

# API GUIDE

## 1.WEB API

The Web API contains a set of REST API in order to interact with LIVERA in a convenient way through AJAX or URL .

**AJAX Example:**the below could be run in general font-end javascript client side.

```
var xhr = new XMLHttpRequest(){
  xhr.open('GET', '/hicat/videoServer?mode=0', true)
  xhr.onload = function() {
    answer = JSON.parse(xhr.responseText)
    console.log(answer1)
  }
  xhr.send()
}
```

**Websocket and bridge system:** bridge is a websocket server that runs on the linux core , help to accept websocket connections to Livera on port 7681 and transfer the information to MCU(32u4) .

**Websocket Example:**we've built a websocket bridge server to help to transport text message from front-end web client to Arduino.

```
//server on port 7681
var WebSocketURL = 'ws://' + window.location.hostname + ':7681'

ws = new WebSocket(WebSocketURL);

ws.onopen = function(event) {
  console.log('ws connection opened:' + WebSocketURL);
}
ws.send(`msg`);
ws.close();
```

**1.Set up Video Streaming Mode:** This API allows you to swith between RTSP and MJPEG streaming mode, be careful that Video Record API only works under RTSP Mode ,and Screen Shot API only works under MJPEG Mode .

```
/hicat/videoServer?mode=0
```

mode=0 : RTSP Server Start mode=1 : MJPEG Mode

**2.Station Mode:** Let Livera link to local wifi router

```
/hicat/stationMode?ssid=xxxxx&password=xxxxxxx
```

return messages: success : none return error :{"result":"ERROR"}

**3.WIFI Access Point:** Make Livera to AP mode

```
/hicat/apMode?ssid=xxxxx&password=xxxxxxx
```

return messages(object): success : none return error : {"result":"ERROR"}

**4.Set Livera time:** Synchronous time with Livera, the time also affect to the default video name.

```
/hicat/setTime?time=%d-%d-%d-%d-%d-%d
```

Example : /hicat/setTime?time=2018-09-09-01-01-01 success : {"result":"OK"}

error : {"result":"ERROR"}

**5.Get Video Download Links:** This API will return a JASON format( {"result":"OK", "files": [ "name": "xxxxx", "name": "xxxxx" ] } ) message about all the file within video folder in SD Card. Once you get the fileName you could just go to this url <http://192.168.1.1/mmc/video/xxxxx.264> to download it(might figure your IP address under station mode).

```
/hicat/files
```

success : {"result":"OK", "files": [ "name": "xxxxx", "name": "xxxxx" ] }

error : {"result":"ERROR"}

**6.Video Record:** This API allows you to record and delete the video and auto save into the default video folder in SD card.

```
/hicat/record?save=1  
/hicat/record?del=xxxxxxx.h264
```

save=1 : start recording save=0 : stop recording and save del=xxxxxxx.h264 : delete xxxxxxx.h264 success : {"result":"OK"}

error : {"result":"ERROR"}

**7.Video Setting:** Switch video angle, video resolution and video fps.

```
/hicat/setCamera?resolve=1&rotate=0&fps=20
```

success : {"result":"OK"}

error : {"result":"ERROR"}

//only work under mjpg mode

**8.Snapshot:** snap shot form one of the frame form MJPEG stream. We suggest to make the direction to /www/mmc/video/ coz the photo info could be receive under /hicat/files API.

```
/hicat/snapshot?name=xxx.jpg&dir=/www/mmc/video/
```

success : {"result":"OK"}

**9.For test:** test api, do nothing but test.

```
/hicat/test
```

success : {"result":"OK"}

## 2.Embedded Linux API

Detail information could be view under our github libhicat (<https://github.com/9crk/libhicat/blob/master/include/libhisiv.h>), There are instructions guide you through development tools set up and workflow, please have a look.

```
#ifndef _LIBHISIV_H_
#define _LIBHISIV_H_

#ifdef __cplusplus
extern "C" {
#endif
int venc_exit(int n);
int venc_init(int resolve);//0:720P 1:QVGA(320*240) 2:VGA(640*480)
int venc_init_more(int resolve,int mode,int fps);//resolve: 0/1/2 1280*720/320*240/640*480 mode: 0/1 H264/MJPEG
int venc_requestIDR();//request IDR frame
int venc_getFrame(char* buffer,int *datalen,int *pts,int *type);
int venc_snap(char* buff,int xRes,int yRes);
int venc_getYUV(int mode,char*buff);//mode=0 Y mode=1 UV mode = 3 YUV420(SP)
int venc_rotate(int dir);

//for audio
extern int aenc_init(int mode);// 0/1 PT_LPCM/AAC/
extern int aenc_getFrame(char* buff);
extern int aenc_exit();
//jpeg to http
extern int libyuvdist_startYuvDistService(int port);
extern int libyuvdist_updateYuv(int iHandle,char* data,int len,int seq,unsigned long timeStamp);
extern int libyuvdist_stopYuvDistService(int iHandle);
extern int libyuvdist_setSettingCallback(int iHandle,int func);//int func(int resX,int resY,int fps)

#ifdef __cplusplus
}
#endif
#endif
```

## 3.Arduino Libray API

The Arduino library wraps the web api, using serial communication and curl to communicate and send command to the video core(HI3518E). There are also example codes could be find in github ([https://github.com/hicat-tech/hicat\\_arduino\\_library](https://github.com/hicat-tech/hicat_arduino_library))

```

#ifndef _HICAT_H_
#define _HICAT_H_

#include <Arduino.h>

class HiCat
{
public:
    HiCat();
    int begin(void);

    /**
     * Take a picture
     *
     * @param file_name    picture name
     * @return 0 - OK, otherwise - error code
     */
    int snapshot(const char *file_name);

    /**
     * Start to record a video which is saved at /www/mmc/video/
     *
     * @return 0 - OK, otherwise - error code
     */
    int record();

    /**
     * Stop to record a video
     *
     * @return 0 - OK, otherwise - error code
     */
    int stop_recording();

    /**
     * Set camera format
     *
     * @param resolution    0 - 1280*720, 2 - 640*480, 1 - 320*240
     * @param rotation      0 - no rotation, 1 - 180 degree rotation
     * @param fps           frame per second from 1 to 25
     * @return 0 - OK, otherwise - error code
     */
    int set_camera(int resolution, int rotation, int fps);

    /**
     * Set Wi-Fi mode
     *
     * @param mode    0 - AP mode, 1 - Station mode
     * @param ssid    SSID
     * @param password password
     * @return 0 - OK, otherwise - error code
     */
    int set_wifi(int mode, const char *ssid, const char *password);

    /**
     * Run a shell command
     *
     * @param command    shell command
     * @return 0 - OK, otherwise - error code
     */
    int run(const char *command);

private:
    void prepare_web_command();
    int read_result();
};

extern HiCat hicat;

#endif // _HICAT_H_

```