Developer World
developer.sony.com

September 2014

How to use the Camera Remote API beta

Camera Remote API Development Guide

SONY

First edition (December 2013)

Preface

Developer World

For the latest technical Sony 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 describe how to develop apps using the Camera Remote API provided by Sony. For API specifications, please see the API Reference document available in the Camera Remote API SDK.

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 1.0.0	First version
2013-12-01	Version 1.0.1	Improved instructions and new template
2014-04-15	Version 1.1.0	Added "system" API service to "API service" in Step 3.
2014-09-01	Version 1.2.0	Added "avContent" API service to "API service" in Step 3.

Contents

Introduction

The Camera Remote API allows you to extend or develop apps and services that can control a number of Sony cameras wirelessly from another device. This opens up for a number of exciting use cases, where you for example can remote control the zoom, viewfinder, image capture and video recording from an app on your phone, tablet or computer.

Since the Camera Remote API beta is platform agnostic, you can develop applications using it for any platform. The Camera Remote API beta SDK comes with sample code, a developer guide and API specifications to make it easy for you to get started.

Figure 1. Typical use case for a Camera Remote API enabled app: watch the viewfinder while somebody is taking a picture of you.

Publishing your app

Helping end users find your application

Once you have created your app and are about to upload it to an app store, please include the following string in your description (for example, in the bottom, on its own line):

```
Camera Remote API by Sony
```

This will help users find your application.

Compatibility with cameras

We recommend you to **add a note about the need for the end user to update the camera firmware in the description for your app**, when you publish it in an app store.

Access Sony cameras wirelessly with the Camera Remote API

Some Wi-Fi® enabled Sony cameras support Camera Remote API, which allows access to camera functionality wirelessly. This enables you to build remote control applications using Camera Remote API. In this document you'll learn all the steps necessary to enable the Camera Remote API in your app. For information about the API specification, please see the API Reference document also available in the Camera Remote API SDK.

Requirements

Camera Remote API is based on commonly-used protocols such as HTTP and JSON. You can develop your applications for your choice of operating systems.

Supported devices

For a full list of supported devices, please see the Device section on http://developer.sony.com. To see which APIs are supported for each compatible camera, please see the API Reference document also available in the Camera Remote API SDK.

How to access camera functions

There are three steps to access the camera functions from your app:

1) Connect the phone or tablet via Wi-Fi® to the Camera which acts as an access point.
2) From your app use SSDP (Simple Service Discovery Protocol) to discover the camera and get the URL needed to call the APIs.
3) Now your app can call the APIs by HTTP POST with JSON.

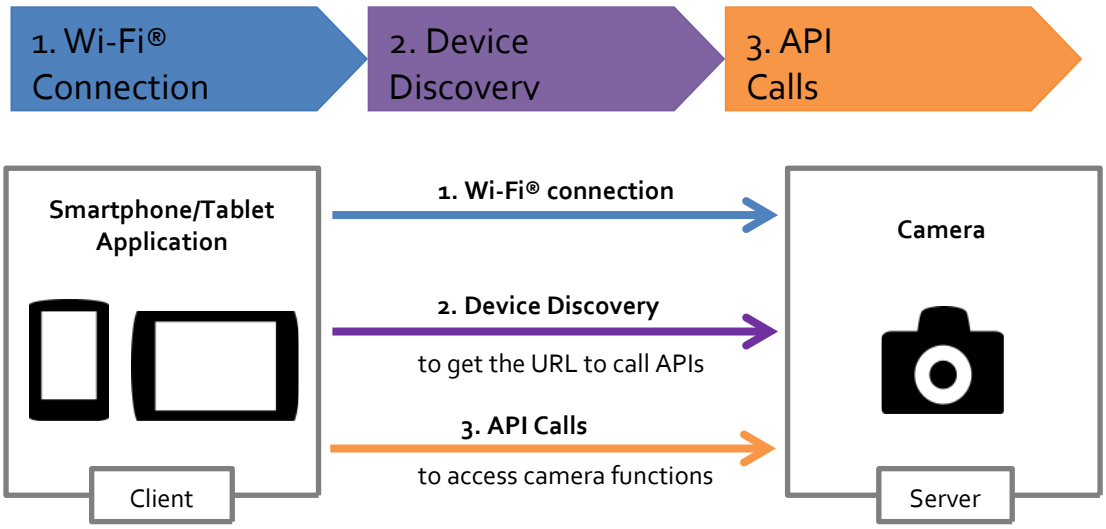


Figure 2. Overview of how to access the camera functions.

Step 1. Connect the phone or tablet to the camera over Wi-Fi®

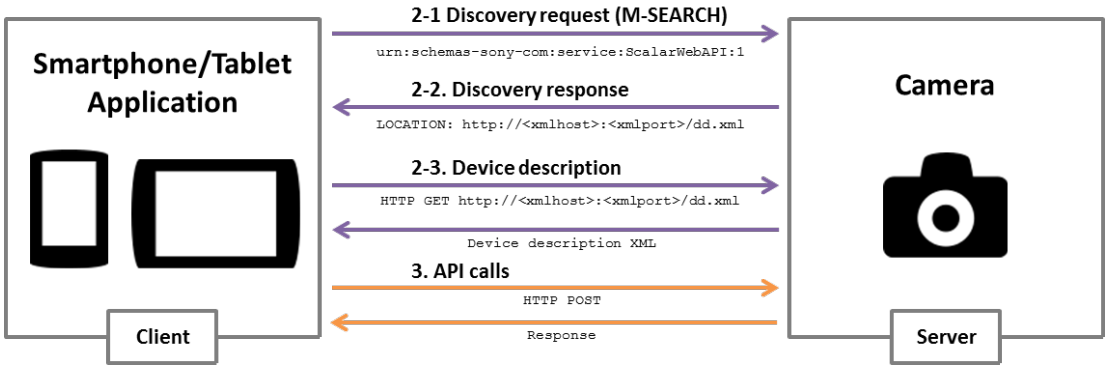
First, you need to enable the remote control function on your camera. The camera will then act as an access point.

Then you should connect your phone or tablet to the camera access point. On the phone or tablet, select the camera access point and input the password displayed on the camera or described in the manuals.

The remote device, for example your smartphone or tablet can now establish a one-to-one connection with the camera via Wi-Fi®. Your application is now be ready to work as a client.

Step 2. Use SSDP to discover the camera (Device Discovery)

Any camera supporting the Camera Remote API can be discovered using SSDP (Simple Service Discovery Protocol). Your smartphone or tablet client app can get the endpoint URL of the API, and information about supported functionalities, over device discovery according to the below illustration. For details please see the sample code included in the Camera Remote API SDK.



Please note that an entire UPnP stack is not required to support Camera Remote API. Only the SSDP portion of UPnP is required.

Discovery request (SSDP M-SEARCH)

Cameras supporting Camera Remote API are discovered via the SSDP M-SEARCH method, using the following specific search target (ST) header:

```
ST: urn:schemas-sony-com:service:ScalarWebAPI:1
```

All multicast messages are sent to the reserved address and port 239.255.255.250:1900.
Here is an example SSDP M-SEARCH for devices that support Camera Remote API. Please see UPnP specification for details.

Request example:

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: seconds to delay response (ex. MX: 1)
ST: urn:schemas-sony-com:service:ScalarWebAPI:1
USER-AGENT: OS/version product/version
```

For your reference, see also sample source code.

Discovery response

Cameras that implement the restrictions specified in the search target (ST) field will respond with a unicast response. Here is an example of the response.

Response example:

```
HTTP/1.1 200 OK
LOCATION: http://192.168.122.1:64321/dd.xml
CACHE-CONTROL: max-age=1800
EXT:
SERVER: OS/version UPnP/1.0 product/version
ST: urn:schemas-sony-com:service:ScalarWebAPI:1
```

The value of the LOCATION header points to an UPnP Device Description. The Device Description is an XML document. The value of the ST header is identical to the one sent in the M-SEARCH request. **Note that the value of LOCATION header may vary depending on camera models.**

Device description

This is the example of the Device description. ("X_ScalarWebAPI_DeviceInfo" section only.)

```
<av:X_ScalarWebAPI_DeviceInfo xmlns:av="urn:schemas-sony-com:av">
  <av:X_ScalarWebAPI_Version>1.0</av:X_ScalarWebAPI_Version>
  <av:X_ScalarWebAPI_ServiceList>
    <av:X_ScalarWebAPI_Service>
      <av:X_ScalarWebAPI_ServiceType>guide</av:X_ScalarWebAPI_ServiceType>
      <av:X_ScalarWebAPI_ActionList_URL>http://192.168.122.1:8080/sony</av:X_ScalarWebAPI_ActionList_URL>
    </av:X_ScalarWebAPI_Service>
    <av:X_ScalarWebAPI_Service>
      <av:X_ScalarWebAPI_ServiceType>camera</av:X_ScalarWebAPI_ServiceType>
      <av:X_ScalarWebAPI_ActionList_URL>http://192.168.122.1:8080/sony</av:X_ScalarWebAPI_ActionList_URL>
    </av:X_ScalarWebAPI_Service>
  </av:X_ScalarWebAPI_ServiceList>
</av:X_ScalarWebAPI_DeviceInfo>
```

The information of Camera Remote API is described by the following tags in "X_ScalarWebAPI_DeviceInfo" element.

Tag Name	Explanation
X_ScalarWebAPI_ServiceList	This element has the all provided "API service" information. Each "API service" is described by "*X_ScalarWebAPI_Service*" tag.
X_ScalarWebAPI_Service	This element has "API service" type and its URL of the service. This tag is sub element of "*X_ScalarWebAPI_ServiceList*".
X_ScalarWebAPI_ServiceType	Provided "API service" name. This tag is sub element of "*X_ScalarWebAPI_Service*". e.g) camera
X_ScalarWebAPI_ActionList_URL	Each "API service" is set under this URL. This tag is sub element of "*X_ScalarWebAPI_Service*". e.g.) http://192.168.122.1:8080/sony

For details about the "API service", see the API service section in this document.

After receiving the discovery response, the client should retrieve the Device Description (e.g. HTTP GET http://192.168.122.1:64321/dd.xml).

Cameras supporting Camera Remote API have "X_ScalarWebAPI_DeviceInfo" tag in Device Description. Its name space is "urn:schemas-sony-com:av".

```
<av:X_ScalarWebAPI_DeviceInfo xmlns:av="urn:schemas-sony-com:av">
```

The client can get the URL to call APIs from the information in the "X_ScalarWebAPI_DeviceInfo" element. For details, see the Endpoint URL section in this document.

Step 3. Call the APIs

Camera Remote API uses JSON-RPC over HTTP. **HTTP POST** is used for unidirectional request from client to camera as a server.

API service

Camera Remote API is categorized into API services based on functionalities.

API service	Explanation
`guide`	To get supported API services.
`camera`	To get/set camera settings, and access shooting functions.
`system`	To get/set system settings.
`avContent`	To access still images and movies stored in a media.

When the client calls the APIs to access shooting functionality, the client must use "camera" as API service.

For more information about APIs which belong to each service, see the API Reference document.

Endpoint URL

Endpoint URL is the URL to call APIs by HTTP POST. It is composed of two parts, "*ActionList_URL*" and "*API service*".

```
URL:  <ActionList_URL>/<API service>
```

Both ActionList_URL and API service are obtained through Device Discovery. The ActionList_URL is defined by the content of "`av:X_ScalarWebAPI_ActionList_URL`" tag in the Device Description. The API Service is defined by the content of "`av:X_ScalarWebAPI_ServiceType`" tag.

The ActionList_URL is determined by the same "`av:X_ScalarWebAPI_Service`" element that content of "`av:X_ScalarWebAPI_ServiceType`" tag is "*camera*".

Note that the ActionList_URL may vary depending on camera models.

Example:

```
<av:X_ScalarWebAPI_Service>
  <av:X_ScalarWebAPI_ServiceType>camera</av:X_ScalarWebAPI_ServiceType>
  <av:X_ScalarWebAPI_ActionList_URL>http://192.168.122.1:8080/sony</av:X_ScalarWebAPI_ActionList_URL>
</av:X_ScalarWebAPI_Service>
```

URL: **http://192.168.122.1:8080/sony/camera**

JSON-RPC and Extensions / Restrictions

Camera Remote API uses JSON-RPC with some extensions and restrictions. JSON-RPC is simple and light communication protocol. Below are some examples.

JSON-RPC Request example

```
{
    "method": "echo",
    "params": ["Hello Camera Remote API"],
    "id": 1,
    "version": "1.0"
}
```

JSON-RPC Response example

```
{
    "result": ["Hello Camera Remote API"],
    "error": null,
    "id": 1
}
```

Camera Remote API adapts some extensions and restrictions to keep APIs simple and easy-to-use.

Extensions

- If the request succeeds, "**error**" member is skipped in the response. On the other hand, if the request fails, "**result**" member is skipped.
- The "**error**" value is array and defined as **[`error_code`,`error_message`].**
  error_code is integer, error_message is string.
- **The client must set "`version`" member in the request**. The type of "version" value is string and the version can be set as 2 numbers separated by dot (e.g. "1.0"). For details about "version", see API Reference document.

Restrictions

- The "**params**"/"**result**" value is an array of fixed length. The length is defined on each API specification.
- Most of APIs will be replied by "**result**" member in those responses. Some of APIs will reply "**results**" member in those responses. See API Reference document.
- The "**id**" member is integer and "0" MUST NOT be used. The client can set "**id**" in the "1" to "2147483647" (0x00000001 to 0x7FFFFFFF).

Calling Camera Remote APIs

Finally, to call the APIs, the client can send JSON data to the endpoint URL of the camera by **HTTP POST**. After calling APIs, the client will get the response from camera.
For more information about APIs, please see the API Reference document.

Setting self-timer example

`Endpoint URL: http://192.168.122.1:8080/sony/camera`

Request example:

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

Response example (case of success):

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

Response example (case of failure):

```
{
    "error": [5, "Illegal Request"],
    "id": 1
}
```

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 Sony Camera Remote API SDK updates, go to Developer World, available at http://developer.sony.com.

Trademarks and acknowledgements