

September 2014

API references for Camera Remote API beta

Camera Remote API API Reference

*All implied warranties, including without limitation the implied warranties of merchantability or fitness for a particular purpose, are excluded. In no event shall Sony Corporation or its licensors be liable for incidental or consequential damages of any nature, including but not limited to lost profits or commercial loss, arising out of the use of the information in this document.

SONY

© Copyright 2013 Sony Corporation. All rights reserved. Brands, company or product names mentioned herein are trademarks of their respective owners. You are hereby granted a limited license to download and/or print a copy of this document for personal use. Any rights not expressly granted herein are reserved.

First edition (December 2013)

This document is published by Sony Corporation without any warranty*. Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by Sony Corporation at any time and without notice. Such changes will, however, be incorporated into new editions of this document. Printed versions are to be regarded as temporary reference copies only.

Preface

Developer World

For the latest Sony technical news, tutorials and development tools go to developer.sony.com

About this document

The "Camera Remote API beta" will be referred to as the "Camera Remote API" in this document.
The purpose of this document is to list the API specifications for the Camera Remote API provided by Sony.

Document conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

| | |
|----------|--|
| HTTP | HyperText Transfer Protocol (RFC 2616) |
| JSON | JavaScript Object Notation (RFC 4627) |
| JSON-RPC | Remote Procedure Call encoded in JSON |
| SSDP | Simple Server Discovery Protocol |

Document history

| Change history | | |
|----------------|--------------|---|
| 2013-09-01 | Version 1.0 | First version |
| 2013-09-25 | Version 1.10 | Added 6 compatible cameras to section 2.3. Modified "API name parameters" in section 3.11 to support more compatible cameras. Corrected some typographical errors. |
| 2013-11-05 | Version 1.20 | Added 3 compatible cameras to section 2.3. Added 2 audio recording APIs. Added "audio" shoot mode parameter. |
| 2013-12-01 | Version 1.21 | Updated to new template |
| 2014-01-06 | Version 1.30 | Added 1 compatible camera. Added zoom support to the PlayMemories Camera Apps compatible cameras. |
| 2014-02-18 | Version 1.40 | Added 4 compatible cameras. |
| 2014-04-15 | Version 1.50 | Added 2 compatible cameras (the firmware update). Added the following API groups and related parameters. <ul style="list-style-type: none"> - Half-press shutter - Touch AF position - Exposure mode - Focus mode - Exposure compensation - F number - Shutter speed - ISO speed rate - White balance - Still size - Beep mode - Date/time setting ("system" API service) - Event notification (added "getEvent" ver.1.1) Modified the following API groups. <ul style="list-style-type: none"> - Event notification (added parameters of "getEvent" ver.1.0) - Server information (added "system" API service support) |
| 2014-06-18 | Version 1.60 | Added 3 new compatible cameras. <ul style="list-style-type: none"> - A7S - DSC-RX100M3 - HDR-AS20 Added more API support to the following compatible cameras. <ul style="list-style-type: none"> - HDR-AS30 (firmware update) - HDR-AS100 (firmware update) - PlayMemories Camera Apps compatible cameras Added the following API groups and related parameters. <ul style="list-style-type: none"> - Interval still recording - Liveview size - Program shift - Flash mode - Movie quality - Steady mode - View angle Added some parameters to the following parameter group. <ul style="list-style-type: none"> - Shoot mode - Touch AF position |

| | | |
|------------|--------------|--|
| | | <ul style="list-style-type: none"> - Exposure mode - White balance - Event |
| 2014-09-01 | Version 1.70 | <p>Added 4 new compatible cameras.</p> <ul style="list-style-type: none"> - QX1 - A5100 - DSC-QX30 - HDR-AZ1 <p>Added the following API groups and related parameters.</p> <ul style="list-style-type: none"> - Still capture (for continuous shooting) - Liveview frame - Zoom setting - Tracking focus - Continuous shooting mode - Continuous shooting speed - Still quality - Movie file format - Scene selection - Color setting - Interval time - Flip setting - TV color system - Camera function - Transferring images - Remote playback - Delete contents - IR remote control - Auto power off - Storage information - Event notification (added "getEvent" ver.1.2) <p>Modified the following API groups.</p> <ul style="list-style-type: none"> - Event notification (added parameters of "getEvent" ver.1.0) - Server information (added "avContent" API service support) <p>Added some parameters to the following parameter group.</p> <ul style="list-style-type: none"> - Event <p>Added liveview frame to liveview data format.</p> <p>Added streaming data format for remote playback.</p> |
| | | |

Contents

| | |
|---|-----------|
| Introduction | 8 |
| API versioning, supported and available APIs | 9 |
| Camera Remote API versioning | 9 |
| Supported APIs and available APIs | 9 |
| Supported API groups for each compatible cameras | 10 |
| API list | 12 |
| Shooting and transferring images | 18 |
| Checking availability of transferring images function | 18 |
| Related APIs and sample sequences | 18 |
| API Reference | 20 |
| Shoot mode | 21 |
| Still capture | 25 |
| Movie recording | 31 |
| Audio recording | 34 |
| Interval still recording | 37 |
| Liveview | 40 |
| Liveview size | 42 |
| Liveview frame | 46 |
| Zoom | 48 |
| Zoom setting | 50 |
| Half-press shutter | 54 |
| Touch AF position | 56 |
| Tracking focus | 60 |
| Continuous shooting mode | 66 |
| Continuous shooting speed | 70 |
| Self-timer | 74 |
| Exposure mode | 78 |
| Focus mode | 82 |
| Exposure compensation | 86 |
| F number | 92 |
| Shutter speed | 97 |
| ISO speed rate | 102 |
| White balance | 107 |
| Program shift | 114 |
| Flash mode | 116 |
| Still size | 120 |
| Still quality | 125 |
| Postview image size | 129 |
| Movie file format | 133 |
| Movie quality | 137 |
| Steady mode | 141 |
| View angle | 145 |
| Scene selection | 149 |
| Color setting | 153 |
| Interval time | 157 |
| Flip setting | 161 |
| TV color system | 165 |
| Camera setup | 169 |
| Camera function | 171 |
| Transferring images | 176 |
| Remote playback | 187 |

| | |
|---|------------|
| Delete contents..... | 195 |
| IR remote control | 196 |
| Auto power off | 200 |
| Beep mode..... | 204 |
| Date/time setting | 208 |
| Storage information..... | 210 |
| Event notification..... | 212 |
| Server information..... | 230 |
| Parameter description..... | 235 |
| Liveview data format | 245 |
| Streaming data format..... | 248 |
| Status code & Error | 251 |
| JSON data types | 253 |
| Sample Sequence..... | 254 |
| Displaying liveview and capturing picture..... | 254 |
| Recording movie | 255 |
| Getting event | 256 |
| Checking API version, supported APIs and available APIs | 257 |
| Changing camera function to transferring images | 258 |
| Transferring images (Date view)..... | 259 |
| Transferring images (Flat view) | 260 |
| Remote playback | 261 |
| More information..... | 262 |
| Trademarks and acknowledgements..... | 262 |

Introduction

The purpose of this document is to describe the API specifications for the Camera Remote API. If you want to get information about how to access camera functions and the procedure to establish connection to use the APIs, please see the Camera Remote API Development Guide available in the Camera Remote API SDK.

API versioning, supported and available APIs

Camera Remote API versioning

Camera Remote API includes versioning on two levels. The Camera Remote API itself has one version, and then each API has their own version. The client app may check those versions and change its behavior accordingly.

Camera Remote API version

Camera Remote API has its version defined by its specifying functions. The version will be changed if an API was added or deleted. The version also will be changed if a supporting function in any APIs was changed. The Camera Remote API version can be obtained by the "[getApplicationInfo](#)" API. For details, please see the "[getApplicationInfo](#)" API specification.

This document only cover Camera Remote API version 2.0.0 or greater. The client app should check whether the API version is "2.0.0" or greater to confirm the server function compatibility with this document.

Version for each API

Each API available in the Camera Remote API has its version also. This version is represented as two numbers separated by "." (dot), X.Y. Basically, every API will be defined from version "1.0". This number will be incremented when the definition of the API is changed.

The client app can get what versions a camera supports by using the "[getVersions](#)" API. The versions received from the "[getVersions](#)" API depend on what APIs are supported on an API service of that camera and depend on the camera's API capability. **The client app MUST set the "version" parameter in the request to specify the version of the API.** If you want to know what versions each API supports, please see each API specification.

Supported APIs and available APIs

The client app should check:

- Supported API list by using "[getMethodTypes](#)" API for all API services except "camera" API service.
- Available API list instead of supported API list when it uses APIs that belong to "camera" API service.

The camera features can vary between models. Therefore the APIs that each camera supports may vary by models. The camera provides its supported API list via the "[getMethodTypes](#)" API.

In addition, the camera status can be changed by user operations and calling APIs. Available APIs of "camera" API service in the camera will be changed by camera status. The camera also provides a list of available APIs via the "[getAvailableApiList](#)" API or the "availableApiList" object of the "[getEvent](#)" API callback for "camera" API service. For more information, please see API specification.

Supported API groups for each compatible cameras

| | HDR-AS15 *2 | HDR-AS20 HDR-AS30 *2 HDR-AS100 *2 | HDR-AZ1 | HDR-MV1 | A7 *3 A7R *3 A7S *3 A5000 *3 A5100 *3 A6000 *3 NEX-5R *3 NEX-5T *3 NEX-6 *3 DSC-HX60 *3 DSC-HX400 *3 DSC-RX100M3 *3 | QX1 DSC-QX30 | DSC-QX10 *2 DSC-QX100 *2 |
|---------------------------|-------------|---|---------|---------|--|-----------------|-----------------------------|
| Shoot mode | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Still capture | No | Yes *4 | Yes | No | Yes | Yes | Yes |
| Movie recording | Yes | Yes | Yes | Yes | Yes *5 | Yes | Yes |
| Audio recording | No | No | No | Yes | No | No | No |
| Interval still recording | No | Yes | Yes | No | No | No | No |
| Liveview | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Liveview size | No | No | No | No | Yes | No | No |
| Liveview frame | No | No | No | No | No | Yes | No |
| Zoom | No | No | No | No | Yes | Yes | Yes |
| Zoom setting | No | No | No | No | No | Yes | No |
| Half-press shutter | No | No | No | No | No | Yes | Yes |
| Touch AF position | No | No | No | No | Yes | Yes | Yes |
| Tracking focus | No | No | No | No | No | Yes | No |
| Continuous shooting mode | No | No | Yes *11 | No | No | Yes *11 | No |
| Continuous shooting speed | No | No | Yes | No | No | Yes *13 | No |
| Self-timer | No | No | Yes | No | Yes | Yes | Yes |
| Exposure mode | No | No | No | No | Yes *6 | Yes | Yes |
| Focus mode | No | No | No | No | No | Yes *12 | Yes *7 |
| Exposure compensation | No | No | No | No | Yes | Yes | Yes |
| F number | No | No | No | No | Yes | Yes | Yes *7 |
| Shutter speed | No | No | No | No | Yes | Yes | Yes *7 |
| ISO speed rate | No | No | No | No | Yes | Yes | Yes |
| White balance | No | No | No | No | Yes | Yes | Yes |
| Program shift | No | No | No | No | Yes | Yes | No |
| Flash mode | No | No | No | No | Yes | Yes *12 | No |
| Still size | No | No | No | No | No | Yes | Yes |
| Still quality | No | No | No | No | No | Yes | No |
| Postview image size | No | No | Yes | No | Yes | Yes | Yes |
| Movie file format | No | No | Yes | No | No | No | No |
| Movie quality | No | Yes | Yes | No | No | Yes *13 | No |

| | | | | | | | |
|---------------------|-----|--------|---------|-----|-----|---------|--------|
| Steady mode | No | Yes | Yes | No | No | No | No |
| View angle | No | Yes *8 | No | No | No | No | No |
| Scene selection | No | No | Yes | No | No | No | No |
| Color setting | No | No | Yes | No | No | No | No |
| Interval time | No | No | Yes | No | No | No | No |
| Flip setting | No | No | Yes | No | No | No | No |
| TV color system | No | No | Yes | No | No | No | No |
| Camera setup *1 | No | No | No | No | Yes | No | No |
| Camera function | No | No | Yes | No | No | Yes | No |
| Transferring images | No | No | Yes | No | No | Yes | No |
| Remote playback | No | No | Yes | No | No | Yes *14 | No |
| Delete contents | No | No | Yes | No | No | Yes | No |
| IR remote control | No | No | Yes | No | No | No | No |
| Auto power off | No | No | Yes | No | No | No | No |
| Beep mode | No | No | Yes | No | No | Yes | Yes |
| Date/time setting | No | No | Yes | No | No | Yes | Yes |
| Storage information | No | No | Yes | No | No | Yes | No |
| Event notification | Yes | Yes | Yes *10 | Yes | Yes | Yes *10 | Yes *9 |
| Server information | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

API groups - Compatible cameras

- *1: Some camera models need "Camera setup" API call before accessing shooting functions.
- *2: The latest firmware update is needed.
- *3: These cameras are compatible with the PlayMemories Camera Apps "[Smart Remote Control](#)" application.
The latest version of the application should be installed and started to use the APIs.
- *4: These cameras support only "[actTakePicture](#)".
- *5: Only A7S, A5100 and DSC-RX100M3 support the APIs group.
- *6: The setting in still mode is not available to cameras which have a hardware mode dial.
- *7: Only DSC-QX100 supports the APIs group.
- *8: Only HDR-AS30 supports the APIs group.
- *9: These cameras support "[getEvent \(v1.1\)](#)" in addition to "[getEvent \(v1.0\)](#)".
- *10: These cameras support "[getEvent \(v1.2\)](#)" in addition to "[getEvent \(v1.0\)](#)" and "[getEvent \(v1.1\)](#)".
- *11: QX1 and DSC-QX30 support continuous shooting mode and HDR-AZ1 supports "Burst"/"MotionShot" shooting mode. Please see API specification to handle these types of continuous shooting mode.
- *12: Only QX1 supports the APIs group.
- *13: Only DSC-QX30 supports the APIs group.
- *14: These cameras don't support "[seekStreamingPosition](#)" API.

API list

The APIs are listed below. There are three categories which are shooting function, transferring images function and general function.

Shooting function

| API groups | APIs | API service |
|--|---|-------------|
| Shoot mode | setShootMode | camera |
| | getShootMode | camera |
| | getSupportedShootMode | camera |
| | getAvailableShootMode | camera |
| Still capture | actTakePicture | camera |
| | awaitTakePicture | camera |
| | startContShooting | camera |
| | stopContShooting | camera |
| Movie recording | startMovieRec | camera |
| | stopMovieRec | camera |
| Audio recording | startAudioRec | camera |
| | stopAudioRec | camera |
| Interval still recording | startIntervalStillRec | camera |
| | stopIntervalStillRec | camera |
| Liveview | startLiveview | camera |
| | stopLiveview | camera |
| Liveview size | startLiveviewWithSize | camera |
| | getLiveviewSize | camera |
| | getSupportedLiveviewSize | camera |
| | getAvailableLiveviewSize | camera |
| Liveview frame | setLiveviewFrameInfo | camera |
| | getLiveviewFrameInfo | camera |
| Zoom | actZoom | camera |
| Zoom setting | setZoomSetting | camera |
| | getZoomSetting | camera |
| | getSupportedZoomSetting | camera |
| | getAvailableZoomSetting | camera |
| Half-press shutter | actHalfPressShutter | camera |
| | cancelHalfPressShutter | camera |
| Touch AF position | setTouchAFPosition | camera |
| | getTouchAFPosition | camera |
| | cancelTouchAFPosition | camera |
| Tracking focus | actTrackingFocus | camera |
| | cancelTrackingFocus | camera |
| | setTrackingFocus | camera |
| | getTrackingFocus | camera |
| | getSupportedTrackingFocus | camera |
| | getAvailableTrackingFocus | camera |

| | | |
|---|--|--------|
| Continuous shooting mode | setContShootingMode | camera |
| | getContShootingMode | camera |
| | getSupportedContShootingMode | camera |
| | getAvailableContShootingMode | camera |
| Continuous shooting speed | setContShootingSpeed | camera |
| | getContShootingSpeed | camera |
| | getSupportedContShootingSpeed | camera |
| | getAvailableContShootingSpeed | camera |
| Self-timer | setSelfTimer | camera |
| | getSelfTimer | camera |
| | getSupportedSelfTimer | camera |
| | getAvailableSelfTimer | camera |
| Exposure mode | setExposureMode | camera |
| | getExposureMode | camera |
| | getSupportedExposureMode | camera |
| | getAvailableExposureMode | camera |
| Focus mode | setFocusMode | camera |
| | getFocusMode | camera |
| | getSupportedFocusMode | camera |
| | getAvailableFocusMode | camera |
| Exposure compensation | setExposureCompensation | camera |
| | getExposureCompensation | camera |
| | getSupportedExposureCompensation | camera |
| | getAvailableExposureCompensation | camera |
| F number | setFNumber | camera |
| | getFNumber | camera |
| | getSupportedFNumber | camera |
| | getAvailableFNumber | camera |
| Shutter speed | setShutterSpeed | camera |
| | getShutterSpeed | camera |
| | getSupportedShutterSpeed | camera |
| | getAvailableShutterSpeed | camera |
| ISO speed rate | setIsoSpeedRate | camera |
| | getIsoSpeedRate | camera |
| | getSupportedIsoSpeedRate | camera |
| | getAvailableIsoSpeedRate | camera |
| White balance | setWhiteBalance | camera |
| | getWhiteBalance | camera |
| | getSupportedWhiteBalance | camera |
| | getAvailableWhiteBalance | camera |
| Program shift | setProgramShift | camera |
| | getSupportedProgramShift | camera |
| Flash mode | setFlashMode | camera |
| | getFlashMode | camera |
| | getSupportedFlashMode | camera |

| | | |
|-------------------------------------|---|--------|
| | getAvailableFlashMode | camera |
| Still size | setStillSize | camera |
| | getStillSize | camera |
| | getSupportedStillSize | camera |
| | getAvailableStillSize | camera |
| Still quality | setStillQuality | camera |
| | getStillQuality | camera |
| | getSupportedStillQuality | camera |
| | getAvailableStillQuality | camera |
| Postview image size | setPostviewImageSize | camera |
| | getPostviewImageSize | camera |
| | getSupportedPostviewImageSize | camera |
| | getAvailablePostviewImageSize | camera |
| Movie file format | setMovieFileFormat | camera |
| | getMovieFileFormat | camera |
| | getSupportedMovieFileFormat | camera |
| | getAvailableMovieFileFormat | camera |
| Movie quality | setMovieQuality | camera |
| | getMovieQuality | camera |
| | getSupportedMovieQuality | camera |
| | getAvailableMovieQuality | camera |
| Steady mode | setSteadyMode | camera |
| | getSteadyMode | camera |
| | getSupportedSteadyMode | camera |
| | getAvailableSteadyMode | camera |
| View angle | setViewAngle | camera |
| | getViewAngle | camera |
| | getSupportedViewAngle | camera |
| | getAvailableViewAngle | camera |
| Scene selection | setSceneSelection | camera |
| | getSceneSelection | camera |
| | getSupportedSceneSelection | camera |
| | getAvailableSceneSelection | camera |
| Color setting | setColorSetting | camera |
| | getColorSetting | camera |
| | getSupportedColorSetting | camera |
| | getAvailableColorSetting | camera |
| Interval time | setIntervalTime | camera |
| | getIntervalTime | camera |
| | getSupportedIntervalTime | camera |
| | getAvailableIntervalTime | camera |
| Flip setting | setFlipSetting | camera |
| | getFlipSetting | camera |
| | getSupportedFlipSetting | camera |
| | getAvailableFlipSetting | camera |

| | | |
|---------------------------------|---|--------|
| TV color system | setTvColorSystem | camera |
| | getTvColorSystem | camera |
| | getSupportedTvColorSystem | camera |
| | getAvailableTvColorSystem | camera |
| Camera setup | startRecMode | camera |
| | stopRecMode | camera |

Transferring images function

| API groups | APIs | API service |
|-------------------------------------|--|-------------|
| Camera function | setCameraFunction | camera |
| | getCameraFunction | camera |
| | getSupportedCameraFunction | camera |
| | getAvailableCameraFunction | camera |
| Transferring images | getSchemeList | avContent |
| | getSourceList | avContent |
| | getContentCount (v1.2) | avContent |
| | getContentList (v1.3) | avContent |
| Remote playback | setStreamingContent | avContent |
| | startStreaming | avContent |
| | pauseStreaming | avContent |
| | seekStreamingPosition | avContent |
| | stopStreaming | avContent |
| | requestToNotifyStreamingStatus | avContent |
| Delete contents | deleteContent (v1.1) | avContent |

General function

| API groups | APIs | API service |
|-------------------------------------|---|---------------------------|
| IR remote control | setInfraredRemoteControl | camera |
| | getInfraredRemoteControl | camera |
| | getSupportedInfraredRemoteControl | camera |
| | getAvailableInfraredRemoteControl | camera |
| Auto power off | setAutoPowerOff | camera |
| | getAutoPowerOff | camera |
| | getSupportedAutoPowerOff | camera |
| | getAvailableAutoPowerOff | camera |
| Beep mode | setBeepMode | camera |
| | getBeepMode | camera |
| | getSupportedBeepMode | camera |
| | getAvailableBeepMode | camera |
| Date/time setting | setCurrentTime | system |
| Storage Information | getStorageInformation | camera |
| Event notification | getEvent (v1.0) | camera |
| | getEvent (v1.1) | camera |
| | getEvent (v1.2) | camera |
| Server information | getAvailableApiList | camera |
| | getApplicationInfo | camera |
| | getVersions | camera, system, avContent |
| | getMethodTypes | camera, system, avContent |

Shooting and transferring images

Camera Remote API provides two main functions. One is shooting function, and the other is transferring images function.

Shooting function

In shooting function, the client can take pictures and record movie via APIs. Changing camera settings such as F number, shutter speed and ISO speed rate is available for some cameras. APIs of shooting function belong to "camera" API service basically. To use shooting function APIs, the client should change camera function to "Remote Shooting" via "[setCameraFunction](#)". "Remote Shooting" is default function after the camera connects with the client via Wi-Fi.

Transferring images function

In transferring images function, the client can retrieve list of stored still images and movies from camera. Also, the client can download the contents and play movie remotely via APIs. APIs of transferring images belong to "avContent" API service. To use transferring images function APIs, the client should change camera function to "Contents Transfer" via "[setCameraFunction](#)". The transferring images function is supported by specific camera models.

Checking availability of transferring images function

Device Description

The functionalities are defined in the Device Description which can be found via the Device Discovery (SSDP). When the client calls the APIs to use shooting function, the client checks the presence or absence of "**camera**" **API service** in Device Description. When the client uses transferring images APIs, the client should check the presence or absence of "**avContent**" **API service** in Device Description in the same way as "camera" API service. Please refer to "Development Guide" for details about the Device Description and Device Discovery (SSDP).

Availability of changing camera function

The client can change camera function between "Remote Shooting" and "Contents Transfer" via "[setCameraFunction](#)" API. This API belongs to "camera" API service and the client can check the availability of calling this API at the moment via the "[getAvailableApiList](#)" API or the "availableApiList" object of the "[getEvent](#)" API callback for "camera" API service.

Supported features of transferring images

The transferring images function allows the client to download the contents from the camera, play movie remotely and deleting contents. To use the features, the client should call the APIs in "avContent" API service. The camera features can vary between models. Therefore the APIs that each camera supports may vary by models. Regarding APIs in "avContent" API service, the client should check if the camera supports APIs via "[getVersions](#)" and "[getMethodTypes](#)" APIs of "avContent". Note that APIs in "avContent" have various API versions (not only ver.1.0 but also ver.1.1, ver.1.2, ver.1.3).

Related APIs and sample sequences

The following API groups are related to transferring images function.

- [Camera function APIs](#)
- [Transferring images APIs](#)
- [Remote playback APIs](#)
- [Delete contents API](#)

And, please refer to following sample sequences to build client application.

- [Changing camera function](#)
- [Transferring images \(Date view\)](#)
- [Transferring images \(Flat view\)](#)
- [Remote playback](#)

For details please see the sample code included in the Camera Remote API SDK.

API Reference

This chapter provides the detailed API specification of Camera Remote API using the below format.

Sample

setSelfTimer

API name

Overview

This API provides a function to set a value of self-timer.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Endpoint URL

Endpoint URL is composed of two parts, ActionList_URL and API service. For details, see Development Guide document.

Version

It must be set as a value of "version" in request JSON data.

Request

Elements of "params"

| Order | type | explanation |
|-------|---------|--|
| 0 | integer | Self-timer (unit: sec) (See Self-timer parameter) |

Request parameters and JSON Example

Camera Remote API uses JSON-RPC over HTTP. **HTTP POST** is used for uni-direction request from client to server. For details, see Development Guide document.

JSON Example

```
{
  "method": "setSelfTimer",
  "params": [2],
  "id": 1,
  "version": "1.0"
}
```

Response result, JSON Example and Error Codes

For details on JSON format, see Development Guide document. Refer to "Status code & Error" section about Error Codes.

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

JSON in response

If request succeeds, "error" is skipped in the response. On the other hand, if request fails, "result" is skipped. Result of API can be replied by "result" in the response. Some of API replies "results" in the response.

Error Codes

See Status code & Error

Related API

- getSelfTimer
- getSupportedSelfTimer
- getAvailableSelfTimer

Related API

This part shows a list of APIs related to this API.

Special note (details)

None in particular.

Special note (details)

This part shows how to use this API and special instruction.

Shoot mode

setShootMode

Overview

This API provides a function to set a value of shooting mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|--|
| 0 | string | Shoot mode (See Shoot mode parameters of Parameter description) |

JSON Example

```
{
  "method": "setShootMode",
  "params": ["movie"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getShootMode](#)
- [getSupportedShootMode](#)
- [getAvailableShootMode](#)

Special note (details)

The camera has the concept of shoot mode. Some of APIs are only available on specific shoot mode. For details, see "[Supported APIs and available APIs](#)".

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "shootMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

Some camera models need "[startRecMode](#)" API call before accessing camera settings. See "[startRecMode](#)" for details.

getShootMode

Overview

This API provides a function to get current camera shooting mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getShootMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|--|
| 0 | string | Current shoot mode (See Shoot mode parameters of Parameter description) |

JSON Example

```
{
  "result": ["still"],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShootMode](#)
- [getSupportedShootMode](#)
- [getAvailableShootMode](#)

Special note (details)

None in particular.

getSupportedShootMode

Overview

This API provides a function to get the supported shoot modes.

The client should use "[getAvailableShootMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedShootMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string-array | A list of supported shoot modes (See Shoot mode parameters of Parameter description) |

JSON Example

```
{
  "result": [
    ["still", "movie"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShootMode](#)
- [getShootMode](#)
- [getAvailableShootMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableShootMode

Overview

This API provides a function to get current shoot mode and the available shoot modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableShootMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| Order | type | explanation |
|-------|--------------|---|
| 0 | string | Current shoot mode (See Shoot mode parameters of Parameter description) |
| 1 | string-array | A list of available shoot modes (See Shoot mode parameters of Parameter description) |

JSON Example

```
{
  "result": [
    "still",
    ["still", "movie"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShootMode](#)
- [getShootMode](#)
- [getSupportedShootMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "shootMode" object in "[getEvent](#)" callback.

Still capture

actTakePicture

Overview

This API provides a function to take picture.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "actTakePicture",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | Array of URLs of postview. The postview is captured image data by camera. The postview image can be used for storing it as the taken picture, and showing it to the client display. |

JSON Example

```
{
  "result": [
    ["http://ip:port/postview/postview.jpg"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [awaitTakePicture](#)
- [getEvent](#)

Special note (details)

This API instructs the server side to shoot still image. When this API is called and the server starts shooting still image, the camera status will change as follows. The camera status can be obtained by ["getEvent"](#).

Camera status: "IDLE" -> "StillCapturing" -> "StillSaving" -> "IDLE"

Note that this sequence is the example of typical case.

The client should check the ["getEvent"](#) parameter ("cameraStatus") and check if it is "IDLE" before calling this API. The camera needs to prepare for the next shot, therefore it may take time to start next capturing after changing to "IDLE". The camera will start capturing as soon as possible.

In case of long exposure, the server will return "40403" error ("Still Capturing Not Finished") within several tens of seconds. If status code "40403" is received, capturing is not completed. Use the ["awaitTakePicture"](#) API to receive status on capture. If status code "40403" is received for ["awaitTakePicture"](#) again, the client can call ["awaitTakePicture"](#) until the capture is done.

This API is only available when the shoot mode is "still".

Some camera models need ["startRecMode"](#) API call before capturing still image. See ["startRecMode"](#) for details.

awaitTakePicture

Overview

This API provides a function to wait while the camera is taking the picture.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "awaitTakePicture",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | Array of URLs of postview. The postview is captured image data by camera. The postview image can be used for storing it as the taken picture, and showing it to the client display. |

JSON Example

```
{
  "result": [
    ["http://ip:port/postview/postview.jpg"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [actTakePicture](#)
- [getEvent](#)

Special note (details)

Please see "[actTakePicture](#)".

startContShooting

Overview

This API provides a function to start continuous shooting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "startContShooting",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

None.

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopContShooting](#)
- [getEvent \(v1.2\)](#)

Special note (details)

The continuous shooting mode and speed should be set properly via "[setContShootingMode](#)" and "[setContShootingSpeed](#)" to use this API. This API is only available when the shoot mode is "still".

This API instructs the server side to start continuous shooting. When this API is called and the server starts continuous shooting, the camera status will change as follows. The camera status will be notified by "[getEvent \(v1.2\)](#)".

[Client calls "[startContShooting](#)"]

Camera status: "IDLE" -> "StillCapturing".

After the shooting has started, the client may stop the shooting. To stop the shooting, "[stopContShooting](#)" must be called, and the camera status will change as follows.

[Client calls "[stopContShooting](#)"]

Camera status: "StillCapturing" -> "StillSaving" -> "IDLE".

Note that this sequence is the example of typical case.

The client should check the "[getEvent \(v1.2\)](#)" parameter ("cameraStatus") and check if it is "IDLE" before calling "[startContShooting](#)". The camera needs to prepare for the next shot, therefore it may take time to start next capturing after changing to "IDLE". The camera will start capturing as soon as possible.

The client can get the list of postviews after the shooting via "[getEvent \(v1.2\)](#)" parameter ("contShooting").

stopContShooting

Overview

This API provides a function to stop continuous shooting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "stopContShooting",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

None

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startContShooting](#)
- [getEvent \(v1.2\)](#)

Special note (details)

The continuous shooting mode and speed should be set properly via "[setContShootingMode](#)" and "[setContShootingSpeed](#)" to use this API.

This API is only available when the shoot mode is "still".

Even if this API is successful, the server may not be ready to start the next shooting. The next shooting is prohibited until the client could make sure, that the server is ready to start the next shooting, through the "[getEvent \(v1.2\)](#)" callback parameter "cameraStatus". See "[startContShooting](#)" API specification for the detail.

The client can get the postview images via "contShooting" object of "[getEvent \(v1.2\)](#)" callback parameter if the server supports.

Movie recording

startMovieRec

Overview

This API provides a function to start recording movie.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "startMovieRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopMovieRec](#)
- [getEvent](#)

Special note (details)

This API instructs the server side to start recording movie. When this API is called and the server starts recording movie, the camera status will change as follows. The camera status can be obtained by ["getEvent"](#).

[Client calls ["startMovieRec"](#)]

Camera status: "IDLE" -> "MovieWaitRecStart" -> "MovieRecording".

After the recording has started, the client may stop the recording. To stop the recording, "[stopMovieRec](#)" must be called, and the camera status will change as follows.

[Client calls "[stopMovieRec](#)"]

Camera status: "MovieRecording" -> "MovieWaitRecStop" -> "MovieSaving" -> "IDLE".

Note that this sequence is the example of typical case. For example, some servers may skip "MovieWaitRecStop".

The client should check the "[getEvent](#)" parameter ("cameraStatus") and check if it is "IDLE" before calling "[startMovieRec](#)".

This API is only available when the shoot mode is "movie".

Some camera models need "[startRecMode](#)" API call before recording movie. See "[startRecMode](#)" for details.

stopMovieRec

Overview

This API provides a function to stop recording movie.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "stopMovieRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|--|
| 0 | string | Reserved. Empty string will be set. |

JSON Example

```
{
  "result": [
    ""
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startMovieRec](#)

Special note (details)

This API is only available when the shoot mode is "movie".

Even if this API is successful, the server may not be ready to start the next shot. The next shot is prohibited until the client could make sure, that the server is ready to start the next shot, through the "[getEvent](#)" callback parameter "cameraStatus". See "[startMovieRec](#)" API specification for the detail.

Audio recording

startAudioRec

Overview

This API provides a function to start audio recording.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "startAudioRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopAudioRec](#)
- [getEvent](#)

Special note (details)

This API instructs the server side to start audio recording. When this API is called and the server starts audio recording, the camera status will change as follows. The camera status can be obtained by ["getEvent"](#).

[Client calls ["startAudioRec"](#)]

Camera status: "IDLE" -> "AudioWaitRecStart" -> "AudioRecording".

After the recording has started, the client may stop the recording. To stop the recording, "[stopAudioRec](#)" must be called, and the camera status will change as follows.

[Client calls "[stopAudioRec](#)"]

Camera status: "AudioRecording" -> "AudioWaitRecStop" -> "AudioSaving" -> "IDLE".

Note that this sequence is the example of typical case.

The client should check the "[getEvent](#)" parameter ("cameraStatus") and check if it is "IDLE" before calling "[startAudioRec](#)".

This API is only available when the shoot mode is "audio".

Note that the server may disable the liveview function when the shoot mode is "audio". The client should check liveview availability by "liveviewStatus" of "[getEvent](#)". The APIs availability will also be changed. The client should check the APIs availability by available API list. When the client switches the shoot mode from "audio" to others, the client can restart the liveview by calling "[startLiveview](#)".

stopAudioRec

Overview

This API provides a function to stop audio recording.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "stopAudioRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is return. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0]
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startAudioRec](#)

Special note (details)

This API is only available when the shoot mode is "audio".

Even if this API is successful, the server may not be ready to start the next recording. The next recording is prohibited until the client could make sure, that the server is ready to start the next recording, through the "[getEvent](#)" callback parameter "cameraStatus".

Interval still recording

startIntervalStillRec

Overview

This API provides a function to start interval still recording.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "startIntervalStillRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopIntervalStillRec](#)
- [getEvent](#)

Special note (details)

This API instructs the server side to start interval still recording. When this API is called and the server starts interval still recording, the camera status will change as follows. The camera status will be notified by "[getEvent](#)".

[Client calls "[startIntervalStillRec](#)"]

Camera status: "IDLE" -> "IntervalWaitRecStart" -> "IntervalRecording".

After the recording has started, the client may stop the recording. To stop the recording, "[stopIntervalStillRec](#)" must be called, and the camera status will change as follows.

[Client calls "[stopIntervalStillRec](#)"]

Camera status: "IntervalRecording" -> "IntervalWaitRecStop" -> "IDLE".

Note that this sequence is the example of typical case.

The client should check the "[getEvent](#)" parameter ("cameraStatus") and check if it is "IDLE" before calling "[startIntervalStillRec](#)".

This API is only available when the shoot mode is "intervalstill".

stopIntervalStillRec

Overview

This API provides a function to stop interval still recording.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "stopIntervalStillRec",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startIntervalStillRec](#)

Special note (details)

This API is only available when the shoot mode is "intervalstill".

Even if this API is successful, the server may not be ready to start the next shot. The next shot is prohibited until the client could make sure, that the server is ready to start the next shot, through the "[getEvent](#)" callback parameter "cameraStatus". See "[startIntervalStillRec](#)" API specification for the detail.

Liveview

startLiveview

Overview

This API provides a function to start liveview.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "startLiveview",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------|-----------------|
| 0 | string | URL of liveview |

JSON Example

```
{
  "result": [
    "http://ip:port/liveview/liveviewstream"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopLiveview](#)

Special note (details)

This API commands the server to start the liveview function. When this API is successful, the client can obtain the URL for downloading the liveview data. The method to obtain liveview is HTTP GET. The liveview comes in one data stream. Refer to [Liveview Data Format](#).

The server may stop liveview by itself. The client should check the status of the liveview, which can be obtained by "[getEvent](#)" callback parameter "liveviewStatus". When this API is called and the server cannot start liveview, this API returns the callback with error. The client should wait for an appropriate interval, and then recall this API and restart. Some camera models need "[startRecMode](#)" API call before starting the liveview. See "[startRecMode](#)" for details.

Please see [Sample Sequence](#) for more information about the procedure of calling APIs.

stopLiveview

Overview

This API provides a function to stop liveview.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "stopLiveview",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startLiveview](#)

Special note (details)

This API instructs the server side to stop the liveview function.

After calling this API, the client should not access the URL obtained in "[startLiveview](#)" before. If the client would like to start liveview again, it should obtain the liveview URL again by calling "[startLiveview](#)".

Liveview size

startLiveviewWithSize

Overview

This API provides a function to start liveview with specific liveview size.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Liveview size (See Liveview size parameter of Parameter description) |

JSON Example

```
{
  "method": "startLiveviewWithSize",
  "params": ["M"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|--------|------|-----------------|
| string | 1 | URL of liveview |

JSON Example

```
{
  "result": [
    "http://ip:port/liveview/liveviewstream"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startLiveview](#)
- [stopLiveview](#)

Special note (details)

The basic concepts are common with "[startLiveview](#)". This API allows the client to set specific liveview size in addition when it starts liveview. The client can get the server's supported sizes using "[getSupportedLiveviewSize](#)" and "[getAvailableLiveviewSize](#)". The actual liveview size depends on the server and the client can get the actual size by decoding liveview data. Some camera models change the liveview quality instead of making the size larger when the client sets larger size parameter. Note that the client should call "[stopLiveview](#)" to change the liveview size before calling this API.

getLiveviewSize

Overview

This API provides a function to get current liveview size.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getLiveviewSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current liveview size (See Liveview size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "M"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSupportedLiveviewSize](#)
- [getAvailableLiveviewSize](#)
- [startLiveviewWithSize](#)

Special note (details)

None in particular.

getSupportedLiveviewSize

Overview

This API provides a function to get the supported liveview sizes.

The client should use "[getAvailableLiveviewSize](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedLiveviewSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported liveview sizes (See Liveview size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["L", "M"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getLiveviewSize](#)
- [getAvailableLiveviewSize](#)
- [startLiveviewWithSize](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableLiveviewSize

Overview

This API provides a function to get current liveview size and the available liveview sizes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableLiveviewSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current liveview size (See Liveview size parameter of Parameter description) |
| 1 | string-array | A list of available liveview sizes (See Liveview size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "M",
    ["L", "M"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getLiveviewSize](#)
- [getSupportedLiveviewSize](#)
- [startLiveviewWithSize](#)

Special note (details)

This API returns current set value and available values at the moment.

Liveview frame

setLiveviewFrameInfo

Overview

This API provides a function to switch the liveview frame information transferring. The liveview frame information includes focus frames, face detection frames and tracking frames on the liveview.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-------------|---------|--|
| "frameInfo" | boolean | true - Transfer the liveview frame information false - Not transfer |

JSON Example

```
{
  "method": "setLiveviewFrameInfo",
  "params": [
    {
      "frameInfo": true
    }
  ],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getLiveviewFrameInfo](#)

Special note (details)

If the parameter is true, the liveview frame information will be transferred on the liveview data. Please refer to [Liveview Data Format](#) for details.

getLiveviewFrameInfo

Overview

This API provides a function to get current setting of the liveview frame information transferring.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getLiveviewFrameInfo",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-------------|---------|--|
| "frameInfo" | boolean | true - Transfer the liveview frame information false - Not transfer |

JSON Example

```
{
  "result": [
    {
      "frameInfo": true
    }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setLiveviewFrameInfo](#)

Special note (details)

None in particular.

Zoom

actZoom

Overview

This API provides a function to zoom.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Direction (See Zoom parameters of Parameter description) |
| 1 | string | Movement (See Zoom parameters of Parameter description) |

JSON Example

```
{
  "method": "actZoom",
  "params": ["in", "start"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getEvent](#)

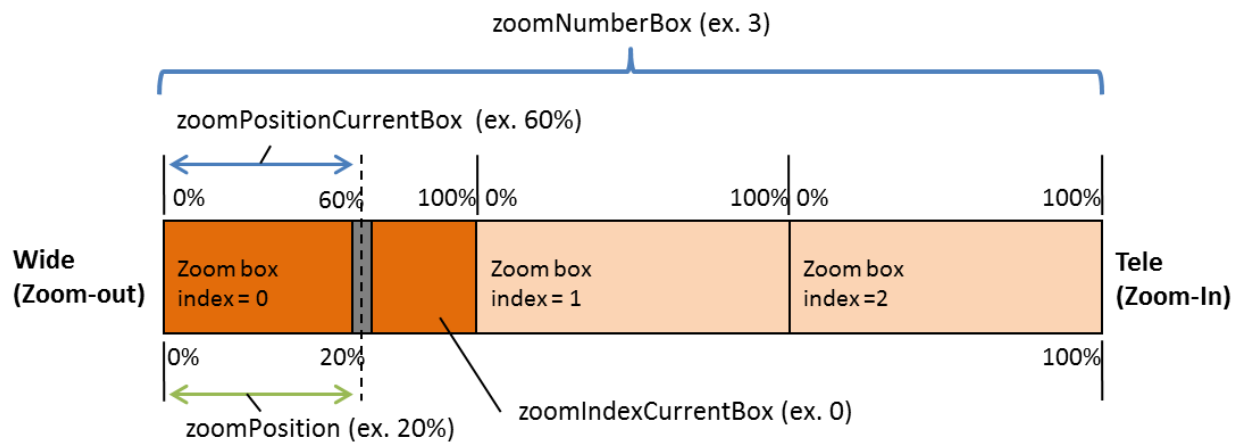
Special note (details)

When the client set "stop" as movement parameter, the direction should be the same as the last "start". For example, if the client set "in"- "start" at the beginning, "in"- "stop" are necessary. If the client set "in"- "start", and set "out"- "stop" later, then the callback will include error.

When the client set "start" as movement parameter, zoom operation will be continued until "stop" or reaching the termination. When "1shot" is set, zoom operation will be stopped after a certain position.

The client can check the zoom information using "[getEvent](#)". The zoom information consists of four parameters, "zoomPosition", "zoomNumberBox", "zoomIndexCurrentBox", and

"zoomPositionCurrentBox" as follows. "Zoom box" represents the type of zoom such as optical and digital.



| type | name | explanation |
|---------|------------------------|---|
| integer | zoomPosition | Zoom position to the whole (0 - 100, unit: percentage) |
| integer | zoomNumberBox | Number of zoom box |
| integer | zoomIndexCurrentBox | Index of current zoom box (starts from 0) |
| integer | zoomPositionCurrentBox | Zoom position in the current zoom box (0 - 100, unit: percentage) |

Zoom setting

setZoomSetting

Overview

This API provides a function to set a value of zoom setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|--------|--------|--|
| "zoom" | string | Zoom setting (See Zoom parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setZoomSetting",
  "params": [
    {
      "zoom": "Optical Zoom Only"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getZoomSetting](#)
- [getSupportedZoomSetting](#)
- [getAvailableZoomSetting](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "zoomSetting" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getZoomSetting

Overview

This API provides a function to get current zoom setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getZoomSetting",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|--------|--------|--|
| "zoom" | string | Current zoom setting (See Zoom parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "zoom": "Optical Zoom Only"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setZoomSetting](#)
- [getSupportedZoomSetting](#)
- [getAvailableZoomSetting](#)

Special note (details)

None in particular.

getSupportedZoomSetting

Overview

This API provides a function to get the supported zoom settings.

The client should use "[getAvailableZoomSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedZoomSetting",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported zoom settings (See Zoom parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "Optical Zoom Only",
        "On:Clear Image Zoom"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setZoomSetting](#)
- [getZoomSetting](#)
- [getAvailableZoomSetting](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableZoomSetting

Overview

This API provides a function to get current zoom setting and the available zoom settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableZoomSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "zoom" | string | Current zoom setting (See Zoom parameters of Parameter description) |
| "candidate" | string-array | A list of available zoom settings (See Zoom parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "zoom": "Optical Zoom Only",
      "candidate": [
        "Optical Zoom Only",
        "On:Clear Image Zoom"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setZoomSetting](#)
- [getZoomSetting](#)
- [getSupportedZoomSetting](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "zoomSetting" object in "[getEvent \(v1.2\)](#)" callback.

Half-press shutter

actHalfPressShutter

Overview

This API provides a function to half-press shutter.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "actHalfPressShutter",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [cancelHalfPressShutter](#)
- [getEvent \(v1.1\)](#)

Special note (details)

This API starts camera auto adjustment function like pressing the shutter button halfway down. After calling this API, the server will start auto focus and auto exposure mainly depending on the camera settings. The client should cancel the function using "[cancelHalfPressShutter](#)" API. The client can get the focus status by checking "focusStatus" object in the response of "[getEvent \(v1.1\)](#)" API.

cancelHalfPressShutter

Overview

This API provides a function to cancel half-press shutter.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "cancelHalfPressShutter",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

None

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [actHalfPressShutter](#)
- [getEvent \(v1.1\)](#)

Special note (details)

None in particular.

Touch AF position

setTouchAFPosition

Overview

This API provides a function to enable touch AF and the position.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|-----------------|
| 0 | double | X-axis position |
| 1 | double | Y-axis position |

JSON Example

```
{
  "method": "setTouchAFPosition",
  "params": [23.4, 45.6],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | key | type | explanation |
|-------|------------|---------|---|
| 0 | | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |
| 1 | | object | The result of touch AF |
| | "AFResult" | boolean | AF result true: AF is done successfully. false: Failure to AF |
| | "AFType" | string | AF type (See Touch AF position parameter of Parameter description |

JSON Example

```
{
  "result": [
    0,
    {
      "AFResult": true,
      "AFType": "Touch"
    }
  ],
  "id": 1
}
```


Error Codes

See [Status code & Error](#)

Related API

- [getTouchAFPosition](#)
- [cancelTouchAFPosition](#)

Special note (details)

The X-axis and Y-axis position are expressed in percentage. The origin of coordinates is upper left of the liveview which is transferred from the server.

The server may fail to touch AF. It also may take time to focus and response time-out error. Even if time-out error occurs, the server will continue to focus. The client should cancel touch AF using "[cancelTouchAFPosition](#)" and get current status of touch AF using "[getEvent](#)".

getTouchAFPosition

Overview

This API provides a function to get current touch AF position.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getTouchAFPosition",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

Following object.

| order | key | type | explanation |
|-------|--------------------|--------------|--|
| 0 | | object | Current touch AF position |
| | "set" | boolean | Set or not. true: Touch AF is set and focused successfully. false: Touch AF is not set or failed to focus. |
| | "touchCoordinates" | double-array | Touch coordinates. This parameter is reserved and the camera will return empty array. |

JSON Example

```
{
  "result": [
    { "set": true, "touchCoordinates": [] }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTouchAFPosition](#)
- [cancelTouchAFPosition](#)

Special note (details)

None in particular.

cancelTouchAFPosition

Overview

This API provides a function to cancel Touch AF.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "cancelTouchAFPosition",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

None

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTouchAFPosition](#)
- [getTouchAFPosition](#)

Special note (details)

None in particular.

Tracking focus

actTrackingFocus

Overview

This API provides a function to start tracking focus.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-------------|--------|-----------------|
| "xPosition" | double | X-axis position |
| "yPosition" | double | Y-axis position |

JSON Example

```
{
  "id": 1,
  "method": "actTrackingFocus",
  "params": [
    {
      "xPosition": 23.4,
      "yPosition": 45.6
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [cancelTrackingFocus](#)
- [getEvent \(v1.2\)](#)

Special note (details)

The X-axis and Y-axis position are expressed in percentage. The origin of coordinates is upper left of liveview which is transferred from the server. The tracking focus function may not operate well in some situations. The client can cancel tracking focus by using "cancelTrackingFocus" API. The server may cancel tracking focus function by itself. The client can check if the tracking focus is operating or not by using "trackingFocusStatus" object of [getEvent \(v1.2\)](#).

cancelTrackingFocus

Overview

This API provides a function to cancel tracking focus.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "cancelTrackingFocus",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

None

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [actTrackingFocus](#)
- [getEvent \(v1.2\)](#)

Special note (details)

The server may cancel tracking focus function by itself. The client can check if the tracking focus is operating or not by using "trackingFocusStatus" object of "[getEvent \(v1.2\)](#)".

setTrackingFocus

Overview

This API provides a function to set a value of tracking focus setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-----------------|--------|--|
| "trackingFocus" | string | Tracking focus setting (See Tracking focus parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setTrackingFocus",
  "params": [
    {
      "trackingFocus": "On"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getTrackingFocus](#)
- [getSupportedTrackingFocus](#)
- [getAvailableTrackingFocus](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "trackingFocus" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getTrackingFocus

Overview

This API provides a function to get current tracking focus setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getTrackingFocus",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-----------------|--------|--|
| "trackingFocus" | string | Current tracking focus setting (See Tracking focus parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "trackingFocus": "On"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTrackingFocus](#)
- [getSupportedTrackingFocus](#)
- [getAvailableTrackingFocus](#)

Special note (details)

None in particular.

getSupportedTrackingFocus

Overview

This API provides a function to get the supported tracking focus settings.

The client should use "[getAvailableTrackingFocus](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedTrackingFocus",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported tracking focus settings (See Tracking focus parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTrackingFocus](#)
- [getTrackingFocus](#)
- [getAvailableTrackingFocus](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableTrackingFocus

Overview

This API provides a function to get current tracking focus setting and the available tracking focus settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableTrackingFocus",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-----------------|--------------|---|
| "trackingFocus" | string | Current tracking focus setting (See Tracking focus parameters of Parameter description) |
| "candidate" | string-array | A list of available tracking focus settings (See Tracking focus parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "trackingFocus": "On",
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTrackingFocus](#)
- [getTrackingFocus](#)
- [getSupportedTrackingFocus](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "trackingFocus" object in "[getEvent \(v1.2\)](#)" callback.

Continuous shooting mode

setContShootingMode

Overview

This API provides a function to set a value of continuous shooting mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|--------------------|--------|--|
| "contShootingMode" | string | Continuous shooting mode (See Continuous shooting mode parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setContShootingMode",
  "params": [
    {
      "contShootingMode": "Spd Priority Cont."
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getContShootingMode](#)
- [getSupportedContShootingMode](#)
- [getAvailableContShootingMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "contShootingMode" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

The client should call appropriate APIs according to the continuous shooting mode. When the continuous shooting mode is "Single", "Burst" or "MotionShot", the client should call "[actTakePicture](#)" API to take still image. When the mode is "Continuous" or "Spd Priority Cont.", the client can call "[startContShooting](#)" and "[stopContShooting](#)" APIs. The client can check the API availability using "getAvailableApiList" or "availableApiList" object of "[getEvent \(v1.2\)](#)" API.

getContShootingMode

Overview

This API provides a function to get current continuous shooting mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getContShootingMode",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|--------------------|--------|--|
| "contShootingMode" | string | Current continuous shooting mode (See Continuous shooting mode parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "contShootingMode": "Spd Priority Cont."
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingMode](#)
- [getSupportedContShootingMode](#)
- [getAvailableContShootingMode](#)

Special note (details)

None in particular.

getSupportedContShootingMode

Overview

This API provides a function to get the supported continuous shooting modes.

The client should use "[getAvailableContShootingMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedContShootingMode",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported continuous shooting modes (See Continuous shooting mode parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "Single",
        "Continuous",
        "Spd Priority Cont."
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingMode](#)
- [getContShootingMode](#)
- [getAvailableContShootingMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableContShootingMode

Overview

This API provides a function to get current continuous shooting mode and the available continuous shooting modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableContShootingMode",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|--------------------|--------------|---|
| "contShootingMode" | string | Current continuous shooting mode (See Continuous shooting mode parameters of Parameter description) |
| "candidate" | string-array | A list of available continuous shooting modes (See Continuous shooting mode parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "contShootingMode": "Spd Priority Cont.",
      "candidate": [
        "Single",
        "Continuous",
        "Spd Priority Cont."
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingMode](#)
- [getContShootingMode](#)
- [getSupportedContShootingMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "contShootingMode" object in "[getEvent \(v1.2\)](#)" callback.

Continuous shooting speed

setContShootingSpeed

Overview

This API provides a function to set a value of continuous shooting speed.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|---------------------|--------|--|
| "contShootingSpeed" | string | Continuous shooting speed (See Continuous shooting speed parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setContShootingSpeed",
  "params": [
    {
      "contShootingSpeed": "Hi"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getContShootingSpeed](#)
- [getSupportedContShootingSpeed](#)
- [getAvailableContShootingSpeed](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "contShootingSpeed" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getContShootingSpeed

Overview

This API provides a function to get current continuous shooting speed.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getContShootingSpeed",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|---------------------|--------|--|
| "contShootingSpeed" | string | Current continuous shooting speed (See Continuous shooting speed parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "contShootingSpeed": "Hi"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingSpeed](#)
- [getSupportedContShootingSpeed](#)
- [getAvailableContShootingSpeed](#)

Special note (details)

None in particular.

getSupportedContShootingSpeed

Overview

This API provides a function to get the supported continuous shooting speeds.

The client should use "[getAvailableContShootingSpeed](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedContShootingSpeed",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported continuous shooting speeds (See Continuous shooting speed parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "Hi",
        "Low"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingSpeed](#)
- [getContShootingSpeed](#)
- [getAvailableContShootingSpeed](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableContShootingSpeed

Overview

This API provides a function to get current continuous shooting speed and the available continuous shooting speeds at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableContShootingSpeed",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|---------------------|--------------|---|
| "contShootingSpeed" | string | Current continuous shooting speed (See Continuous shooting speed parameters of Parameter description) |
| "candidate" | string-array | A list of available continuous shooting speeds (See Continuous shooting speed parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "contShootingSpeed": "Hi",
      "candidate": [
        "Hi",
        "Low"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setContShootingSpeed](#)
- [getContShootingSpeed](#)
- [getSupportedContShootingSpeed](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "contShootingSpeed" object in "[getEvent \(v1.2\)](#)" callback.

Self-timer

setSelfTimer

Overview

This API provides a function to set a value of self-timer.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Self-timer (unit: second) (See Self-timer parameters of Parameter description) |

JSON Example

```
{
  "method": "setSelfTimer",
  "params": [2],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSelfTimer](#)
- [getSupportedSelfTimer](#)
- [getAvailableSelfTimer](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "selfTimer" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

Some camera models need "[startRecMode](#)" API call before accessing camera settings. See "[startRecMode](#)" for details.

getSelfTimer

Overview

This API provides a function to get current self-timer setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSelfTimer",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Current self-timer setting (unit: second) (See Self-timer parameters of Parameter description) |

JSON Example

```
{
  "result": [2],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSelfTimer](#)
- [getSupportedSelfTimer](#)
- [getAvailableSelfTimer](#)

Special note (details)

None in particular.

getSupportedSelfTimer

Overview

This API provides a function to get the supported self-timer settings.

The client should use "[getAvailableSelfTimer](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedSelfTimer",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|--|
| 0 | integer-array | A list of supported self-timer settings (unit: second) (See Self-timer parameters of Parameter description) |

JSON Example

```
{
  "result": [
    [0,2,10]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSelfTimer](#)
- [getSelfTimer](#)
- [getAvailableSelfTimer](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableSelfTimer

Overview

This API provides a function to get current self-timer setting and the available self-timer settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableSelfTimer",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|--|
| 0 | integer | Current self-timer setting (unit: second) (See Self-timer parameters of Parameter description) |
| 1 | integer-array | A list of available self-timer settings (unit: second) (See Self-timer parameters of Parameter description) |

JSON Example

```
{
  "result": [
    0,
    [0,2,10]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSelfTimer](#)
- [getSelfTimer](#)
- [getSupportedSelfTimer](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "selfTimer" object in "[getEvent](#)" callback.

Exposure mode

setExposureMode

Overview

This API provides a function to set a value of exposure mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Exposure mode (See Exposure mode parameter of Parameter description) |

JSON Example

```
{
  "method": "setExposureMode",
  "params": ["Intelligent Auto"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getExposureMode](#)
- [getSupportedExposureMode](#)
- [getAvailableExposureMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "exposureMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getExposureMode

Overview

This API provides a function to get current exposure mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getExposureMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current exposure mode (See Exposure mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "Intelligent Auto"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureMode](#)
- [getSupportedExposureMode](#)
- [getAvailableExposureMode](#)

Special note (details)

None in particular.

getSupportedExposureMode

Overview

This API provides a function to get the supported exposure modes.

The client should use "[getAvailableExposureMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedExposureMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported exposure modes (See Exposure mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["Intelligent Auto", "Superior Auto"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureMode](#)
- [getExposureMode](#)
- [getAvailableExposureMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableExposureMode

Overview

This API provides a function to get current exposure mode and the available exposure modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableExposureMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current exposure mode (See Exposure mode parameter of Parameter description) |
| 1 | string-array | A list of available exposure modes (See Exposure mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "Intelligent Auto",
    ["Intelligent Auto", "Superior Auto"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureMode](#)
- [getExposureMode](#)
- [getSupportedExposureMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "exposureMode" object in ["getEvent"](#) callback.

Focus mode

setFocusMode

Overview

This API provides a function to set a value of focus mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Focus mode (See Focus mode parameter of Parameter description) |

JSON Example

```
{
  "method": "setFocusMode",
  "params": ["MF"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getFocusMode](#)
- [getSupportedFocusMode](#)
- [getAvailableFocusMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "focusMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getFocusMode

Overview

This API provides a function to get current focus mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getFocusMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current focus mode (See Focus mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "MF"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFocusMode](#)
- [getSupportedFocusMode](#)
- [getAvailableFocusMode](#)

Special note (details)

None in particular.

getSupportedFocusMode

Overview

This API provides a function to get the supported focus modes.

The client should use "[getAvailableFocusMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedFocusMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | Supported focus modes (See Focus mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["AF-S", "MF"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFocusMode](#)
- [getFocusMode](#)
- [getAvailableFocusMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableFocusMode

Overview

This API provides a function to get current focus mode and the available focus modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableFocusMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current focus mode (See Focus mode parameter of Parameter description) |
| 1 | string-array | A list of available focus modes (See Focus mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "MF",
    ["AF-S", "MF"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFocusMode](#)
- [getFocusMode](#)
- [getSupportedFocusMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "focusMode" object in "[getEvent](#)" callback.

Exposure compensation

setExposureCompensation

Overview

This API provides a function to set a value of exposure compensation.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Index value of exposure compensation (See getSupportedExposureCompensation) |

JSON Example

```
{
  "method": "setExposureCompensation",
  "params": [2],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getExposureCompensation](#)
- [getSupportedExposureCompensation](#)
- [getAvailableExposureCompensation](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "exposureCompensation" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

See [getSupportedExposureCompensation](#) for details of the request parameter.

getExposureCompensation

Overview

This API provides a function to get current exposure compensation value.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getExposureCompensation",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Current index value of exposure compensation (See getSupportedExposureCompensation) |

JSON Example

```
{
  "result": [
    2
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureCompensation](#)
- [getSupportedExposureCompensation](#)
- [getAvailableExposureCompensation](#)

Special note (details)

See [getSupportedExposureCompensation](#).

getSupportedExposureCompensation

Overview

This API provides a function to get the supported exposure compensation values.

The client should use "[getAvailableExposureCompensation](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedExposureCompensation",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|---|
| 0 | integer-array | Upper limit of exposure compensation index values list that supported by the server. |
| 1 | integer-array | Lower limit of exposure compensation index values list that supported by the server. |
| 2 | integer-array | Exposure compensation index step list of values that supported by the server. 1: 1/3 EV 2: 1/2 EV 0: invalid |

JSON Example

```
{
  "result": [
    [6,6],
    [-6,-6],
    [1,2]
  ],
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureCompensation](#)
- [getExposureCompensation](#)
- [getAvailableExposureCompensation](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

The callback parameter "Upper limit", "Lower limit" and "index step" of this API will be expanded (Declared in the list in order to handle multiple values). For example, in case that Upper limit[0]=6, Lower limit[0]=-6, index step[0]=1(1/3EV) or index step[0]=2(1/2EV), correspondence between exposure index value and UI value is below.

| Exposure index value | UI [EV] | |
|----------------------|----------------------------|----------------------------|
| | index step[0] =1(1/3EV) | index step[0] =2(1/2EV) |
| 6 | 2.0 | 3.0 |
| 5 | 1.7 | 2.5 |
| 4 | 1.3 | 2.0 |
| 3 | 1.0 | 1.5 |
| 2 | 0.7 | 1.0 |
| 1 | 0.3 | 0.5 |
| 0 | 0.0 | 0.0 |
| -1 | -0.3 | -0.5 |
| -2 | -0.7 | -1.0 |
| -3 | -1.0 | -1.5 |
| -4 | -1.3 | -2.0 |
| -5 | -1.7 | -2.5 |
| -6 | -2.0 | -3.0 |

Example: Upper limit[0]=6, Lower limit[0]=-6

getAvailableExposureCompensation

Overview

This API provides a function to get current exposure compensation value and the available exposure compensation values at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableExposureCompensation",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Current exposure compensation index value |
| 1 | integer | Upper limit of available exposure compensation index value. |
| 2 | integer | Lower limit of available exposure compensation index value. |
| 3 | integer | Exposure compensation index step value |

JSON Example

```
{
  "result": [
    0,
    6,
    -6,
    1
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setExposureCompensation](#)
- [getExposureCompensation](#)
- [getSupportedExposureCompensation](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "exposureCompensation" object in "[getEvent](#)" callback. See [getSupportedExposureCompensation](#).

F number

setFNumber

Overview

This API provides a function to set a value of F number.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|-------------|
| 0 | string | F number |

JSON Example

```
{
  "method": "setFNumber",
  "params": ["5.4"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getFNumber](#)
- [getSupportedFNumber](#)
- [getAvailableFNumber](#)

Special note (details)

The parameter "F number" can be set in the range of supported F numbers. The client can get available F numbers at the moment using "[getAvailableFNumber](#)". Note that available F numbers often may vary on lenses.

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "fNumber" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getFNumber

Overview

This API provides a function to get current F number.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getFNumber",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current F number (See setFNumber) |

JSON Example

```
{
  "result": [
    "5.4"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFNumber](#)
- [getSupportedFNumber](#)
- [getAvailableFNumber](#)

Special note (details)

None in particular.

getSupportedFNumber

Overview

This API provides a function to get the supported F numbers.

The client should use "[getAvailableFNumber](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedFNumber",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported F numbers (See setFNumber) |

JSON Example

```
{
  "result": [
    ["1.4", "2.0", "2.8", "4.0", "5.6"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFNumber](#)
- [getFNumber](#)
- [getAvailableFNumber](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableFNumber

Overview

This API provides a function to get current F number and the available F numbers at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableFNumber",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current F number (See setFNumber) |
| 1 | string-array | A list of available F numbers (See setFNumber) |

JSON Example

```
{
  "result": [
    "1.4",
    ["1.4", "2.0", "2.8", "4.0"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFNumber](#)
- [getFNumber](#)
- [getSupportedFNumber](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "fNumber" object in [getEvent](#) callback.

Shutter speed

setShutterSpeed

Overview

This API provides a function to set a value of shutter speed.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---------------|
| 0 | string | Shutter speed |

JSON Example

```
{
  "method": "setShutterSpeed",
  "params": ["1/2"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getShutterSpeed](#)
- [getSupportedShutterSpeed](#)
- [getAvailableShutterSpeed](#)

Special note (details)

The parameter "Shutter speed" can be set in the range of supported shutter speeds. The client can get available shutter speeds at the moment using "[getAvailableShutterSpeed](#)".

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "shutterSpeed" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getShutterSpeed

Overview

This API provides a function to get current shutter speed.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getShutterSpeed",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current shutter speed (See setShutterSpeed) |

JSON Example

```
{
  "result": [
    "1/2"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShutterSpeed](#)
- [getSupportedShutterSpeed](#)
- [getAvailableShutterSpeed](#)

Special note (details)

None in particular.

getSupportedShutterSpeed

Overview

This API provides a function to get the supported shutter speeds.

The client should use "[getAvailableShutterSpeed](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedShutterSpeed",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported shutter speeds (See setShutterSpeed) |

JSON Example

```
{
  "result": [
    ["30\"", "1\"", "1/2", "1/30", "1/250"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShutterSpeed](#)
- [getShutterSpeed](#)
- [getAvailableShutterSpeed](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableShutterSpeed

Overview

This API provides a function to get current shutter speed and the available shutter speeds at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableShutterSpeed",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current shutter speed (See setShutterSpeed) |
| 1 | string-array | A list of available shutter speeds (See setShutterSpeed) |

JSON Example

```
{
  "result": [
    "1/2",
    ["30\"", "1\"", "1/2", "1/30", "1/250"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setShutterSpeed](#)
- [getShutterSpeed](#)
- [getSupportedShutterSpeed](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "shutterSpeed" object in "[getEvent](#)" callback.

ISO speed rate

setIsoSpeedRate

Overview

This API provides a function to set a value of ISO speed rate.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|----------------|
| 0 | string | ISO speed rate |

JSON Example

```
{
  "method": "setIsoSpeedRate",
  "params": ["400"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getIsoSpeedRate](#)
- [getSupportedIsoSpeedRate](#)
- [getAvailableIsoSpeedRate](#)

Special note (details)

The parameter "ISO speed rate" can be set in the range of supported ISO speed rates. The client can get available ISO speed rates at the moment using "[getAvailableIsoSpeedRate](#)".

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "isoSpeedRate" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getIsoSpeedRate

Overview

This API provides a function to get current ISO speed rate value.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getIsoSpeedRate",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|--|
| 0 | string | Current ISO speed rate value (See setIsoSpeedRate) |

JSON Example

```
{
  "result": [
    "400"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIsoSpeedRate](#)
- [getSupportedIsoSpeedRate](#)
- [getAvailableIsoSpeedRate](#)

Special note (details)

None in particular.

getSupportedIsoSpeedRate

Overview

This API provides a function to get the supported ISO speed rate values.

The client should use "[getAvailableIsoSpeedRate](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedIsoSpeedRate",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string-array | A list of supported ISO speed rate values (See setIsoSpeedRate) |

JSON Example

```
{
  "result": [
    ["100", "400", "3200"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIsoSpeedRate](#)
- [getIsoSpeedRate](#)
- [getAvailableIsoSpeedRate](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableIsoSpeedRate

Overview

This API provides a function to get current ISO speed rate value and the available ISO speed rate values at the moment.

The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableIsoSpeedRate",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string | Current ISO speed rate value (See setIsoSpeedRate) |
| 1 | string-array | A list of available ISO speed rate values (See setIsoSpeedRate) |

JSON Example

```
{
  "result": [
    "400",
    ["100", "400", "3200"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIsoSpeedRate](#)
- [getIsoSpeedRate](#)
- [getSupportedIsoSpeedRate](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "isoSpeedRate" object in "[getEvent](#)" callback.

White balance

setWhiteBalance

Overview

This API provides a function to set a value of white balance.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|--|
| 0 | string | White balance mode (See White balance parameter of Parameter description) |
| 1 | boolean | Color temperature enabled flag true: enabled false: not enabled |
| 2 | integer | Color temperature (See White balance parameter of Parameter description) |

JSON Example

```
{
  "method": "setWhiteBalance",
  "params": ["Color Temperature", true, 2500],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getWhiteBalance](#)
- [getSupportedWhiteBalance](#)
- [getAvailableWhiteBalance](#)

Special note (details)

The parameter "White balance mode" and "color temperature" can be set in the range of supported modes and temperatures. The client can get available range at the moment using "[getAvailableWhiteBalance](#)".

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "whiteBalance" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getWhiteBalance

Overview

This API provides a function to get current white balance.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getWhiteBalance",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

Following object.

| order | key | type | explanation |
|-------|--------------------|---------|--|
| 0 | | object | Current white balance |
| | "whiteBalanceMode" | string | White balance mode (See White balance parameter of Parameter description) |
| | "colorTemperature" | integer | Color temperature (See White balance parameter of Parameter description) |

JSON Example

```
{
  "result": [
    { "whiteBalanceMode": "Color Temperature", "colorTemperature": 2500 }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setWhiteBalance](#)
- [getSupportedWhiteBalance](#)
- [getAvailableWhiteBalance](#)

Special note (details)

None in particular.

getSupportedWhiteBalance

Overview

This API provides a function to get the supported white balances.

The client should use "[getAvailableWhiteBalance](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedWhiteBalance",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | key | type | explanation |
|-------|-------------------------|---------------|---|
| 0 | | object-array | A list of supported white balances |
| | "whiteBalanceMode" | string | White balance mode (See White balance parameter of Parameter description) |
| | "colorTemperatureRange" | integer-array | Color temperature range (See White balance parameter of Parameter description) |

JSON Example

```
{
  "result": [
    [
      { "whiteBalanceMode": "Auto WB", "colorTemperatureRange": [] },
      { "whiteBalanceMode": "Daylight", "colorTemperatureRange": [] },
      { "whiteBalanceMode": "Color Temperature", "colorTemperatureRange": [9900, 2500, 100] }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setWhiteBalance](#)
- [getWhiteBalance](#)
- [getAvailableWhiteBalance](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableWhiteBalance

Overview

This API provides a function to get current white balance and the available white balances at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getAvailableWhiteBalance",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|-------|-------------------------|---------------|---|
| 0 | | object | Current white balance |
| | "whiteBalanceMode" | string | White balance mode (See White balance parameter of Parameter description) |
| | "colorTemperature" | integer | Color temperature (See White balance parameter of Parameter description) (When -1 is set, this parameter is invalid.) |
| 1 | | object-array | A list of available white balances |
| | "whiteBalanceMode" | string | White balance mode (See White balance parameter of Parameter description) |
| | "colorTemperatureRange" | integer-array | Color temperature range (See White balance parameter of Parameter description) |

JSON Example

```
{
  "result": [
    {
      "whiteBalanceMode": "Color Temperature", "colorTemperature": 2500 },
    [
      {
        "whiteBalanceMode": "Auto WB", "colorTemperatureRange": [] },
        {
          "whiteBalanceMode": "Daylight", "colorTemperatureRange": [] },
        {
          "whiteBalanceMode": "Color Temperature", "colorTemperatureRange": [9900, 2500, 100] }
      ]
    ],
    "id": 1
  ]
}
```


Error Codes

See [Status code & Error](#)

Related API

- [setWhiteBalance](#)
- [getWhiteBalance](#)
- [getSupportedWhiteBalance](#)

Special note (details)

This API returns current set value and available values at the moment.

Program shift

setProgramShift

Overview

This API provides a function to set program shift. The client can change the aperture (F number) and shutter speed combination using this API.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|--------------|
| 0 | integer | Shift amount |

JSON Example

```
{
  "method": "setProgramShift",
  "params": [1],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSupportedProgramShift](#)

Special note (details)

The parameter "Shift amount" can be set in the range of supported amounts. The client can get supported amounts using [getSupportedProgramShift](#). The client can check "programShift" object in [getEvent](#) callback to recognize the timing of a change of the server. When the "isShifted" parameter in "programShift" object in [getEvent](#) callback is false, it means that the program shift is canceled. The client also can check the changes of "fNumber" and "shutterSpeed" values after setting the program shift.

getSupportedProgramShift

Overview

This API provides a function to get the supported program shift amounts.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedProgramShift",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|--|
| 0 | integer-array | Range of supported shift amounts [<maximum value>, <minimum value>] |

JSON Example

```
{
  "result": [
    [5, -5]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setProgramShift](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

For example, when the response is [2, -2], the client can set 2, 1, 0, -1 or -2 as the shift amount using "[setProgramShift](#)". Typically, the client can set -1 or 1 to shift the aperture and shutter speed combination step by step.

Flash mode

setFlashMode

Overview

This API provides a function to set a value of flash mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Flash mode (See Flash mode parameter of Parameter description) |

JSON Example

```
{
  "method": "setFlashMode",
  "params": ["off"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getFlashMode](#)
- [getSupportedFlashMode](#)
- [getAvailableFlashMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "flashMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getFlashMode

Overview

This API provides a function to get current flash mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getFlashMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current flash mode (See Flash mode parameter of Parameter description) |

JSON Example

```
{
  "result": ["off"],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlashMode](#)
- [getSupportedFlashMode](#)
- [getAvailableFlashMode](#)

Special note (details)

None in particular.

getSupportedFlashMode

Overview

This API provides a function to get the supported flash modes.

The client should use "[getAvailableFlashMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedFlashMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported flash modes (See Flash mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["off", "on"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlashMode](#)
- [getFlashMode](#)
- [getAvailableFlashMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableFlashMode

Overview

This API provides a function to get current flash mode and the available flash modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableFlashMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current flash mode (See Flash mode parameter of Parameter description) |
| 1 | string-array | A list of available flash modes (See Flash mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "off",
    ["off", "on"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlashMode](#)
- [getFlashMode](#)
- [getSupportedFlashMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "flashMode" object in "[getEvent](#)" callback.

Still size

setStillSize

Overview

This API provides a function to set a value of still size.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Still aspect (See Still size parameter of Parameter description) |
| 1 | string | Still size (See Still size parameter of Parameter description) |

JSON Example

```
{
  "method": "setStillSize",
  "params": ["4:3", "5M"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|---------|------|---|
| integer | 1 | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getStillSize](#)
- [getSupportedStillSize](#)
- [getAvailableStillSize](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "stillSize" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getStillSize

Overview

This API provides a function to get current still size.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getStillSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|-------|----------|--------|---|
| 0 | | object | Current still size |
| | "aspect" | string | still aspect (See Still size parameter of Parameter description) |
| | "size" | string | still size (See Still size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    {"aspect": "4:3", "size": "5M"}
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillSize](#)
- [getSupportedStillSize](#)
- [getAvailableStillSize](#)

Special note (details)

None in particular.

getSupportedStillSize

Overview

This API provides a function to get the supported still sizes.

The client should use "[getAvailableStillSize](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedStillSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | key | type | explanation |
|-------|----------|--------------|---|
| 0 | | object-array | A list of supported still sizes |
| | "aspect" | string | Still aspect (See Still size parameter of Parameter description) |
| | "size" | string | Still size (See Still size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    [
      { "aspect": "16:9", "size": "17M" },
      { "aspect": "16:9", "size": "7.5M" },
      { "aspect": "4:3", "size": "18M" }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillSize](#)
- [getStillSize](#)
- [getAvailableStillSize](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableStillSize

Overview

This API provides a function to get current still size and the available still sizes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getAvailableStillSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|-------|----------|--------------|---|
| 0 | | object | Current still size |
| | "aspect" | string | Still aspect (See Still size parameter of Parameter description) |
| | "size" | string | Still size (See Still size parameter of Parameter description) |
| 1 | | object-array | A list of available still sizes |
| | "aspect" | string | Still aspect (See Still size parameter of Parameter description) |
| | "size" | string | Still size (See Still size parameter of Parameter description) |

JSON Example

```
{
  "result": [
    { "aspect": "4:3", "size": "5M" },
    [
      { "aspect": "16:9", "size": "17M" },
      { "aspect": "16:9", "size": "7.5M" },
      { "aspect": "4:3", "size": "18M" }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillSize](#)
- [getStillSize](#)
- [getSupportedStillSize](#)

Special note (details)

This API returns current set value and available values at the moment.

Still quality

setStillQuality

Overview

This API provides a function to set a value of still quality.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|----------------|--------|--|
| "stillQuality" | string | Still quality (See Still quality parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setStillQuality",
  "params": [
    {
      "stillQuality": "RAW+JPEG"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getStillQuality](#)
- [getSupportedStillQuality](#)
- [getAvailableStillQuality](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "stillQuality" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getStillQuality

Overview

This API provides a function to get current still quality.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getStillQuality",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|----------------|--------|--|
| "stillQuality" | string | Current still quality (See Still quality parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "stillQuality": "RAW+JPEG"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillQuality](#)
- [getSupportedStillQuality](#)
- [getAvailableStillQuality](#)

Special note (details)

None in particular.

getSupportedStillQuality

Overview

This API provides a function to get the supported still quality.

The client should use "[getAvailableStillQuality](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedStillQuality",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|--|
| "candidate" | string-array | A list of supported still quality (See Still quality parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "RAW+JPEG",
        "Fine",
        "Standard"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillQuality](#)
- [getStillQuality](#)
- [getAvailableStillQuality](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableStillQuality

Overview

This API provides a function to get current still quality and the available still quality at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableStillQuality",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|----------------|--------------|--|
| "stillQuality" | string | Current still quality (See Still quality parameters of Parameter description) |
| "candidate" | string-array | A list of available still quality (See Still quality parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "stillQuality": "Fine",
      "candidate": [
        "RAW+JPEG",
        "Fine",
        "Standard"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStillQuality](#)
- [getStillQuality](#)
- [getSupportedStillQuality](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "stillQuality" object in "[getEvent \(v1.2\)](#)" callback.

Postview image size

setPostviewImageSize

Overview

This API provides a function to set a value of postview image size. The postview image can be used for storing it as the taken picture, and showing it to the client display.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|--|
| 0 | string | Postview image size (See Postview image size parameters of Parameter description) |

JSON Example

```
{
  "method": "setPostviewImageSize",
  "params": ["Original"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getPostviewImageSize](#)
- [getSupportedPostviewImageSize](#)
- [getAvailablePostviewImageSize](#)

Special note (details)

The postview is the still image data that can be received as the response of "[actTakePicture](#)" and "[awaitTakePicture](#)".

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "postviewImageSize" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

Some camera models need "[startRecMode](#)" API call before accessing camera settings. See "[startRecMode](#)" for details.

getPostviewImageSize

Overview

This API provides a function to get current postview image size.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getPostviewImageSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------|--|
| 0 | string | Current postview image size (See Postview image size parameters of Parameter description) |

JSON Example

```
{
  "result": [
    "Original"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setPostviewImageSize](#)
- [getSupportedPostviewImageSize](#)
- [getAvailablePostviewImageSize](#)

Special note (details)

None in particular.

getSupportedPostviewImageSize

Overview

This API provides a function to get the supported postview image sizes.

The client should use "[getAvailablePostviewImageSize](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedPostviewImageSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string-array | A list of supported postview image sizes (See Postview image size parameters of Parameter description) |

JSON Example

```
{
  "result": [
    ["Original", "2M"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setPostviewImageSize](#)
- [getPostviewImageSize](#)
- [getAvailablePostviewImageSize](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailablePostviewImageSize

Overview

This API provides a function to get current postview image size and the available postview image sizes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailablePostviewImageSize",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string | Current postview image size (See Postview image size parameters of Parameter description) |
| 1 | string-array | A list of available postview image sizes (See Postview image size parameters of Parameter description) |

JSON Example

```
{
  "result": [
    "Original",
    ["Original", "2M"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setPostviewImageSize](#)
- [getPostviewImageSize](#)
- [getSupportedPostviewImageSize](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "postviewImageSize" object in "[getEvent](#)" callback.

Movie file format

setMovieFileFormat

Overview

This API provides a function to set a value of movie file format.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-------------------|--------|--|
| "movieFileFormat" | string | Movie file format (See Movie file format parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setMovieFileFormat",
  "params": [
    {
      "movieFileFormat": "XAVC S"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getMovieFileFormat](#)
- [getSupportedMovieFileFormat](#)
- [getAvailableMovieFileFormat](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "movieFileFormat" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getMovieFileFormat

Overview

This API provides a function to get current movie file format.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getMovieFileFormat",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------------|--------|--|
| "movieFileFormat" | string | Current movie file format (See Movie file format parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "movieFileFormat": "XAVC S"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieFileFormat](#)
- [getSupportedMovieFileFormat](#)
- [getAvailableMovieFileFormat](#)

Special note (details)

None in particular.

getSupportedMovieFileFormat

Overview

This API provides a function to get the supported movie file formats.

The client should use "[getAvailableMovieFileFormat](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedMovieFileFormat",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported movie file formats (See Movie file format parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "MP4",
        "XAVC S"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieFileFormat](#)
- [getMovieFileFormat](#)
- [getAvailableMovieFileFormat](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableMovieFileFormat

Overview

This API provides a function to get current movie file format and the available movie file formats at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableMovieFileFormat",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------------|--------------|---|
| "movieFileFormat" | string | Current movie file format (See Movie file format parameters of Parameter description) |
| "candidate" | string-array | A list of available movie file formats (See Movie file format parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "MP4",
        "XAVC S"
      ],
      "movieFileFormat": "XAVC S"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieFileFormat](#)
- [getMovieFileFormat](#)
- [getSupportedMovieFileFormat](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "movieFileFormat" object in "[getEvent \(v1.2\)](#)" callback.

Movie quality

setMovieQuality

Overview

This API provides a function to set a value of movie quality.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Movie quality (See Movie quality parameter of Parameter description) |

JSON Example

```
{
  "method": "setMovieQuality",
  "params": ["HQ"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getMovieQuality](#)
- [getSupportedMovieQuality](#)
- [getAvailableMovieQuality](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "movieQuality" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getMovieQuality

Overview

This API provides a function to get current movie quality.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getMovieQuality",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current movie quality (See Movie quality parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "HQ"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieQuality](#)
- [getSupportedMovieQuality](#)
- [getAvailableMovieQuality](#)

Special note (details)

None in particular.

getSupportedMovieQuality

Overview

This API provides a function to get the supported movie qualities.

The client should use "[getAvailableMovieQuality](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedMovieQuality",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string-array | A list of supported movie qualities (See Movie quality parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["HQ", "STD", "VGA", "SLOW", "SSLOW"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieQuality](#)
- [getMovieQuality](#)
- [getAvailableMovieQuality](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableMovieQuality

Overview

This API provides a function to get current movie quality and the available movie qualities at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableMovieQuality",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string | Current movie quality (See Movie quality parameter of Parameter description) |
| 1 | string-array | A list of available movie qualities (See Movie quality parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "HQ",
    ["HQ", "STD", "VGA", "SLOW", "SSLOW"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setMovieQuality](#)
- [getMovieQuality](#)
- [getSupportedMovieQuality](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "movieQuality" object in [getEvent](#) callback.

Steady mode

setSteadyMode

Overview

This API provides a function to set a value of steady mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Steady mode (See Steady mode parameter of Parameter description) |

JSON Example

```
{
  "method": "setSteadyMode",
  "params": ["off"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSteadyMode](#)
- [getSupportedSteadyMode](#)
- [getAvailableSteadyMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "steadyMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getSteadyMode

Overview

This API provides a function to get current steady mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSteadyMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current steady mode (See Steady mode parameter of Parameter description) |

JSON Example

```
{
  "result": ["off"],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSteadyMode](#)
- [getSupportedSteadyMode](#)
- [getAvailableSteadyMode](#)

Special note (details)

None in particular.

getSupportedSteadyMode

Overview

This API provides a function to get the supported steady modes.

The client should use "[getAvailableSteadyMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedSteadyMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported steady modes (See Steady mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["off", "on"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSteadyMode](#)
- [getSteadyMode](#)
- [getAvailableSteadyMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableSteadyMode

Overview

This API provides a function to get current steady mode and the available steady modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableSteadyMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current steady mode (See Steady mode parameter of Parameter description) |
| 1 | string-array | A list of available steady modes (See Steady mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "off",
    ["off", "on"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSteadyMode](#)
- [getSteadyMode](#)
- [getSupportedSteadyMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "steadyMode" object in "[getEvent](#)" callback.

View angle

setViewAngle

Overview

This API provides a function to set a value of view angle.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | View angle (See View angle parameter of Parameter description) |

JSON Example

```
{
  "method": "setViewAngle",
  "params": [120],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getViewAngle](#)
- [getSupportedViewAngle](#)
- [getAvailableViewAngle](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "viewAngle" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getViewAngle

Overview

This API provides a function to get current view angle.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getViewAngle",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Current view angle (See View angle parameter of Parameter description) |

JSON Example

```
{
  "result": [
    120
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setViewAngle](#)
- [getSupportedViewAngle](#)
- [getAvailableViewAngle](#)

Special note (details)

None in particular.

getSupportedViewAngle

Overview

This API provides a function to get the supported view angles.

The client should use "[getAvailableViewAngle](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedViewAngle",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|--|
| 0 | integer-array | A list of supported view angles (See View angle parameter of Parameter description) |

JSON Example

```
{
  "result": [
    [120,170]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setViewAngle](#)
- [getViewAngle](#)
- [getAvailableViewAngle](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableViewAngle

Overview

This API provides a function to get current view angle and the available view angles at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableViewAngle",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------------|--|
| 0 | integer | Current view angle (See View angle parameter of Parameter description) |
| 1 | integer-array | A list of available view angles (See View angle parameter of Parameter description) |

JSON Example

```
{
  "result": [
    120,
    [120,170]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setViewAngle](#)
- [getViewAngle](#)
- [getSupportedViewAngle](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "viewAngle" object in "[getEvent](#)" callback.

Scene selection

setSceneSelection

Overview

This API provides a function to set a value of scene selection.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|---------|--------|--|
| "scene" | string | Scene selection (See Scene selection parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setSceneSelection",
  "params": [
    {
      "scene": "Under Water"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSceneSelection](#)
- [getSupportedSceneSelection](#)
- [getAvailableSceneSelection](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "sceneSelection" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getSceneSelection

Overview

This API provides a function to get current scene selection.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSceneSelection",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|---------|--------|--|
| "scene" | string | Current scene selection (See Scene selection parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "scene": "Normal"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSceneSelection](#)
- [getSupportedSceneSelection](#)
- [getAvailableSceneSelection](#)

Special note (details)

None in particular.

getSupportedSceneSelection

Overview

This API provides a function to get the supported scene selections.

The client should use "[getAvailableSceneSelection](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedSceneSelection",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported scene selections (See Scene selection parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "Normal",
        "Under Water"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSceneSelection](#)
- [getSceneSelection](#)
- [getAvailableSceneSelection](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableSceneSelection

Overview

This API provides a function to get current scene selection and the available scene selections at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableSceneSelection",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "scene" | string | Current scene selection (See Scene selection parameters of Parameter description) |
| "candidate" | string-array | A list of available scene selections (See Scene selection parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "scene": "Under Water",
      "candidate": [
        "Normal",
        "Under Water"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setSceneSelection](#)
- [getSceneSelection](#)
- [getSupportedSceneSelection](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "sceneSelection" object in "[getEvent \(v1.2\)](#)" callback.

Color setting

setColorSetting

Overview

This API provides a function to set a value of color setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|----------------|--------|--|
| "colorSetting" | string | Color setting (See Color setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setColorSetting",
  "params": [
    {
      "colorSetting": "Vivid"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getColorSetting](#)
- [getSupportedColorSetting](#)
- [getAvailableColorSetting](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "colorSetting" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getColorSetting

Overview

This API provides a function to get current color setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getColorSetting",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|----------------|--------|--|
| "colorSetting" | string | Current color setting (See Color setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "colorSetting": "Vivid"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setColorSetting](#)
- [getSupportedColorSetting](#)
- [getAvailableColorSetting](#)

Special note (details)

None in particular.

getSupportedColorSetting

Overview

This API provides a function to get the supported color settings.

The client should use "[getAvailableColorSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedColorSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported color settings (See Color setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "Neutral",
        "Vivid"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setColorSetting](#)
- [getColorSetting](#)
- [getAvailableColorSetting](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableColorSetting

Overview

This API provides a function to get current color setting and the available color settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableColorSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|----------------|--------------|---|
| "colorSetting" | string | Current color setting (See Color setting parameters of Parameter description) |
| "candidate" | string-array | A list of available color settings (See Color setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "colorSetting": "Vivid",
      "candidate": [
        "Neutral",
        "Vivid"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setColorSetting](#)
- [getColorSetting](#)
- [getSupportedColorSetting](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "colorSetting" object in "[getEvent \(v1.2\)](#)" callback.

Interval time

setIntervalTime

Overview

This API provides a function to set a value of interval time.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-------------------|--------|---|
| "intervalTimeSec" | string | Interval time (unit: second) (See Interval time parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setIntervalTime",
  "params": [
    {
      "intervalTimeSec": "10"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getIntervalTime](#)
- [getSupportedIntervalTime](#)
- [getAvailableIntervalTime](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "intervalTime" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getIntervalTime

Overview

This API provides a function to get current interval time.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getIntervalTime",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-------------------|--------|---|
| "intervalTimeSec" | string | Current interval time (unit: second) (See Interval time parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "intervalTimeSec": "10"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIntervalTime](#)
- [getSupportedIntervalTime](#)
- [getAvailableIntervalTime](#)

Special note (details)

None in particular.

getSupportedIntervalTime

Overview

This API provides a function to get the supported interval times.

The client should use "[getAvailableColorSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedIntervalTime",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|--|
| "candidate" | string-array | A list of supported interval times (unit: second) (See Interval time parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "1",
        "2",
        "5",
        "10",
        "30",
        "60"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIntervalTime](#)
- [getIntervalTime](#)
- [getAvailableIntervalTime](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableIntervalTime

Overview

This API provides a function to get current interval time and the available interval times at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableIntervalTime",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------------|--------------|--|
| "intervalTimeSec" | string | Current interval time (unit: second) (See Interval time parameters of Parameter description) |
| "candidate" | string-array | A list of available interval times (unit: second) (See Interval time parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "intervalTimeSec": "10",
      "candidate": [
        "1",
        "2",
        "5",
        "10",
        "30",
        "60"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setIntervalTime](#)
- [getIntervalTime](#)
- [getSupportedIntervalTime](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "intervalTime" object in "[getEvent \(v1.2\)](#)" callback.

Flip setting

setFlipSetting

Overview

This API provides a function to set a value of flip setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|--------|--------|--|
| "flip" | string | Flip setting (See Flip setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setFlipSetting",
  "params": [
    {
      "flip": "On"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getFlipSetting](#)
- [getSupportedFlipSetting](#)
- [getAvailableFlipSetting](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "flipSetting" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getFlipSetting

Overview

This API provides a function to get current flip setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getFlipSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|--------|--------|--|
| "flip" | string | Current flip setting (See Flip setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "flip": "On"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlipSetting](#)
- [getSupportedFlipSetting](#)
- [getAvailableFlipSetting](#)

Special note (details)

None in particular.

getSupportedFlipSetting

Overview

This API provides a function to get the supported flip settings.

The client should use "[getAvailableFlipSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedFlipSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported flip settings (See Flip setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlipSetting](#)
- [getFlipSetting](#)
- [getAvailableFlipSetting](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableFlipSetting

Overview

This API provides a function to get current flip setting and the available flip settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableFlipSetting",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "flip" | string | Current flip setting (See Flip setting parameters of Parameter description) |
| "candidate" | string-array | A list of available flip settings (See Flip setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "flip": "On",
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setFlipSetting](#)
- [getFlipSetting](#)
- [getSupportedFlipSetting](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "flipSetting" object in "[getEvent \(v1.2\)](#)" callback.

TV color system

setTvColorSystem

Overview

This API provides a function to set a value of TV color system.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-----------------|--------|--|
| "tvColorSystem" | string | TV color system (See TV color system parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setTvColorSystem",
  "params": [
    {
      "tvColorSystem": "NTSC"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getTvColorSystem](#)
- [getSupportedTvColorSystem](#)
- [getAvailableTvColorSystem](#)

Special note (details)

Note that the server will restart after setting the TV color system via this API.

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "tvColorSystem" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getTvColorSystem

Overview

This API provides a function to get current TV color system.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getTvColorSystem",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|-----------------|--------|--|
| "tvColorSystem" | string | Current TV color system (See TV color system parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "tvColorSystem": "NTSC"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTvColorSystem](#)
- [getSupportedTvColorSystem](#)
- [getAvailableTvColorSystem](#)

Special note (details)

None in particular.

getSupportedTvColorSystem

Overview

This API provides a function to get the supported TV color systems.

The client should use "[getAvailableFlipSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedTvColorSystem",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported TV color systems (See TV color system parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "NTSC",
        "PAL"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTvColorSystem](#)
- [getTvColorSystem](#)
- [getAvailableTvColorSystem](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableTvColorSystem

Overview

This API provides a function to get current TV color system and the available TV color systems at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableTvColorSystem",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-----------------|--------------|---|
| "tvColorSystem" | string | Current TV color system (See TV color system parameters of Parameter description) |
| "candidate" | string-array | A list of available TV color systems (See TV color system parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "tvColorSystem": "NTSC",
      "candidate": [
        "NTSC",
        "PAL"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setTvColorSystem](#)
- [getTvColorSystem](#)
- [getSupportedTvColorSystem](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "tvColorSystem" object in "[getEvent \(v1.2\)](#)" callback.

Camera setup

startRecMode

Overview

This API provides a function to set up camera for shooting function. Some camera models need this API call before starting liveview, capturing still image, recording movie, or accessing all other camera shooting functions.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "startRecMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [stopRecMode](#)

Special note (details)

Some camera models need this API call before starting liveview, capturing still image, recording movie, or accessing all other camera shooting functions. The client must check if the server needs this API call. The check can be done by checking the availability of this API in "[getAvailableApiList](#)" or "[getMethodTypes](#)" callback. The client should call this API just once before accessing camera shooting functions if the server needs.

stopRecMode

Overview

This API provides a function to stop shooting functions.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "stopRecMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [startRecMode](#)

Special note (details)

None in particular.

Camera function

setCameraFunction

Overview

This API provides a function to set a value of camera function.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Camera function (See Camera function parameter of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setCameraFunction",
  "params": [
    "Remote Shooting"
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getCameraFunction](#)
- [getSupportedCameraFunction](#)
- [getAvailableCameraFunction](#)

Special note (details)

The execution of this API may take time. The client should check the parameter "cameraFunctionResult" of "[getEvent \(v1.0\)](#)" to get result of setting camera function. The callback of this API is for starting execution, not for getting result of setting.

When the client switches "Remote Shooting" to "Contents Transfer" via this API and switching is done successfully, the server status will change as follows.

Camera status : "IDLE" -> "ContentsTransfer"

The client should check the parameter "cameraStatus" of "[getEvent](#)" to get the camera status. The client should run initial sequence for transferring images after switching to "ContentsTransfer" status. The client should check if the media is inserted in the camera via "[getStorageInformation](#)" API.

When the client switches "Contents Transfer" to "Remote Shooting" via this API, the server status will change as follows. The client should restart the liveview after getting camera function back to "Remote Shooting".

Camera status : "ContentsTransfer" -> "IDLE"

Note that the server may take time to switch function and the client should monitor camera status.

The camera needs to prepare for the next shot, therefore it may take time to start capturing after changing to "IDLE". But the camera will start capturing as soon as possible.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

getCameraFunction

Overview

This API provides a function to get current camera function.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getCameraFunction",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current camera function (See Camera function parameter of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    "Remote Shooting"
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setCameraFunction](#)
- [getSupportedCameraFunction](#)
- [getAvailableCameraFunction](#)

Special note (details)

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

getSupportedCameraFunction

Overview

This API provides a function to get the supported camera function s.

The client should use "[getAvailableCameraFunction](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedCameraFunction",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported camera functions (See Camera function parameter of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    [
      "Remote Shooting",
      "Contents Transfer"
    ]
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setCameraFunction](#)
- [getCameraFunction](#)
- [getAvailableCameraFunction](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

getAvailableCameraFunction

Overview

This API provides a function to get current camera function and the available camera functions at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableCameraFunction",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current camera function (See Camera function parameter of Parameter description) |
| 1 | string-array | A list of available camera functions (See Camera function parameter of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    "Remote Shooting",
    [
      "Remote Shooting",
      "Contents Transfer"
    ]
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setCameraFunction](#)
- [getCameraFunction](#)
- [getSupportedCameraFunction](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "cameraFunction" object in "[getEvent](#)" callback.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

Transferring images

getSchemeList

Overview

This API provides the list of schemes that device can handle. In Camera Remote API, standard URI structure, as defined by RFC 3986, is used for representing device's resources. Schemes are used to refer to device resources. URI is provided from the server.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSchemeList",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| order | key | type | explanation |
|-------|----------|--------------|-------------------------|
| 0 | | object-array | A list of scheme names. |
| | "scheme" | string | Schema name. |

JSON Example

```
{
  "result": [
    [
      {
        "scheme": "storage"
      }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSourceList](#)
- [getContentCount \(v1.2\)](#)
- [getContentList \(v1.3\)](#)

Special note (details)

The camera supports only "storage" as a scheme.

Please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#) about how to use this API.

getSourceList

Overview

This API provides the list of sources under the scheme. The source is included in URI to access stored contents in the camera. The camera supports specific source.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request**Elements of "params"**

| key | type | explanation |
|----------|--------|-------------|
| "scheme" | string | Scheme name |

JSON Example

```
{
  "method": "getSourceList",
  "params": [
    {
      "scheme": "storage"
    }
  ],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|-------|----------|--------------|---|
| 0 | | object-array | A list of source names under specific scheme. |
| | "source" | string | Source name. |

JSON Example

```
{
  "result": [
    [
      {
        "source": "storage:memoryCard1"
      }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSchemeList](#)
- [getContentCount \(v1.2\)](#)
- [getContentList \(v1.3\)](#)

Special note (details)

The camera supports only "storage:memoryCard1" as a source.

Please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#) about how to use this API.

getContentCount (v1.2)

Overview

This API provides a function to get content count under specific URI.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.2 |

Request**Elements of "params"**

| key | type | explanation |
|----------|--------------|---|
| "uri" | string | URI to identify the content. |
| "type" | string-array | Optional parameter to narrow down result within specified URI in the request. Following values are defined. Only for "date" view. Not available if "target" parameter is "all". "still" - Still image. "movie_mp4" - MP4 movie. "movie_xavcs" - XAVC S movie. null - Not specified. |
| "target" | string | Optional parameter to widen result within specified URI in the request. Following values are defined. "all" - Return the number of all contents. |
| "view" | string | View type "date" - Date view "flat" - Flat view |

JSON Example

```
{
  "method": "getContentCount",
  "params": [
    {
      "uri": "storage:memoryCard1",
      "target": "all",
      "view": "date"
    }
  ],
  "id": 1,
  "version": "1.2"
}
```

Response**Elements of "result"**

| key | type | explanation |
|---------|--------|---------------------------------------|
| "count" | string | The number of contents under the URI. |

JSON Example

```
{
  "result": [
    {
      "count": 7
    }
  ],
  "id": 1
}
```

Error Codes

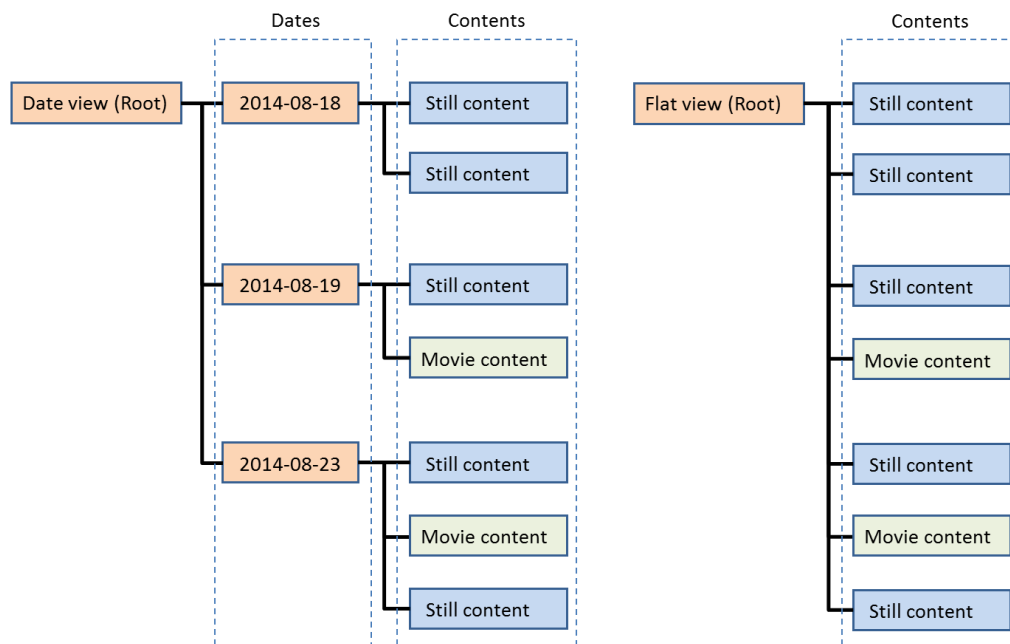
See [Status code & Error](#)

Related API

- [getSchemeList](#)
- [getSourceList](#)
- [getContentList \(v1.3\)](#)

Special note (details)

The transferring images function supports two types of view to retrieve count/list of contents. One in "date" view, the other is "flat" view. Please see following figure.



The "date" view has date folder structure and the contents belong to each date folder. The Camera Remote API handles date folder as the content and the client can get count of dates via "[getContentCount \(v1.2\)](#)" API as well. On the other hand, the "flat" view has no folder structure.

[Number of all contents in "date" view or "flat" view]

Request Parameters

| key | comment | example |
|----------|---|-----------------------|
| "uri" | Specify the response of " getSchemeList " | "storage:memoryCard1" |
| "type" | Not available | |
| "target" | Specify "all" | "all" |
| "view" | Specify view type | "date" or "flat" |

Request JSON Example

```

{
  "method": "getContentCount",
  "params": [
    {
      "uri": "storage:memoryCard1",
      "target": "all",
      "view": "date"
    }
  ],
  "id": 1,
  "version": "1.2"
}

```

[Number of date folders in "date" view]

Request Parameters

| key | comment | example |
|----------|---|-----------------------|
| "uri" | Specify the response of " getSchemeList " | "storage:memoryCard1" |
| "type" | Not available | |
| "target" | Not specified | |
| "view" | Specify view type | "date" or "flat" |

Request JSON Example

```
{
  "method": "getContentCount",
  "params": [
    {
      "uri": "storage:memoryCard1",
      "view": "date"
    }
  ],
  "id": 1,
  "version": "1.2"
}
```

[Number of contents in a date folder in "date" view]

Request Parameters

| key | comment | example |
|----------|--|---|
| "uri" | Specify the response of " getContentList (v1.3) " for date folder. | "storage:memoryCard1?path=2014-08-18" |
| "type" | Specify list of types if the client wants to filter. | ["still", "movie_mp4", "movie_xavcs"], null ... |
| "target" | Not specified | |
| "view" | Specify view type | "date" |

Request JSON Example

```
{
  "method": "getContentCount",
  "params": [
    {
      "uri": "storage:memoryCard1?path=2014-08-18",
      "type": [
        "still",
        "movie_mp4"
      ],
      "view": "date"
    }
  ],
  "id": 1,
  "version": "1.2"
}
```

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

getContentList (v1.3)

Overview

This API provides a function to get content list under specific URI.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.3 |

Request**Elements of "params"**

| key | type | explanation |
|---------|--------------|---|
| "uri" | string | URI to identify the content. |
| "stIdx" | integer | Start index to get list items. |
| "cnt" | integer | Count of the maximum number of items that can be listed, starting from "stIdx". Maximum number is 100. |
| "type" | string-array | Optional parameter to narrow down result within specified URI in the request. Following values are defined. Only for "date" view. "still" - Still image. "movie_mp4" - MP4 movie. "movie_xavcs" - XAVC S movie. null - Not specified. |
| "view" | string | View type "date" - Date view "flat" - Flat view |
| "sort" | string | Sort type "ascending" - Ascending "descending" - Descending "" - Not specified |

JSON Example

```
{
  "method": "getContentList",
  "params": [
    {
      "uri": "storage:memoryCard1",
      "stIdx": 0,
      "cnt": 50,
      "view": "date",
      "sort": ""
    }
  ],
  "id": 1,
  "version": "1.3"
}
```

Response

Elements of "result"

An array of objects composed by following pairs.

| key | type | explanation |
|------------------|--------------|--|
| "uri" | string | URI to identify the content. |
| "title" | string | Title of this content to be recognized by user. |
| "content" | object | Content information |
| "original" | object-array | List of original content information |
| "url" | string | Original content URL |
| "fileName" | string | File name |
| "stillObject" | string | Still object type "jpeg" - jpeg image "raw" - raw image "mpo" - mpo image "" - Unknown |
| "largeUrl" | string | Resized content URL (2M pixel scale) |
| "smallUrl" | string | Resized content URL (VGA scale) |
| "thumbnailUrl" | string | Thumbnail URL |
| "createdTime" | string | Created Time (ISO8601) |
| "contentKind" | string | The kind of content "still" - Still image "movie_mp4" - MP4 movie "movie_xavcs" - XAVC S movie "directory" - directory "" - Unknown |
| "folderNo" | string | Folder number |
| "fileNo" | string | File number |
| "isPlayable" | string | Playable status on the camera display "true" - The content is playable "false" - The content is not playable "" - Unknown |
| "isBrowsable" | string | Browsable status "true" - The content is browsable "false" - The content is not browsable "" - Unknown |
| "isProtected" | string | Protect status "true" - The content is protected "false" - The content is not protected "" - Unknown |
| "remotePlayType" | string-array | Remote play type "simpleStreaming" - Simple streaming null - Unknown |

JSON Example

```
{
  "result": [
    [
      {
        "uri": "storage:memoryCard1?path=2014-08-18",
        "title": "20140818",
        "content": null,
        "createdTime": "",
        "contentKind": "directory",
        "folderNo": "",
        "fileNo": "",
        "isPlayable": "false",
        "isBrowsable": "true",
        "isProtected": "",
        "remotePlayType": null
      }
    ]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getSourceList](#)
- [getSchemeList](#)
- [getContentCount \(v1.2\)](#)

Special note (details)

Please see "[getContentCount \(v1.2\)](#)" about basic conception of transferring images function and contents structure.

Regarding how to specify "uri", "type" and "view" in request parameters, please see "[getContentCount \(v1.2\)](#)".

When the client gets list of dates in "date" view, the camera returns the URI of date to specify date folder as "uri" of the response. When the client gets list of contents in a date folder in "date" view, the camera returns the URI of still or movie as "uri" of the response.

The client can download images via "url" in "original" object, "largeUrl", "smallUrl" and "thumbnailUrl". These URLs can be empty string ("") if the content doesn't support the size.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#). And see following response parameters and JSON examples.

[List of date folders in "date" view]

Response Parameters

| key | comment | example |
|------------------|--------------------------------|---------------------------------------|
| "uri" | URI indicating a date folder | "storage:memoryCard1?path=2014-08-18" |
| "title" | Name of date folder (YYYYMMDD) | "20140818" |
| "content" | Not available | null |
| "createdTime" | Not available | "" |
| "contentKind" | Date folder | "directory" |
| "folderNo" | Not available | "" |
| "fileNo" | Not available | "" |
| "isPlayable" | Not playable | "false" |
| "isBrowsable" | Browsable | "true" |
| "isProtected" | Unknown | "" |
| "remotePlayType" | Not available | null |

Response JSON Example

```
{
  "result": [
    [
      {
        "uri": "storage:memoryCard1?path=2014-08-18",
        "title": "20140818",
        "content": null,
        "createdTime": "",
        "contentKind": "directory",
        "folderNo": "",
        "fileNo": "",
        "isPlayable": "false",
        "isBrowsable": "true",
        "isProtected": "",
        "remotePlayType": null
      }
    ]
  ],
  "id": 1
}
```


[List of contents (still image)]

Response Parameters

| key | comment | example |
|------------------|--|--|
| "uri" | URI indicating still image content | "image:content?contentId=XXXXXX XXXX" |
| "title" | Not available | "" |
| "content" | Includes "original", "smallUrl", "largeUrl" and "thumbnailUrl" typically. Some cameras return not only "jpeg" but also "raw" in "original" object. | Please see JSON example. |
| "createdTime" | Created time of still image | "2014-08-18T12:34:56+09:00" |
| "contentKind" | Still image | "still" |
| "folderNo" | Indicates folder number in which still image file is stored. (Ex. 100MSDCF) | "100" |
| "fileNo" | Indicates file number of still image file in memory card. (Ex. DSC00001.JPG) | "0001" |
| "isPlayable" | Not playable (There is no way to play still image on the camera). | "false" |
| "isBrowsable" | Not browsable. | "false" |
| "isProtected" | Unknown | "" |
| "remotePlayType" | Not available | null |

Response JSON Example

```
{
  "result": [
    [
      {
        "uri": "image:content?contentId=XXXXXXXXXX",
        "title": "",
        "content": {
          "original": [
            {
              "fileName": "DSC00001.JPG",
              "stillObject": "jpeg",
              "url": "http://ip:port/contentstransfer/orgjpeg/xxxxxxxx-xxxxxxxx"
            }
          ],
          "smallUrl": "http://ip:port/contentstransfer/vga/xxxxxxxx-xxxxxxxx",
          "largeUrl": "http://ip:port/contentstransfer/scn/xxxxxxxx-xxxxxxxx",
          "thumbnailUrl": "http://ip:port/contentstransfer/thumb/xxxxxxxx-xxxxxxxx"
        },
        "createdTime": "2014-08-18T12:34:56+09:00",
        "contentKind": "still",
        "folderNo": "100",
        "fileNo": "0001",
        "isPlayable": "false",
        "isBrowsable": "false",
        "isProtected": "",
        "remotePlayType": null
      }
    ]
  ],
  "id": 1
}
```

[List of contents (movie)]

Response Parameters

| key | comment | example |
|------------------|---|--|
| "uri" | URI indicating movie | "video:content?contentId=XXXXXXX XXX" |
| "title" | Not available | "" |
| "content" | Includes "original" and "thumbnailUrl" typically. | Please see JSON example. |
| "createdTime" | Created time of movie | "2014-08-18T12:34:56+09:00" |
| "contentKind" | MP4 movie or XAVC S movie | "movie_mp4", "movie_xavcs" |
| "folderNo" | Indicates folder number in which movie file is stored. (Ex. 100ANV01) | "100" |
| "fileNo" | Indicates file number of movie file in memory card. (Ex. MAH00002.MP4) | "0002" |
| "isPlayable" | Not playable (There is no way to play movie on the camera. Remote playback is available.) | "false" |
| "isBrowsable" | Not browsable. | "false" |
| "isProtected" | Unknown | "" |
| "remotePlayType" | Indicates remote playback if possible | ["simpleStreaming"] |

Response JSON Example

```
{
  "result": [
    [
      {
        "uri": "video:content?contentId=XXXXXXXXXX",
        "title": "",
        "content": {
          "original": [
            {
              "fileName": "MAH00002.MP4",
              "stillObject": "",
              "url": "http://ip:port/contentstransfer/org/xxxxxxx-xxxxxxx"
            }
          ],
          "smallUrl": "",
          "largeUrl": "",
          "thumbnailUrl": "http://ip:port/contentstransfer/thumb/xxxxxxx-xxxxxxx"
        },
        "createdTime": "2014-08-18T12:34:56+09:00",
        "contentKind": "movie_mp4",
        "folderNo": "100",
        "fileNo": "0002",
        "isPlayable": "false",
        "isBrowsable": "false",
        "isProtected": "",
        "remotePlayType": [
          "simpleStreaming"
        ]
      }
    ]
  ],
  "id": 1
}
```

Remote playback

setStreamingContent

Overview

This API provides a function to set streaming content for remote playback.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|------------------|--------|---|
| "uri" | string | URI of content. |
| "remotePlayType" | string | Remote playback type. "simpleStreaming" - Simple streaming "" - unknown |

JSON Example

```
{
  "id": 1,
  "method": "setStreamingContent",
  "params": [
    {
      "remotePlayType": "simpleStreaming",
      "uri": "video:content?contentId=XXXXXXXXXX"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|---------------|--------|---|
| "playbackUrl" | string | URL for streaming. Refer to Streaming data format . |

JSON Example

```
{
  "result": [
    {
      "playbackUrl": "http://ip:port/streaming/playbackstream"
    }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getContentList \(v1.3\)](#)
- [startStreaming](#)
- [pauseStreaming](#)
- [seekStreamingPosition](#)
- [stopStreaming](#)
- [requestToNotifyStreamingStatus](#)

Special note (details)

The remote playback allows the client to play movie streaming from the camera and control streaming via APIs.

The camera supports only "simpleStreaming" as "remotePlayType". The client can get "remotePlayType" parameter from the response of ["getContentList \(v1.3\)"](#) and remote playback will be available if the response of ["getContentList \(v1.3\)"](#) for the content includes "simpleStreaming".

To download from the URL of streaming and parse the data stream, please refer to [Streaming data format](#). The client can display the playback data and playback time position on its UI.

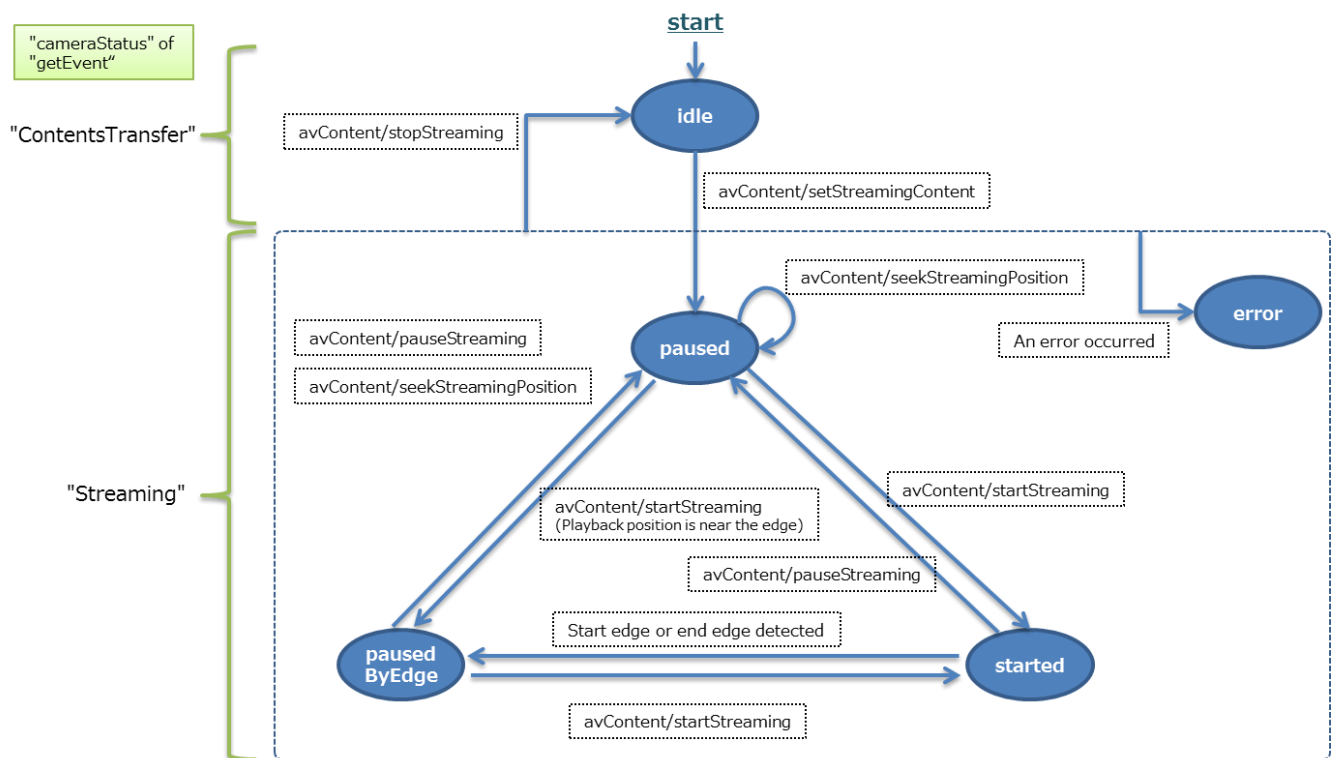
[Preparation]

The client should switch the camera function from "Remote Shooting" to "Contents Transfer" using ["setCameraFunction"](#) API. The client can check if the server supports the remote playback via ["getMethodTypes"](#) API of "avContent" API service.

[Status transition]

The client can check the camera status via "cameraStatus" object of ["getEvent"](#) API response. After calling ["setStreamingContent"](#) API, the camera status will be changed from "ContentsTransfer" to "Streaming".

The client can monitor transition of detailed status "streaming status" for remote playback via ["requestToNotifyStreamingStatus"](#) API. The "streaming status" will be changed based on following diagram and "streaming status" is sub-status for "cameraStatus" of ["getEvent"](#).



Streaming status transition for remote playback

The APIs related to remote playback are available only when "cameraStatus" of ["getEvent"](#) is "ContentsTransfer" or "Streaming". Please see each API specifications for details. This ["setStreamingContent"](#) API can be called in only "ContentsTransfer" camera status.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

startStreaming

Overview

This API provides a function to start streaming.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "startStreaming",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStreamingContent](#)
- [pauseStreaming](#)
- [seekStreamingPosition](#)
- [stopStreaming](#)
- [requestToNotifyStreamingStatus](#)

Special note (details)

This API can be called only when "cameraStatus" of "[getEvent](#)" is "Streaming". This API is related to "Streaming status". See [diagram of streaming status transition](#).

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

pauseStreaming

Overview

This API provides a function to pause streaming.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "id": 1,
  "method": "pauseStreaming",
  "params": [],
  "version": "1.0"
}
```

Response**Elements of "result"**

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStreamingContent](#)
- [startStreaming](#)
- [seekStreamingPosition](#)
- [stopStreaming](#)
- [requestToNotifyStreamingStatus](#)

Special note (details)

This API can be called only when "cameraStatus" of "[getEvent](#)" is "Streaming". This API is related to "Streaming status". See [diagram of streaming status transition](#).

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

seekStreamingPosition

Overview

This API provides a function to seek streaming position while streaming content.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|----------------|---------|------------------------------------|
| "positionMsec" | integer | Seek position (unit: millisecond). |

JSON Example

```
{
  "method": "seekStreamingPosition",
  "params": [
    {
      "positionMsec": 1500
    }
  ],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStreamingContent](#)
- [startStreaming](#)
- [pauseStreaming](#)
- [stopStreaming](#)
- [requestToNotifyStreamingStatus](#)

Special note (details)

Some camera models don't support this API, therefore the client should check supported APIs via ["getMethodTypes"](#) API of "avContent" API service.

This API can be called only when "cameraStatus" of ["getEvent"](#) is "Streaming". This API is related to "Streaming status". See [diagram of streaming status transition](#).

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

The client can get duration of movie via [Streaming data format](#) and specify seek position in millisecond within the duration.

stopStreaming

Overview

This API provides a function to stop streaming.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "stopStreaming",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStreamingContent](#)
- [startStreaming](#)
- [pauseStreaming](#)
- [seekStreamingPosition](#)
- [requestToNotifyStreamingStatus](#)

Special note (details)

This API can be called only when "cameraStatus" of "[getEvent](#)" is "Streaming". When the client calls this API, the camera status will be changed from "Streaming" to "ContentsTransfer". It means that the server finishes the remote playback function.

This API is related to "Streaming status". See [diagram of streaming status transition](#).

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

requestToNotifyStreamingStatus

Overview

This API provides a function to get streaming status from the server.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.0 |

Request**Elements of "params"**

| key | type | explanation |
|-----------|---------|---|
| "polling" | boolean | Long polling flag true: Callback when timeout or change point detection. false: Callback immediately. |

JSON Example

```
{
  "method": "requestToNotifyStreamingStatus",
  "params": [
    {
      "polling": true
    }
  ],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| key | type | explanation |
|----------|--------|--|
| "status" | string | Streaming status "idle" - No playback content set "paused" - Paused "started" - Started "pausedByEdge" - Paused by start edge/end edge "error" - Error occurred "" - Empty string indicates invalid status |
| "factor" | string | Factor of streaming status "startEdge" - Paused by start edge. This parameter is only for "pausedByEdge" status. "endEdge" - Paused by end edge. This parameter is only for "pausedByEdge" status. "fileError" - Cannot play movie with file problem. This parameter is only for "error" status. "mediaError" - Cannot play movie with media problem. This parameter is only for "error" status. "otherError" - Other error. This parameter is only for "error" status. "" - Unknown |

JSON Example

```
{
  "result": [
    {
      "status": "started",
      "factor": ""
    }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setStreamingContent](#)
- [startStreaming](#)
- [pauseStreaming](#)
- [seekStreamingPosition](#)
- [stopStreaming](#)

Special note (details)

The client can handle this API like "[getEvent](#)" API of "camera" API service. This API can be called only when "cameraStatus" of "[getEvent](#)" is "ContentsTransfer" or "Streaming". This API is related to "Streaming status". See [diagram of streaming status transition](#).

The purpose of this API is sending the streaming status from the server actively. When the client calls some APIs, the client can monitor the status updates from the camera by using this API.

If this API is executed, the server times out or does not return a response until the parameter is updated. (If the input parameter "polling" is false, the server replies immediately.) If the client gets the response, the client should execute this API immediately. When the execution of this API is unsuccessful or timeout, the server will notify error in response.

In case this API is called with "polling=true", the server will not return a response until the server is ready, and streaming status has changed in the server. So, by calling with "polling=true", the client can recognize the timing of a change in the parameter of the server.

In case this API is called with "polling=false", the server will return a response immediately. So, by calling with "polling=false", the client can recognize the server state snapshot at the time. The client can call this API with "polling=false" at the beginning. (The information obtained in the response may be useful for the client to build UI layouts and so on.)

Delete contents

deleteContent (v1.1)

Overview

This API provides a function to delete contents.

| | |
|--------------|----------------------------|
| Endpoint URL | <ActionList_URL>/avContent |
| Version | 1.1 |

Request

Elements of "params"

| key | type | explanation |
|-------|--------------|---|
| "uri" | string-array | List of URI to delete. Maximum number is 100. |

JSON Example

```
{
  "id": 1,
  "method": "deleteContent",
  "params": [
    {
      "uri": [
        "image:content?contentId=XXXXXXXXXX",
        "video:content?contentId=XXXXXXXXXX"
      ]
    }
  ],
  "version": "1.1"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getContentList \(v1.3\)](#)

Special note (details)

This API can be called only when "cameraStatus" of "[getEvent](#)" is "ContentsTransfer". After calling this API, the camera status will be changed to "Deleting". And the status will be back to "ContentsTransfer" after deletion is done.

The client can specify the content URI (not URL) of still image or movie. The client gets the URIs from the response of "[getContentList \(v1.3\)](#)". If the camera cannot complete the deletion for some reason, for example some contents are protected, the response will return "41003" error (Some content could not be deleted.). When "41003" error is returned, the camera skips the contents the camera cannot delete.

IR remote control

setInfraredRemoteControl

Overview

This API provides a function to set a value of IR remote control setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|-------------------------|--------|--|
| "infraredRemoteControl" | string | IR remote control setting (See IR remote control setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "method": "setInfraredRemoteControl",
  "params": [
    {
      "infraredRemoteControl": "Off"
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getInfraredRemoteControl](#)
- [getSupportedInfraredRemoteControl](#)
- [getAvailableInfraredRemoteControl](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "infraredRemoteControl" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getInfraredRemoteControl

Overview

This API provides a function to get current IR remote control setting.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getInfraredRemoteControl",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------------------|--------|--|
| "infraredRemoteControl" | string | Current IR remote control setting (See IR remote control setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "infraredRemoteControl": "Off"
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setInfraredRemoteControl](#)
- [getSupportedInfraredRemoteControl](#)
- [getAvailableInfraredRemoteControl](#)

Special note (details)

None in particular.

getSupportedInfraredRemoteControl

Overview

This API provides a function to get the supported IR remote control settings.

The client should use "[getAvailableFlipSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedInfraredRemoteControl",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|--------------|---|
| "candidate" | string-array | A list of supported IR remote control settings (See IR remote control setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setInfraredRemoteControl](#)
- [getInfraredRemoteControl](#)
- [getAvailableInfraredRemoteControl](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableInfraredRemoteControl

Overview

This API provides a function to get current IR remote control setting and the available IR remote control settings at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableInfraredRemoteControl",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------------------|--------------|---|
| "infraredRemoteControl" | string | Current IR remote control setting (See IR remote control setting parameters of Parameter description) |
| "candidate" | string-array | A list of available IR remote control settings (See IR remote control setting parameters of Parameter description) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "infraredRemoteControl": "Off",
      "candidate": [
        "On",
        "Off"
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setInfraredRemoteControl](#)
- [getInfraredRemoteControl](#)
- [getSupportedInfraredRemoteControl](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "infraredRemoteControl" object in "[getEvent \(v1.2\)](#)" callback.

Auto power off

setAutoPowerOff

Overview

This API provides a function to set a value of auto power off time.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| key | type | explanation |
|----------------|---------|---|
| "autoPowerOff" | integer | Auto power off time (unit: second) (0 means that the auto power off function does not activate.) |

JSON Example

```
{
  "id": 1,
  "method": "setAutoPowerOff",
  "params": [
    {
      "autoPowerOff": 60
    }
  ],
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "id": 1,
  "result": []
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getAutoPowerOff](#)
- [getSupportedAutoPowerOff](#)
- [getAvailableAutoPowerOff](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "autoPowerOff" object in "[getEvent \(v1.2\)](#)" callback to recognize the timing of a change in the parameter of the server.

getAutoPowerOff

Overview

This API provides a function to get current auto power off time.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAutoPowerOff",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|----------------|---------|---|
| "autoPowerOff" | integer | Current auto power off time (unit: second) (0 means that the auto power off function does not activate.) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "autoPowerOff": 60
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setAutoPowerOff](#)
- [getSupportedAutoPowerOff](#)
- [getAvailableAutoPowerOff](#)

Special note (details)

None in particular.

getSupportedAutoPowerOff

Overview

This API provides a function to get the supported auto power off times.

The client should use "[getAvailableFlipSetting](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getSupportedAutoPowerOff",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|-------------|---------------|--|
| "candidate" | integer-array | Auto power off times (unit: second) (0 means that the auto power off function does not activate.) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "candidate": [
        0,
        60
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setAutoPowerOff](#)
- [getAutoPowerOff](#)
- [getAvailableAutoPowerOff](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableAutoPowerOff

Overview

This API provides a function to get current auto power off time and the available auto power off times at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getAvailableAutoPowerOff",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| key | type | explanation |
|----------------|---------------|--|
| "autoPowerOff" | integer | Current auto power off time (unit: second) (0 means that the auto power off function does not activate.) |
| "candidate" | integer-array | A list of available auto power off times (unit: second) (0 means that the auto power off function does not activate.) |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "autoPowerOff": 60,
      "candidate": [
        0,
        60
      ]
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setAutoPowerOff](#)
- [getAutoPowerOff](#)
- [getSupportedAutoPowerOff](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "autoPowerOff" object in "[getEvent \(v1.2\)](#)" callback.

Beep mode

setBeepMode

Overview

This API provides a function to set a value of beep mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|--------|---|
| 0 | string | Beep mode (See Beep mode parameter of Parameter description) |

JSON Example

```
{
  "method": "setBeepMode",
  "params": ["On"],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|---------|---|
| 0 | integer | Return parameter When the execution of the API is successful, 0 is set. If API is not successful, "result" member is not returned, and "error" member is returned. See Status code & Error for error detail. |

JSON Example

```
{
  "result": [0],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getBeepMode](#)
- [getSupportedBeepMode](#)
- [getAvailableBeepMode](#)

Special note (details)

Even if the response is successful, the setting may not be finished on the server. Therefore, the client can check "beepMode" object in "[getEvent](#)" callback to recognize the timing of a change in the parameter of the server.

getBeepMode

Overview

This API provides a function to get current beep mode.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getBeepMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Current beep mode (See Beep mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "On"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setBeepMode](#)
- [getSupportedBeepMode](#)
- [getAvailableBeepMode](#)

Special note (details)

None in particular.

getSupportedBeepMode

Overview

This API provides a function to get the supported beep modes.

The client should use "[getAvailableBeepMode](#)" to get the available parameters at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getSupportedBeepMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | A list of supported beep modes (See Beep mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    ["Off", "On", "Shutter Only"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setBeepMode](#)
- [getBeepMode](#)
- [getAvailableBeepMode](#)

Special note (details)

This method returns the same value for the response parameter only when this method is successful and the function is available. The result of the response depends on camera models.

getAvailableBeepMode

Overview

This API provides a function to get current beep mode and the available beep modes at the moment. The available parameters can be changed by user operations and calling APIs.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableBeepMode",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string | Current beep mode (See Beep mode parameter of Parameter description) |
| 1 | string-array | A list of available beep modes (See Beep mode parameter of Parameter description) |

JSON Example

```
{
  "result": [
    "On",
    ["Off", "On", "Shutter Only"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [setBeepMode](#)
- [getBeepMode](#)
- [getSupportedBeepMode](#)

Special note (details)

This API returns current set value and available values at the moment. The client can get the same response of this API via "beepMode" object in "[getEvent](#)" callback.

Date/time setting

setCurrentTime

Overview

This API provides a function to set current time with timezone information.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/system |
| Version | 1.0 |

Request

Elements of "params"

| order | key | type | explanation |
|-------|------------------------|---------|--|
| 0 | | object | Current time |
| | "dateTime" | string | Data time (ISO8601) |
| | "timeZoneOffsetMinute" | integer | Timezone offset (unit: minute, range: $\pm(23*60+59)$) |
| | "dstOffsetMinute" | integer | DST offset (unit: minute, range: $\pm(23*60+59)$) |

JSON Example

```
{
  "method": "setCurrentTime",
  "params": [
    {
      "dateTime": "2014-04-01T21:35:43Z",
      "timeZoneOffsetMinute": 540,
      "dstOffsetMinute": 0
    }
  ],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

None.

JSON Example

```
{
  "result": [],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

None.

Special note (details)

Only UTC time can apply to "Date time" parameter using YYYY-MM-DDTHH:mm:ssZ format. Please see also JSON example.

After the client set the date and time using this API, the date and time information on the camera will be set. The timing of adjusting the date and time depends on camera models.

Note that this API belongs to "system" API service. The client can check the availability of this API by checking if "system" API service is in the device description and by using "[getMethodTypes](#)" of "system" API service. See also "Development Guide" for more details.

Storage information

getStorageInformation

Overview

This API provides a function to get storage information.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "id": 1,
  "method": "getStorageInformation",
  "params": [],
  "version": "1.0"
}
```

Response

Elements of "result"

| order | key | type | explanation |
|-------|----------------------------|--------------|--|
| 0 | | object-array | A list of storage information. |
| | "storageID" | string | Storage ID (See Storage parameter of Parameter description) |
| | "recordTarget" | string | Recording target (If true, the storage is recording target.) |
| | "numberOfRecordableImages" | string | Number of recordable images. (-1: undefined) |
| | "recordableTime" | string | Recordable time. (unit: minute) (-1: undefined) |
| | "storageDescription" | string | Description of the storage. |

JSON Example

```
{
  "id": 1,
  "result": [
    {
      "storageDescription": "",
      "numberOfRecordableImages": 100,
      "recordTarget": true,
      "storageID": "Memory Card 1",
      "recordableTime": -1
    }
  ]
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getEvent \(v1.0\)](#)

Special note (details)

When no memory card is inserted, the API returns "No Media" as "storageID". The client can get the same response of this API via "storageInformation" object in "[getEvent \(v1.0\)](#)" callback.

For details please see the sample code included in the Camera Remote API SDK and [Sample Sequence](#).

Event notification

getEvent (v1.0)

Overview

This API provides a function to get event from the server.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

| order | type | explanation |
|-------|---------|---|
| 0 | boolean | Long polling flag true: Callback when timeout or change point detection. false: Callback immediately. |

JSON Example

```
{
  "method": "getEvent",
  "params": [true],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | Key | type | explanation |
|-------|--------------------------|--------------|---|
| 0 | | object | Available API list |
| | "type" | string | Name of object type ("availableApiList") |
| | "names" | string-array | A list of available API names (See API name parameters and getAvailableApiList) |
| 1 | | object | Camera status |
| | "type" | string | Name of object type ("cameraStatus") |
| | "cameraStatus" | string | Camera status (See Event parameters) |
| 2 | | object | Zoom information |
| | "type" | string | Name of object type ("zoomInformation") |
| | "zoomPosition" | integer | Zoom position to the whole (See actZoom) (When -1 is set, this parameter is invalid.) |
| | "zoomNumberBox" | integer | Number of zoom box (See actZoom) (When -1 is set, this parameter is invalid.) |
| | "zoomIndexCurrentBox" | integer | Index of current zoom box (See actZoom) (When -1 is set, this parameter is invalid.) |
| | "zoomPositionCurrentBox" | integer | Zoom position in the current zoom box (See actZoom) (When -1 is set, this parameter is invalid.) |

| | | | |
|--------|----------------------------|--------------|--|
| 3 | | object | Liveview status |
| | "type" | string | Name of object type ("liveviewStatus") |
| | "liveviewStatus" | boolean | Liveview status true: Ready to transfer Liveview images. false: Not ready to transfer Liveview images. |
| 4 | | object | Liveview orientation |
| | "type" | string | Name of object type ("liveviewOrientation") |
| | "liveviewOrientation" | string | Liveview orientation (See Event parameters) |
| 5 | | object-array | Array of postview URLs taken by camera body |
| | "type" | string | Name of object type ("takePicture") |
| | "takePictureUrl" | string-array | Array of URLs of taken picture by camera body. |
| 6 to 9 | | object/array | Reserved |
| 10 | | object-array | Storage information |
| | "type" | string | Name of object type ("storageInformation") |
| | "storageID" | string | Storage ID (See Storage parameter of Parameter description) |
| | "recordTarget" | string | Recording target (If true, the storage is recording target.) |
| | "numberOfRecordableImages" | string | Number of recordable images. (-1: undefined) |
| | "recordableTime" | string | Recordable time. (unit: minute) (-1: undefined) |
| | "storageDescription" | string | Description of the storage. |
| 11 | | object | Beep mode |
| | "type" | string | Name of object type ("beepMode") |
| | "currentBeepMode" | string | Current beep mode (See Beep mode parameter) |
| | "beepModeCandidates" | string-array | A list of available beep modes (See Beep mode parameter) |
| 12 | | object | Camera function |
| | "type" | string | Name of object type ("cameraFunction") |
| | "currentCameraFunction" | string | Current camera function (See Camera function parameter) |
| | "cameraFunctionCandidates" | string-array | A list of available camera functions (See Camera function parameter) |

| | | | |
|----|--------------------------|--------------|---|
| 13 | | object | Movie quality |
| | "type" | string | Name of object type ("movieQuality") |
| | "currentMovieQuality" | string | Current movie quality (See Movie quality parameter) |
| | "movieQualityCandidates" | string-array | A list of available movie qualities (See Movie quality parameter) |
| 14 | | object | Still size |
| | "type" | string | Name of object type ("stillSize") |
| | "checkAvailability" | boolean | If true, the client should check the change of available parameters by calling " getAvailableStillSize ". |
| | "currentAspect" | string | Current still aspect (See Still size parameter) |
| | "currentSize" | string | Current still size (See Still size parameter) |
| 15 | | object | Result of setting camera function |
| | "type" | string | Name of object type ("cameraFunctionResult") |
| | "cameraFunctionResult" | string | Result of setting camera function. "Success" - Success. "Failure" - Failed to changing function. |
| 16 | | object | Steady mode |
| | "type" | string | Name of object type ("steadyMode") |
| | "currentSteadyMode" | string | Current steady mode (See Steady mode parameter) |
| | "steadyModeCandidates" | string-array | A list of available steady modes (See Steady mode parameter) |
| 17 | | object | View angle |
| | "type" | string | Name of object type ("viewAngle") |
| | "currentViewAngle" | string | Current view angle (See View angle parameter) |
| | "viewAngleCandidates" | string-array | A list of available view angles (See View angle parameter) |
| 18 | | object | Exposure mode |
| | "type" | string | Name of object type ("exposureMode") |
| | "currentExposureMode" | string | Current exposure mode (See Exposure mode parameter) |
| | "exposureModeCandidates" | string-array | A list of available exposure modes (See Exposure mode parameter) |
| 19 | | object | Postview image size |
| | "type" | string | Name of object type ("postviewImageSize") |

| | | | |
|----------|-----------------------------------|---------------|---|
| | "currentPostviewImageSize" | string | Current postview image size (See Postview image size parameters) |
| | "postviewImageSizeCandidates" | string-array | A list of available postview image sizes (See Postview image size parameters) |
| 20 | | object | Self-timer |
| | "type" | string | Name of object type ("selfTimer") |
| | "currentSelfTimer" | integer | Current self-timer setting (unit: second) (See Self-timer parameters) (When -1 is set, this parameter is invalid.) |
| | "selfTimerCandidates" | integer-array | A list of available self-timer settings (unit: second) (See Self-timer parameters) |
| 21 | | object | Shoot mode |
| | "type" | string | Name of object type ("shootMode") |
| | "currentShootMode" | string | Current shoot mode (See Shoot mode parameters) |
| | "shootModeCandidates" | string-array | A list of available shoot modes (See Shoot mode parameters) |
| 22 to 24 | | object/array | Reserved |
| 25 | | object | Exposure compensation |
| | "type" | string | Name of object type ("exposureCompensation") |
| | "currentExposureCompensation" | integer | Current exposure compensation index value (See getSupportedExposureCompensation) |
| | "maxExposureCompensation" | integer | Max value of exposure compensation index (See getSupportedExposureCompensation) |
| | "minExposureCompensation" | integer | Min value exposure compensation index (See getSupportedExposureCompensation) |
| | "stepIndexOfExposureCompensation" | integer | Exposure compensation step index (See getSupportedExposureCompensation) |
| 26 | | object | Flash mode |
| | "type" | string | Name of object type ("flashMode") |
| | "currentFlashMode" | string | Current flash mode (See Flash mode parameters) |
| | "flashModeCandidates" | string-array | A list of available flash modes (See Flash mode parameters) |
| 27 | | object | F number |
| | "type" | string | Name of object type ("fNumber") |
| | "currentFNumber" | string | Current F number value (See setFNumber) |
| | "fNumberCandidates" | string-array | A list of available F number values (See setFNumber) |
| 28 | | object | Focus mode |

| | | | |
|----|---------------------------|--------------|---|
| | "type" | string | Name of object type ("focusMode") |
| | "currentFocusMode" | string | Current focus mode (See Focus mode parameter) |
| | "focusModeCandidates" | string-array | A list of available focus modes (See Focus mode parameter) |
| 29 | | object | ISO speed rate |
| | "type" | string | Name of object type ("isoSpeedRate") |
| | "currentIsoSpeedRate" | string | Current ISO speed rate value (See setIsoSpeedRate) |
| | "isoSpeedRateCandidates" | string-array | A list of available ISO speed rate values (See setIsoSpeedRate) |
| 30 | | object | Reserved |
| 31 | | object | Program shift |
| | "type" | string | Name of object type ("programShift") |
| | "isShifted" | boolean | Program shift status true: Shifted false: Not shifted |
| 32 | | object | Shutter speed |
| | "type" | string | Name of object type ("shutterSpeed") |
| | "currentShutterSpeed" | string | Current shutter speed value (See setShutterSpeed) |
| | "shutterSpeedCandidates" | string-array | A list of available shutter speed values (See setShutterSpeed) |
| 33 | | object | White balance |
| | "type" | string | Name of object type ("whiteBalance") |
| | "checkAvailability" | boolean | If true, the client should check the change of available parameters by calling "getAvailableWhiteBalance" . |
| | "currentWhiteBalanceMode" | string | Current white balance mode (See White balance parameter) |
| | "currentColorTemperature" | integer | Current color temperature (See White balance parameter) |
| 34 | | object | Touch AF position |
| | "type" | string | Name of object type ("touchAFPosition") |
| | "currentSet" | boolean | Set or not. true: Touch AF is set and focused successfully. false: Touch AF is not set or failed to focus. |
| | "currentTouchCoordinates" | double-array | Touch coordinates. This parameter is reserved and the camera will return empty array. |

JSON Example

```
{
  "result": [
    {
      "type": "availableApiList",
      "names": ["startLiveview", "stopLiveview", "setSelfTimer", ...]
    },
    {
      "type": "cameraStatus",
      "cameraStatus": "IDLE"
    },
    {
      "type": "zoomInformation",
      "zoomPosition": 0,
      "zoomNumberBox": 1,
      "zoomIndexCurrentBox": 0,
      "zoomPositionCurrentBox": 0
    },
    {
      "type": "liveviewStatus",
      "liveviewStatus": true
    },
    {
      "type": "liveviewOrientation",
      "liveviewOrientation": "90"
    },
    [
      {
        "type": "takePicture",
        "takePictureUrl": [
          "http://ip:port/postview/postview.jpg"
        ]
      }
    ],
    [],
    null,
    null,
    null,
    [
      {
        "storageDescription": "Storage Media",
        "numberOfRecordableImages": 123,
        "type": "storageInformation",
        "storageID": "Memory Card 1",
        "recordTarget": true,
        "recordableTime": 30
      }
    ],
    {
      "type": "beepMode",
      "currentBeepMode": "On",
      "beepModeCandidates": ["Off", "On", "Shutter Only"]
    },
  ],
}
```

```

{
  "type": "cameraFunction",
  "currentCameraFunction": "Remote Shooting",
  "cameraFunctionCandidates": [
    "Contents Transfer",
    "Remote Shooting"
  ]
},
{
  "type": "movieQuality",
  "currentMovieQuality": "HQ",
  "movieQualityCandidates": ["HQ", "STD", "VGA", "SSLOW"]
},
{
  "type": "stillSize",
  "checkAvailability": true,
  "currentAspect": "4:3",
  "currentSize": "18M"
},
{
  "type": "cameraFunctionResult",
  "cameraFunctionResult": "Success"
},
{
  "type": "steadyMode",
  "currentSteadyMode": "off",
  "steadyModeCandidates": ["off", "on"]
},
{
  "type": "viewAngle",
  "currentViewAngle": 120,
  "viewAngleCandidates": [120, 170]
},
{
  "type": "exposureMode",
  "currentExposureMode": "Intelligent Auto",
  "exposureModeCandidates": ["Intelligent Auto", "Aperture", "Shutter"]
},
{
  "type": "postviewImageSize",
  "currentPostviewImageSize": "2M",
  "postviewImageSizeCandidates": ["Original", "2M"]
},
{
  "type": "selfTimer",
  "currentSelfTimer": 0,
  "selfTimerCandidates": [0, 2, 10]
},
{
  "type": "shootMode",
  "currentShootMode": "still",
  "shootModeCandidates": ["still", "movie"]
},
null,
null,
null,

```

```

    {
      "type": "exposureCompensation",
      "currentExposureCompensation": 1,
      "maxExposureCompensation": 9,
      "minExposureCompensation": -9,
      "stepIndexOfExposureCompensation": 1
    },
    {
      "type": "flashMode",
      "currentFlashMode": "auto",
      "flashModeCandidates": ["off", "auto", "on"]
    },
    {
      "type": "fNumber",
      "currentFNumber": "2.8",
      "fNumberCandidates": ["2.8", "4.0", "5.6", "22.0", "32.0"]
    },
    {
      "type": "focusMode",
      "currentFocusMode": "AF-S",
      "focusModeCandidates": ["AF-S", "MF"]
    },
    {
      "type": "isoSpeedRate",
      "currentIsoSpeedRate": "100",
      "isoSpeedRateCandidates": ["AUTO", "100", "24000"]
    },
    null,
    {
      "type": "programShift",
      "isShifted": true
    },
    {
      "type": "shutterSpeed",
      "currentShutterSpeed": "1/2",
      "shutterSpeedCandidates": ["2\\", "1\\", "1/2", "1/3", "1/4"]
    },
    {
      "type": "whiteBalance",
      "checkAvailability": true,
      "currentWhiteBalanceMode": "Color Temperature",
      "currentColorTemperature": 2500
    },
    {
      "type": "touchAFPosition",
      "currentSet": true,
      "currentTouchCoordinates": []
    }
  ],
  "id": 1
}

```

Error CodesSee [Status code & Error](#)**Related API**

None.

Special note (details)

The purpose of this API is sending the event from the server actively. When the client calls some APIs and/or the user operates the camera directly, the client can get the parameter updates from the camera by using this API.

If `getEvent` is executed, the server times out or does not return a response until the parameter is updated. (If the input parameter "polling" is false, the server replies immediately.) If the client gets the response, the client should execute `getEvent` immediately. When the execution of this API is unsuccessful or timeout, the server will notify error in response.

When each object in the response is null or empty array as JSON, it means that no update was happened in the server for the object type or the server does not support the object type. In case of empty string or empty array in supported object, it means that the parameter is invalid. In regard to some objects related to `getAvailableXXX` API, the client can check if the object is valid via "[getAvailableApiList](#)" or "availableApiList" object.

The callback parameter value could be the same as the previous one, so the client should evaluate the value of callback. The client can detect each object in the callback using "Name of object type" which is in each object. The client must ignore objects which are not described in this document.

In case this API is called with "polling=true", the server will not return a response until the server is ready, and any value has changed in the server. So, by calling with `polling=true`, the client can recognize the timing of a change in the parameter of the server. For each parameter, if the current value or available candidates doesn't update, the parameter object including "type" will be null object or empty array. Otherwise, the object will be notified. This API with "polling=true" is a notifying type method and acts with other requiring type methods in parallel.

In case this API is called with "polling=false", the server will return a response immediately. So, by calling with `polling=false`, the client can recognize the server state snapshot at the time. The client can call this API with "polling=false" at the beginning. (The information obtained in the response may be useful for the client to build UI layouts and so on.)

For more information about how to use this API, see [Sample Sequence](#).

getEvent (v1.1)

Overview

This API provides a function to get event from the server.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.1 |

Request**Elements of "params"**

| order | type | explanation |
|-------|---------|---|
| 0 | boolean | Long polling flag true: Callback when timeout or change point detection. false: Callback immediately. |

JSON Example

```
{
  "method": "getEvent",
  "params": [true],
  "id": 1,
  "version": "1.1"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|---------|---------------|--------------|--|
| 0 to 34 | | object/array | Same as v1.0. Please see " getEvent (v1.0) " for details. |
| 35 | | object | Focus status |
| | "type" | string | Name of object type ("focusStatus") |
| | "focusStatus" | string | Focus status (See Event parameters) |

JSON Example

```
{
  "result": [
    ...,
    {
      "type": "focusStatus",
      "focusStatus": "Focused"
    }
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getEvent \(v1.0\)](#)

Special note (details)

Some camera models support version 1.1 of "getEvent" in addition to v1.0. The client can check if the server supports version 1.1 using "[getVersions](#)" and "[getMethodTypes](#)" for "camera" API service. The client app must set "1.1" as the "version" parameter in the request when it uses "getEvent" version 1.1 function.

The version 1.1 of the API supports "focusStatus" object as the response in addition to v1.0 of the API. Please see "[getEvent \(v1.0\)](#)" for more details.

getEvent (v1.2)

Overview

This API provides a function to get event from the server.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.2 |

Request**Elements of "params"**

| order | type | explanation |
|-------|---------|---|
| 0 | boolean | Long polling flag true: Callback when timeout or change point detection. false: Callback immediately. |

JSON Example

```
{
  "method": "getEvent",
  "params": [true],
  "id": 1,
  "version": "1.2"
}
```

Response**Elements of "result"**

| order | key | type | explanation |
|---------|--------------------|--------------|---|
| 0 to 35 | | object/array | Same as v1.1. Please see " getEvent (v1.1) " for details. |
| 36 | | object | Zoom setting |
| | "type" | string | Name of object type ("zoomSetting") |
| | "zoom" | string | Current zoom setting (See Zoom parameters) |
| | "candidate" | string-array | A list of available zoom settings (See Zoom parameters) |
| 37 | | object | Still quality |
| | "type" | string | Name of object type ("stillQuality") |
| | "stillQuality" | string | Current still quality (See Still quality parameters) |
| | "candidate" | string-array | A list of available still qualities (See Still quality parameters) |
| 38 | | object | Continuous shooting mode |
| | "type" | string | Name of object type ("contShootingMode") |
| | "contShootingMode" | string | Current continuous shooting mode (See Continuous shooting mode parameters) |
| | "candidate" | string-array | A list of available continuous shooting modes (See Continuous shooting mode parameters) |

| | | | |
|----|---------------------|--------------|---|
| 39 | | object | Continuous shooting speed |
| | "type" | string | Name of object type ("contShootingSpeed") |
| | "contShootingSpeed" | string | Current continuous shooting speed (See Continuous shooting speed parameters) |
| | "candidate" | string-array | A list of available continuous shooting speeds (See Continuous shooting speed parameters) |
| 40 | | object | URLs of continuous shooting |
| | "type" | string | Name of object type ("contShooting") |
| | "contShootingUrl" | object-array | Array of URL of continuous shooting. When more than one URL notifies, the last one is the latest. |
| | "postviewUrl" | string | The URL of thumbnail of postview. |
| | "thumbnailUrl" | string | The URL of postview (the size depends on "postviewImageSize"). |
| 41 | | object | Flip setting |
| | "type" | string | Name of object type ("flipSetting") |
| | " flip" | string | Current flip setting (See Flip setting parameters) |
| | "candidate" | string-array | A list of available flip settings (See Flip setting parameters) |
| 42 | | object | Scene selection |
| | "type" | string | Name of object type ("sceneSelection") |
| | "scene" | string | Current scene selection (See Scene selection parameters) |
| | "candidate" | string-array | A list of available scene selections (See Scene selection parameters) |
| 43 | | object | Interval time |
| | "type" | string | Name of object type ("intervalTime") |
| | "intervalTimeSec" | string | Current interval time (unit: second) (See Interval time parameters) |
| | "candidate" | string-array | A list of available interval times (See Interval time parameters) |
| 44 | | object | Color setting |
| | "type" | string | Name of object type ("colorSetting") |
| | "colorSetting" | string | Current color setting (See Color setting parameters) |
| | "candidate" | string-array | A list of available color settings (See Color setting parameters) |

| | | | |
|----------|-------------------------|--------------|---|
| 45 | | object | Movie file format |
| | "type" | string | Name of object type ("movieFileFormat") |
| | "movieFileFormat" | string | Current movie file format (See Movie file format parameters) |
| | "candidate" | string-array | A list of available movie file formats (See Movie file format parameters) |
| 46 to 51 | | object | Reserved |
| 52 | | object | IR remote control setting |
| | "type" | string | Name of object type ("infraredRemoteControl") |
| | "infraredRemoteControl" | string | Current IR remote control setting (See IR remote control setting parameters) |
| | "candidate" | string-array | A list of available IR remote control settings (See IR remote control setting parameters) |
| 53 | | object | TV color system |
| | "type" | string | Name of object type ("tvColorSystem") |
| | "tvColorSystem" | string | Current TV color system (See TV color system parameters) |
| | "candidate" | string-array | A list of available TV color systems (See TV color system parameters) |
| 54 | | object | Tracking focus status |
| | "type" | string | Name of object type ("trackingFocusStatus") |
| | "trackingFocusStatus" | string | Tracking focus status (See Tracking focus parameters) |
| 55 | | object | Tracking focus setting |
| | "type" | string | Name of object type ("trackingFocus") |
| | "trackingFocus" | string | Current tracking focus setting (See Tracking focus parameters) |
| | "candidate" | string-array | A list of available tracking focus settings (See Tracking focus parameters) |
| 56 | | object | Battery information |
| | "type" | string | Name of object type ("batteryInfo") |
| | "batteryInfo" | object-array | Array of URL of continuous shooting. When more than one URL notifies, the last one is the latest. |
| | "batteryID" | string | Battery ID (See Battery information parameters) |
| | "status" | string | Battery status (See Battery information parameters) |
| | "additionalStatus" | string | Additional battery status (See Battery information parameters) |

| | | | | |
|----|-----------------|---------------|---------------|---|
| | | "levelNumer" | integer | Battery level (numerator). (When -1 is set, this parameter becomes invalid.) |
| | | "levelDenom" | integer | Battery level (denominator). (When -1 is set, this parameter becomes invalid.) |
| | | "description" | string | Description of the battery |
| 57 | | | object | Recording time |
| | "type" | | string | Name of object type ("recordingTime") |
| | "recordingTime" | | integer | Recording time of the movie. (unit: second) (When -1 is set, this parameter is invalid.) |
| 58 | | | object | Number of shots |
| | "type" | | string | Name of object type ("numberOfShots") |
| | "numberOfShots" | | integer | Number of shots. (When -1 is set, this parameter becomes invalid.) Supported only when the shoot mode is "intervalstill". |
| 59 | | | object | Auto power off time |
| | "type" | | string | Name of object type ("autoPowerOff") |
| | "autoPowerOff" | | integer | Auto power off time (unit: second) (0 means that the auto power off function does not activate.) |
| | "candidate" | | integer-array | A list of available auto power off times |

JSON Example

```
{
  "result": [
    ...,
    {
      "type": "zoomSetting",
      "candidate": [
        "Optical Zoom Only",
        "On:Clear Image Zoom"
      ],
      "zoom": "Optical Zoom Only"
    },
    {
      "stillQuality": "Fine",
      "type": "stillQuality",
      "candidate": [
        "Fine",
        "Standard"
      ]
    }
  ],
}
```

```
{
  "type": "contShootingMode",
  "candidate": [
    "Single",
    "Continuous"
  ],
  "contShootingMode": "Single"
},
{
  "type": "contShootingSpeed",
  "contShootingSpeed": "Hi",
  "candidate": [
    "Hi",
    "Low"
  ]
},
{
  "type": "contShooting",
  "contShootingUrl": [
    {
      "postviewUrl": "http://ip:port/continuous/postview1.jpg",
      "thumbnailUrl": "http://ip:port/continuous/thumbnail1.jpg"
    },
    {
      "postviewUrl": "http://ip:port/continuous/postview2.jpg",
      "thumbnailUrl": "http://ip:port/continuous/thumbnail2.jpg"
    }
  ]
},
{
  "type": "flipSetting",
  "candidate": [
    "On",
    "Off"
  ],
  "flip": "Off"
},
{
  "type": "sceneSelection",
  "scene": "Normal",
  "candidate": [
    "Normal",
    "Under Water"
  ]
},
{
  "type": "intervalTime",
  "candidate": [
    "1",
    "2",
    "5",
    "10",
    "30",
    "60"
  ],
  "intervalTimeSec": "1"
},
}
```

```
{
  "type": "colorSetting",
  "candidate": [
    "Neutral",
    "Vivid"
  ],
  "colorSetting": "Vivid"
},
{
  "movieFileFormat": "MP4",
  "type": "movieFileFormat",
  "candidate": [
    "MP4",
    "XAVC S"
  ]
},
null,
null,
null,
null,
null,
null,
{
  "type": "infraredRemoteControl",
  "infraredRemoteControl": "On",
  "candidate": [
    "On",
    "Off"
  ]
},
{
  "type": "tvColorSystem",
  "tvColorSystem": "NTSC",
  "candidate": [
    "NTSC",
    "PAL"
  ]
},
{
  "type": "trackingFocusStatus",
  "trackingFocusStatus": "Not Tracking"
},
{
  "type": "trackingFocus",
  "candidate": [
    "Off",
    "On"
  ],
  "trackingFocus": "Off"
},
}
```

```
{
  "type": "batteryInfo",
  "batteryInfo": [
    {
      "levelDenom": 4,
      "levelNumer": 4,
      "status": "active",
      "description": "",
      "additionalStatus": "",
      "batteryID": "externalBattery1"
    }
  ]
},
{
  "type": "recordingTime",
  "recordingTime": 2
},
{
  "numberOfShots": 6,
  "type": "numberOfShots"
},
{
  "type": "autoPowerOff",
  "candidate": [
    60,
    0
  ],
  "autoPowerOff": 0
}
],
"id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getEvent \(v1.0\)](#)
- [getEvent \(v1.1\)](#)

Special note (details)

Some camera models support version 1.2 of "getEvent" in addition to v1.0 and v1.1. The client can check if the server supports version 1.2 using "[getVersions](#)" and "[getMethodTypes](#)" for "camera" API service. The client app must set "1.2" as the "version" parameter in the request when it uses "getEvent" version 1.2 function.

Please see "[getEvent \(v1.0\)](#)" and "[getEvent \(v1.1\)](#)" for more details.

Server information

getAvailableApiList

Overview

This API provides a function to get the available API names that the server supports at the moment.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getAvailableApiList",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|---|
| 0 | string-array | A list of available API names (See API name parameters of Parameter description) |

JSON Example

```
{
  "result": [
    ["startLiveview", "stopLiveview", "setSelfTimer", ...]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getMethodTypes](#)

Special note (details)

The client gets the list of API names that can be executed at the moment from callback parameter. The callback parameter "A list of available API names" varies depending on the camera status at the moment. For example, when the server is in "movie" mode, "[setSelfTimer](#)" will not be in the callback parameter.

The client should ignore API names which are not described in this document.

getApplicationInfo

Overview

This API provides a function to get name and "Camera Remote API" version of the server.

| | |
|--------------|-------------------------|
| Endpoint URL | <ActionList_URL>/camera |
| Version | 1.0 |

Request**Elements of "params"**

Empty.

JSON Example

```
{
  "method": "getApplicationInfo",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "result"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Application name of the server |
| 1 | string | "Camera Remote API" version of the server |

JSON Example

```
{
  "result": [
    "Smart Remote Control",
    "2.0.0"
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getVersions](#)

Special note (details)

The client should call this API during initialization sequence. The client can get the following "Camera Remote API" version of server. The version definition: "M.m.x".

[M]: Major number (Any number)

[m]: Minor number (Any number)

[x]: Server release number (Any number)

The client should check whether the API version is "2.x.x" ("2.0.0" or greater) to confirm the server function compatibility with this document.

getVersions

Overview

This API provides supported versions on the "API service". The client can get the list of API names for specific version using "[getMethodTypes](#)" API. The client can get list of versions, which the server supports, using this API.

| | |
|--------------|--|
| Endpoint URL | <ActionList_URL>/camera <ActionList_URL>/system <ActionList_URL>/avContent |
| Version | 1.0 |

Request

Elements of "params"

Empty.

JSON Example

```
{
  "method": "getVersions",
  "params": [],
  "id": 1,
  "version": "1.0"
}
```

Response

Elements of "result"

| order | type | explanation |
|-------|--------------|--|
| 0 | string-array | Supported versions on the API service. |

JSON Example

```
{
  "result": [
    ["1.0"]
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getMethodTypes](#)

Special note (details)

The version in the response is defined for each API in the "API service". For more information about "API service", see Development Guide. When the client calls the APIs to use camera functionality, API service is basically "camera". If the server supports "system" API service, it can call this API for "system".

getMethodTypes

Overview

This API provides a function to get the supported APIs for the version. The client can get the list of API names for specific version using this API. The client can get list of versions, which the server supports, using "[getVersions](#)" API.

| | |
|--------------|--|
| Endpoint URL | <ActionList_URL>/camera <ActionList_URL>/system <ActionList_URL>/avContent |
| Version | 1.0 |

Request**Elements of "params"**

| order | type | explanation |
|-------|--------|---|
| 0 | string | Version for each API. In case this parameter is empty string, response includes methods of all versions supported in camera. |

JSON Example

```
{
  "method": "getMethodTypes",
  "params": ["1.0"],
  "id": 1,
  "version": "1.0"
}
```

Response**Elements of "results"**

| order | type | explanation |
|--------|--------------|--------------------------|
| 0 to N | array | Array of elements |
| 0 | string | API name |
| 1 | string-array | Array of parameter types |
| 2 | string-array | Array of response types |
| 3 | string | Version for each API |

JSON Example

```
{
  "results": [
    ["getAvailableApiList", [], ["string*"], "1.0"],
    ["setShootMode", ["string"], ["int"], "1.0"],
    ...
  ],
  "id": 1
}
```

Error Codes

See [Status code & Error](#)

Related API

- [getVersions](#)

Special note (details)

The definition of "parameter types" is below.

| value | explanation |
|----------------------------------|--|
| "bool" | True or false value is stored. |
| "bool*" | Multiple true or false values are stored in an array. |
| "int" | Integer number. |
| "int*" | Multiple integer numbers are stored in an array. |
| "double" | Double number. |
| "double*" | Multiple double numbers are stored in an array. |
| "string" | String data is stored. |
| "string*" | Multiple string data are stored in an array. |
| "{"object name":"object type"}" | An Object is stored. Double quotes in object are escaped to store object structure into a string. |
| "{"object name":"object type"}*" | Multiple objects (of the same type) are stored in an array. Double quotes in object are escaped to store object structure into a string. |

Most of APIs will be replied by "result" in those responses. Note that result of this API will be replied by "results" in the response.

The camera status can be changed by user operations and calling APIs. Available APIs in the camera will be changed by camera status. Therefore, the client should check available APIs at the moment via ["getAvailableApiList"](#) API or "availableApiList" object of ["getEvent"](#) API callback.

The client should ignore API names which are not described in this document.

Parameter description

API name parameters

The API names which belong to "camera" API service, for example "setSelfTimer" and "getSelfTimer", are listed as API name parameters for the response of "[getAvailableApiList](#)" and "availableApiList" object of "[getEvent](#)". Note that API names of other than "camera" API service are not included in the available API list.

Note that the client can check the availability of "[actTakePicture](#)", "[awaitTakePicture](#)", "[startMovieRec](#)", "[stopMovieRec](#)", "[startAudioRec](#)", "[stopAudioRec](#)", "[startIntervalStillRec](#)", and "[stopIntervalStillRec](#)" by checking current shooting mode. Current shooting mode can be obtained by "[getShootMode](#)", "[getAvailableShootMode](#)" or "[getEvent](#)".

Shoot mode parameters

| value | explanation |
|-----------------|---------------------------|
| "still" | Still image shoot mode |
| "movie" | Movie shoot mode |
| "audio" | Audio shoot mode |
| "intervalstill" | Interval still shoot mode |

Liveview size parameter

| value | explanation |
|-------|--|
| "L" | XGA size scale (the size varies depending on the camera models, and some camera models change the liveview quality instead of making the size larger.) |
| "M" | VGA size scale (the size varies depending on the camera models) |

Zoom parameters

Zoom direction parameter

| Value | explanation |
|-------|-------------|
| "in" | Zoom-In |
| "out" | Zoom-Out |

Zoom movement parameter

| Value | explanation |
|---------|-------------|
| "start" | Long push |
| "stop" | Stop |
| "1shot" | Short push |

Zoom setting parameter

| Value | explanation |
|-----------------------|----------------------|
| "Optical Zoom Only" | Optical zoom only. |
| "On:Clear Image Zoom" | On:Clear Image Zoom. |

Touch AF position parameter

| value | explanation |
|---------|---|
| "Touch" | Focus on around touch area |
| "Wide" | Focus on over a wide range including touch area |

Tracking focus parameter

Tracking focus setting parameter

| value | explanation |
|-------|--|
| "Off" | Does not track a subject to be focused on. |
| "On" | Tracks a subject to be focused on. |

Tracking focus status parameter

| value | explanation |
|----------------|------------------------|
| "Tracking" | Tracking a subject |
| "Not Tracking" | Not tracking a subject |

Continuous shooting mode parameter

| value | explanation |
|----------------------|------------------------------------|
| "Single" | Single shooting |
| "Continuous" | Continuous shooting |
| "Spd Priority Cont." | Speed priority continuous shooting |
| "Burst" | Burst shooting |
| "MotionShot" | MotionShot |

Continuous shooting speed parameter

| value | explanation |
|-------------|---------------------------|
| "Hi" | Hi |
| "Low" | Low |
| "8fps 1sec" | 10 frames in 1.25 seconds |
| "5fps 2sec" | 10 frames in 2 seconds |
| "2fps 5sec" | 10 frames in 5 seconds |

Self-timer parameters

| value | explanation |
|-------|-------------|
| 0 | Off |
| 2 | 2 seconds |
| 10 | 10 seconds |

Exposure mode parameter

| value | explanation |
|--------------------|-------------------|
| "Program Auto" | Program Auto |
| "Aperture" | Aperture Priority |
| "Shutter" | Shutter Priority |
| "Manual" | Manual Exposure |
| "Intelligent Auto" | Intelligent Auto |
| "Superior Auto" | Superior Auto |

Focus mode parameter

| value | explanation |
|--------|---------------|
| "AF-S" | Single AF |
| "AF-C" | Continuous AF |
| "MF" | Manual Focus |

White balance parameter

White balance mode parameter

| value | explanation |
|--------------------------------|------------------------------|
| "Auto WB" | Auto WB |
| "Daylight" | Daylight |
| "Shade" | Shade |
| "Cloudy" | Cloudy |
| "Incandescent" | Incandescent |
| "Fluorescent: Warm White (-1)" | Fluorescent: Warm White (-1) |
| "Fluorescent: Cool White (0)" | Fluorescent: Cool White (0) |
| "Fluorescent: Day White (+1)" | Fluorescent: Day White (+1) |
| "Fluorescent: Daylight (+2)" | Fluorescent: Daylight (+2) |
| "Flash" | Flash |
| "Color Temperature" | Color Temperature |
| "Custom 1" | Custom 1 |
| "Custom 2" | Custom 2 |
| "Custom 3" | Custom 3 |

White balance color temperature parameter

When a value of color temperature is -1, the parameter is invalid.

The below is a sample of color temperature range.

| value | explanation |
|-------------------|--|
| [9900, 2500, 100] | [Maximum value, minimum value, step value] In this case, it means that from 2500K to 9900K with 100K increments in between. |

Flash mode parameter

| value | explanation |
|------------|--------------|
| "off" | OFF |
| "auto" | Auto flash |
| "on" | Forced flash |
| "slowSync" | Slow synchro |
| "rearSync" | Rear synchro |
| "wireless" | Wireless |

Still size parameter

Still aspect parameter

| value | explanation |
|--------|-------------|
| "16:9" | 16:9 |
| "4:3" | 4:3 |
| "3:2" | 3:2 |
| "1:1" | 1:1 |

Still size parameter

Note that below is example and the supported parameters may vary depending on the servers.

| value | explanation |
|--------|-------------|
| "20M" | 20M pixels |
| "18M" | 18M pixels |
| "17M" | 17M pixels |
| "13M" | 13M pixels |
| "7.5M" | 7.5M pixels |
| "5M" | 5M pixels |
| "4.2M" | 4.2M pixels |
| "3.7M" | 3.7M pixels |

Still quality parameters

| value | explanation |
|------------|-----------------|
| "RAW+JPEG" | RAW+JPEG |
| "Fine" | JPEG (Fine) |
| "Standard" | JPEG (Standard) |

Postview image size parameters

| value | explanation |
|------------|---|
| "Original" | Original size |
| "2M" | 2M-pixel size (the actual size depends on camera models.) |

Movie file format parameters

| value | explanation |
|----------|-------------|
| "MP4" | MP4 |
| "XAVC S" | XAVC S |

Movie quality parameter

| value | explanation |
|-----------|---|
| "PS" | MP4, 1920x1080 60p/50p |
| "HQ" | MP4, 1920x1080 30p/25p |
| "STD" | MP4, 1280x720 30p/25p |
| "VGA" | MP4, 640x480 30p/25p |
| "SLOW" | MP4, 1280x720 30p (Imaging frame rate: 60p) |
| "SSLOW" | MP4, 1280x720 30p/25p (Imaging frame rate: 120p/100p) |
| "HS120" | MP4, 1280x720 120p |
| "HS100" | MP4, 1280x720 100p |
| "HS240" | MP4, 800x480 240p |
| "HS200" | MP4, 800x480 200p |
| "50M 60p" | XAVC S, 1920x1080 60p |
| "50M 50p" | XAVC S, 1920x1080 50p |
| "50M 30p" | XAVC S, 1920x1080 30p |
| "50M 25p" | XAVC S, 1920x1080 25p |
| "50M 24p" | XAVC S, 1920x1080 24p |

Steady mode parameter

| value | explanation |
|-------|-------------|
| "off" | Off |
| "on" | On |

View angle parameter

Note that below is example and the supported parameters may vary depending on the servers.

| value | explanation |
|-------|------------------|
| 120 | 120-degree angle |
| 170 | 170-degree angle |
| -1 | invalid |

Scene selection parameter

| value | explanation |
|---------------|-------------|
| "Normal" | Normal |
| "Under Water" | Under Water |

Color setting parameter

| value | explanation |
|-----------|---------------|
| "Neutral" | Neutral color |
| "Vivid" | Vivid color |

Interval time parameter

| value | explanation |
|-------|-------------|
| "1" | 1 second |
| "2" | 2 seconds |
| "5" | 5 seconds |
| "10" | 10 seconds |
| "30" | 30 seconds |
| "60" | 60 seconds |

Flip setting parameter

| value | explanation |
|-------|---|
| "On" | Flips the image vertically and swaps the left and right sound channels. |
| "Off" | Does not flip the image. |

IR remote control setting parameter

| value | explanation |
|-------|---------------------------------|
| "On" | Using IR remote controller. |
| "Off" | Not using IR remote controller. |

TV color system parameter

| value | explanation |
|--------|-------------|
| "NTSC" | NTSC |
| "PAL" | PAL |

Beep mode parameter

| value | explanation |
|----------------|-----------------------------------|
| "Off" | Turns off the beep/shutter sound. |
| "On" | Turns on the beep/shutter sound. |
| "Shutter Only" | Turns on the shutter sound only. |

Storage parameter

| value | explanation |
|-----------------|---|
| "Memory Card 1" | Memory Card 1 (The card is inserted in the camera.) |
| "No Media" | No Media |

Camera function parameter

| value | explanation |
|---------------------|------------------------------|
| "Remote Shooting" | Shooting function |
| "Contents Transfer" | Transferring images function |

Battery information parameter

Battery ID parameter

| value | explanation |
|--------------------|---|
| "externalBattery1" | External battery 1 (The battery is inserted in the camera.) |
| "noBattery" | No battery |

Battery status parameter

| value | explanation |
|------------|----------------------------|
| "active" | The battery is in use. |
| "inactive" | The battery is not in use. |
| "unknown" | The status is unknown |

Additional battery status parameter

| value | explanation |
|------------------|--|
| "batteryNearEnd" | The battery power will be discharged soon. |
| "charging" | The battery is charging. |
| "" | Not set additional status. |

Event parameters

Camera status parameters

| value | explanation |
|------------------------|---|
| "Error" | Error at the server (ex. high temperature, no memory card) |
| "NotReady" | The server cannot start recording (ex. during initialization, mode transitioning) |
| "IDLE" | Ready to record |
| "StillCapturing" | Capturing still images |
| "StillSaving" | Saving still images |
| "MovieWaitRecStart" | Preparing to start recording movie |
| "MovieRecording" | Recording movie |
| "MovieWaitRecStop" | Stopping the movie recording |
| "MovieSaving" | Saving movie |
| "AudioWaitRecStart" | Preparing to start recording audio |
| "AudioRecording" | Recording audio |
| "AudioWaitRecStop" | Stopping the audio recording |
| "AudioSaving" | Saving audio |
| "IntervalWaitRecStart" | Preparing to capture interval still images |
| "IntervalRecording" | Capturing interval still images |
| "IntervalWaitRecStop" | Stopping interval still images |
| "ContentsTransfer" | The status ready to transferring images |
| "Streaming" | Streaming the movie |
| "Deleting" | Deleting the content |

Liveview orientation parameter

| value | explanation |
|-------|-------------------------------|
| "0" | Not rotated |
| "90" | Rotated 90 degrees clockwise |
| "180" | Rotated 180 degrees clockwise |
| "270" | Rotated 270 degrees clockwise |

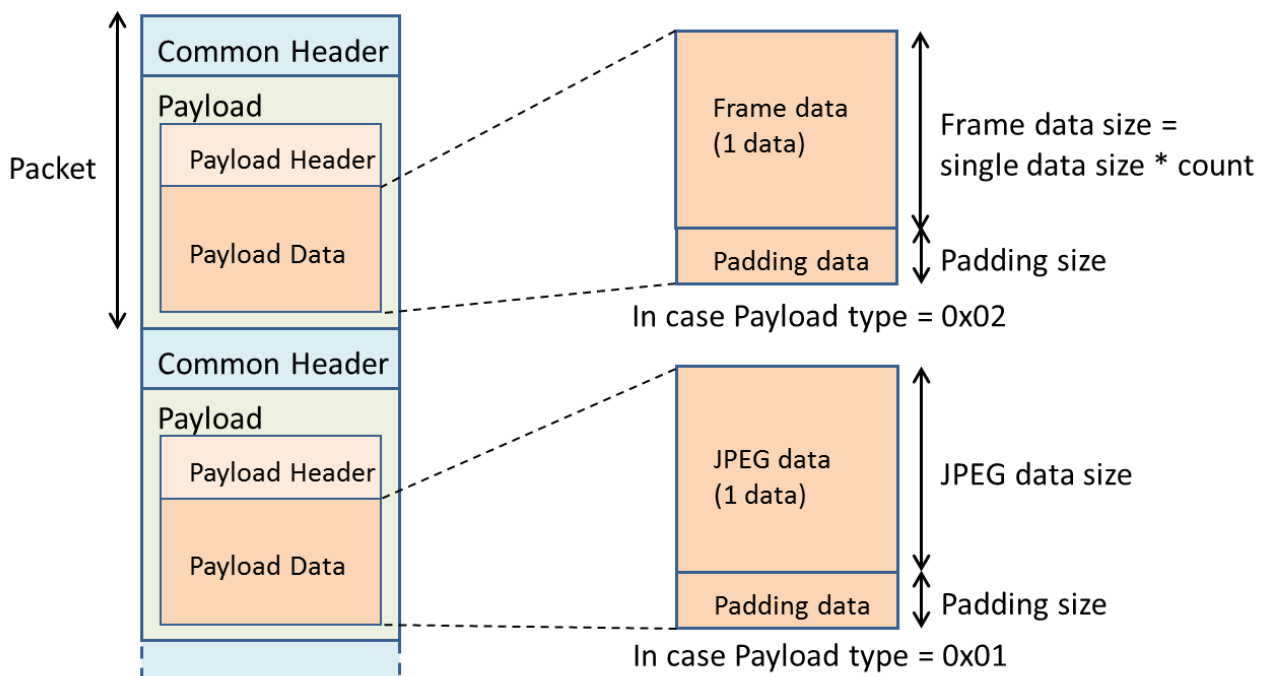
Focus status parameter

| value | explanation |
|----------------|---------------------------|
| "Not Focusing" | The focus is not working. |
| "Focusing" | The focus is in progress. |
| "Focused" | The focus is locked. |
| "Failed" | The focus has failed. |

Liveview data format

Format of the liveview data JPEG container

- ✓ The liveview data is downloaded as one data stream, by HTTP GET.
 - The smallest unit is called "Packet" as the diagram below indicates. During the download, this "Packet" will be repeated.
 - The client should keep on downloading the data of "Packet", and extract JPEG image from one "Packet", and decode it, and show it to the display.
 - The client may not display all the JPEG because of the decoding time. In that case, the client should skip some JPEG images.
- ✓ The endian of the data is network byte order.
- ✓ Refer to the following sections for more information about Common Header and Payload.



Common Header

Common Header is constructed of the following 8 Bytes.

- ✓ Start byte : 1 [B]
 - 0xFF, fixed
- ✓ Payload type : 1 [B]
 - indicates type of the Payload
 - **0x01 = For liveview images**
 - **0x02 = For Liveview Frame Information**
- ✓ Sequence number : 2 [B]
 - Frame No, 2 bytes integer and increments every frame
 - This frame no will be repeated.
- ✓ Time stamp : 4 [B]
 - 4 bytes integer, the unit will be indicated by Payload type
 - In case Payload type = 0x01, the unit of the Time stamp of the Common Header is milliseconds. The start time may not start from zero and depends on the server.

Payload Header

Payload header format will be as following 128 Bytes.

- ✓ Start code : 4[B]
 - fixed (0x24, 0x35, 0x68, 0x79)
 - This can be used for detection of the payload header.
- ✓ Payload data size without padding size : 3[B]
 - Bytes.
 - In case Payload Type = 0x01, the size indicates the size of JPEG in Payload data.
 - In case Payload Type = 0x02, the size indicates the size of Frame information data in Payload data.
- ✓ Padding size : 1[B]
 - Padding size of the Payload data after the JPEG data, Bytes.

In case Payload Type = 0x01, the header format will be as following.

- ✓ Reserved : 4[B]
- ✓ Flag : 1[B]
 - This value is set to 0x00
 - Other values is reserved
- ✓ Reserved : 115[B]
 - All fixed, 0x00

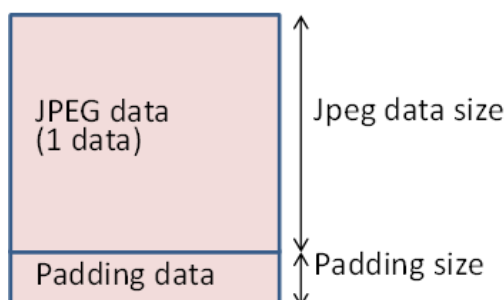
In case Payload Type = 0x02, the header format will be as following.

- ✓ Frame information data version: 2[B]
 - The data version of the frame information data.
 - The high order 1 byte indicates the major version and the low order 1 byte indicates the minor version.
 - 0x01, 0x00: version 1.0
 - The client should ignore the data that it does not understand by using the data version.
- ✓ Frame count : 2[B]
 - Number of the frame data.
- ✓ Single Frame data size: 2[B]
 - Single size of the frame data. In case the data version is 1.0, the data size is 16[B].
 - The client can read each frame by using the data size.
- ✓ Reserved : 114[B]
 - All fixed, 0x00

Payload Data

In case Payload type = 0x01

- ✓ JPEG data(1 data)
 - This size is indicated as "JPEG data size" in Payload Header.
- ✓ Padding data
 - This size is indicated as "Padding size" in Payload Header (No padding if 0 bytes was indicated).



In case Payload type = 0x02, single frame information data format is as following. The rectangular frame data will be repeated for the number of the frame information.

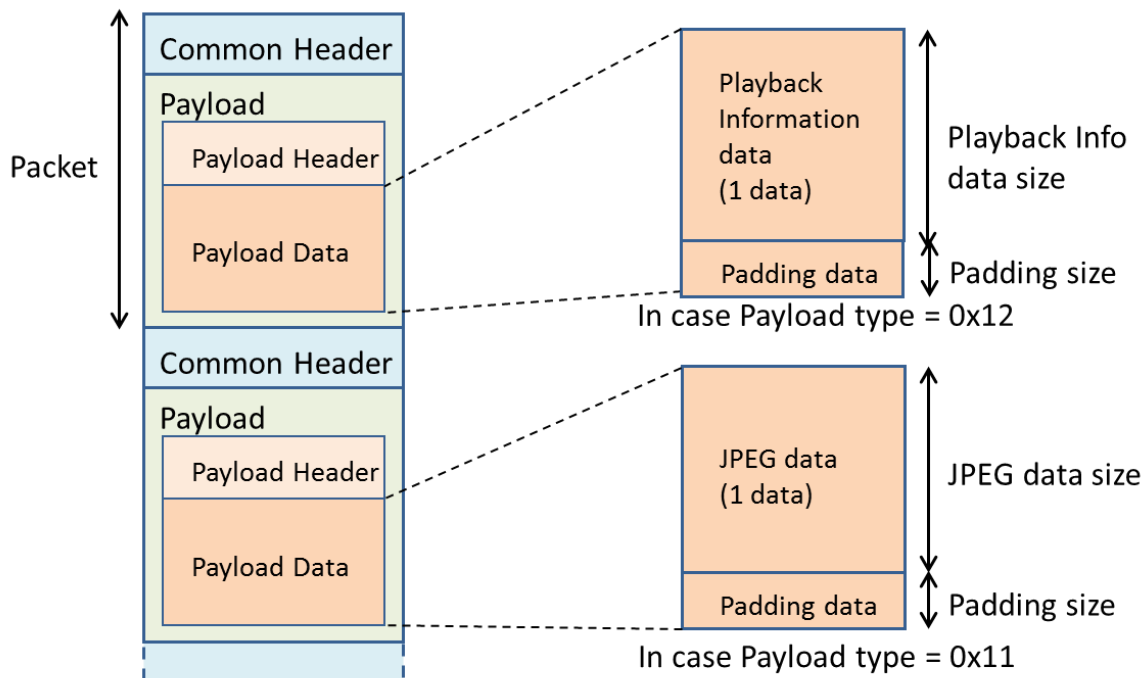
- ✓ Frame information data (single)
 - Top-left corner position of the frame : 4[B]
 - Bottom-right corner position of the frame : 4[B]
 - The high order 16 bits: X-axis position, the low order 16 bits: Y-axis position.
 - The corner position uses the upper left position of the liveview image as the coordinate origin.
 - The position is expressed between 0-10000 (when the width/height of the liveview is assumed to be 10,000.)
 - category : 1[B]
 - Refer to the following table for details.
 - status: 1[B]
 - Refer to the following table for details.
 - additional status: 1[B]
 - Refer to the following table for details.
 - Reserved : 5[B]
 - All fixed, 0x00
- ✓ Padding data
 - Byte size indicated as "Padding size" in Payload Header (No padding if 0[B] was indicated)

| category | status | additional status |
|--------------------------|------------------|-------------------|
| 0x00: Invalid | 0x00: Invalid | 0x00: Invalid |
| 0x01: Contrast AF | 0x01: Normal | 0x01: Selected |
| 0x02: Phase Detection AF | 0x02: Main | 0x02: Large Frame |
| 0x03: (Reserved) | 0x03: Sub | |
| 0x04: Face | 0x04: Focused | |
| 0x05: Tracking | 0x05: (Reserved) | |
| | 0x06: (Reserved) | |
| | 0x07: (Reserved) | |

Streaming data format

Format of the streaming data JPEG container

- ✓ Similar to [Liveview data format](#).
- ✓ The streaming data is downloaded as one data stream, by HTTP GET.
 - The smallest unit is called "Packet" as the diagram below indicates. During the download, this "Packet" will be repeated.
 - The client should keep on downloading the data of "Packet", and extract JPEG image from one "Packet", and decode it, and show it to the display.
 - The client may not display all the JPEG because of the decoding time. In that case, the client should skip some JPEG images.
- ✓ The endian of the data is network byte order.
- ✓ Refer to the following sections for more information about Common Header and Payload.



Common Header

Common Header is constructed of the following 8 Bytes.

- ✓ Start byte : 1 [B]
 - 0xFF, fixed
- ✓ Payload type : 1 [B]
 - indicates type of the Payload
 - **0x11 = For Streaming Images**
 - **0x12 = For Streaming Playback Information**
- ✓ Sequence number : 2 [B]
 - Frame No, 2 bytes integer and increments every frame
 - This frame no will be repeated.
- ✓ Time stamp : 4 [B]
 - 4 bytes integer, the unit will be indicated by Payload type

Payload Header

Payload header format will be as following 128 Bytes.

- ✓ Start code : 4[B]
 - fixed (0x24, 0x35, 0x68, 0x79)
 - This can be used for detection of the payload header.
- ✓ JPEG data size : 3[B]
 - Size of JPEG in Payload data, Bytes.
- ✓ Padding size : 1[B]
 - Padding size of the Payload data after the JPEG data, Bytes.

In case Payload Type = 0x11, the header format after the padding size will be as following.

- ✓ Image width: 2[B]
- ✓ Image height: 2[B]
- ✓ Reserved : 116[B]
 - All fixed, 0x00

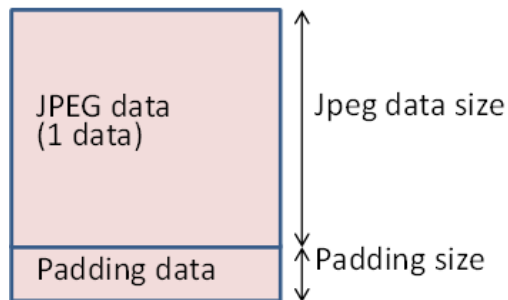
In case Payload Type = 0x12, the header format after the padding size will be as following.

- ✓ Playback information data version: 2[B]
 - The data version of the playback information data.
 - The high order 1 byte indicates the major version and the low order 1 byte indicates the minor version.
 - 0x01, 0x00: version 1.0
 - The client should ignore the data that it does not understand by using the data version.
- ✓ Reserved : 118[B]
 - All fixed, 0x00

Payload Data

In case Payload type = 0x11

- ✓ JPEG data(1 data)
 - This size is indicated as "JPEG data size" in Payload Header.
- ✓ Padding data
 - This size is indicated as "Padding size" in Payload Header (No padding if 0 bytes was indicated).



In case Payload type = 0x12

- ✓ Playback information data
 - Duration : 4[B]
Unit: millisecond
 - Playback position : 4[B]
Unit: millisecond
 - Reserved : 24[B]
All fixed, 0x00
- ✓ Padding data
 - Byte size indicated as "Padding size" in Payload Header (No padding if 0[B] was indicated)

Status code & Error

Major status codes are below. The "error" member is defined as [error_code, error_message].
The error_message may vary depending on the camera models.

- OK

```
"error": [0, "OK"]
```

- Any

A generic error code which can be used with any error.

```
"error": [1, "Any"]
```

- Timeout

```
"error": [2, "Timeout"]
```

- Illegal Argument

Parameters in "params" are illegal.

```
"error": [3, "Illegal Argument"]
```

- Illegal Data Format

```
"error": [4, "Illegal Data Format"]
```

- Illegal Request

When request body is empty, has no id or invalid id, has no method, has no parameter, or when "params" is not an array.

```
"error": [5, "Illegal Request"]
```

- Illegal Response

```
"error": [6, "Illegal Response"]
```

- Illegal State

```
"error": [7, "Illegal State"]
```

- Illegal Type

```
"error": [8, "Illegal Type"]
```

- Index Out Of Bounds

```
"error": [9, "Index Out Of Bounds"]
```

- No Such Element

```
"error": [10, "No Such Element"]
```

- No Such Field

```
"error": [11, "No Such Field"]
```

- No Such Method

For cases method is unmatched.

```
"error": [12, "No Such Method"]
```

- NULL Pointer

```
"error": [13, "Null Pointer"]
```

- Unsupported Version

```
"error": [14, "Unsupported Version"]
```

- Unsupported Operation

```
"error": [15, "Unsupported Operation"]
```

- Shooting fail

```
"error": [40400, "Shooting fail"]
```

- Camera Not Ready

```
"error": [40401, "Camera Not Ready"]
```

- Already Running Polling API

```
"error": [40402, "Already Running Polling Api"]
```

- Still Capturing Not Finished

```
"error": [40403, "Still Capturing Not Finished"]
```

- Some content could not be deleted.

```
"error": [41003, "Some content could not be deleted"]
```

In case HTTP status code is other than 200 OK, the error_code will be same as HTTP status code.

Here, major status codes are listed up.

- 401 Unauthorized

```
"error": [401, "Unauthorized"]
```

- 403 Forbidden

```
"error": [403, "Forbidden"]
```

- 404 Not Found

```
"error": [404, "Not Found"]
```

- 406 Not Acceptable

```
"error": [406, "Not Acceptable"]
```

- 413 Request Entity Too Large

```
"error": [413, "Request Entity Too Large"]
```

- 414 Request-URI Too Long

```
"error": [414, "Request-URI Too Long"]
```

- 501 Not Implemented

```
"error": [501, "Not Implemented"]
```

- 503 Service Unavailable

```
"error": [503, "Service Unavailable"]
```

JSON data types

For basic information about JSON data types, refer to RFC 4627. Camera Remote API adapts some extensions to keep APIs simple and easy-to-use.

Camera Remote API defines custom data types as below.

[boolean]: true or false value.

[integer]: Integer number, ranging from -2147483648 to 2147483647.

[double]: Double number, ranging from 2.2250738585072014e-308 to 1.7976931348623157e+308.

These three data types, plus [string], are primitive data types.

Camera Remote API also defines custom array types which only contains each of the above data types.

[boolean-array]: Multiple true or false values are stored in an array.

[integer-array] : Multiple integer numbers are stored in an array.

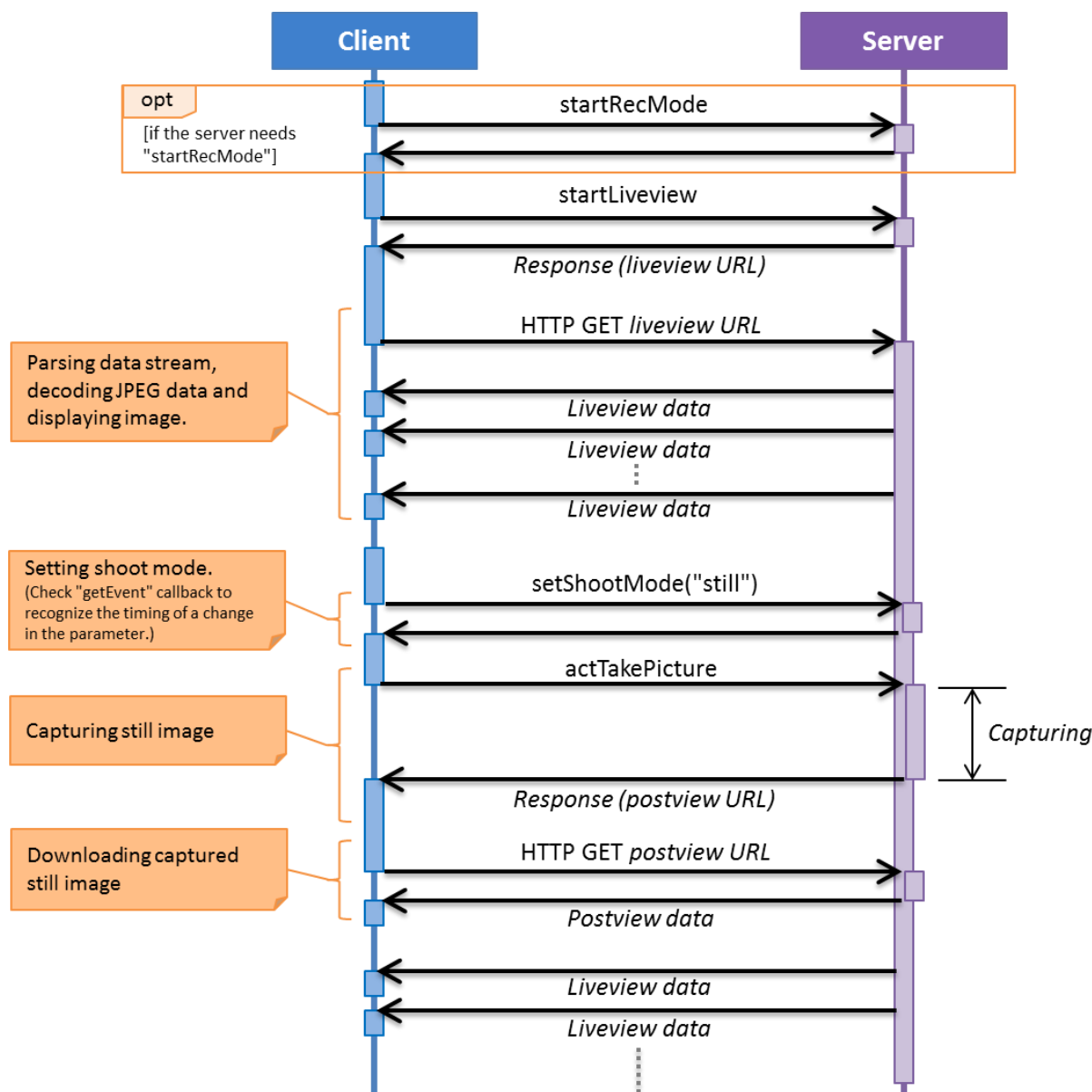
[double-array] : Multiple double numbers are stored in an array.

[string-array] : Multiple string data are stored in an array.

Sample Sequence

Here are sample sequences for some use cases.

Displaying liveview and capturing picture

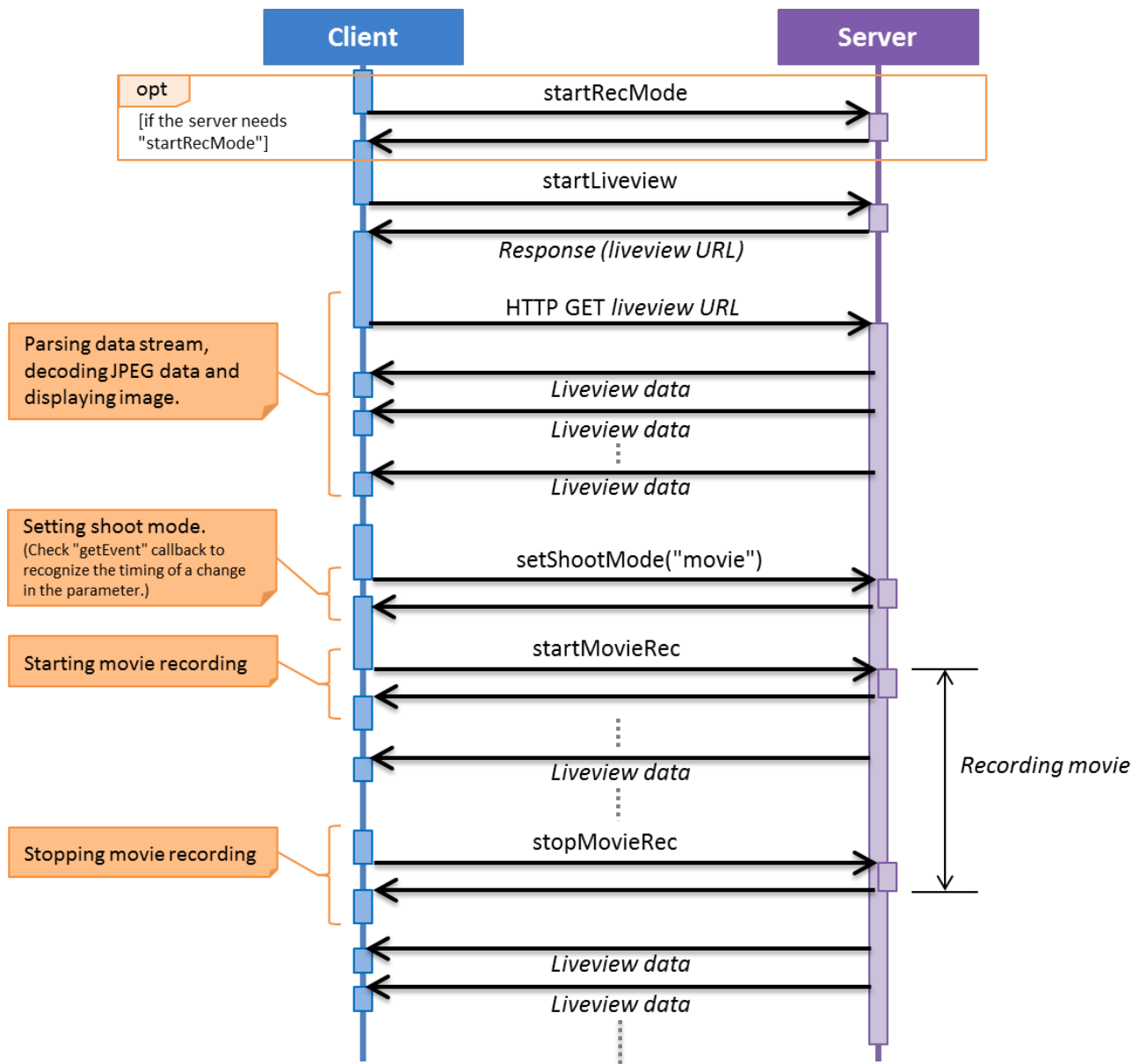


Some camera models need "[startRecMode](#)" API call before accessing camera shooting functions. The client must check if the server needs "[startRecMode](#)" API call. The check can be done by checking the availability of "[startRecMode](#)" API in "[getAvailableApiList](#)" or "[getMethodTypes](#)" callback.

To start liveview, the client should call "[startLiveview](#)" API and get liveview image data via the liveview URL in the response. The liveview data is kind of continuous images and the client can parse this data stream and decode each JPEG data. For more information about liveview data, see [Liveview data format](#).

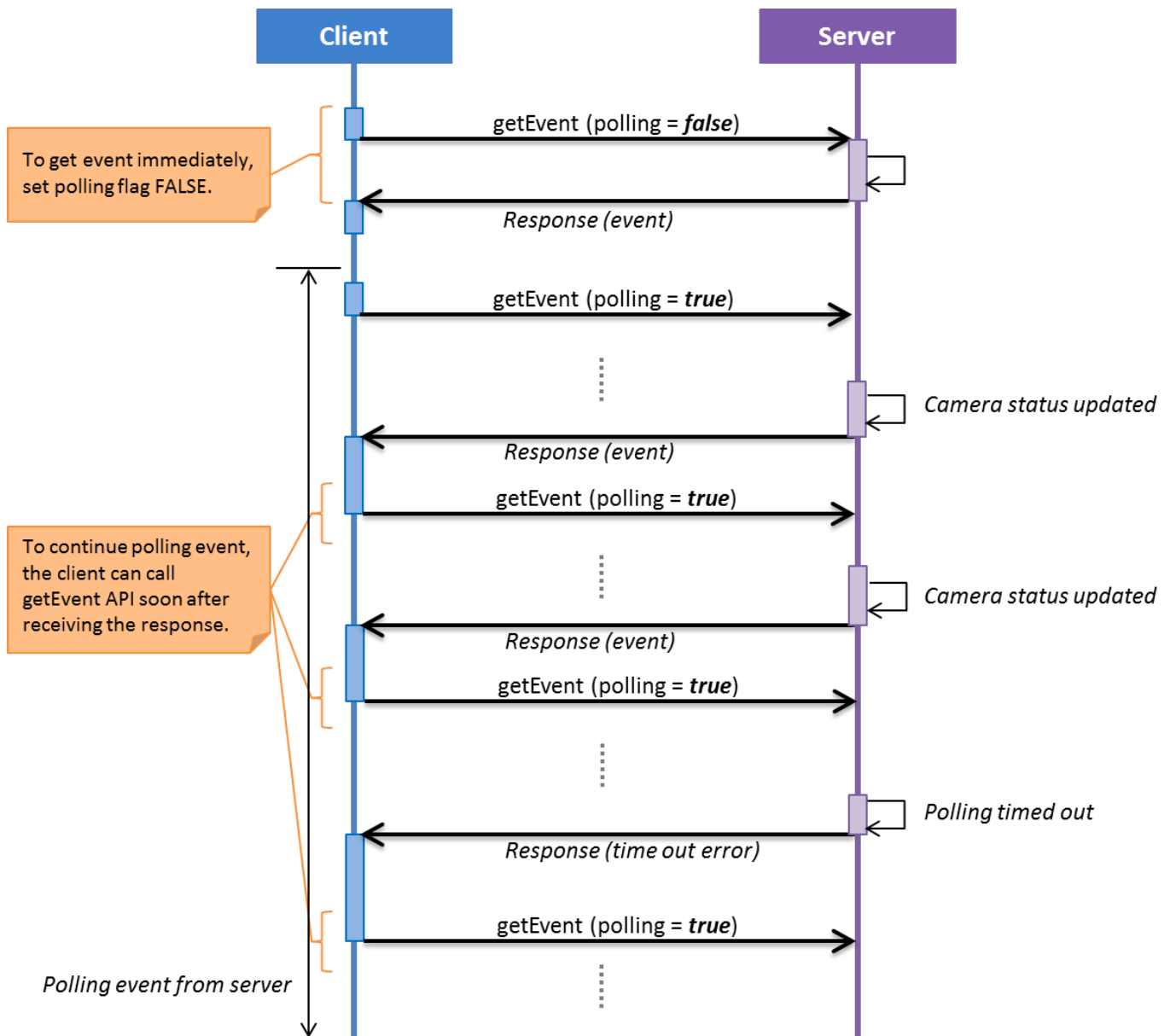
To capture still image, the client should call "[actTakePicture](#)" API in "still" shoot mode. The camera will provide the URL of postview in the response after capturing and the client can use it to display the postview image on the display and save it as the captured picture.

Recording movie



To record movie, the client should change shoot mode to "movie". The client should call "[startMovieRec](#)" API to start recording movie and call "[stopMovieRec](#)" API to stop.

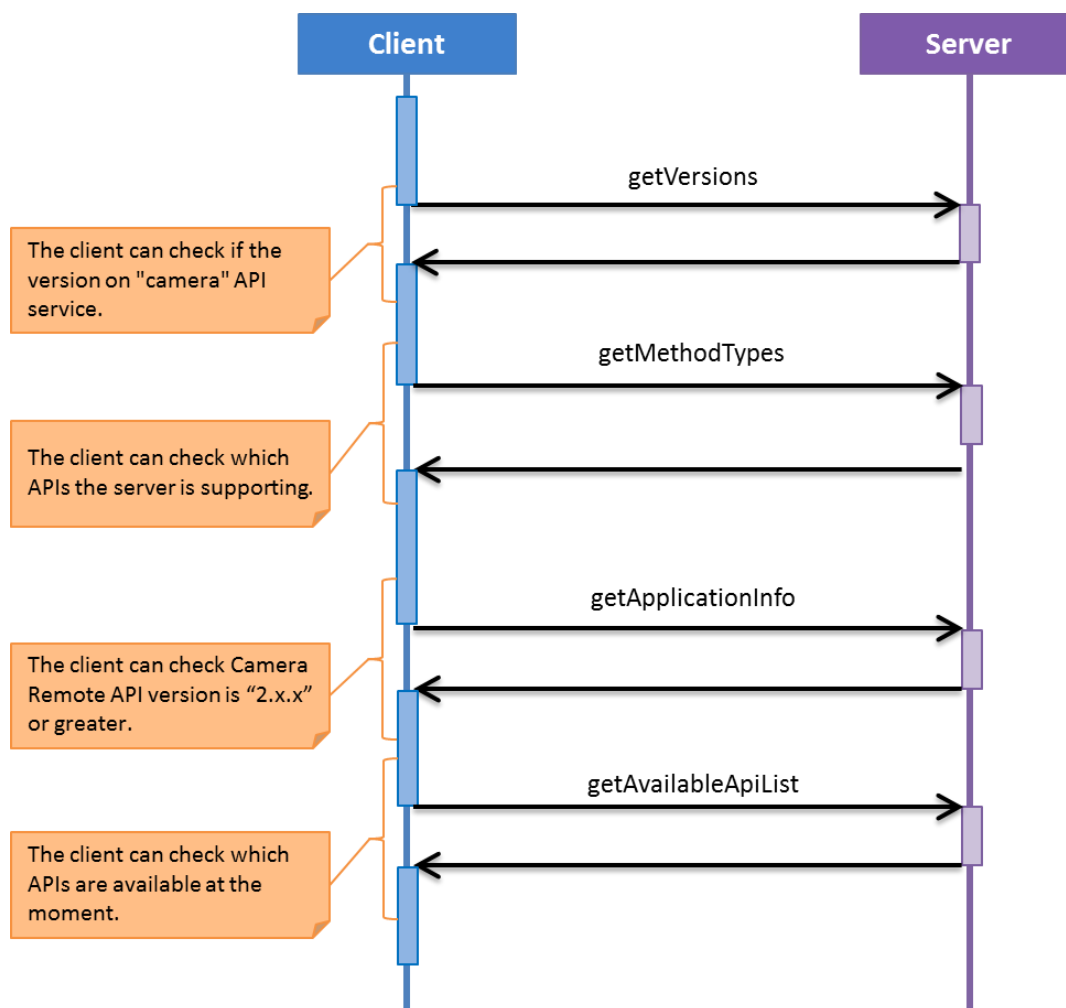
Getting event



The purpose of "[getEvent](#)" API is sending the event from the server actively. The client can recognize the current camera status. For example, when the client calls "Zoom" API, the server will send the event including the information about zoom position.

If "[getEvent](#)" is executed with "polling=false", the server replies immediately. If "[getEvent](#)" is executed with "polling=true", the server will response when one of server parameter is updated, or timed out.

Checking API version, supported APIs and available APIs

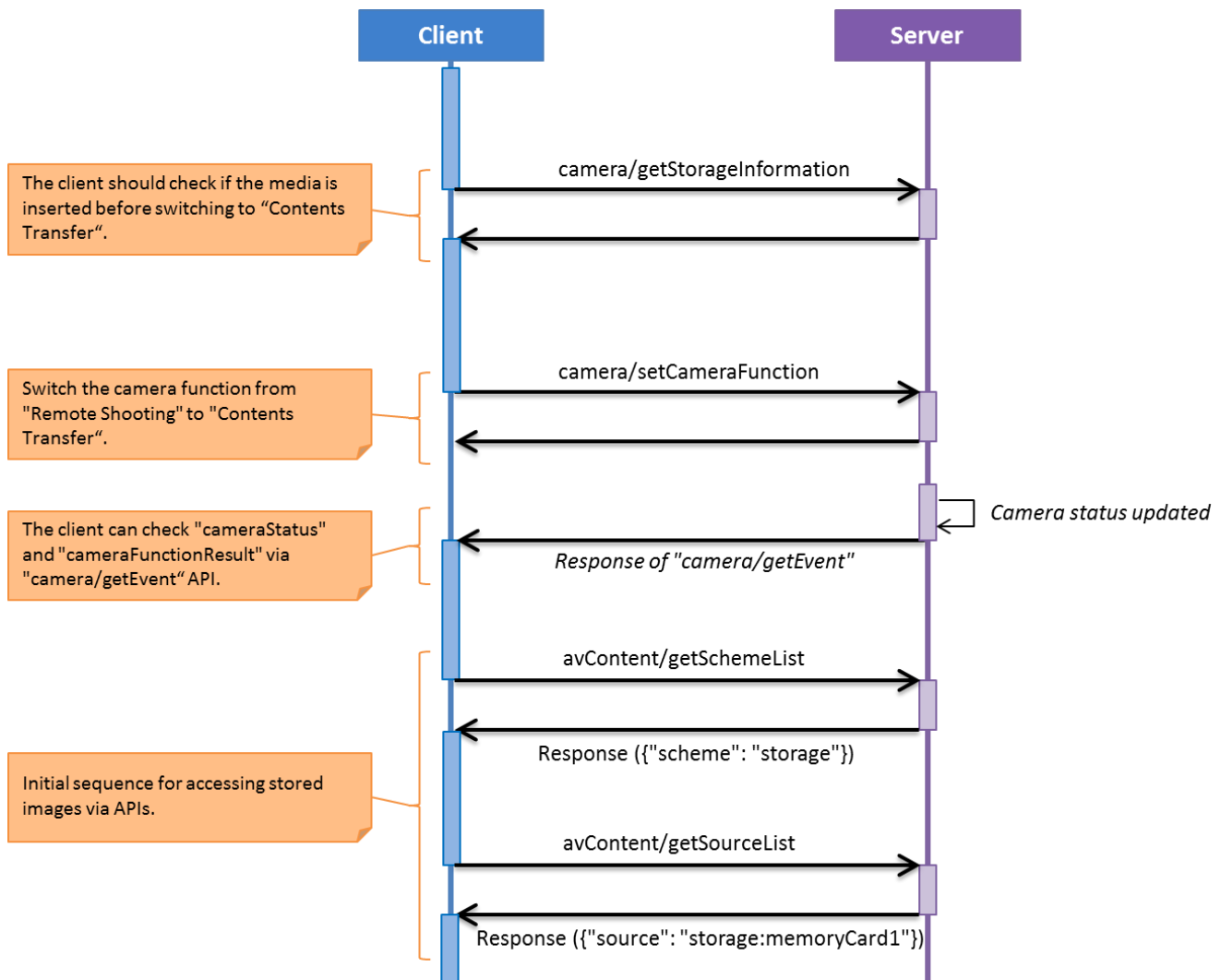


The "[getApplicationInfo](#)" API provides the Camera Remote API version which the camera supports. The client should check whether the API version is "2.x.x" ("2.0.0" or greater) to confirm the server function compatibility with this document.

The functions of camera vary by the model. Therefore the APIs that the camera supports vary by the model. The camera provides its supported API list via "[getMethodTypes](#)" API.

In addition, the camera status can be changed by user operations and calling APIs. Available APIs in the camera will be changed by camera status. The camera also provides available API list via "[getAvailableApiList](#)" API or "availableApiList" object of "[getEvent](#)" API callback. For more information, please see API specification.

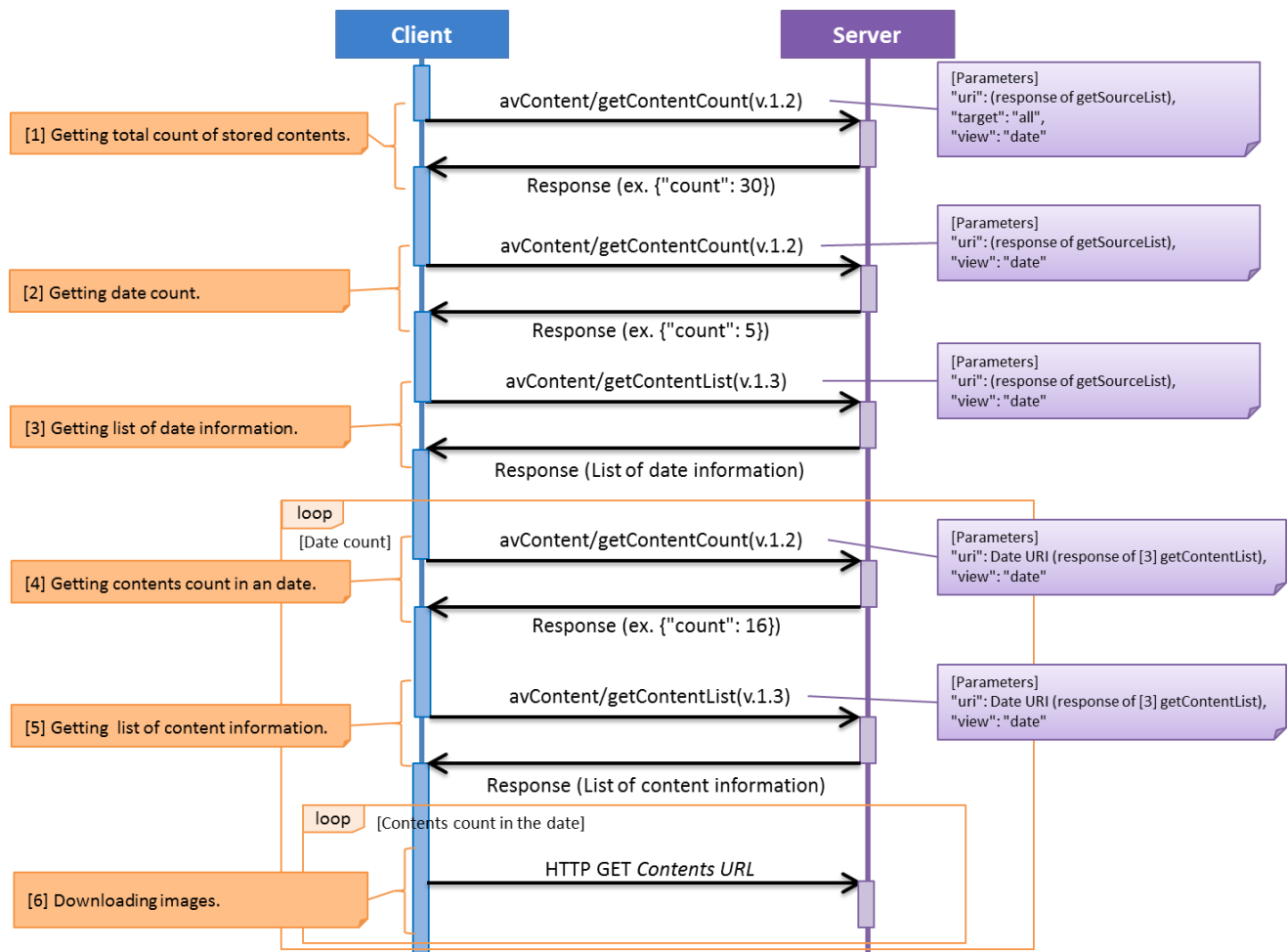
Changing camera function to transferring images



To access stored images, the client should change the camera function from "Remote Shooting" to "Contents Transfer" via "[setCameraFunction](#)" API. The client should check if the media is inserted in the camera via "[getStorageInformation](#)" API.

After that, "[getSchemeList](#)" and "[getSourceList](#)" of "avContent" API service will provide the source name to retrieve contents count and contents list.

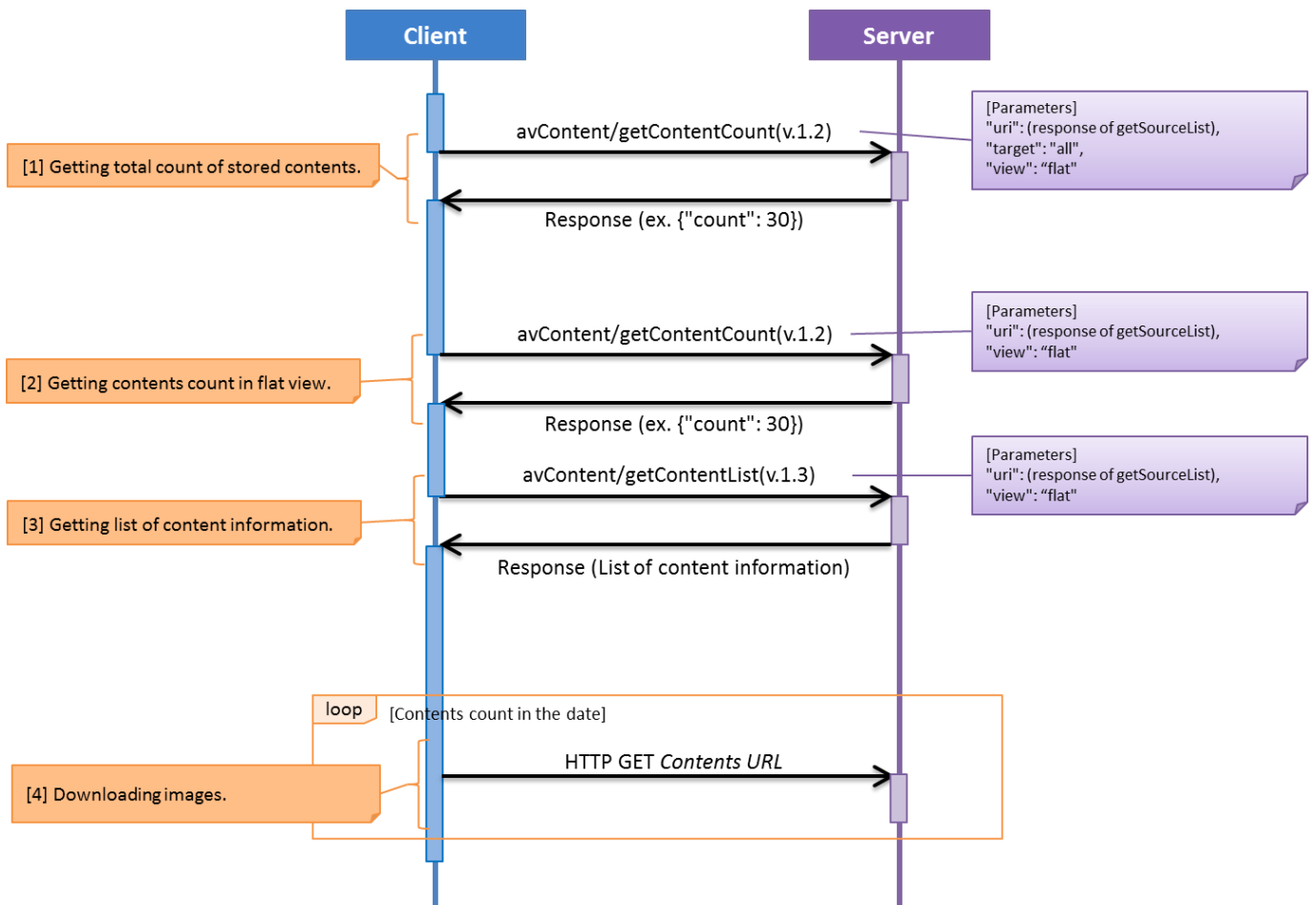
Transferring images (Date view)



In "date" view, the client can get count of dates via "[getContentCount \(v1.2\)](#)" and list of dates via "[getContentList \(v1.3\)](#)". Using the "uri" for specific date, the client can get count of contents and list of contents in the date.

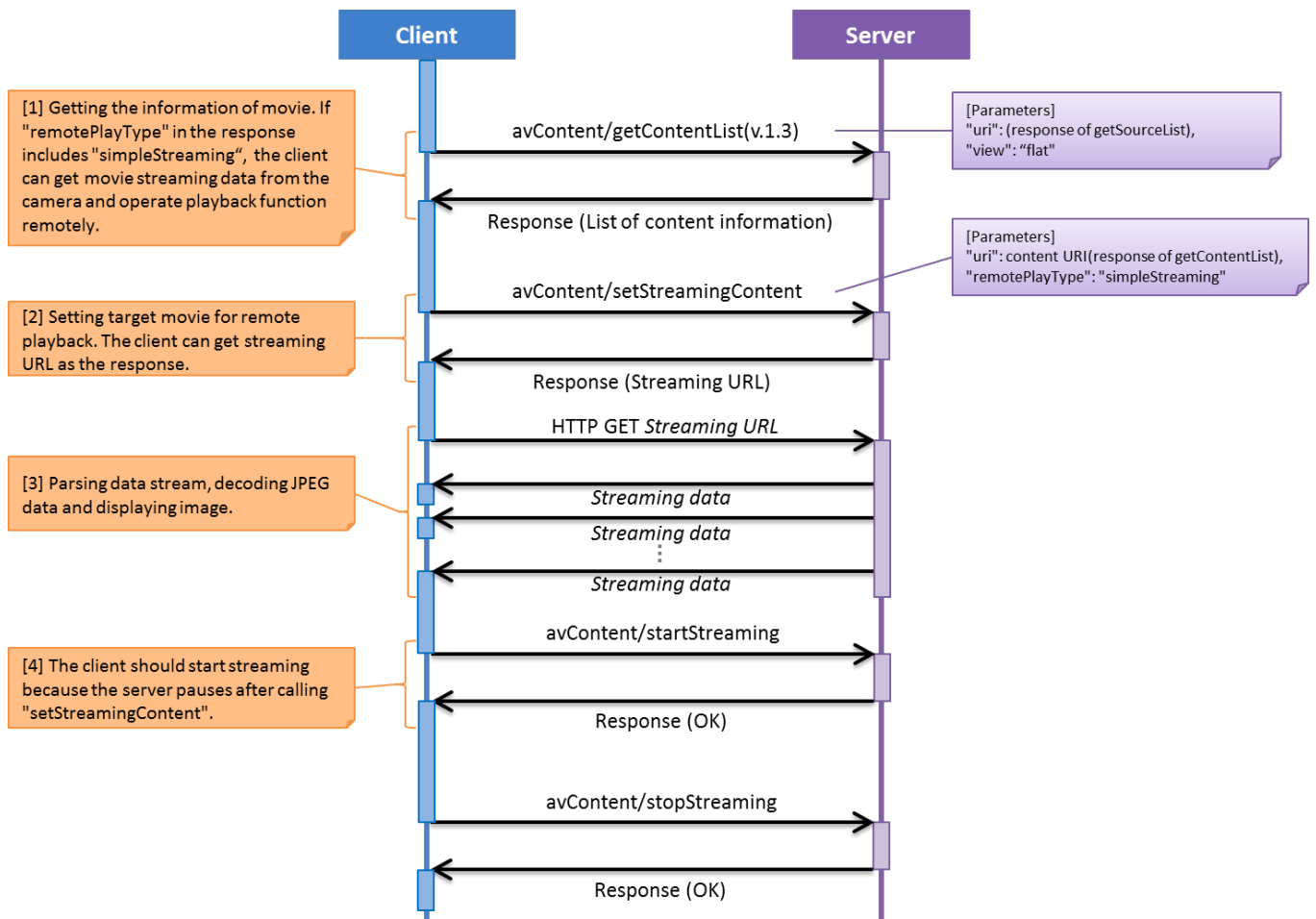
There are URLs for original still image or movie, resized data and thumbnails in the content information. The client can download images from camera via HTTP GET request.

Transferring images (Flat view)



The "flat" view has simple structure. Please refer to [the structure diagram](#).

Remote playback



The remote playback allows the client to play movie streaming from camera. The target movie is specified by "uri" which the client can get using "[getContentList \(v1.3\)](#)" API.

The streaming data format is similar to the liveview data format. Please refer to [Streaming data format](#).

More information

- IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels
<http://www.ietf.org/rfc/rfc2119.txt>
- IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1
<http://www.ietf.org/rfc/rfc2616.txt>
- IETF RFC 4627, The application/json Media Type for JavaScript Object Notation (JSON)
<http://www.ietf.org/rfc/rfc4627.txt>
- JSON-RPC
<http://json-rpc.org/>

For information regarding the latest Camera Remote API SDK updates, go to Developer World available at <http://developer.sony.com>

Trademarks and acknowledgements

Sony is a trademark or registered trademark of Sony Corporation.

Java is a trademark or registered trademark of Oracle Corporation.

Wi-Fi is a trademark or registered trademark of Wi-Fi Alliance.

All other trademarks and copyrights are the property of their respective owners