

# IPCamLive JSON API v2

Document version: v1.11

IPCamLive JSON API lets you integrate IPCamLive services into your web application.

## Accessing the API

IPCamLive JSON API requires the use of an API secret. This identifies your application for IPCamLive. API secret is a key that must be provided to IPCamLive as a parameter when an API method is called. If you do not have API secret, please contact the support team at [support@ipcamlive.com](mailto:support@ipcamlive.com).

## Contents

Accessing the API.....	1
API usage .....	3
API endpoint.....	3
Result in case of success.....	3
Result in case of failure .....	4
API functions.....	5
Get the state of the camera .....	5
Get the stream info of the camera.....	6
Get camera viewer info .....	8
Get HLS stream URL.....	9
Get server state .....	10
Get snapshot image URL .....	11
Get time-lapse video URL.....	12
Get camera list .....	13
Get camera PIN codes .....	14
Set camera PIN codes.....	15
Get camera storage info.....	16
Get camera storage files.....	17

## API usage

You can access the functions by sending an HTTP `GET` request to its URI.

## API endpoint

The address and the format of the API call is:

```
https://ipcamlive.com/api/v2/functionname?apisecret=myapisecret&parameter=data
```

## Result in case of success

If the request **succeeds**, the servers sends the response data in the following format:

```
{
  "result": "ok",
  "data":
  {
    ...
  }
}
```

In this case, the *data* token contains the result of the execution.

## Result in case of failure

If the request **fails**, the servers sends the response data in the following format:

```
{
  "result": "err",
  "errcode": -1,
  "errdata":
  {
    ...
  }
}
```

Note: *errdata* is an optional token

In this case, the *errcode* member contains the error code of the execution.

General error codes:

Error code	Description
-1	Missing or invalid parameters
-2	Invalid data provided (API secret is not valid, account is disabled or e.g. alias is not valid)
-3	This call is not available in Basic package

## API functions

### Get the state of the camera

You can get info about the current state of the camera by invoking *getcamerastate* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
servicetype	Service type of the camera (B – Basic, S – Standard, P – Professional, U – Ultimate)
streamavailable	State of the camera stream (1 – connected, 0 - disconnected)

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getcamerastate?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data":
  {
    "servicetype": "P",
    "streamavailable": "0"
  }
}
```

## Get the stream info of the camera

You can get info about the video stream of the camera by invoking *getstreaminfo* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
video	Video stream info data structure
width	Width of the video stream
height	Height of the video stream
format	Video compression format (e.g. H264, MJPEG, MPEG4, MPGE2, ect.)
fps	Frame rate of the video stream
keyinterval	Number of the frames between key frames. This key exists if non-still image based compression is used.
audio	Audio stream info data structure. This token exists if audio support is enabled and the video stream contains audio channel.
bitrate	Bit rate of the audio stream
samplerate	Sample rate of the video stream
format	Audio compression format (e.g. AAC, PCMA, PCMU, MP3, L16, AC3, AMR, G726, ect.)
bandwidth	Bandwidth data structure
mbps	Bandwidth of the video stream in mbps

Additional error codes:

Error code	Description
-100	Camera is not available

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getstreaminfo?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    "video": {
      "width": "1280",
      "height": "720",
      "format": "h264",
      "fps": "24.87",
      "keyinterval": "41.80"
    },
    "audio": {
      "bitrate": "64000",
      "samplerate": "8000",
      "format": "PCMU"
    },
    "bandwidth": {
      "mbps": "0.81"
    }
  }
}
```

## Get camera viewer info

You can get the current viewer info of the camera by invoking *getcameraviewerinfo* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
hls	Active viewers using HLS streaming
ws	Active viewers using low latency websocket streaming

Additional error codes:

Error code	Description
-100	Camera is not available

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getcameraviewerinfo?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    "viewers": {
      "hls": 0,
      "ws": 0
    }
  }
}
```



## Get HLS stream URL

You can get the URL of the HLS stream by invoking *getstreamhlsurl* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
url	URL to the snapshot image

Additional error codes:

Error code	Description
-100	Camera is not available

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getstreamhlsurl?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    {
      "url": "http://s21.ipcamlive.com/streams/01589983c7c13bfc6/stream.m3u8"
    }
  }
}
```

Please note that the stream URL may change for some reasons (e.g. server failure, load balancer, ect.). This means that the actual URL must be queried before opening the video stream. In addition, during the playback, current URL must be queried from time to time (recommended interval is 1 min) and the new URL must be opened in case it has changed.

Please also note that our video player component (ipcamliveplayer.js) does some extra administration over playing the HLS video stream (e.g. viewer registration, on-demand stream management, viewers distribution between servers, ect.). When playing the HLS stream directly these functions will not work.

## Get server state

You can check the state of the streaming server related to the camera by invoking *getserverstate* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
state	State of the server (Online / Offline)

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getserverstate?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data":
  {
    "state": "online"
  }
}
```

## Get snapshot image URL

You can get the URL of the snapshot image from the live video stream by invoking *getsnapshoturl* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
url	URL to the snapshot image

Additional error codes:

Error code	Description
-100	Camera is not available

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getsnapshoturl?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data":
  {
    "url": "http://s21.ipcamlive.com/streams/01589983c7c13bfc6/snapshot.jpg"
  }
}
```

## Get time-lapse video URL

You can get the URL of the time-lapse video clip by invoking *gettimelapseurl* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera
date	Date of the clip. Date format is ISO 8601 (e.g. 2017-03-01) or relative date (e.g. -2 days). If date parameter is not specified, then path of the long-lapse video will be provided.

Result consists of the following field(s):

Parameter	Description
url	URL to the MP4 video file

Additional error codes:

Error code	Description
-100	Time-lapse function is not enabled on this camera

Example of invoking this method:

```
https://ipcamlive.com/api/v2/gettimelapseurl?apisecret=myapisecret&alias=myalias
&date=2017-01-01
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    {
      "url": "http://s21.ipcamlive.com/streams/01589983c7c13bfc6/snapshot.jpg"
    }
  }
}
```

## Get camera list

You can get the list of the cameras by invoking *getcameras* method.

Result consists of the following field(s):

Parameter	Description
cameras[]	
alias	Alias of the camera
name	Name of the camera
servicetype	Service type of the camera
url	URL of the video stream (e.g. rtsp://ipaddress/main)
enabled	0 – disabled, 1 – enabled

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getcameras?apisecret=myapisecret
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    "cameras": [
      {
        "alias": "55c9f405218d1",
        "name": "55c9f405218d1",
        "servicetype": "P",
        "url": "rtsp://1.2.3.4:554/live4.sdp",
        "enabled": "1"
      },
      {
        "alias": "55f1d712e8195",
        "name": "<b>55f1d712e8195</b>",
        "servicetype": "P",
        "url": "http://1.2.3.4:554/mjpeg",
        "enabled": "1"
      }
    ]
  }
}
```

## Get camera PIN codes

You can get the PIN code settings of the camera by invoking *getcamerapins* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
pinlockenabled	PIN lock feature enabled on the camera (0 – disabled, 1 – enabled)
pincodes	PIN codes valid on the camera

Additional error codes:

Error code	Description
-100	This feature is not available on the camera (not sufficient service type)

Example of invoking this method:

```
https://ipcamlive.com/api/v2/getcamerapins?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    {
      "pinlockenabled": "1",
      "pincodes": "6108:Pin-1;8471:Pin-2"
    }
  }
}
```

Explanation:

PIN1 code: 6108

PIN1 comment: Pin-1

PIN2 code: 8471

PIN2 comment: Pin-2

## Set camera PIN codes

You can set the PIN code settings of the camera by invoking *setcamerapins* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera
pinlockenabled	Enable or disable the PIN lock feature on the camera (0 – disabled, 1 – enabled)
pincodes	New PIN codes

Additional error codes:

Error code	Description
-100	This feature is not available on the camera (not sufficient service type)

Example of invoking this method:

```
https://ipcamlive.com/api/v2/setcamerapins?apisecret=myapisecret&alias=myalias&pinlockenabled=1&pincodes=6108:Pin-1;8472:Pin-2
```

Example of the result data:

```
{  
  "result": "ok"  
}
```

## Get camera storage info

You can get storage info of the camera by invoking *getcamerastorageinfo* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
storageinfo	
totalsize	Total size of the storage files in MB
fromdate	Timestamp of the first video recorded
todate	Timestamp of the last video recorded
filecount	Count of the storage files

Additional error codes:

Error code	Description
-100	Storage feature is enabled on this camera

Example of invoking this method:

```
https://ipcamlive.com/api/v2/setcamerastorageinfo?apisecret=myapisecret&alias=myalias
```

Example of the result data:

```
{
  "result": "ok",
  "data": {
    "storageinfo": {
      "totalsize": "6914",
      "fromdate": "2022-09-08 10:00:09",
      "todate": "2022-09-09 09:15:09",
      "filecount": "93"
    }
  }
}
```



## Get camera storage files

You can get the list of the MP4 storage files associated to the camera by invoking *getcamerastoragefiles* method.

This method requires the following parameter(s):

Parameter	Description
alias	Alias of the camera

Result consists of the following field(s):

Parameter	Description
storageinfo	
[]	
fileid	Unique ID of the file
url	Direct HTTP access of the MP4 files
fromdate	Timestamp of the first video recorded
todate	Timestamp of the last video recorded
filesize	Size of the storage file in MB

Additional error codes:

Error code	Description
-100	Storage feature is enabled on this camera

Example of invoking this method:

```
https://ipcamlive.com/api/v2/setcamerastoragefiles?apisecret=myapisecret&alias=myalias
```

## Example of the result data:

```
{
  "result": "ok",
  "data": {
    "storageinfo": [
      {
        "fileid": "141617",
        "url": "https://s86.ipcamlive.com/storages/cameraalias/S20220908100009_E20220908101504.mp4",
        "fromdate": "2022-09-08 10:00:09",
        "todate": "2022-09-08 10:15:04",
        "filesize": "33"
      },
      {
        "fileid": "141647",
        "url": "https://s86.ipcamlive.com/storages/cameraalias/S20220908101504_E20220908103000.mp4",
        "fromdate": "2022-09-08 10:15:04",
        "todate": "2022-09-08 10:30:00",
        "filesize": "77"
      },
      {
        "fileid": "141667",
        "url": "https://s86.ipcamlive.com/storages/cameraalias/S20220908103000_E20220908104502.mp4",
        "fromdate": "2022-09-08 10:30:00",
        "todate": "2022-09-08 10:45:02",
        "filesize": "64"
      }
    ]
  }
}
```