

linkingvision

H5STREAM

API Manual

Copyright © 2018 linkingvision, All rights reserved

Revision History

Revision	Date	Description
1.00	2018/04/03	First version
1.01	2018/04/26	r4.2 update
1.0.2	2018/05/05	r5.0
1.0.3	2018/07/14	r6.0 split API document

Table of Contents

1.0	Release Notes	6
1.1	Version 1.00	6
2.0	Scope	6
3.0	References.....	6
4.0	Terms and Definitions.....	7
5.0	HTTP/HTTPS RESTFUL API	8
5.1	Protocol Syntax.....	8
5.2	System	8
5.3	Video source management.....	13
5.4	Extend ONVIF command	21
5.5	Record management	23
5.6	Cloud video management	30
6.0	JS API	33
6.1	JS object Parameter	33
6.2	Websocket	33
6.3	WebRTC.....	33
6.4	HLS.....	33
6.5	RTMP	34

1.0 Release Notes

1.1 Version 1.00

Initial version.

2.0 Scope

This specification defines live streaming scenario, and include install guide, development API interface. And introduce the best live streaming method for different browser, and then can achieve sub seconds low latency.

3.0 References

ONVIF <https://www.onvif.org/>

4.0 Terms and Definitions

5.0 HTTP/HTTPS RESTFUL API

RESTFUL API is based HTTP, all the API use HTTP GET method, and all the return of the API is JSON format. When develop or debug, suggest use HTTP, For the production, suggest use the HTTPS instead of HTTP.

5.1 Protocol Syntax

All HTTP request is based on GET, and all the response is JSON. And here is the basic syntax and example, for better understanding, some command will use the real value.

Syntax:

```
Request : http://server/api/v1/<resources>?para=xxx
          &<...>=<...>
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
\r\n
```

<... >=<... >stand for multi parameter.

Example:

```
Request :
http://localhost:8080/api/v1/AddSrcONVIF?name=name1&token=token4&user=admin&password=12345&ip=192.168.0.234&port=80&onvifaddr=/onvif/device_service&session=c1782caf-b670-42d8-ba90-2244d0b0ee83
Response:
{
  "bStatus": true,
  "strCode": "Add successfully"
}
\r\n
```

5.2 System

5.2.1 Login

URL: /api/v1/Login

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/Login?user=xxx&password=xxx
Response:
{
  "bStatus": true,
  "strSession": "c1782caf-b670-42d8-ba90-2244d0b0ee83",
  "nTimeout": 600
}
or
{
  "bStatus": false,
  "strCode": "xxxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>user</i>	<i>must</i>	<i>User name default is admin</i>
<i>password</i>	<i>must</i>	<i>pass default is 12345, the password is md5 hash of the real password.</i>

5.2.2 Logout

URL: /api/v1/Logout

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/Logout?session=xxxxxxxxxxxxxxxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.2.3 Keepalive

URL: /api/v1/Keepalive

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/Keepalive?session=xxxxxxxxxxxxxxxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.2.4 Get system info

URL: /api/v1/GetSystemInfo

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/GetSystemInfo?session=xxxxxxxxxxxxxxxxxxxx
Response:
{
  "strVersion": "r4.0.0403.18",
  "strHostId": "xxxxxxxxxx",
  "strLicenseType": "None",
  "strLicenseFull": "",
  "strChannelLimit": "unlimited",
  "strEndtime": "unlimited"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.2.5 Get run info

URL: /api/v1/GetRunInfo

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/GetRunInfo?session=xxxxxxxxxxxxxxxxxxxx
Response:
{
  "strRunTime": "0H 23MIN",
  "strCPU": "16%",
  "strMemory": "39%",
  "strNetworkIn": "0kpbs",
  "strNetworkOut": "0kpbs"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.2.6 Update User

URL: /api/v1/UpdateUser

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/
UpdateUser?user=admin&oldpassword=827ccb0eea8a706c4c34a16891f84e7b&newpa
ssword=xxxxxxx&session=xxxxxxx
Response:
{
  "bStatus": true,
  "strCode": "Update user successfully"
}\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>user</i>	<i>must</i>	<i>User name</i>
<i>oldpassword</i>	<i>must</i>	<i>Old password</i>
<i>newpassword</i>	<i>must</i>	<i>New password</i>

5.3 Video source management

5.3.1 Get source list

URL: /api/v1/GetSrc

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/GetSrc?token=xxx&session=xxxxxxx
Response:
{
  "src": [
    {
      "nType": "H5_ONVIF",
      "strName": "name1",
      "strToken": "token1",
      "strUrl": "rtsp://192.168.0.1/stream",
      "strUser": "admin",
      "strPasswd": "*****",
      "bPasswdEncrypt": false,
      "bEnableAudio": false,
      "nConnectType": "H5_AUTO",
      "nRTSPType": "H5_RTSP_AUTO",
      "strSrcIpAddress": "192.168.100.184",
      "strSrcPort": "80",
      "nChannelNumber": 1,
      "bOnvifProfileAuto": true,
      "strOnvifAddr": "/onvif/device_service",
      "strOnvifProfileMain": "Profile_1",
      "strOnvifProfileSub": "Profile_2",
      "bOnline": true,
      "strSnapshotUrl": "/api/v1/GetImage?token=token1",
      "bRec": false,
      "strServerToken": "",//the token is the register h5stream
      "strOriginalToken": "",
      "strServerUuid": "" }
    ]
  }
}
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>optional</i>	<i>The source token need to get, is no this parameter, stand for get all.</i>

5.3.1 Add file source

URL: /api/v1/AddSrcFile

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/
AddSrcFile?name=xxx&token=xxx&url=xxx&session=xxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>source token</i>
<i>name</i>	<i>must</i>	<i>source name</i>
<i>url</i>	<i>must</i>	<p>File path in the server, need use url encode to encode</p> <p>https://meyerweb.com/eric/tools/dencoder/ C:\xdev\h5s-r4.0.0403.18-win64-release\h5sample.mp4 encode to</p> <p>C%3a%5cxdev%5ch5s-r4.0.0403.18-win64-release%5ch5sample.mp4</p>
<i>servertoken</i>	<i>optional</i>	<p>Added to h5stream that register to this.(next version)</p> <p>And the token will be map to another one, need get again.</p>

5.3.2 Add RTSP RTMP source

URL: /api/v1/AddSrcRTSP

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/AddSrcRTSP?name=xxx&token=xxx
        &user=admin&password=12345&url=rtsp://192.168.0.1/stream&session=x
xxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>source token</i>
<i>name</i>	<i>must</i>	<i>source name</i>
<i>user</i>	<i>optional</i>	<i>RTSP username</i>
<i>password</i>	<i>must</i>	<i>RTSP password</i>
<i>url</i>	<i>must</i>	<p><i>File path in the server, need use url encode to encode</i></p> <p>https://meyerweb.com/eric/tools/dencoder/</p> <p><i>rtsp://192.168.0.23:554/Streaming/Channels/101</i></p> <p><i>?transportmode=unicast&profile=Profile_1</i></p> <p><i>Encode to:</i></p> <p><i>rtsp%3a%2f%2f192.168.0.23%3a554%2fStreaming</i></p> <p><i>%2fChannels%2f101%3ftransportmode%3dunicast</i></p> <p><i>%26profile%3dProfile_1</i></p>
<i>servertoken</i>	<i>optional</i>	<i>Add to register h5stream(next version)</i>

5.3.3 Add ONVIF source

URL: /api/v1/AddSrcONVIF

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/AddSrcONVIF?name=name1&token=token1
&user=admin&password=12345&ip=192.168.0.1
&port=80&onvifaddr=/onvif/device_service&main=Profile_1&sub=Profile_2&se
ssion=xxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>source token</i>
<i>name</i>	<i>must</i>	<i>source name</i>
<i>user</i>	<i>must</i>	<i>ONVIF username</i>
<i>password</i>	<i>must</i>	<i>ONVIF password</i>
<i>ip</i>	<i>must</i>	<i>ONVIF device ip address</i>
<i>port</i>	<i>must</i>	<i>ONVIF device port</i>
<i>onvifaddr</i>	<i>optional</i>	<i>ONIVF service address</i> <i>Default is /onvif/device_service</i>
<i>main</i>	<i>optional</i>	<i>Main stream token</i>
<i>sub</i>	<i>optional</i>	<i>Sub stream token</i>
<i>servertoken</i>	<i>optional</i>	<i>Add to register h5stream(next version)</i>

5.3.4 Delete source

URL: /api/v1/DelSrc

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/DelSrc?token=token1&session=xxxxxx  
Response:  
{  
  "bStatus": true/false,  
  "strCode": "xxxxxx"  
}  
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>deleted source token</i>

5.3.5 PTZ

URL: /api/v1/Ptz

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/  
Ptz?token=token1&action=left&speed=0.5&session=xxxxxx  
Response:  
{  
  "bStatus": true/false,  
  "strCode": "xxxxxx"  
}  
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>ONIVF source token</i>
<i>action</i>	<i>must</i>	<i>PTZ command</i> <i>up/down/left/right/</i> <i>zoomin/zoomout/stop</i> <i>/preset</i>
<i>speed</i>	<i>optional</i>	<i>stop does not need speed</i> <i>speed can be double 0-1</i>
<i>preset</i>	<i>optional</i>	<i>The preset goto</i>

5.3.1 Set preset

URL: /api/v1/ SetPreset

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/ SetPreset?token=token1&
&presetname=2&presettoken=2&session=xxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>source token</i>
<i>presetname</i>	<i>must</i>	<i>Preset name, such as 1, 2, 3</i>
<i>presettoken</i>	<i>must</i>	<i>Preset token, such as 1, 2, 3</i>

5.3.2 Delete preset

URL: /api/v1/DelPreset

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/DelPreset?token=token1&&
presettoken=2&session=xxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>source token</i>
<i>presettoken</i>	<i>must</i>	<i>preset token, such as 1, 2, 3</i>

5.4 Extend ONVIF command

5.4.1 Search ONVIF Device

URL: /api/v1/OnvifSearch

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/OnvifSearch?timeout=10&session=xxxxxx
Response:
{
  "device": [
    {
      "strIp": "192.168.0.23",
      "strPort": "80",
      "strOnvifAddr": "/onvif/device_service",
      "strModel": "xxx-xxx"
    }
  ]
}
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>timeout</i>	<i>optional</i>	<i>Searched time, in second</i> <i>Default is 10s</i>
<i>servertoken</i>	<i>optional</i>	<i>Search in the register h5stream(next version)</i>

5.4.2 Probe ONVIF device

URL: /api/v1/OnvifProbe

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```

Request :http://server/api/v1/
OnvifProbe?ip=192.168.0.1&port=80&user=admin&password=12345&onvifaddr=/o
nvif/device_service&session=xxxxxx
Response:
{
  "profile": [
    {
      "strToken": "Profile_1",
      "strCodecName": "H264",
      "strName": "H264 1920 x 1080 25fps 3072bps",
      "strRtspUrl":
"rtsp://192.168.0.23:554/Streaming/Channels/101?transportmode=unicast&pr
ofile=Profile_1",
      "nWidth": 1920,
      "nHeight": 1080,
      "nFps": 25,
      "nBandwidth": 0,
      "bGotUrl": true
    }
  ]
}
Or
{
  "bStatus": false,
  "strCode": "can't connect device"
}

```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>user</i>	<i>must</i>	<i>ONVIF device username</i>
<i>password</i>	<i>must</i>	<i>ONVIF device password</i>
<i>ip</i>	<i>must</i>	<i>ONVIF device ip address</i>
<i>port</i>	<i>must</i>	<i>ONVIF device port</i>
<i>onvifaddr</i>	<i>optional</i>	<i>ONVIF service address</i>
<i>servertoken</i>	<i>optional</i>	<i>Probe in the register h5stream(next version)</i>

5.5 Record management

5.5.1 Record

URL: /api/v1/Record

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/Record?token=token1&duration=100&filename=
xxxxx&&session=xxxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>
<i>duration</i>	<i>must</i>	<i>Record time</i>
<i>filename</i>	<i>optional</i>	<i>If the file name is blank, the server auto gen the file name. the file name does not have extend file name. all the record file is mp4.</i>

5.5.2 Manual Record Start

URL: /api/v1/ManualRecordStart

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/ManualRecordStart?token=token1&session=xxx  
xxx  
Response:  
{  
  "bStatus": true/false,  
  "strCode": "xxxxxxx"  
}  
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>

5.5.3 Manual Record Stop

URL: /api/v1/ManualRecordStop

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/ManualRecordStop?token=token1&session=xxxx  
xx  
Response:  
{  
  "bStatus": true/false,  
  "strCode": "xxxxxxx"  
}  
\r\n
```


Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>

5.5.4 Path record

URL: /api/v1/PathRecord

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/PathRecord?token=token1&start1=2018-3-29&start2=3-3-5&tz=8&duration=100&filename=xxxxx&alwayscreate=true
&session=xxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>
<i>duration</i>	<i>must</i>	<i>Record time</i>
<i>start1</i>	<i>must</i>	<i>Start time 1</i> <i>2018-3-29 y-m-d</i>
<i>start2</i>	<i>must</i>	<i>Start time 2</i> <i>20-3-29 h-m-s</i> <i>Client can access the video by</i> <i>blow url</i> <i>/mediastore/record/token1/2018-</i> <i>3-29TZ8/20-3-29/filename.mp4</i>
<i>tz</i>	<i>optional</i>	<i>8 or -8, default use the server</i> <i>time zone</i>
<i>filename</i>	<i>must</i>	<i>Record file name, does not have</i> <i>extend name.</i>
<i>alwayscreate</i>	<i>optional</i>	<i>true or false, default is false,</i> <i>always record when the</i> <i>recording is on-going, if false,</i> <i>just append the duration to</i> <i>current session</i>

5.5.5 Snapshot

URL: /api/v1/Snapshot

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/
Snapshot?token=token1&filename=xxxxx&&session=xxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>
<i>filename</i>	<i>optional</i>	<i>If the file name is blank, the server auto gen the file name. the file name does not have extend file name. all the record file is mp4.</i>

5.5.6 Path snapshot

URL: /api/v1/PathSnapshot

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/ PathSnapshot?token=token1&start1=2018-3-
29&start2=3-3-5&tz=8&filename=xxxxx&&session=xxxxxx
Response:
{
  "bStatus": true/false,
  "strCode": "xxxxxx"
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>
<i>start1</i>	<i>must</i>	<i>Start time 1</i> <i>2018-3-29 y-m-d</i>
<i>start2</i>	<i>must</i>	<i>Start time 2</i> <i>20-3-29 h-m-s</i> <i>Client can access the snapshot</i> <i>by blow url</i> <i>/mediastore/snapshot/token1/20</i> <i>18-3-29TZ8/20-3-</i> <i>29/filename.jpg</i>
<i>tz</i>	<i>optional</i>	<i>8 or -8, default use the server</i> <i>time zone</i>
<i>filename</i>	<i>must</i>	<i>Snapshot file name, does not</i> <i>have extend name.</i>

5.5.7 Search

URL: /api/v1/Search

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

Request :<http://server/api/v1/Search?type=xxx&start=2018-03-07T120101%2b08&end=2018-04-26T130101%2b08&token=xxx&session=xxxxxxx>
 Response:

```
{
  "record": [
    {
      "strToken": "token1",
      "strStartTime": "2018-04-25T19:32:05+08:00",
      "strDuration": "301",
      "strPath": "/mediastore/record/token1/2018-4-25TZ8/19-32-5/1234.mp4"
    },
    {
      "strToken": "token1",
      "strStartTime": "2018-04-25T19:37:05+08:00",
      "strDuration": "0",
      "strPath": "/mediastore/record/token1/2018-4-25TZ8/19-37-5/1234.mp4"
    }
  ]
}
```

\r\n

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>
<i>token</i>	<i>must</i>	<i>Source token</i>
<i>type</i>	<i>must</i>	<i>record or snapshot</i>
<i>start</i>	<i>must</i>	<i>ISO8601 format time</i> <i>2018-03-07T120101+08</i>
<i>end</i>	<i>must</i>	<i>ISO8601 format time</i> <i>2018-03-07T130101+08</i>

5.6 Cloud video management

5.6.1 Get cloud client info

URL: /api/v1/GetCloudClientInfo for internal network H5STREAM register status get.

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/GetCloudClientInfo&session=xxxxxx
Response:
{
  "bEnable": true,
  "strServerName": "Server 1",
  "strCloudIp": "10.0.0.1",
  "strCloudPort": "8080",
  "bSSL": false,
  "strUser": "admin",
  "bOnline": true
}
\r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.6.2 Cloud server list get

URL: /api/v1/GetServerList For cloud H5STREAM to get the server registered to the cloud.

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```

Request :http://server/api/v1/GetServerList&session=xxxxxxx
Response:
{
  "server": [
    {
      "strServerToken": "fc2f0aa4-86ca-4d75-ad29-59b91326dfae",
      "strServerIp": "192.168.0.1",
      "bOnline": true
    }
  ]
}
\r\n

```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.7 Stream management

5.7.1 Get WS connect status

URL: /api/v1/h5swsapi/Status

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```

Request :http://server/api/v1/h5swsapi/Status&session=xxxxxxx
Response:
{
  "nConnectCount": 2
} \r\n

```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

5.7.2 Get RTC connect status

URL: /api/v1/h5srtcapi/Status

Security level: Administrator, Operator, Viewer

Method: GET

Syntax:

```
Request :http://server/api/v1/h5srtcapi/Status&session=xxxxxxx
Response:
{
  "nConnectCount": 2
} \r\n
```

Parameter	Optional/Must	Description
<i>session</i>	<i>must</i>	<i>session id</i>

6.0 JS API

6.1 JS object Parameter

```
/**
@param
var conf = {
  videoid:'h5sVideo1', //{string} - id of the video element tag
  videodom: h5svideodom1, //{object} - video dom. if there has videoid, just use the videoid
  protocol: window.location.protocol, // {string} - http: or https:
  host: window.location.host, //{string} - localhost:8080
  rootpath>window.location.pathname, // {string} - path of the app running
  token:'token1', // {string} - token of stream
  hlsver:'v1', //{string} - v1 is for ts, v2 is for fmp4
  session:'c1782caf-b670-42d8-ba90-2244d0b0ee83' //{string} - session got from login
};
*/
```

6.2 Websocket

```
/**
* Interface with h5s websocket player API
* @constructor
*/
function H5sPlayerWS(conf)
H5sPlayerWS.prototype.connect
H5sPlayerWS.prototype.disconnect
```

6.3 WebRTC

```
/**
* Interface with h5s WebRTC player API
* @constructor
*/
function H5sPlayerRTC(conf)
H5sPlayerRTC.prototype.connect
H5sPlayerRTC.prototype.disconnect
```

6.4 HLS

```
/**
 * Interface with h5s websocket player API
 * @constructor
 */
function H5sPlayerHls(conf)
H5sPlayerHls.prototype.connect
H5sPlayerHls.prototype.disconnect
```

6.5 RTMP

For support old version web browser such as IE11, and H5STREAM Flash RTMP player based on videojs, and you can refer www.rtmp.html to a demo.