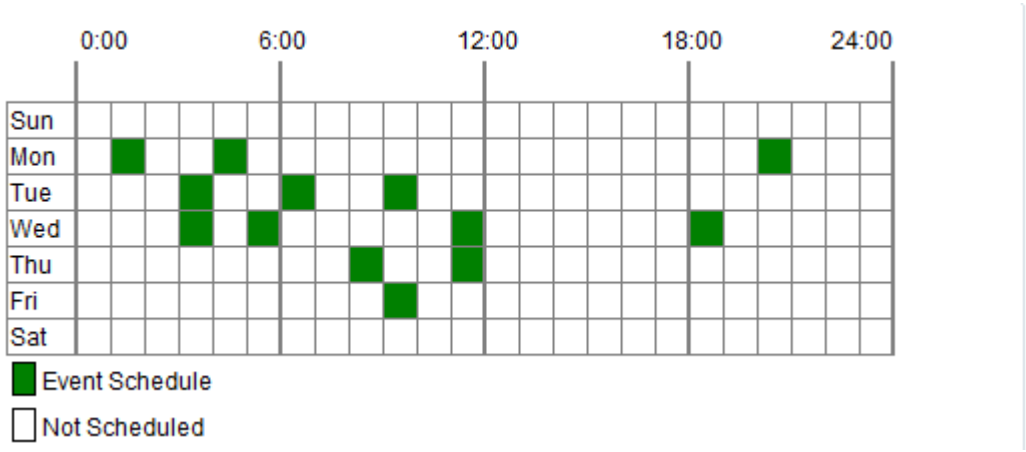
Start 20 : 23 Duration 300 Minutes

URL: http://192.168.1.1/cgi-bin/event.cgi

HTML body:

action=addEventSetting&name=123&enabled=1&sched.type=2&sched.time=0:2023-300,1:2023-300,2:2023-300,3:2023-300,4:2023-300,5:2023-300&eventID=SCHEDULE&actions=SAVESTREAM:SD

4. EventID: DI ; schedule type: Weekly
   schedule time :



Days	Hours presented in binary format	Hours presented in Hex format
Sun	0000 0000 0000 0000 0000 0000	000000
Mon	0100 1000 0000 0000 0000 1000	480008
Tue	0001 0010 0100 0000 0000 0000	124000
Wed	0001 0100 0001 0000 0010 0000	141020
Thu	0000 0000 1001 0000 0000 0000	009000
Fri	0000 0000 0100 0000 0000 0000	004000
Sat	0000 0000 0000 0000 0000 0000	000000

URL: http://192.168.1.1/cgi-bin/event.cgi
HTML body:
action=addEventSetting&name=sss&enabled=1&sched.type=1&sched.time=000000,480008,124000,141020,009000,004000,000000&eventID=DI&actions=DO:NONE,HTTP:NONE

5. URL: http://192.168.1.1/cgi-bin/event.cgi
   HTML body:
   action=addEventSetting&name=xxx&enabled=1&sched.type=0&eventID=AD&actions=DO:NONE,HTTP:NONE

14.2 UpdateEventSetting

ActionEvent: updateEventSetting

URL Syntax	http://<*IP*>/cgi-bin/event.cgi

HTML Body	action=**updateEventSetting** &name=<*value*> [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
name	<*String*> (128)	***unique ID***
enabled	[0,1]	0:disabled 1:enabled
eventID	<*String*> (128) MD, SCHEDULE, DI, AD, PIR, IVAPC, IVAMD, IVAFZ	1.MD : motion detection 2. SCHEDULE: schedule event, only support Save Stream to SD. 3.DI: 4.AD :audio detection 5.PIR: 6.IVAPC :IVA people count 7.IVAMD :IVA motion detection 8.IVAFZ :IVA Only support one eventID. Depends on capability. See example.
sched.type	[0,1,2]	0:Always 1:Weekly 2: Durative
sched.time	<*String*> (128)	
actions	<*String*> (128)	SAVESTREAM SNAPSHOT EMAIL LIGHTLED SPEAKER DO UDP HTTP MULTICAST UDP Depends on capability. See example.
speakerName	<*String*> (64)	Depends on capability.(SPEAKER
receiverAddress1	<*String*> (128)	

receiverAddress2	<*String*> (128)	
senderAddress	<*String*> (128)	
senderName	<*String*> (64)	
subject	<*String*> (64)	
udp.ipAddress	<*String*> (32)	
udp.portNo	[1025-65534]	
multicast.ipAddress	<*String*> (32)	
multicast.portNo	[1025-65534]	

14.3 RemoveEventSetting

ActionEvent: removeEventSetting

Syntax	http://<*IP*>/cgi-bin/event.cgi
HTML Body	action=**removeEventSetting** **&**name=<*value*>
Response	
Comment	
HTTP Method	POST

14.4 GetEventPolicy

ActionEvent: getEventPolicy

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action=**getEventPolicy**
Response	size=4 R1enabled=1 R1name=MD_test R1eventID=MD R1sched.type=0 R1sched.time= R1actions=SAVESTREAM:FTP,SNAPSHOT:FTP R1speakerName= R1receiverAddress1= R1receiverAddress2= R1senderAddress= R1senderName= R1subject= R2enabled=1 R2name=SD_test R2eventID=SCHEDULE R2sched.type=0 R2sched.time=

	R2actions=SAVESTREAM:SD R2speakerName= R2receiverAddress1= R2receiverAddress2= R2senderAddress= R2senderName= R2subject= R3enabled=1 R3name=DI_test R3eventID=DI R3sched.type=1 R3sched.time=100000,100000,100000,100000,100000,100000,100000 R3actions=DO:NONE R3speakerName= R3receiverAddress1= R3receiverAddress2= R3senderAddress= R3senderName= R3subject= R4enabled=0 R4name=AD_test R4eventID=AD R4sched.type=2 R4sched.time=0:0141-300,3:0141-300,4:0141-300,6:0141-300 R4actions=SAVESTREAM:SAMBA,SNAPSHOT:SD,DO:NONE R4speakerName= R4receiverAddress1= R4receiverAddress2= R4senderAddress= R4senderName= R4subject=
Comment	
HTTP Method	GET

14.5 SetEmailSetting

ActionEvent: setEmailSetting

URL Syntax	http://<*IP*>/cgi-bin/event.cgi
HTML Body	action=**setEmailSetting** [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
authenticationMode1	[0,1,2]	
port1	[25, 1025-65534]	

smtpServerHostName1	<*String*> (64)	
password1	<*String*> (64)	
authenticationMode2	<*String*> (64)	
port2	[25,1025-65534]	
smtpServerHostName2	<*String*> (64)	
accountName2	<*String*> (64)	0:PLAIN 1:LOGIN 2: LOGIN with TLS
password2	<*String*> (64)	

Example:

URL: http://192.168.1.1/cgi-bin/event.cgi

HTML body:

action=setEmailSetting&authenticationMode1=1&port1=25&smtpServerHostName1=brickcom.com.tw&accountName1=brick&password1=12345678&authenticationMode2=1&port2=25&smtpServerHostName2=&accountName2=&password2=

14.6 GetEmailSetting

ActionEvent: getEmailSetting

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action=**getEmailSetting**
Response	authenticationMode1=1 port1=25 smtpServerHostName1= accountName1= password1= authenticationMode2= port2=25 smtpServerHostName2= accountName2= password2=
Comment	
HTTP Method	GET

14.7 SetFTPSetting

ActionEvent: setFTPSetting

Syntax	http://<*IP*>/cgi-bin/event.cgi
HTML Body	action=**setFTPSetting** [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
addressType1	[0,1]	0:IPv4 address 1:hostName
hostName1	<*String*> (64)	
ipAddress1	<*String*> (32)	
port1	[21, 1025-65534]	
accountName1	<*String*> (64)	
password1	<*String*> (64)	
ftpShareDIR1	<*String*> (32)	
passiveMode1	[0,1]	0:enabled 1:disabled
addressType2	[0,1]	0:IPv4 address 1:hostName
hostName2	<*String*> (64)	
ipAddress2	<*String*> (32)	
port2	[21, 1025-65534]	
accountName2	<*String*> (64)	
password2	<*String*> (64)	
ftpShareDIR2	<*String*> (32)	
passiveMode2	[0,1]	0:enabled 1:disabled

Example:
URL:
http://192.168.1.1/cgi-bin/event.cgi
HTML body:
action=setFTPSetting&addressType1=0&hostName1=&ipAddress1=192.168.1.11&port1=21&accountName1=brick&password1=1234567&ftpShareDIR1=other&passiveMode1=1&addressType2=0&hostName2=&ipAddress2=0.0.0.0&port2=21&accountName2=&password2=&ftpShareDIR2=&passiveMode2=1

14.8 GetFTPSetting

ActionEvent: getFTPSetting

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action=**getFTPSetting**
Response	addressType1=0 hostName1= ipAddress1=192.168.1.11 ipv6Address1= port1=21 accountName1=brick password1=1234567 passiveMode1=1 addressType2=0 hostName2= ipAddress2=0.0.0.0 ipv6Address2=

	port2=21 accountName2= password2= passiveMode2=1
Comment	
HTTP Method	GET

14.9 SetAlarmMediaInfo

ActionEvent: setAlarmMediaInfo

Syntax	http://<*IP*>/cgi-bin/event.cgi
HTML Body	action=**setAlarmMediaInfo** [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
timeBeforeEvent	[0-10]	Depends on hardware.
timeAfterEvent	[0-30]	Depends on hardware.
maxBufferSize	[128-1024]	

Example:
URL:
http://192.168.1.1/cgi-bin/event.cgi
HTML body:
action=setAlarmMediaInfo&timeAfterEvent=5&maxBufferSize=1024

14.10 GetAlarmMediaInfo

ActionEvent: getAlarmMediaInfo

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action= **getAlarmMediaInfo**
Response	timeBeforeEvent= timeAfterEvent= maxBufferSize=
Comment	
HTTP Method	GET

14.11 SetSamba

ActionEvent: setSamba

URL Syntax	http://<IP>/cgi-bin/event.cgi
HTML Body	action=**setSamba** [&<*argument*>=<*value*>…]

Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
addressType	[0,1]	0:IPV4 1:hostName
hostDns	<*String*> (32)	
ipAddress	<*String*> (32)	
userName	<*String*> (16)	
password	<*String*> (16)	
workGroup	<*String*> (32)	
shareDIR	<*String*> (32)	

Example:

URL:

http://192.168.1.1/cgi-bin/event.cgi

HTML body:

action=setSamba&addressType=0&hostDns=&ipAddress=192.168.1.111&userName=brickcom&password=12345678&workGroup=&shareDIR=tmp

14.12 GetSamba

ActionEvent: getSamba

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action=**getSamba**
Response	addressType= hostDns= ipAddress= userName= password= shareDIR= workGroup=
Comment	
HTTP Method	GET

14.13 SetHttp

ActionEvent: setHttp

Syntax	http://<IP>/cgi-bin/event.cgi
HTML Body	action=**setHttp** [&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
HttpUrl	<*String*> (128)	
Message	<*String*> (128)	
username	<*String*> (16)	
password	<*String*> (16)	

Example
URL:
http://192.168.1.1/xxxx.cgi
Message: name1=value1&name2=vlaue2
Result: http://192.168.1.1/xxxx.cgi? name1=value1&name2=vlaue2

Example:
http://192.168.1.1/notification.cgi?event=MD&camera=FB-100A

14.14 GetHttp

ActionEvent: getHttp

URL Syntax	http://<*IP*>/cgi-bin/event.cgi?action=**getHttp**
Response	username= password= HttpUrl= Message=
Comment	
HTTP Method	GET

15. I/O Control

I/O Control API allows applications to view/adjust the GPIO setting

15.1 SetGPIOSetting

ActionEvent: setGPIOSetting

URL Syntax	http://<*IP*>/cgi-bin/gpio.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
di1.triggerType	[0,1]	**0:Low** **1:High**
do1.triggerType	[2,3]	**2:Open** **3:Ground**
do1. triggerTime	>0	second

Example

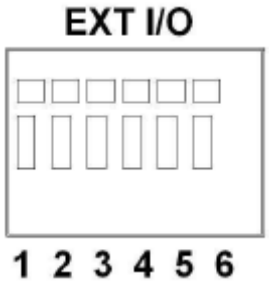
di1.triggerType =1
do1.triggerType= 3
do1.triggerTime=5 second

Example:

URL: http://192.168.1.1/cgi-bin/gpio.cgi
HTTP body:
action=set&di1.triggerType=1&do1.triggerType=3&do1.triggerTime=5

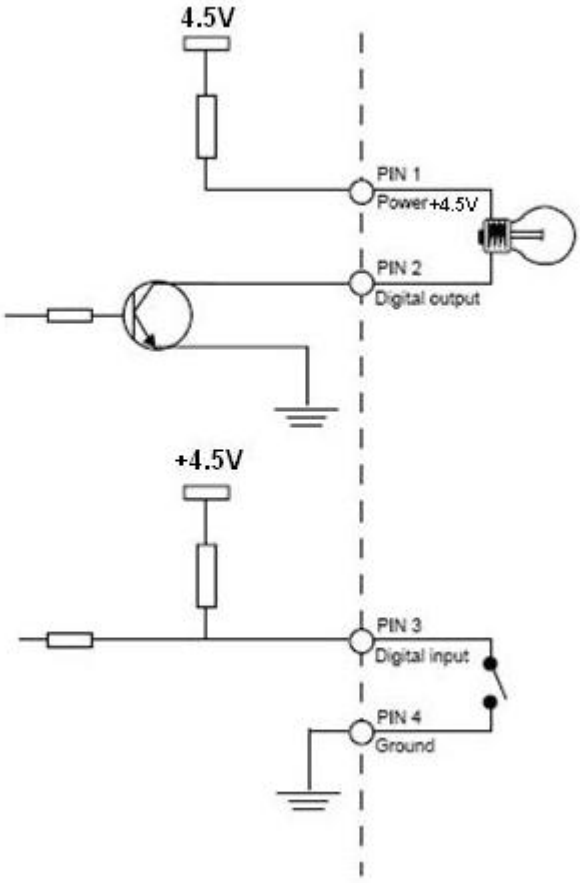
Extension I/O Terminal Block

The Network Camera provides an extension I/O terminal block to connect external input/output devices. The definition of the pins are listed below:



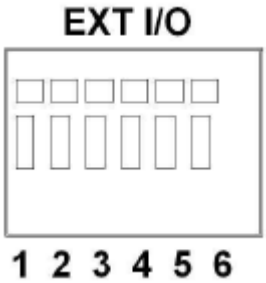
Pin	Function
1	Power +4.5V
2	Digital Output
3	Digital Input
4	Ground
5	RS-485 -
6	RS-485 +

DI/DO Diagram



For example: LED (5V)

Set DI: High DO: GROUND Result: LED light up



Line ①：Led Positive (Power +4.5V)

Line ②：Led Negative (DO)

Line ③：DI

Line ④：Ground

The LED will light up when the following configurations are set:
1. In the GUI DI/DO configuration settings, set Digital Input to High, Digital Output to Ground, and Duration for five seconds.
2. LED positive to pin1, and LED negative to pin2.
When the LED lights up, the Digital Input status will change to High and the Digital Output status will change to Low.

If Line 3 and Line 4 are connected, the status of Digital Input and Digital Output change to Low, the LED will fade off after 5 seconds.

When LED fades off, the Digital Input status will change to Low and Digital Output status will change to High.

15.2 GetGPIOStatus

ActionEvent: getGPIOStatus

URL Syntax	http://<*IP*>/cgi-bin/gpio.cgi?action=**get**
Response	di1.status=1 (current status **0:Low , 1:High**) di1.triggerType=0 do1.status=0 (current status **0:Low , 1:High**) do1.triggerType=2 do1.triggerTime=5
Comment	
HTTP Method	GET

15.3 TriggerDO

ActionEvent: triggerDO

URL Syntax	http://<*IP*>/cgi-bin/gpio.cgi
HTML Body	action=triggerDO [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
status	[high,low]	

Example:
URL: http://192.168.1.1/cgi-bin/gpio.cgi
HTML body: action=triggerDO&status=low
URL: http://192.168.1.1/cgi-bin/gpio.cgi
HTML body: action=triggerDO&status=high

15.4 TurnAllLedOff

ActionEvent: turnAllLedOff

URL Syntax	http://<*IP*>/cgi-bin/led.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
off	[0,1]	0:enabled 1:disabled Depends on hardware.

15.5 Getledstatus

ActionEvent: getGPIOStatus

URL Syntax	http://<*IP*>/cgi-bin/led.cgi?action=**get**
Response	Off=
Comment	
HTTP Method	GET

16. PIR sensor and White LED

PIR sensor and White LED API allows applications to adjust the PIR sensor and white LED controls.

16.1 SetPIRsensor

ActionEvent: set (Depends on hardware)

URL Syntax	http://<*IP*>/cgi-bin/pirsensor.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
sensitivity	[1-10]	
enabled	[0,1]	0:enabled 1:disabled

16.2 GetPIRsenor

ActionEvent: get

URL Syntax	http://<*IP*>/cgi-bin/pirsensor.cgi?action=**get**
Response	sensitivity = enabled=
Comment	
HTTP Method	GET

16.3 Getwledall (Depends on hardware)

ActionEvent: getall

URL Syntax	http://<*IP*>/cgi-bin/wled.cgi?action=**getall**
Response	size= led1.name= led1.method = led1.active= led1.inactive= led1.duringtime=

Comment	
HTTP Method	GET

16.4 UpdateWled

ActionEvent: updateWled (Depends on hardware)

URL Syntax	http://<*IP*>/cgi-bin/wled.cgi
HTML Body	action=**update** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	Setting whilte LED.
HTTP Method	POST

Argument	Valid values	Description
name	PIR	
method	[1,2,3]	1: ON_OFF 2: SLIDER (Reserve) 3: PULSE
duringtime	[1-10]	second
active		1:ON 10:FADE_TO_10 20:FADE_TO_20 30:FADE_TO_30 40:FADE_TO_40 50:FADE_TO_50 60:FADE_TO_60 70:FADE_TO_70 80:FADE_TO_80 90:FADE_TO_90 100:FADE_TO_100
inactive	[0,2]	0:OFF 2:FADE_TO_OFF

Example:
Setting whilte LED configuration for trigger event.
URL:
http://192.168.1.1/cgi-bin/wled.cgi
HTML body:
update&method=3&duringtime=5&active=100&inactive=2&name=PIR

Add configuration to event
URL:
http://192.168.1.1/cgi-bin/event.cgi
HTML body:
action=updateEventSetting&index=3&actions=lightled&enabled=1

16.5 SetLightCTL

ActionEvent: set

URL Syntax	http://<*IP*>/cgi-bin/wledctl.cgi
HTML Body	action=**set** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
method	1	1: ON_OFF
level	[0,10]	0:OFF 2:20% 4:40% 6:60% 8:80% 10:100%

16.6 GetLightCTL

ActionEvent: get

URL Syntax	http://<IP>/cgi-bin/ wledctl.cgi?action=**get**
Response	method= level=
Comment	
HTTP Method	GET

17. Intelligence

Intelligence API allows applications to control the Intelligent Video Analytics (IVA) module of the IP Camera.

17.1 GetViMDSetting

ActionEvent: getViMDSetting

URL Syntax	http://<*IP*>/cgi-bin/motion.cgi?action=**getMD**
Response	Mode= ICExist= Enabled= DrawFlag= FrameWidth= FrameHeight= UpdateFrame= EffWidth= EffHeight= SenseMode= SensePara1= SensePara2= SensePara3= SensePara4= SensePara5= SensePara6=
Comment	
HTTP Method	GET

17.2 SetViMDSetting

ActionEvent: setViMDSetting

URL Syntax	http://<*IP*>/cgi-bin/motion.cgi
HTML Body	action=**setMD** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	**Valid values**	**Description**
Mode	[0,1]	0:Motion detect and Forbidden zone 1:People counting
Enabled	[0,1]	0:disabled 1:enabled
DrawFlag	[0,1]	draw the rectangular boundary for each detected object on image.
FrameWidth		depends on resolution
FrameHeight		depends on resolution
UpdateFrame	>0	default 450 Update the detected stationary objects to background model after n frames
EffWidth	>=30	default 30 effective width of the motion object
EffHeight	>=30	default 30 effective height of the motion object
SenseMode	[1,2,3,4,5,6]	Profile index of SensePara 1: 10 25 10 20 10 20 2: 10 40 10 40 10 40 3: 10 50 10 50 10 50 4: 10 60 10 60 10 60 5: 10 70 10 70 10 70 6: 10 80 10 80 10 80 (user define)
SensePara6	<*String*> (32)	YDark YBright UDark UBright VDark VBright User define default : 10 80 10 80 10 80

Note:

a. Sequence of sensitivity parameter (SensePara): Y_Dark Y_Bright U_Dark U_Bright V_Dark V_Bright
b. Y_Dark, U_Dark, V_Dark: Dark sensitivities of YUV, range between 10 to 30
c. Y_Bright, U_Bright, V_Bright: Bright sensitivities of YUV, range between 30 to 100

Example:

1. URL:
   http://192.168.1.1/cgi-bin/motion.cgi
   HTML body:
   action=setMD&Mode=0&Enabled=1&SenseMode=4
2. URL:
   http://192.168.1.1/cgi-bin/motion.cgi
   HTML body:
   action=setMD&Mode=0&Enabled=1&SenseMode=6&SensePara6=10 79 10 79 10 79

17.3 GetViFZSetting

ActionEvent: getViFZSetting

URL Syntax	http://<IP>/cgi-bin/forbidden.cgi?action=**getFZ**
Response	Mode= ICExist= Enabled= DrawFlag= RegionSize= r1Index= r1Point= r1RegionX= r1RegionY= … r10Index= r10Point= r10RegionX= r10RegionY=
Comment	
HTTP Method	GET

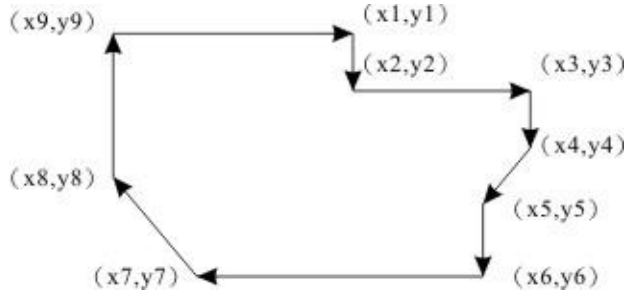
17.4 SetViFZSetting

ActionEvent: setViFZSetting

URL Syntax	http://<*IP*>/cgi-bin/motion.cgi
HTML Body	action=**setFZ** [&<*argument*>=<*value*>&<*argument*>=<*value*>]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
Mode	[0,1]	0:Motion detect and Forbidden zone 1:People counting
Enabled	[0,1]	0:disabled 1:enabled
DrawFlag	[0,1]	draw the rectangular boundary for each detected object on image.
r1Index	[1-10]	Region index
r1Point	[3-10]	n points
r1RegionX	<*String*> (32)	X1 X2 … Xn
r1RegionY	<*String*> (32)	Y1 Y2 … Yn
r***N***Index	[1-10]	Region index N=[1,10]
r***N***Point	[3-10]	n points N=[1,10]
r***N***RegionX	<*String*> (32)	X1 X2 … Xn N=[1,10]
r***N***RegionY	<*String*> (32)	Y1 Y2 … Yn N=[1,10]

Example for FZRegion: 10 points



Example:

1. add two region to detect
   URL:
   http://192.168.1.1/cgi-bin/motion.cgi
   HTML body:
   action=setFZ&Enabled=1&Mode=0&DrawFlag=1&RegionSize=1&r1Index=1&r1Point=4&r1RegionX=536 586 816 698&r1RegionY=334 140 278 404&r2Index=2&r2Point=5&r2RegionX=1070 1038 822 916 1030&r2RegionY=482 590 528 412 324
2. add third region to region
   URL:
   http://192.168.1.1/cgi-bin/motion.cgi
   HTML body:
   action=setFZ&Enabled=1&Mode=0&DrawFlag=1&RegionSize=0&r1Index=1&r1Point=4&r1RegionX=536 586 816 698&r1RegionY=334 140 278 404&r2Index=2&r2Point=5&r2RegionX=1070 1038 822 916 1030&r2RegionY=482 590 528 412 324&r3Index=3&r3Point=5&r3RegionX=364 216 164 270 364&r3RegionY=544 610 456 386 428

17.5 GetViPCSetting

ActionEvent: getViPCSetting

URL Syntax	http://<*IP*>/cgi-bin/peoplecount.cgi?action=**getPC**
Response	Mode= ICExist= Enabled= DrawFlag= UpLine= DownLine= ObjWidth= ObjHeight= HeadMinWH= HeadMaxWH=
Comment	
HTTP Method	GET

17.6 SetViPCSetting

ActionEvent: setViPCSetting

URL Syntax	http://<*IP*>/cgi-bin/peoplecount.cgi
HTML Body	action=**setPC** [&<*argument*>=<*value*>&<*argument*>=<*value*>…]
Response	
Comment	
HTTP Method	POST

Argument	Valid values	Description
Mode	[0,1]	0: Motion detect and Forbidden zone 1:People counting
Enabled	[0,1]	0: disabled 1: enabled
DrawFlag	[0,1]	Draw the rectangular boundary for each detected object in the monitored area.
UpLine		The y-coordinate of the upward scan line.
DownLine		The y-coordinate of the downward scan line.
ObjWidth		Average width of object.
ObjHeight		Average height of object.
HeadMinWH	>30	Minimum effective width of the motion object
HeadMaxWH	>30	Maximum effective width of the motion object

Example:

URL:

http://192.168.1.1/cgi-bin/motion.cgi

HTML body:

action=setPC&Mode=1&Enabled=1&DrawFlag=1&UpLine=177&DownLine=296&HeadMinWH=256&HeadMaxWH=320

18. Modification History

Revision	Date	Originator	Comments
0		Steve	Initial version 1.0
1	2009/9/9	Kenny	add API: 1.getSnapshot 2.getRtsp and setRtsp 3.getVideoCodecs getResolutions getAudioCodecs 4.getinboundChannel 5.Event Notify to HttpServer modify API: 1.getcapability 2.setAudioDevice 3.setGPIOSetting getGPIOStatus
2	2009/9/28	Kenny	1. Remove parameter level from whiteBalance Structure. 2. Add SshutterSpeedSetting and SgainSetting structure and api 3. Modify getCameraSetting and setCameraSetting
3	2009/11/12	Kenny	1. Add PTZ API and parameter 2. Check other API and parameter.
4	2009/11/17	Kenny	1. Modify IO control error. 2. Modify AppendixA InboundChanel url error
5	2009/11/18	Kenny	1. Add IO control information table.
6	2009/11/27	Kenny	1. Modify setCameraSetting getCameraSetting setEffect getEffect API
7	2010/2/2	Gimmy	1. add Intelligence API
8	2010/2/11	kenny	1. 5.13 setVideoRecord 2. 5.14 getVideoRecord 3. 6.19 setIRCutFilter 4. 6.20 getIRCutFilter 5. 6.27 setCameraSetting 6. 6.28 getCameraSetting 7. 6.35 setlightSensor 8. 6.36 getlightSensor 9. 8.13 setWPSBtnEnabled 10. 10.8 setOperationSetting 11. 10.9 getOperationSetting 12. 15.3 triggerDO 13. 15.4 turnAllLedOff

19. AppendixA RTSP

This document specifies the external RTSP-based application programming interface of the camera and video servers. The RTSP URL is rtsp://<*server name*>/channelX where <*server name*> is the IP address of the server. The DESCRIBE, SETUP, OPTIONS, PLAY, PAUSE and TEARDOWN methods are supported. The RTSP protocol is described in RFC 2326.