

FF JSON Protocol

Version 1.0

Updated:	2018-03-29
Latest version always at:	https://goo.gl/ZLxizr *

Event details packed into JSON document are sent with multipart POST HTTP requests.

HTTP POST request parts:

- Event JSON
- Image associated with event

JSON Fields	1
JSON example	3
Example of C-code for sending events	4
HTTP headers dump	5
Simple php receiver	6
Sending messages with CURL executable	7

JSON Fields

Field name	Type	Format / Value	Description
packetCounter	Unsigned int	0	Counter of packets. Starts from "1". Counter starts after successful connection or after start of device. It does not matter.
datetime	string	YYYYMMDD HHMMSSFFF+HHMM	Event timestamp (UTC). YYYY - year 4 digits MM - month 2 digits DD - day 2 digits <space> HH - hours MM - minutes SS - seconds FFF - milliseconds +-HHMM - time zone (optional) All values above are zero-padded.
plateText	Unicode string	\u0041\u0041....	Plate number in the Unicode translation https://www.branah.com/unicode-converter
plateCountry	ASCII string	DNK UKR FRA ...	Country code, ISO 3166, ALPHA3
plateConfidence	float	0.808824	Recognition confidence level

carState	ASCII string	"new" "updated" "lost"	Event type. NumberOk does not send "lost" events.
geotag: { "lat":, "lon": }	Json object with floats as values	50.418114,30.476213	Latitude, longitude
imageType	ASCII string	"plate" "frame"	The kind of attached image: plate only rectangle with plate number, frame - full frame.
plateImageType	ASCII string	"png" "jpeg" "bmp"	The type of attached image. HTTP header must use the same type with header Content-type: image/...
plateImageSize	Unsigned long	0	The size in bytes of attached image, may be zero.
carMoveDirection	string	"null" "in" "out"	The kind of detected movement
timeProcessing	Unsigned int	459	Time of image processing, ms
plateCoordinates	Array of unsigned int	[800, 545, 150, 20] [left,top,width,height]	Coordinates of license plate bounding box upper left corner and its dimensions
carID	Unsigned long long		Internal runtime plate ID
GEOtarget	Unicode string		Channel description
sensorProviderID	ASCII string		The human-readable name of the sender
channel	Unsigned int		Channel number. Starts from 1. Camera apps set the value to 1 or omit the field. NumberOk server based app sets the value as per channel configuration.
zone	Unsigned int		Zone number per given channel. Starts from 1. Camera apps may omit.

JSON example

```
{
  "packetCounter": "0",
  "datetime": "20170221 081331000",
  "plateText": "\u0059\u0058\u0034\u0035\u0036\u0037\u0031",
  "plateCountry": "DNK",
  "plateConfidence": "0.808824",
  "carState": "new",
  "geotag": {
    "lat": 50.418114,
    "lon": 30.476213
  },
  "imageType": "plate",
  "plateImageType": "png",
  "plateImageSize": "0",
  "carMoveDirection": "in",
  "timeProcessing": "409",
  "plateCoordinates": [0, 680, 440, 272],
  "carID": "27",
  "GEOtarget": "",
  "sensorProviderID": "SAMSUNG00",
  "channel": "1"
}
```

JSON validation tool: <http://jsonlint.com/>

Example of C-code for sending events

This example uses cross platform library **curl** (<https://curl.haxx.se/>).

Before sending event (JSON data) is saved to file "json_name".

Image is in lp_image.

Url is in cld.cloud_url.

```
CURL *curl;
CURLcode res;
struct curl_httppost *formpost=NULL;
struct curl_httppost *lastptr=NULL;
std::string cloud_url = "sovabi-dev.cloudapp.net:8081/receiver/";
std::string lp_image = "etalon_hd.jpg";
std::string json_name = "test_json.json";

curl_global_init(CURL_GLOBAL_ALL);

/* Fill in the file upload field */
curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "event",
             CURLFORM_FILE, json_name.c_str(),
             CURLFORM_END);

curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "image",
             CURLFORM_FILE, lp_image.c_str(),
             CURLFORM_END);

curl = curl_easy_init();
curl_easy_setopt(curl, CURLOPT_URL, cloud_url.c_str() );
curl_easy_setopt(curl, CURLOPT_HTTPPOST, formpost );

res = curl_easy_perform( curl );

If (res != CURLE_OK)
{
    fprintf(stderr, "curl_easy_perform() failed: %s\n",curl_easy_strerror( res ) );
}

curl_easy_cleanup(curl);
curl_formfree(formpost);
```

HTTP headers dump

```
=> Send header, 227 bytes (0xe3)
0000: POST /ff_post.php HTTP/1.1
0025: User-Agent: curl/7.35.0
003e: Host: 10.0.3.160
0050: Accept: */*
005d: Content-Length: 377541
0075: Expect: 100-continue
008b: Content-Type: multipart/form-data; boundary=-----
00cb: ----c83536ed0d118f9a
00e1:..
<= Recv header, 23 bytes (0x17)
0000: HTTP/1.1 100 Continue
=> Send data, 152 bytes (0x98)
0000: -----c83536ed0d118f9a
002c: Content-Disposition: form-data; name="event"; filename="14703115
006c: 85.json"
0076: Content-Type: application/json
0096:..
=> Send data, 337 bytes (0x151)
0000: {"packetCounter":"1","datetime":"20160805 145305000","plateText"
0040: :"\u4100\u4100\u3400\u3200\u3900\u3500\u4f00\u4b00","plateCountr
0080: y":"UKR","plateConfidence":"0.852113","carState":"new","carMoved
00c0: irection":"IN","timeProcessing":"20","plateCoordinates":[1233, 1
0100: 429, 104, 144],"carID":"1","GEOtarget":"Office","sensorProviderI
0140: D":"CameraTest2"}
=> Send data, 146 bytes (0x92)
0000:..
0002: -----c83536ed0d118f9a
002e: Content-Disposition: form-data; name="image"; filename="etalon_h
006e: d.jpg"
0076: Content-Type: image/jpeg
0090:..
=> Send data, 16384 bytes (0x4000)
0000: .....JFIF.....H.H.....C.....
0040: .....!"$.$.8.....
0080: .....J.....!..1AQ"a..2q#B..R.....3.$Cb....
....
....
3f80: f...kq.q{1..).~...(.j.FU..J..K(..E..d.3..|..tp.....#yV9...>..cqH.
3fc0: ..$.f.Bh..v*.N..P....I#.9.....Dc.Z...&.9].....]..#.v%..R...
=> Send data, 48 bytes (0x30)
0000:..
0002: -----c83536ed0d118f9a--
```

Simple php receiver

The following simple receiver may be used with any HTTP server that supports PHP extension.

ff_post.php

```
<?php
$uploaddir = 'log/';
echo "<pre>\n";
$uploadfile = $uploaddir . basename($_FILES['event']['name']);
if (move_uploaded_file($_FILES['event']['tmp_name'], $uploadfile)) {
    echo "File json uploaded.\n";
} else {
    echo "Upload report error!\n";
}
$uploadfile = $uploaddir . basename($_FILES['image']['name']);

if (move_uploaded_file($_FILES['image']['tmp_name'], $uploadfile)) {
    echo "File image uploaded.\n";
} else {
    echo "Upload image error!\n";
}

echo 'Debug info:\n';
print_r($_FILES);
echo "</pre>\n";
?>
```

Simple Web form to send events:

ff_post_form.html

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
</head>
<form method="post" enctype="multipart/form-data" action="ff_post.php">
    Enter event file: <input type="file" name="event" size="40">
    Enter image file: <input type="file" name="image" size="40">
    <input type="submit" value="send" name="submit">
</form>
</html>
```

Sending messages with CURL executable

Provided we have following files

1470311582.json - event

etalon_hd.jpg - image

```
curl --trace-ascii headers.txt -i -X POST -F "event=@1470311582.json" -F  
"image=@etalon_hd.jpg" http://sovabi-dev.cloudapp.net:8081/receiver/
```

headers.txt - headers log

Result from data being sent to test server

```
curl --trace-ascii headers.txt -i -X POST -F  
'event=@1470311582.json;type=application/json' -F image=@etalon_hd.jpg  
http://10.0.3.160/ff_post.php
```

```
HTTP/1.1 100 Continue  
HTTP/1.1 200 OK  
Date: Tue, 21 Feb 2017 12:23:22 GMT  
Server: Apache/2.4.7 (Ubuntu)  
X-Powered-By: PHP/5.5.9-1ubuntu4.17  
Vary: Accept-Encoding  
Content-Length: 494  
Content-Type: text/html
```

<pre>

```
File json uploaded.
```

```
File image uploaded.
```

```
Debug info:\nArray
```

```
(  
    [event] => Array  
    (  
        [name] => 1470311582.json  
        [type] => application/json  
        [tmp_name] => /tmp/phpG2utRR  
        [error] => 0  
        [size] => 337  
    )  
  
    [image] => Array  
    (  
        [name] => etalon_hd.jpg  
        [type] => image/jpeg  
        [tmp_name] => /tmp/phpUURZBu  
        [error] => 0  
        [size] => 376858  
    )  
)  
)  
</pre>
```