Watcher manual

2019-09-20

Integration of Flussonic Watcher SDK into Apps for Android
Integration of Flussonic Watcher SDK into Apps for iOS

Introduction to Flussonic Watcher

Flussonic Watcher

is a complete video surveillance software system that works with a distributed IP camera network. This system can be used for video streaming, recording and managing video archives.

Flussonic Watcher is a scalable and ready-to-use system with flexible integration options. This web and mobile-oriented solution can solve various business tasks: from launching a corporate cloud video monitoring to a municipal video surveillance system that covers the entire city or state.

Watcher supports small to medium to large projects with unlimited number of cameras. The number of cameras is limited only by hardware.

When working in a cluster of servers, Watcher ensures the fault-tolerance of stream ingest (failover).

Flussonic Watcher presentation

- Internet providers — to launch a dedicated cloud-based video surveillance service and offer it to their customers for an additional fee. Watcher can be easily integrated with any provider and its existing billing system.
- Management companies — for live video broadcasts from socially significant facilities and construction monitoring.
- Production — to perform audio-video monitoring on factories, courts, polling stations, etc.
- Municipal and federal projects — to provide free access to public cameras, as well as limited access to security organizations.

- The control module is a component that has a database of all users and cameras, user access management, archive quotas, cluster servers, provisioning and camera activation, as well as APIs for integration with client billing or custom module development.
- Web interface is the main interface for users and administrators. It works in all modern browsers and mobile devices. The interface includes a dashboard with cameras and archive, favorites, groups, map, user management tools and cameras, settings, interface branding tools and much more.
- iOS and Android mobile applications give you access to cameras and user archive from mobile devices.
- Firmware for cameras (agent) is an optional component. It can be installed on the cameras to provide access behind NAT with stable encryption and direct video delivery to Watcher.

Flussonic Watcher comes in two editions: Single and Cluster.

Single is good for smaller projects that have no more than 500 cameras and do not need interface branding (company logo, custom colors, etc.).

The Cluster edition is truly scalable with no limitations on the number of cameras. It also includes branding tools for adding a logo, corporate colors, captions, as well as the ability to organize a cluster opf server with stream failover tools.

System requirements for Watcher

Flussonic Watcher works in two versions:

Single - Suitable for those who do not need branding of the interface (their logo, colors), for projects of up to 500 cameras, not more than 1 server.

Cluster - Advanced version. Includes branding tools (changing logo, color, text), the ability to expand to tens of thousands of cameras, including the ability to build a cluster of dozens of servers with redundant streams. Suitable for those who do subscriber service or a large video surveillance system.

Such a cluster configuration requires a minimum of 2 servers (if you plan to use streaming servers). However, cluster mode is supported on a single server too.

On this page:

Single mode

Management + streaming server (two in one):

Operating system: Ubuntu 14.04 or later, Debian 7 or later;

Database: PostgreSQL;

CPU: CPU: Xeon E-3 1230v5 3.4 GHz and higher;

Memory: 32GB RAM;

Dedicated server: Yes.

Hard drive type: HDD / SSD;

Hard drive size: depends on the camera network size and video archive storage requirements.

10 GB of free disk space per 1 Mbps camera per day.
20 GB of free disk space per 2 Mbps camera per day.
70 GB of free disk space per 1 Mbps camera 7 days archive;

These system requirements are suitable for a network of:
500 1Mbps cameras
500 users
Failover turned off
Mosaic turned off

Or

250 2Mbps cameras
500 users
Failover turned off
Mosaic turned off

Servers should be dedicated, so no other software should run on them.

Cluster mode

Management server (endpoint — it runs Flussonic Media Server, Flussonic Watcher, and the database):

- Operating system: Ubuntu 14.04 or later, Debian 7 or later;
- CPU: CPU: 2-core CPU;
- Memory:8Gb RAM;
- Virtual server support: Yes;
- Database: PostgreSQL;
- Hard drive type: SSD;
- Hard drive size: 64GB of free disk space;
- Dedicated server: Yes.

Video streaming server (streamer — it is used for video streaming and video archive storage):

- Operating system: Ubuntu 14.04 or later, Debian 7 or later;
- CPU: CPU: Xeon E-3 1230v5 3.4 GHz and higher;
- Memory: 32GB RAM;
- Dedicated server: Yes.
- Hard drive type: HDD / SSD;
- Hard drive size: depends on the camera network size and video archive storage requirements.

10 GB of free disk space per 1 Mbps camera per day.
20 GB of free disk space per 2 Mbps camera per day.
70 GB of free disk space per 1 Mbps camera 7 days archive.

Important! For correct operation, Flussonic Watcher requires open ports 80, 443, 1935, 554 on all hosts, and the management server must have a real hostname which is resolved from the Internet.

The Watcher mobile app

Operating system:

- iOS 10 or higher
- Android 6 or higher

Installing and Updating Watcher

Flussonic Watcher can work both in cluster (multi-server) mode and in single-server mode. Watcher is installed in almost the same way regardless of the mode (single or cluster). Watcher uses the PostgreSQL database engine.

Important! Watcher is installed from the package flussonic-watcher, and Flussonic Media Server and PostgreSQL are automatically installed with Watcher.

On this page:

Comparing Watcher cluster with Watcher single

Watcher Cluster supports UI branding tools (adding your custom logo, choosing your custom colors, etc.)

Watcher Cluster can work with special firmware for cameras — Flussonic Agent, or simply Agent. Agent makes cameras accessible from behind NAT and greatly simplifies setting up the entire infrastructure. With Agent, you can add cameras in plug-and-play mode. This firmware improves the stability of video delivery and implements data encryption directly from cameras to streaming servers. To start using this firmware, contact our manager who leads your project.

Single-server mode is suitable for small and medium-size projects, where the maximum number of cameras does not exceed 500.

Servers' configuration

If your poject is small (less than 500 IP cameras) and you don't need a cluster of streaming servers, just install Watcher on a single server. In single-server mode, all cameras are connected to a single server, where Flussonic Watcher, Flussonic Media Server and the database are installed, the web interfaces works, streams are ingested, and the archive is written.

The cluster requires at least two servers:

The managing server (endpoint). It has the web interface to Watcher, Flussonic Media Server, the business logic, and the PostgreSQL database engine. Watcher works the managing server only.

Streaming servers (streamers). A streamer is a server machine that has Flussonic Media Server

installed. It stores DVR archives and handles camera streams. You can add from 1 to 100 streaming servers.

All servers must have public IP addresses and the same cluster key (specified in the Flussonic settings). In addition, the host name of the management server must resolve to the IP address.

The following picture shows the parts of cluster and the flow of video streams:

Installation main steps

To install Watcher:

Install Flussonic Watcher on the managing server. PostgreSQL and Flussonic Media Server are installed automatically together with Watcher.

Specify the path to PostgreSQL in the administrator UI of Flussonic Media Server.

Create a user with Watcher administrator privileges by using the Watcher's administrator UI.

That's all for a single-server Watcher.

If you plan to create a cluster, after you've done steps 1-3 continue the installation with the following steps:

Install Flussonic Media Server on all streaming servers.

Create a cluster and register streamers in Watcher by using the Watcher's administrator UI.

All the steps are described later on this page.

On how to update Watcher, see the section Updating Flussonic Watcher.

Installing Flussonic Watcher

On the server where you plan to run Watcher execute the command:

curl -sSf https://flussonic.com/public/install_watcher.sh | sh
After successful installation, the system advises you to start PostgreSQL and suggests the command to do so. Do not start PostgreSQL yet, go to the next step - user creation.

Create the user and the database. First, create the user vsaas by typing this command:

sudo -u postgres -i createuser -P vsaas
The system will prompt you to enter the password that will be used for the user vsaas:

Enter password for new role: (come up with and enter Watcher super admin password)
Type the password again for confirmation:

Enter it again: (re-enter Watcher super admin password)

Create the database vsaas_production with the created user vsaas as the owner:

sudo -u postgres -i createdb -O vsaas -e -E UTF8 -T template0 vsaas_production
The system's response if the database was created successfully:

CREATE DATABASE vsaas_production OWNER vsaas ENCODING 'UTF8' TEMPLATE template0;

In the Flussonic's web UI (http://flussonic:8080/admin), go to IP cameras and specify the path to the database in the Database path box.

Important. Replace VSAAS_PASSWORD with the real password of the vsaas user that you created in previous steps.

For cluster only: Add the mode cluster option in the configuration file /etc/flussonic/flussonic.conf:

```
vsaas {
database postgresql://vsaas:vsaas_password@localhost/vsaas_production;
endpoint enabled;
mode cluster;
}
```

After editing the file restart Flussonic Media Server with the command:

```
service flussonic restart
```

Finally, go to the the Flussonic UI in the browser.

Now if you go to the main page of the Flussonic UI http://FLUSSONIC:8080, the Watcher's web interface will open instead of Flussonic's.

To return to the Flussonic Media Server UI, go to http://FLUSSONIC:8080/admin.

Now it's time to create the Watcher main administrator.

Creating the Watcher administrator

Go to http://FLUSSONIC:8080 and the administrator control panel opens (we also call it the Watcher web UI). On the very first launch, the system will ask you to enter the login and come up with the password to create the first Watcher's administrator account.

The installation for single-server mode is now complete.

If you want to create a cluster, you will need to prepare streamers and set up Watcher to work as part of a cluster (see next steps).

(For cluster only) Creating a cluster (multiple server mode)

Creating a cluster means to add streamers (streaming servers) in the settings of Flussonic Watcher. Streamers (streaming servers) are servers intended to stream video from IP cameras. You must add at least one streamer on which IP cameras are added to configuration. This will allow you to start receiving video from cameras in cluster mode.

The Watcher UI page Settings > Streamers is essential for the cluster.

Pre-conditions

For each streamer, install Flussonic Media Server on a separate server, which will act as a streamer. In other words, besides the server with Flussonic Watcher, you must have at least one other server with a public IP address and Flussonic Media Server installed.

After you install Flussonic on a streamer, immediately change the administrator's login and password on each streamer.

Configure HTTPS on each streamer. It is enough to set the port for HTTPS, and Flussonic will use self-signed SSL certificates. Open the web interface and specify the port for HTTPS in Config > SSL-tunneled protocols, for example, 443. Other ways to configure HTTPS

Set identical date and time on the managing server and on each streamer.

In each streamer settings, specify cluster_key (the key must be the same as the cluster key for Flussonic Watcher). Learn more.

Configure the DNS zone for the managing server.

For cluster only: For Watcher to work correctly in a cluster, you need to add the A record in the DNS zone settings for your domain, where you specify the host name. This hostname must also be registered in the operating system on the server with Flussonic Watcher (the managing server). This is necessary for streamers to access the managing server.

To check that the hostname resolves, run on the managing server the command hostname — it must return the correct hostname specified in the DNS settings, for example, example.com.

When the streaming server is ready for work, you need to add it in the settings of Flussonic Watcher.

Adding streamers to Watcher

Log in to Flussonic Watcher as the administrator.

Go to Settings > Streamers and click the "+" icon to add a streamer:

Hostname – the domain name of the streaming server. Example: streamer2.example.com
Cluster key – the cluster key used in the cluster (the cluster_key option in the configuration file). If the streamer’s and Watcher's cluster keys are identical, there is no need to fill this field.
DVR path – the path to the archive. This field is required for the archive to be recorded. For example: /dvr. Multiple paths are possible (separate the paths with spaces): /dvr1 /dvr2.

Hostname – the domain name of the streaming server. Example: streamer2.example.com

Cluster key – the cluster key used in the cluster (the cluster_key option in the configuration file). If the streamer’s and Watcher's cluster keys are identical, there is no need to fill this field.

DVR path – the path to the archive. This field is required for the archive to be recorded. For example: /dvr. Multiple paths are possible (separate the paths with spaces): /dvr1 /dvr2.

When you have added a number of streamers, you must select the default one. Click the streamer in the list and then click Default. All new cameras will be added to the default stream server.

Important!

In the UI section Streamers you don’t need to add the host where Flussonic Watcher itself is deployed. On all servers in a cluster identical date and time must be set.

For each streamer, you can enable the automatic use of redundant (backup) servers for streams ingest in case this streamer fails (see Failover).

(For cluster only) Installing Flussonic Media Server on streamers

Flussonic Media Server is installed on all streaming servers.

Run the command:

```
curl -sSf https://flussonic.com/public/install.sh | sh
```

Now start Flussonic Media Server:

```
/etc/init.d/flussonic start
```

Learn more about installation of Flussonic Media Server in the Flussonic documentation:

- Quick start with Flussonic Media Server — briefly describes how to install Flussonic and start using it.
- Installing Flussonic Media Server — detailed comments about the installation, system requirements, and more.

On each streamer set up HTTPS and specify the cluster key in Flussonic's settings – see Create a cluster below.

Updating Flussonic Watcher

To update Watcher:

```
apt-get update
apt-get -y install flussonic-watcher
/etc/init.d/flussonic restart
```

During its update, Watcher automatically migrates the database to work with the new version. In rare cases it might be necessary to migrate the database manually. Watcher will show the message about that in the UI.

We strongly recommend that you back up your database every day and before you update Watcher.

Database Migration and Updating

This section provides the instructions on how to maintain the database used by Watcher.

Migrate to PostgreSQL (necessary starting from version 19.03)

Update the database structure manually (migth be necessary in some cases, Watcher will inform you about it)

Migration from SQLite to PostgreSQL

Back up these files:

/etc/flussonic/flussonic.conf

/opt/flussonic/priv/vsaas.db

Install the latest version of Flussonic Watcher with SQLite support (19.05). Run the following commands:

apt update
apt install flussonic-watcher=19.05 flussonic=19.05 flussonic-erlang=21.3.6
/etc/init.d/flussonic restart
Learn more about the update process

Back up all the data by using our migration tool:

/opt/flussonic/contrib/watcher backup create

The tool creates a file like this:
/var/lib/flussonic/watcher-backups/20190215201434-b62d21842ab7-WatcherBackup.gz

Install PostgreSQL

To install PostgreSQL, with root access in the console execute the command:

apt install postgresql

Create a PostgreSQL user and a database. Type two commands in the console one by one. First, create the user vsaas:

sudo -u postgres createuser -P vsaas

The system will prompt you to enter the password that will be used for the user vsaas:
Enter password for new role:

(come up with and enter the password of the Watcher main administrator)
Enter the password one more time:
Enter it again:
(re-enter the password)
Create the database vsaas_production with the created user vsaas as the owner:

```
sudo -u postgres createdb -O vsaas -e -E UTF8 -T template0 vsaas_production
```

System's response if the database was created successfully:
CREATE DATABASE vsaas_production OWNER vsaas ENCODING 'UTF8' TEMPLATE template0

To install PostgreSQL, with root access in the console execute the command:

Create a PostgreSQL user and a database. Type two commands in the console one by one. First, create the user vsaas:

The system will prompt you to enter the password that will be used for the user vsaas:
Enter password for new role:
(come up with and enter the password of the Watcher main administrator)

Enter the password one more time:
Enter it again:
(re-enter the password)

Create the database vsaas_production with the created user vsaas as the owner:

System's response if the database was created successfully:
CREATE DATABASE vsaas_production OWNER vsaas ENCODING 'UTF8' TEMPLATE template0

Edit the database line in the Flussonic configuration file /etc/flussonic/flussonic.conf:

```
database postgresql://vsaas:VSAAS_PASSWORD@localhost/vsaas_production;
```

Replace VSAAS_PASSWORD with the real password of the vsaas user that you created when installing PostgreSQL.
You can edit text files with the text editor nano.

Reload the Flussonic service:

```
/etc/init.d/flussonic restart
```

Restore data from the file that was created by the migration tool:

```
/opt/flussonic/contrib/watcher backup restore -d 20190215201434
```

Open the Watcher web inrweface and check that everything works (the data is present).

Updating the database structure

Some Flussonic Watcher updates include structural changes to the database. In this case, you will see the following message:

The script /opt/flussonic/contrib/watcher_db_migrate.sh adds changes to the database structure.

Important! Back up your database before running the watcher_db_migrate.sh script.

For backing up the database, you can use the watcher tool.

After you have created a database backup, run the script watcher_db_migrate.sh manually on the server with Watcher:

/opt/flussonic/contrib/watcher_db_migrate.sh

Here is a log of the successful execution of the script:

```
# /opt/flussonic/contrib/watcher_db_migrate.sh
INFO  [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO  [alembic.runtime.migration] Will assume non-transactional DDL.
INFO  [alembic.runtime.migration] Running upgrade dfd74e510414 -
> 1822b8f25e20, agent:model, agent:camera
INFO  [alembic.runtime.migration] Running upgrade 1822b8f25e20 -> 1a71a9477bbb, streamer: cluster_key
INFO  [alembic.runtime.migration] Running upgrade 1a71a9477bbb -
> 7a3ab2550cab, streamer_fkey cascade
Restarting Watcher
```

After Flussonic Watcher has restarted, you will be able to use the Watcher web interface.

If while running watcher_db_migrate.sh you encounter an error similar to this one

```
alembic.util.exc.CommandError: Can't locate revision identified by 'ebdce5515b6d
```

then install the previously used version (that you used before Watcher update or downgrade) and run:

```
cd /opt/flussonic/apps/vsaas
/opt/flussonic/contrib/watcher db downgrade
```

Settings

The menu section Settings in the Watcher web interface allows the Watcher administrator to edit the following settings of Flussonic Watcher:

General Watcher settings

Event notification settings

Web interface appearance settings

Streamer settings in the UI

General Watcher settings

Common settings

Mode – is displayed if Watcher works in multi-server mode (Cluster). If Watcher works in single-server mode, DVR path is diplayed here instead of Mode.

Watcher modes are descibed in Installing Watcher

DVR path — the path to the DVR archive storage. It can be a Swift storage, not only a local storage. This setting is displayed only in single-server mode. In cluster mode, the DVR path must be configured for each streamer in the streamer settings on the managing server (see Streamers below).

Watcher modes are descibed in Installing Watcher

Streamers - the link to the page for managing streamers. It becomes active in cluster mode.

API key — a token used for mobile access. You need to use it in your Account on the Flussonic website to activate access to mobile applications.

Language — the default UI language. If no default language is selected, Watcher will use the same language that is used by the browser.

GA key — your Google Analytics key for Watcher. With Google Analytics you can gather statistics on Watcher usage by users.

Guest access – allows users to access Watrcher UI by a link to a UI page, without logging in. A guest has access to public cameras and the map.

Demo access — allows access in demonstration mode, by the link Demo access on the login page. Actions such as modifying Watcher settings are not availabe in this mode. You cannot change the password for the user Demo.

Cameras serial number management – turns on the management of camera serial numbers in Flussonic Watcher. It is necessary when you use the Flussonic Agent software on cameras.

Map

Show map in main menu — show or hide the Map section from the main menu. The map is turned on by default.

Map center — geographic coordinates for map centering.

Map provider — select the map provider: Google, Openstreetmaps or OpenStreetMaps Offline.

Map key — the token of the geo cover in use (Google API key), allows automatically translate an address into coordinates in order to show the camera on the map.

Main page

Registration allowed — turns on the self-registration option (in version 19.07 and higher).

Homepage — specifies what users will see upon login: map or dashboard.

Guest homepage — specifies what guests will see on the home page: map or dashboard.

External authentication – specify the HTTP address or address of RADIUS server that you use to authenticate users.

Peeklio

Operator ID — necessary for the Watcher mobile app. It is the ID of your Watcher, which is used in mobile apps to connect to a specific Watcher server.

Subscribers must enter this ID in their app to get access to their cameras. However, if an app was branded, it is enough to enter a login and password (because branded apps can only work with a specific Watcher server).

Event notification settings

You can select which events received from cameras Watcher will to process. Also, you can instruct Watcher to receive events sent from external systems. Motion detection events and license plate recognition events are supported. (Usage examples will be added here soon.)

External event notification URL — Watcher will receive HTTP events from external sources specified here.

External event filter URL — the URL of your custom event handler. Watcher automatically sends the events received from cameras to this URL at the moment when an event takes place. Your script then receives an event from Watcher and returns (or doesn't return) the event ID. If the ID is returned, the event is considered confirmed and Watcher registers it in its database. Also, an email notification and push notification to the mobile application are sent. The archive interval around the event ([event_utc-10, event.utc+30]) is protected from deletion.

If the script returns no identifier, the event is considered unconfirmed and is not registered in Watcher.

If this field is not filled, then all events are registered in the Watcher.

To sum up, this script is for your custom filtering of events. You can save only those events that you are interested in.

Watcher uses the JSON format to send info about events to your handler:

{
"event":"video_activity",
"camera_id":"test1",
"algorithm":"plate_detector",
"activity_type":"enter",
"number":"ABCDEHKMOTX",

"area_id": "0",
"start_at": 1554883886,
"end_at": 1554883886
}

Disable push notifications — notifications about video analytics events (such as motion events) will not be sent to the Watcher mobile app. This option is independent from your custom backend scripts.

Adding and configuring streamers in the UI

Before you add streamers to Watcher, you need to prepare them. Details

To add streamer servers to Watcher, go to Settings and click the link Streamers. This link is active only in cluster mode:

By default, the local streamer is added to Watcher.

To add a remote streamer, click the + button and specify its settings.

The streamer appears in the list where you can edit its settings.

Watcher UI branding settings

Learn more in Interface branding

Reset password

A user can use the

RESTORE PASSWORD

option on the login page of Watcher UI. A password recovery link will be automatically sent via email. Read how to configure an SMTP server.

An administrator can change the user password with the watcher utility.

Example:

```
/opt/flussonic/contrib/watcher reset_pass support@erlyvideo.org new_password
```

You will see this message if everything went OK:

```
Changing password for support@erlyvideo.org
```

Failover

A failover cluster is a group of servers that work together to maintain the overall service stability and exclude any possible downtime of any part of the system. If one of the servers fails, another cluster server takes over its workload. This process is called

failover

.

In Watcher cluster, in case of a streamer server failure, the camera traffic to this server will automatically redistribute between other cluster servers, also called donors. The video archive becomes unavailable on the streamer server that had a failure. The new video archive will be temporarily recorded on the donor server.

When the connection to the primary server that had an issue is renewed, the traffic is automatically redirected back. Provided that the storage on the primary server wasn't damaged during the failover, the access to the main video archive will be re-established. However, the temporary video archive will be deleted from the donor server.

Turn on the failover option for each streamer server individually in the Streamer section and change mode in /etc/flussonic/flussonic.conf file on Watcher server:

```
vsaas {
  mode cluster+failover=30;
  ...
}
```

Where

30 — how often (in seconds) Watcher checks connection to streamers.

To apply new settings, restart Watcher:

/etc/init.d/flussonic restart

Motion Detection Events Processing

The server Flussonic Watcher can receive events over the SMTP protocol. Cameras send motion detection events over this protocol, and Watcher adds corresponding marks in the archive recordings in the places when motion was detected.

How motion recording works

Flussonic Watcher continuously keeps recording of video received from a camera, with the specified archive depth. When an event arrives, Flussonic Watcher saves the time interval in the database to be able to show the event in the archive player. The record with detected motion is protected from deletion.

The duration of a protected recording is determined by two timestamps, the first of which is calculated as the current time minus 10 seconds, and the second timestamp is the current time plus 30 seconds.

You need to set the depth of the archive, for example, 6 hours, and then enable the reception of events. As a result, you will have a recoding of 6 hours of continuous archive and additionally motion events, which will be stored as long as there is free space on the disk. Recording of new events will delete the old ones.

By calculating the necessary disk space based on the bitrate of the cameras and the frequency of motion events, you can save up to 50-90% of the disk space compared to the normal recording without events.

You can configure motion detection in two steps:

(If you don't use Flussonic Agent) Configuring Flussonic Watcher to receive motion events.

Configuring a camera to send motion events to Flussonic Watcher.

On this page

Configuring motion detection events for cameras without Flussonic Agent

Configuring Wacther to receive events

To activate events receiving, add the camera_alarm plugin into /etc/flussonic/flussonic.conf:

```
plugin camera_alarm {
  catch motion;
  listen smtp://0.0.0.0:1025;
}
```

The catch parameter specifies the word that Flussonic Watcher will search for in the subject of the message.
Most cameras send messages that have the default subject like this: "Camera 123 Motion Detected at 14:21 27-07-2019".

If your camera sends messages with a different subject or allows you to specify your own subject, then you can configure catch as you like.

It is possible to specify several parameters for a catch by listing them separated by commas: catch motion,alarm,detect;

The listen parameter specifies the interface and port for the built-in SMTP server. You can set login and password for SMTP:

```
listen smtp://username:password@0.0.0.0:1025;
```

Restart the server to apply the settings:

```
/etc/init.d/flussonic restart
```

Configuring the camera for sending motion events

To configure the camera, specify the SMTP server address, and the names of sender and recepient.

Use the IP address of your Flussonic Watcher server as the SMTP server address.

The sender and receiver must be specified as full camera names (camera name and ID). You can find the full names in the Watcher UI. Example: cam1-abcdefg@example.com, where cam1 is a camera name in Watcher and abcdefg is a camera ID in Watcher. Besides, the full name of the camera can be found in the browser's address bar when the page with camera settings is open.

Here is an example of camera settings:

Configuring motion detection events for cameras with Flussonic Agent

If you have Flussonic Agent installed on a camera to automatically connect the camera to the server, if it is behind NAT, then you do not need to configure anything on the server. It is enough to configure the camera to send events.

To configure the camera, specify the SMTP server address, and the names of sender and recepient.

Go to the camera's interface to the message sending section and specify SMTP server settings:

```
SMTP Server: 127.0.0.1
SMTP Port: 5025
```

Use the IP address of your Flussonic Watcher server as the SMTP server address.

Fill in other fields. Sender name is the name of the camera.

Note. You can specify only a part of the camera name if the camera has Flussonic Agent installed. The

full name will be provided automatically.

Viewing events in the web interface of Flussonic Watcher

After you have configured a camera to send motion events, you can view the events in the DVR archive.

To view the DVR archive of the camera:

Go to Cameras in the Watcher main menu.

On the top of the page, select the List mode for viewing the list of cameras.

Find the camera (for example, by using the search form on the right).

Open the actions menu for this camera by clicking the icon in the most right column.

Choose View. The player opens and you can jump to the interval with recorded video. If DVR is disabled for a camera, the player shows only live video.

If there were motion events, you will see the marks on the timeline that show points in time when the camera recorded movement:

License Plate Detection Events

Flussonic can detect license plates and recognize Russian car numbers on the video transmitted by an IP camera (including special transport like Emergency and Fire cars). This functionality is known as ANPR (automatic number plate recognition).

Flussonic does the following:

- Creates events of license plate detection.

  Video comes from IP cameras to a streamer (in cluster mode) or to the managing server (in single mode), where the number recognition takes place.
- Provides the Watcher UI for viewing plate detection events.

  You can view registered events and watch the recorded video of each event.
- Provides the API for integration with external services.

To start detecting car numbers:

- Prepare hardware and software for the Flussonic server that will carry out number recognition.
- Turn on and configure the car number recognition. To configure the feature, use the web interface or the configuration file, but please remember that some options can be set only through the file.

On this page:

Installing the ANPR module

To start using number recognition, install Flussonic Media Server and Watcher first (if you didn't have them installed). The ANPR module can work both in cluster and single mode of Flussonic and Watcher installation.

Number recognition takes splace on a streaming server in cluster mode or on a managing server in single mode. To this server, you need to connect cameras that transmit video from the place where you want to detect car numbers. This server must have at least one high-performance videocard GPU NVIDIA with at least 6 Gb video memory.

System requirements for Flussonic's ANPR module

- OS: Ubuntu 16.04 x64
- GPU: Nvidia (Pascal) minimum 6 GB VRAM (for a certain project we could tell more exactly how much memory is required).
- CPU: 4+ cores
- RAM: 8+ GB
- Flussonic Media Server (on streamers in a cluster)

Flussonic Media Server + Watcher (single server)

Attention. In a cluster installation, the number recognition module works on a streamer server, where Flussonic Media Server must be installed first. If you use a single server, install Flussonic Media Server and Watcher before you install the recognition system.

Installation

To install the number recognition module, add the official Nvidia repository to the system and then install the flussonic-vision package from the Flussonic's repository:

```
wget http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_9.2.148-1_amd64.deb
dpkg -i cuda-repo-ubuntu1604_9.2.148-1_amd64.deb
apt-key adv --fetch-keys http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
apt update
apt install flussonic-vision
```

Setting up ANPR in the configuration file

Note. You can do the same through the Watcher UI – the settings will be saved to the config file automatically. But you will still need to check the GPU number in the file and edit it, if necessary.

Open the file /etc/flussonic/flussonic.conf.

Add the line plugin vision;, which enables the recognition system.

Add the vision directive to a stream's settings, and specify the GPU number:

```
stream cam1 {
  url rtsp://192.168.0.11:554/h264;
  vision gpu=0;
}

stream cam2 {
  url rtsp://192.168.0.12:554/h264;
  vision gpu=1;
}

plugin vision;
```

gpu (required) – GPU number. You can get it by using the nvidia-smi tool.

By default, the recognition system searches for car numbers over all the field of view of a camera.

Reload the configuration so that the changes made to the file take effect.

Setting up ANPR in the UI

Pre-condition

Before you configure cameras in the UI, make sure you enable the recognition system. Add the line plugin vision; to the configuration file /etc/flussonic/flussonic.conf.

Enabling license number recognition

To turn on plate detection and number recognition for a camera:

In the Watcher UI, go to Cameras. Find the camera in the list and open its settings by clicking the icon in the upper right corner of the player.

Select the License plate recognition check box and click Save.

Now Flussonic will recognize car numbers that appear in the frame of this camera, and mark the time when the car entered and left the scene.
Flussonic modifies the stream settings in the configuration file /etc/flussonic/flussonic.conf.

You may need to edit the GPU number manually in the configuration file (see the previous section about setting up ANPR in the configuration file).

Viewing ANPR events in Watcher

Flussonic creates events of two types:

enter – a car number appeared in the field of view of a camera

leave – a car number left the field of view.

To see detected car numbers for a camera:

In the Watcher UI, go to Notifications. The list of all events opens.

To find certain events, use filtering and search capabilities on the right:

In Source, select Plate detector.

In From and To, select the date and time of beginning and end of the period when events were detected.

In Search, type the car number.
To delete the specified search criteria, click Clear Filter.

In Source, select Plate detector.

In From and To, select the date and time of beginning and end of the period when events were detected.

In Search, type the car number.

To delete the specified search criteria, click Clear Filter.

The list of events is filtered as you enter search parameters.

To view the recording of an event, click in the line with this event. The player will appear on the bottom right to play the recording. To enlarge the player, just double-click it.

The API of the ANPR module

After you correctly set up the recognition module, you can get events data in the JSON format via the Watcher API.

The detailed API reference is availabe at https://flussonic.github.io/watcher-docs/api.html.

Below is an example of an API request and a response:

```
root@ubuntu:~# curl localhost/vsaas/api/v2/events?type=activity -H 'x-vsaas-api-key: dfb21d1f-3e00-44a2-a706-36d99f9e9d73'
 {
  "start_at": 1538645882872,
  "type": "activity",
  "id": "7ecb0a13-414c-462f-a206-3c5d047baad4",
  "ext_data": null,
  "object_id": "A123AA 77",
  "end_at": null,
  "source": "plate_detector",
  "camera_id": "cam0-00",
  "source_id": "0",
  "object_class": "leave",
  "event_data": null
 }
```

start_at - the start time of an event

id - a unique ID of the record

object_id - car number

camera_id - camera name in Watcher

object_class - the event, which can be enter or leave (a car entered or left the camera's field of view).

Auto-login

Flussonic Watcher allows its users to log in by a special URL (auto-login link), without entering a password. This is useful if you want to simplify access to Flussonic Watcher for your users or prevent the transfer of passwords to third parties.

The auto-login link is issued to an authorized client.

In order to generate a URL for auto-login, you will need to first request a token by using the link /vsaas/api/v2/auth/generate-autologin-token. After that, the user can be authorized by sending the token via the POST method to /vsaas/autologin.

Follow the steps:

1) First, you need to generate a token for the autologin of a particular user. To do this, make a POST request in the JSON format as follows:

```
curl --header "X-Vsaas-Api-Key: API_KEY" --header "Content-Type: application/json" --request POST --
data '{"login": LOGIN, "valid_till": VALID_TILL, "lifetime": LIFETIME}' "http://watcher.com/vsaas/api/v2/auth/ge
nerate-autologin-token"
```

In the request, replace these placeholders with real values:

- API_KEY — the Watcher API key that can be found in the Watcher UI on the Settings page in API key. This key is sent in the HTTP header X-Vsaas-Api-Key.
- LOGIN — the login (the same as the email) of the user to whom you want to give access. Line. Required.
- VALID_TILL — the UTC time in seconds until this token is valid for autologin. Integer. Optional parameter.
- LIFETIME — the duration of the session that will be open at the user's automatic logging in, in seconds. Integer. Optional parameter.

Watcher sends a response in the JSON format:

```
{
   "autologin_token": "demo:1487258314:f8b1:b4bdaac58cbe94638e5b14a3728b8e6d633f3c6e",
   "success": true
}
```

The autologin_token field contains the token for the specified user.

2) The autologin_token received in the previous step can be used in POST requests to Flussonic Watcher. For example:

```
<form action="http://watcher.com/vsaas/autologin" method="POST">
  <input type="hidden" name="autologin_token" value="AUTOLOGIN_TOKEN" />
  <input type="submit" />
</form>
```

At the click on the submit button, the user will be logged in automatically into the Flussonic Watcher web interface.

User Interface

A user can log into Flussonic Watcher using an email and password after adding to the system.

New users can access cameras that were assigned to them by an administrator. Cameras may be public, available after authorization, and private.

A user interface has two screens: the first one shows cameras on a map and the second one features a list of cameras.

The following menus are available:

- Dashboard.
- Map.
- Favorite.
- Cameras.
- Users and Groups. (For Administrator)
- Settings. (For Administrator)

Dashboard

Dashboard — is a primary webpage for users to access cameras.

The following access filters are available:

- Any.
- Public.
- Authorized.
- Private.

Camera display modes:

- Medium Thumbnails.
- Large Thumbnails.
- List.

You can also toggle Compact and Hide inactive cameras.

The list below shows automatically updated screenshots from all cameras. In order to watch a certain camera or show its video archive, click on the respective image or Play button.

You can also click on the little arrow button on the top right corner of each block to access the video archive. To open camera in full screen, click on a camera name:

Mosaic shows up to 8 selected cameras on a single screen. To use a multi-camera view, create a client mosaic in the Cameras section.

Map

The map shows cameras that have coordinates configured.

Favorites

A user can mark a camera with a star. After that, it'll show up in the «Favorite» section.

Cameras

Here you can perform all operations with cameras:

- Add a camera.
- Camera search via ONVIF.
- Import Flussonic Cameras.

Also you can do the following:

- Export a list of files in CSV format.
- View camera serial numbers list.
- Create a Mosaic.
- Go to the DVR archive of a camera

More information on adding cameras: «Camera management» section.

Users and Groups

In this menu, you can add and modify Flussonic Watcher users. For more information see «User Management».

Use groups to simplify user and camera navigation by making logical grouping (for example, by floors, departments, regions, neighborhoods, etc.), as well as simplify the user right management to provide camera visibility. See the «Groups» section for more information.

Settings

This menu is used to configure the Flussonic Watcher. More in the section Settings.

Mobile Applications

Watcher offers mobile applications for real time access to a video surveillance system.

It provides:

- Watching live video from IP cameras with the ultra low latency
- Viewing the archive with no limits on its depth
- The access control based on a fingerprint or a PIN code
- TLS encryption of video streams
- Push notifications about events
- Downloading video screenshots.

Mobile applications need to know the address of your Watcher system to get video from it. Get your Operator ID that stores the URL of your Watcher system beforehand.

By default, the application tries to reach the Flussonic server, so a mobile app can't authenticate without a personal Operator ID.

To get your Operator ID, log in to your account, click on the installation key on the main portal screen and specify the public Watcher URL in the following format: http://APIKEY@watcher-host. The host address should be accessible from the Internet. It's highly recommended to use a domain name, not just IP address.

To start using the application, download it from the Apple Store or Google Play, and then authenticate using your Watcher login, password and the Operator ID that you received from manage.erlyvideo.org.

APIKEY — an authentication key that you can obtain in the Watcher settings UI.

Watcher-Hostname — a publicly visible path to Watcher.

Важно! Make sure that your server is visible from the Internet and that it has a permanent white IP address.

If your server is using NAT or a Firewall, contact our support and we will help with all necessary configuring on the terms of extended support.

Interface Customization

Watcher in cluster mode provides tools for interface customization (branding).

To set your custom branded interface for Watcher:

Go to Settings > Branding and specify:

Common settings

- Custom Logo. Select a graphic file with a logo image to be displayed in the upper right corner. If the file is too large in width and height, the system will reduce it to the required size.
- Login page custom logo. Select a graphic file with a logo image to be displayed on the page where you enter the login and password. If the file is too large in width and height, the system will reduce it to the required size.
- Favicon. The icon that appears in the browser on the tab where Watcher is open. The favicon must be a square PNG image of 64x64 pixels.
- Custom page title. The title that appears in the browser on the tab where Watcher is open.

Color scheme

You can select colors for basic UI elements, and Watcher will defineall other colors automatically based on the specified main colors.

Additional footer text

- Address. Your company postal address.
- Phone. Your company phone number.
- Work hours. Your company working hours.

You can also customize the password recovery email template. Read more in Email customization.

Email customization

Setting the SMTP server

Configure your mailbox settings so that password recovery emails could reach your users. Add parameters of the outgoing mail server to the configuration file.

For example, use these parameters:

Email: email.address@example.com;
Password: xyz123;
Mail server address: smtp.example.com;
Port: 465;
Connection security: SSL

Add these parameters to the configuration file:

```
vsaas {
  database postgresql://vsaas:PASSWORD@localhost/vsaas_production;
  smtp_server smtp.example.com:465;
  smtp_login email.address:xyz123;
  email_from "Flussonic Watcher <email.address@example.com>";
  smtp_opts ssl;
}
```

Changing the template of password recovery emails

To change the template for password recovery emails, follow these steps:

- Go to /opt/flussonic/lib/vsaas/watcher/templates.
- Use password_reset_request.email and password_changed.email text files as an examples.
- Add custom_ file prefix to your custom template versions: custom_password_reset_request.email and custom_password_changed.email.
- If you want to use custom hypertext templates, add the .html extension to these files as follows: custom_password_reset_request.email.html and custom_password_changed.email.html. Save these files in the same directory.

The template consists of two parts: the header and the body. You can also add a subject to the header.

In addition, in the message body you can use variables

- {{data.base_url}}
- {{data.token}}

Text template example:

custom_password_reset_request.email:

```
---
subject: "ABC surveillance password reset"
---
```

Thank you for using the forgot password option. Follow this link to reset your password to the ABC surveillanc
e system:
{{data.base_url}}/vsaas/forgot-password/{{data.token}}

HTML template example:

custom_password_reset_request.email.html:

```
---
subject: "ABC surveillance password reset"
---
<html>
<body>
<p>Thank you for using the forgot password option.to the ABC surveillance system:<br>
<a href="{{data.base_url}}/vsaas/forgot-password/{{data.token}}">Reset your password</a>
</p>
</body>
</html>
```

Using Flussonic Agent

Flussonic Agent (or Agent) is a small-size software that you install on IP cameras or on other devices used in a video surveillance service. It allows cameras or other devices from a local area network to connect to an external Watcher server. An encrypted channel is used.

It is enough to connect a device with Agent to the Internat, and Agent automatically connects to your server where Watcher is running.

The benefits of using Agent

Agent solves the problem of communication between Watcher and devices from behind NAT. It helps if the camera does not have a dedicated IP address or you do not want to perform port forwarding on network equipment so that the camera in the local network could be seen by a remote Watcher server via the Internet. With Agent, a camera itself initiates the connection with Watcher and automatically registers there (without Agent, usually it is the server that initiates the connection with cameras).

In addition, Agent helps if the communication channel between the camera and the server is unstable. Cameras, for the most part, do not know how to buffer video. Agent installed on cameras uses the buffer to resend packets that for some reason did not reach the server.

The Flussonic Agent section provides details about how Agent works and how it is better than other ways of delivering video from cameras to Flussonic Watcher.

In this section:

Devices supported by Agent

You can install Flussonic Agent on the following types of devices:

- On an IP camera
- On a microcomputer Raspberry Pi 3 Model B+
- On a router that supports OpenWRT (coming in future Watcher versions)

Depending on where Agent is installed, it works differently:

- If Agent is installed on an IP camera, it automatically connects a camera to the Watcher server and starts transmitting video from the camera at the moment the camera is connected to the server.
- With Agent installed on a device (router or microcomputer Raspberry Pi), Watcher gets access to all cameras that are located in the same LAN with this device. Watcher, which works on an external server, can take video streams from cameras via Agent. Without Agent, access to cameras would be blocked by NAT.
  With this method of installing Agent, you don't install it on cameras. This method eliminates the risk that after the camera has been updated by the manufacturer, Agent might not start on it.

Installing Flussonic Agent

Flussonic Agent can be installed on both IP cameras and Raspberry Pi 3 Model B+ microcomputers.

Installing Flussonic Agent on IP cameras

Flussonic ships Agent for cameras in the form of modifications of the cameras' original firmware.

To install Flussonic Agent on on an IP camera:

1. Receive Agent for your Watcher server from your personal manager at Flussonic.
2. Go to the camera's web interface.
3. Update the firmware with the new one received from Flussonic.

Installing Flussonic Agent on Raspberry Pi 3 Model B+

To install Flussonic Agent on a Raspberry Pi device:

1. Receive Agent for your Watcher server from your personal manager at Flussonic.
2. Install the Raspbian operating system on your Raspberry Pi device.
3. Install the DEB package received from Flussonic on the device by using a package manager.
4. Configure Agent launching parameters by editing the file /lib/systemd/system/flussonic-peeklio.service (for example, in the nano text editor). Add the following lines to the file:

```
[Unit]
Description=Flussonic Peeklio
After=network.target
```

Flussonic Agent in the Watcher UI

This section describes how to view the list of Agents registered in your Wathcer.

Adding and viewing Agents in Watcher

If Agent is installed on a camera, it connects to Watcher automatically. After you have connected Agent to the Internet, Agent registers and appears in Watcher.

To check that Agent has registered in Watcher successfully, go to Agents and you'll see the list of Agents that have registered in Watcher. The list containsthe following data:

- Agent's status — Watcher uses color to show the state of Agent (red — for not working Agents, green - for working Agents).
- AgentID — a unique identifier of this Agent.
- Camera — the name of channel on a camera that is transmitted via this Agent.
- Info — additional information about Agent (private IP address, version, time of uninterrupted work).
- SN — Agent serial number.
- Streamer — the streaming server to which this Agent is connected.

Adding Cameras to Watcher by Using Agent

This section describes how to add cameras to the Watcher, if Agent is used (it is installed on cameras or on a device in the same local network with cameras).

If Agent is installed on a camera, it adds itself and the camera to Watcher. If Agent is installed on a device in a local network with cameras, then you will need to add the cameras manually.

If you don't use Agent, you will need to add cameras manually.

About the UI for adding cameras

On this page:

Adding a camera with Agent to Watcher

If Agent is installed on a camera, the camera will be added to Watcher automatically.

Adding cameras located in a network with Agent to Watcher

You have installed Agent on a Raspberry Pi device or a router. Now you can connect cameras to a external Watcher server in the Internet with the help of this Agent.

To add cameras to Watcher by using Agent in the same local network with cameras:

Install Agent on a device

Look at and remember its identifier agentID.
The identifier can be found in either of the ways:

at http://[Agent-local-IP]:5680/agent-status
in Watcher in the Agents section, where a newly added Agent appears.

at http://[Agent-local-IP]:5680/agent-status

in Watcher in the Agents section, where a newly added Agent appears.

(For each camera) Go to Watcher UI and add a camera (Cameras > Add a camera > New camera)

In Stream URL enter:
rtsp://USER:PASS@CAM-LOCAL-IP/STREAM via=agent://AGENTID

Here:

* STREAM - the link to the RTSP stream of a camera, can be found in the camera's documentation.
* CAM-LOCAL-IP -
the camera's IP address in the same local network as where the device with installed Agent is located.
* USER:PASS - the login and password for the camera.
* AGENTID - Agent identifier.

Flussonic Agent Status and Log

This section describes how to obtain Agent status and Agent log messages.

Agent status

To make sure that Agent has successfully connected to Watcher, you can use the browser (as an alternative to looking at the list of Agents in the Watcher UI).

Open the following address in the browser:

http://[AgentIP]:5680/agent-status

Replace [AgentIP] with the local IP address of the device where Agent is installed. The current status of Agent will be shown.

Important. Open this address only in the same local network where Agent is installed.

Agent log

For diagnostic purposes, we migth sometimes ask you to send us Agent log messages.

To download Agent logs, open the following address in the browser:

http://[AgentIP]:5680/agent-status?k=1

Replace [AgentIP] with the local IP address of the device where Agent is installed.

Important. Open this address only in the same local network where Agent is installed.

Save the log messages and attach it to the ticket in which you correspond with our support team.

Managing Organizations

In your CCTV subscriber service that uses Watcher, you give each subscriber their own space in Watcher where they can add cameras and give other users access to them. This space is called Organization, and you need to create as many Organizations in Watcher as you have subscribers to your service.

Thus, in your Watcher you will need to create as many Organizations as the nummber of your Subscribers.

If you use Flussonic Watcher as a security system on a protected object, then a single Organization will be enough for you — it is created in Watcher by default.

Note. Organizations will be introduced in Watcher v19.08.

To manage Organizations, go to Organizations in the Watcher main menu.

This page tells you how:

- Add an Organization
- Edit an Organization
- Add cameras to an Organization
- Add users to an Organization
- Assign the owner user to an Organization

Adding Organizations

If you use Flussonic Watcher as a security system on a protected object, then you will be good with a single Organization created by default.

If you use Watcher as a CCTV service, you need to create an Organization that corresponds to each subscriber.

Important. Before you add an Organization, you need to create a user who will be the owner for this Organization. This user must have the full set of permissions for managing the Organization.

To add an Organization for a new subscriber:

- Go to Organizations > create an Organization
- Fill in the form and click Save.

Title — defines the name with which the Organization will be displayed in the list of Organizations.

Owner — a Watcher user who has the maximum permissions for managing the Organization (meaning managing cameras and users) and who is responsible for payments for your service.

Note. The Watcher user - Organization owner must be created in Watcher before you create an Organization.

Note — notes about the Organization.

Camera limit — the maximum number of cameras that can be added to a single Organization.

User limit — the maximum number of users that can be added to a single Organization.

DVR limit — the maximum number of camera-days of the archive that will be available for this Organization. For example, you can set the limit of 10 camera-days. These 10 camera-days of the

archive can be distributed arbitrarily among cameras, for example, you can configure Camera 1 to have 3 days of the archive, Camera 2 — 3 days of the archive, and Camera 3 — 4 days of the archive. So totally we have 10 camera-days of archive for all cameras.

Editing Organizations

To edit an Organization, go to Organizations and in the list click the title of the Organization you want to modify.

Adding cameras to an Organization

The key element of each Organization is the list of IP cameras. One camera can belong only to one Organization.

A camera can be added in Watcher only within an Organization. There are two ways in Watcher to add a camera.

Adding a camera on the Organizations page:

Go to Organizations

In the list of Organizations, find the Organization where you are going to add a camera and click the counter in the Cameras column.

After doing so, you can either add an existing camera to the Organization or create a new camera and then add it to the Organization.

Adding a camera on the Cameras page:

When you manually add a camera you need to specify the Organization and (optionally) Folder.

By default, Watcher uses an Organization marked as Default, and the root folder inside the default Organization.

Deleting a camera from an Organization

To delete a camera from an Organization, go to camera settings and edit Organization and Folder.

Adding users to an Organization

After you have added an Organization to Wathcer, you will probably need to add users who will have access to cameras of this Organization.

To add a user to an Organization:

Go to Organizations.

In the list of Organizations, find the Organization where you are going to add a user and click the counter in the Users column.

The page that opens shows the list of all users of this Organization. Click Add a user to create a new user within this Organization.

When filling in the form, pay attention to these fields:

Internal IP. An IP address that Watcher will use for autologin.

Enabled. Select this to activate the user account.

Internal IP. An IP address that Watcher will use for autologin.

Enabled. Select this to activate the user account.

Camera Management

Cameras in Watcher are created and exist only within a specific Organization. One camera can belong to one Organization only.

To manage cameras, go to Cameras in the main menu.

You can add IP cameras to Watcher in three ways:

- Manually - you manually fill in all the data necessary for connecting the camera.
- Uploading CSV - you create a list of cameras and add them to Watcher at once.
- Searching by ONVIF - the standard ONVIF protocol is used to search for cameras and add them to Watcher. When using this method, the cameras must be in the same local network with the Watcher server.

If you use Agent, the cameras are added automatically or manually.

On this page:

Adding cameras

After Watcher is installed, you will see no cameras in the Watcher UI, even if they are present on Flussonic Media Server. To add a camera to the system, click Add a camera:

In the dialog that opens, choose New camera and enter camera settings:

Camera settings

General settings

- Title. The camera name, which you see in the list of cameras. Use only Latin characters and numbers, as the name will be used in URLs.
- Stream URL. The camera URL. For example: rtsp://mycam.local/stream0. If you know the camera URL, you can import the camera from the Flussonic server or use auto-search to add the camera.
- Substream URL. Additional address. For example: rtsp://mycam.local/stream1; rtsp://mycam.local/stream2
- Organization. The Organization to which the camera will be added. A camera can belong to only one Organization. If you do not select Organization, the default Organization is used.
- Folder. A hierarchical node of the camera tree within the Organization where the camera will be added. If you do not select a folder, the root folder in the selected Organization is used.

Administration

- Streamer. A Flussonic Media Server that is used as a streamer for this camera.

Owner. The camera owner. Owners are configured in the [Users and Groups] menu (manage-users). The owner can access a camera always, regardless of access rights.

Access. The camera availability: to all users, only to registered users or to specific users:

Public (public) — the camera is available to all users of Flussonic Watcher, regardless of the settings of groups and users. When setting the value of public source appears in the list of cameras for all users of your server. Also, without restrictions on access, the camera can be published on a web page with permanent access without authorization.
For customers (authorized) — the camera is available to all users, but you need to authorize them on your Flussonic Watcher.
Private — the camera is available to a user only after the user was given access to it in the Users and Groups section.

Public (public) — the camera is available to all users of Flussonic Watcher, regardless of the settings of groups and users. When setting the value of public source appears in the list of cameras for all users of your server. Also, without restrictions on access, the camera can be published on a web page with permanent access without authorization.

For customers (authorized) — the camera is available to all users, but you need to authorize them on your Flussonic Watcher.

Private — the camera is available to a user only after the user was given access to it in the Users and Groups section.

DVR Depth. The archive depth. In other words, it’s the number of days after recording during which the video archive is available (then it is purged). To enable this option, configure the repository in the System settings or Streamer settings.

DVR Space. The maximum storage space for camera's archive, in Gigabytes.

Additional settings

Enabled. Turning the camera on and off. It means whether video from this camera will be transmitted or not to Watcher.

On-demand. Turn on to make the camera transmit video only on request, turn off to make the camera work constantly.

License plate recognition. Turning on or off license plate detection and Russian car number recognition.

Comment. The text that describes camera positioning or gives any other information.

Location

Coordinates. The coordinates of the camera location. Define the camera placement on the map. You can change its placement. We recommend that you place all cameras to their actual locations on the map to help users find them.

Address. The postal address of the camera.

Saving camera settings

After you have edited the camera settings, click Save. The camera will appear on the management page in the list of cameras and on the Dashboard:

Creating a folder and adding cameras to it

By default, all cameras in an Organization are added to the root folder.

To create a folder and add cameras to it:

1. Go to Organization in the Watcher UI
2. Click Add a folder next to the selected Organization or a subfolder within the Organization
3. Drag a camera from the list of cameras to the folder.

Another way to add a camera to a folder is by editing the Folder field on the camera settings page. To do so, go to camera settings (Cameras > click the camera in the camera list) and choose Folder where you want to move the camera.

Deleting a camera

To delete a camera:

1. Go to Cameras
2. In the list of cameras that opens, find the camera and click the Delete icon.

Another way to delete a camera is by clicking Delete on the camera settings page.

Adding Cameras to Folders

If an Organization has a large number of added cameras, it becomes important to have the means to navigate them easily.

To address this, Watcher provides Folders, which are used to group cameras on some basis, for example, based on their geographical location.

Note. Folders will be introduced in Watcher v19.08.

One folder can include a number of subfolders and cameras. Placing cameras into folders is similar to organizing the storage of documents in the file system.

Creating a folder and adding cameras to it

By default, all cameras in an Organization are added to the root folder.

To create a folder and add cameras to it:

1. Go to Organization in the Watcher UI
2. Click Add a folder next to the selected Organization or a subfolder within the Organization
3. Drag a camera from the list of cameras to the folder.

Another way to add a camera to a folder is by editing the Folder field on the camera settings page. To do so, go to camera settings (Cameras > click the camera in the camera list) and choose Folder where you want to move the camera.

Deleting folders

To delete a folder:

1. Move all the cameras from the folder you are going to delete. You can move them to another folder or to the root folder of the Organization.
2. Click Delete next to the folder.

User Management

After installing Watcher, you will see only one user on the Users page — the Watcher Administrator.

Standard Watcher users exist within Organizations. These are users on the side of your service subscriber. One user can be added to several different Organizations.

The administrator of Watcher can also create users who do not belong to any Organization. These are usually users on the service provider side. For example, these are users who have access only to statistics.

On this page:

Adding users to an Organization

After you have added an Organization to Wathcer, you will probably need to add users who will have access to cameras of this Organization.

To add a user to an Organization:

1. Go to Organizations.
2. In the list of Organizations, find the Organization where you are going to add a user and click the counter in the Users column.
3. The page that opens shows the list of all users of this Organization. Click Add a user to create a new user within this Organization.
4. When filling in the form, pay attention to these fields:

Internal IP. An IP address that Watcher will use for autologin.
Enabled. Select this to activate the user account.

Internal IP. An IP address that Watcher will use for autologin.
Enabled. Select this to activate the user account.

Adding users to Watcher

To create a user who does not belong to any Organization:

1. Go to Users > Add a user.
2. When filling in the form, give the user the permission Can view statistics of Organizations.

Watcher creates a user without access to cameras but with access to all statistics of Organizations.

Granting a user rights to manage Organizations

One Organization can have a number of users that have access to its cameras. You can give each user different permissions to manage the Organization.

To change a user's rights to manage an Organization:

Go to Organizations and click in the column Users next to the Organization where you want to grant users rights to manage the Organization.

In the list of users that opens, click next to each user the permissions that you are giving them:

Can edit cameras - the user can add and edit cameras
Can edit users - the user can add and edit other users
Can view statistics - the user can view watcher resources consumed by the Organization.

Can edit cameras - the user can add and edit cameras

Can edit users - the user can add and edit other users

Can view statistics - the user can view watcher resources consumed by the Organization.

Another way to edit the user permissions to manage the Organization — is in the user settings on the Permissions in the Organization tab.

Granting a user access to cameras

Users can have various permissions to access video received from cameras.

To grant a user access to cameras:

Go to Organizations and click in the column Users next to the Organization where you want to modify user permissions.

In the list of users that opens, click the user whose permissions you want to modify.

In the user settings, go to the tab Access to cameras.

If the Access to cameras tab in inactive, the user was not allowed to manage cameras.

To allow the user to manage cameras, go to the list of Organization's users and click the button Manage cams next to this user.

Select folders containing cameras to which you grant this user access to.

Categories of access to cameras

Access to cameras — the user can view video received from IP cameras of the Organization.

Access to archive — the user can view recorded video in DVR archives.

Access to PTZ — the user can control cameras via PTZ.

Note 1. Granting a user access to a folder means that all its subfolders will also be available to this user.

Note 2. You can give access only to an entire folder, not to individual cameras in the folder. To give access to only one camera, add this camera to a separate folder and give the user access to that folder.

After adding access to cameras, they will be available to the user in the Dashboard.

Deleting users from Organizations

To delete a user from an Organization:

Go to Organizations

Click in the column Users next to the Organization from where you want to delete a user

Click the Delete icon next to the user.

Creating a client mosaic

A

mosaic

allows gathering of up to 16 selected cameras on a single screen. It is used for convenient viewing of selected cameras.

You can create and manage mosaics in two ways:

- In the Cameras section, by dragging and dropping cameras
- In the Mosaics section, by selecting cameras manually

Managing mosaics in the camera list

To create a mosaic of cameras:

1) Log on to Flussonic Watcher as an administrator.

2) Go to Cameras and click in the upper menu the mosaic view icon (on the right):

3) Click Create.

4) In the window that opens, enter the mosaic title and select the dimensions of the mosaic (the number of cameras in it).

5) Drag cameras from the lit in the right. If the list is very long, you can use filters.

6) To delete a camera from the mosaic, click the recycle bin icon in the upper rigth corner of the corresponding player in the mosaic.

All created mosaics are displayed in Mosaics and in Cameras > the mosaic view.

To view a mosaic:

The video from cameras in a mosaic is shown right when you edit it in Cameras > the mosaic view.

Cameras are also shown in Mosaics after you click the mosaic title there.

To change the title and size of a mosaic:

In the line Mosaics > [mosaic title] click the title and select Edit.

To remove a mosaic:

In the line Mosaics > [mosaic title] click the title and select Delete.

Managing mosaics manually

To create a mosaic of cameras:

1) Log on to Flussonic Watcher as an administrator.

2) Go to Mosaics and click Create a mosaic.

3) On the screen that opens, enter the mosaic title and select the dimensions of the mosaic (the number of cameras in it).

4) Now add cameras to the cells of the mosaic. Click in a cell and then click a camera in the list of cameras that has opened. The camera will be added in the cell.

5) Click Save.

All created mosaics are displayed in Mosaics and in Cameras > the mosaic view.

To view a mosaic:

In the Mosaics section, click the row with the name of the mosaic you want to view. The page with the video from the cameras opens.

To change the title and size of a mosaic:

In the Mosaics section, click the settings icon next to the mosaic in the list.

To remove a mosaic:

In the Mosaics section, click the recycle bin icon next to the mosaic in the list. Alternatively, use the Delete button on the editing mosaic screen.

Import Cameras and Search

Import cameras from Flussonic Media Server

On a Flussonic Watcher launch, cameras will be available on the network. You can use either Onvif search or import streams from Flussonic Media Server.

You can import a Flussonic Watcher camera that was previously added to Flussonic Media Server.

Camera Search

Flussonic Watcher uses the WS-Discovery mechanism to search Onvif-compatible cameras. It can detect Ubiquity, Samsung and other cameras.

Some cameras require login and password for authorization. If you cannot find a camera, enter your login and password on the camera and search again.

Many cameras has two or more H264 streams, so you can add all of them as show on the video:

After you selected the camera profile, the correct value will appear in the RTSP URL. Enter a name that will be used for this camera.

See how to set the DVR path in the settings.

Watcher API

You can integrate the Watcher video surveillance system into your system and create custom mobile applications using the flexible API.

Flussonic Watcher API let you import or export users and cameras individually or in bulk. API has the advanced integration options with client billing. It allows you manage the availability of cameras, private archives and billing, change passwords, check camera status and solve other tasks that are available in the GUI.

Also, the API let you configure authentication via RADIUS-server or use an authorization backend.

User Import API Camera Import API integration with billing Change password Backend for user authorization RADIUS authorization

User import API

You can import users and settings from third-party systems, databases or spreadsheets using API. It requires API authorization. To do that, include the APIKEY that can be found in the Watcher settings in the

x-vsaas-api-key

HTTP header.

To import a list of users, create a CSV file and send it to:

http(s)://YOUR_WATCHER_URL/vsaas/api/import/users

CSV file fields:

email — the main user ID for authentication that is also used for password recovery.

password — plain text password.

is_active — 1: active user; 0: blocked user.

is_admin — 1: administrator user; 0: regular user.

groups — a list of groups to which the user belongs. Delimiter is a semicolon ';'.

note — a comment to this user.

Command-line import:

```
curl --data-binary @mydata.csv -H 'Content-type:text/plain; charset=utf-8' -H 'x-vsaas-api-key: API_KEY_ВАШЕГО ВОТЧЕРА' http://WATCHER-HOSTNAME/vsaas/api/import/users
```

mydata.csv example:

```
email,password,is_active,is_admin,note,groups
ivanov@domain.tld,CergitMig,1,0,user1,grp1;grp2
petrov@domain.tld,LajQuolOy,0,1,user2,grp2
```

One string example:

```
echo -e "email,password,is_active,is_admin,note,groups\nivanov@domain.tld,CergitMig,1,0,user1,grp1;grp2\npetrov@domain.tld,LajQuolOy,0,1,user2,grp2" | curl http://127.0.0.1:8080/vsaas/api/import/users?type=csv --data-binary @-  -H 'Content-type:text/plain; charset=utf-8' -H 'x-vsaas-api-key: 3a7d9386-6c3a-440d-a75d-e6b3fdc3368e'
```

Response:

```
{"success": true}
```

Camera Import API

dd { font-style: italic; /* cursive / } dt { margin-top: 1em; / top margin */ }

A POST request is used to import the cameras to the following URL:
http(s)://YOUR_WATCHER_URL/vsaas/api/interchange/usercameras

Console:

curl http://127.0.0.1:8080/vsaas/api/interchange/usercameras --data-binary @mydata.csv -H 'Content-type:text/csv' -H 'x-vsaas-api-key: <your api key>'

mydata.csv example:

stream_url,substream_url,thumbnails,onvif_url,onvif_profile,ptz,dvr_depth,dvr_path,enabled,access,title
rtsp://127.0.0.1:554,,,http://127.0.0.1:8899,000,0,3,storage,1,private,office_cam1
rtsp://127.0.0.2:554,,,http://127.0.0.2:8899,000,0,3,storage,1,private,office_cam2

One string example:

echo -
e "stream_url,substream_url,thumbnails,onvif_url,onvif_profile,ptz,dvr_depth,dvr_path,enabled,access,title\nrtsp://127.0.0.1:554,,,http://127.0.0.1:8899,000,0,3,storage,1,private,office_cam1\nrtsp://127.0.0.2:554,,,http://127.0.0.2:8899,000,0,3,storage,1,private,office_cam2" | curl http://127.0.0.1:8080/vsaas/api/interchange/usercameras?type=csv --data-binary @- -H 'Content-type:text/csv' -H 'x-vsaas-api-key: 3a7d9386-6c3a-440d-a75d-e6b3fdc3368e'

Response:

{"zu": 0, "cameras": [{"name": "office_cam2-689f1b1548", "created": true}, {"name": "office_cam1-c0ce3faa10", "created": true}], "users": [], "success": true, "zc": 2}}

CSV or JSON fields:

title: camera name.

name: stream name. Default is ‘title’ with a random suffix.

static: 1: constantly working stream, 0: on-demand stream.

stream_url: main stream RTSP URL.

substream_url: secondary stream RTSP URL (for multi-bitrate). Turned off by default.

thumbnails (string): camera snapshot URL. Send 0 instead of the URL to reset thumbnails. If you don’t know the camera snapshot URL, use 1. Watcher will turn it on automatically.
Note: we recommend that you use a direct URL. Otherwise, the server load will increase at 10%. If you specify the URL, a direct communication with the camera is established, saving your server processing power.

onvif_url: camera URL to access via the onvif protocol. By default, it’s set to ‘no.’
Onvif_profile — UNFIV profile.

onvif_profile: ONFIV profile

ptz: (0 or 1) — turn PTZ off/on (if camera supports).

access: camera access type (private / public / authorized). Public will be accessible to all users, private — to a camera owner only. Private is a default value.

owner: camera owner login.
enabled: (0 or 1) turns camera on / off.
dvr_path: path to save the archive. Default is no archive.
dvr_depth: (integer in days) — number of days to store the archive. 0 — disables the archive.
coordinates:geographical coordinates.
postal_address: camera’s postal address.
comment: camera comment.
agent_model (string): camera model.
agent_serial (string): camera serial number.
agent_id (string): camera agent unique number.
agent_key (string): special field used for Watcher authorization.
agent_pin (string): special field used for Watcher authorization.

Client billing integration

Camera provisioning automation (i.e., billing redirect) requires a firmware update from Flussonic Agent. Flussonic Agent is an optional module within the Watcher software system that automates the camera activation process through the client billing. This module establishes a secure tunnel between cameras and Watcher that allows connecting through NAT, and also it improves the video transport stability. Firmware modification can be customized for each client's camera model.

Learn how Agent works

When the very first connection is established between the camera and Agent, a request is sent to the provisioning server (i.e., activation server). It redirects Agent to the client’s server that has Watcher up and running.

After sending a request to the activation server, Agent generates a CSV and exports it to a client billing system. The export URL should be predefined.

The list of CSV attributes, sent from the Flussonic activation server to the client billing:

agent_model (string) - camera model.

agent_serial (string) - camera’s serial number.

agent_id (string) - Agent UID on the camera.

agent_key (string) - special field used for Watcher authorization.

agent_pin (string) - special field used for Watcher authorization.

stream_url (string) - primary stream’s RTSP-URL.

substream_url (string) - secondary stream’s RTSP-URL.

thumbnails (string) - URL snapshots from the camera.

onvif_url (string) - URL at which the camera can be accessed via onvif protocol.

onvif_profile (string) -service field.

ptz (0 or 1) - turns PTZ on / off.

The activation server sends information to the client billing to create new objects for each camera. It will append new cameras to others currently registered in the billing system. Another camera registration system can also be used instead of the standard billing. The agent_serial parameter is used for camera registration.

After the registration, other attributes can be added per camera object on the client billing side.

Client-side billing CSV Attributes:

owner (string) — login of the connected account.

dvr_depth (integer) - number of days to store video archives for the camera. 0 — disables the archive.

enabled (0 or 1) - turns camera on or off.

access (private / public) - camera visibility. It can be either accessible by the camera owner or my all users.

Note: These attributes are not transmitted.

The user should develop the mechanism of billing plans and other logic that will affect access to the file and settings, and to a certain video archive’s depth.

After you apply these settings (owner, dvr_depth, enabled, access), the data on all cameras and users will be automatically pushed to a CSV file here by the following:

http(s)://YOUR-WATCHER-URL/vsaas/api/interchange/usercameras?x-vsaas-api-key=(obtain the key in the Watcher settings)

User information also should be exported along with the camera information. User CSV Attributes: http(s)://YOUR-WATCHER-URL/vsaas/api/import/users

login (string) — the user's Watcher login.

is_active (0 limits watcher access) — cameras work, but the user can not enter Watcher.

dvr_allowed (0 limits video archive access) — cameras write to the archives, but users cannot have access to them.

It's possible to add cameras with users before activation. The process is described below:

1) Add a camera to the client billing (or another registration system) as an item with the attribute: agent_serial (serial number on a camera box). The following attributes should be set on the camera purchasing step:

owner — client login

dvr_depth — archive depth

enabled — camera status

access — access type

2) Then, when a technician turns on the camera in the client's home, the camera sends a request to the activation server, receives necessary activation attributes, then pushes the gathered information to the CSV file in the client billing. It will append new items based on the agent_serial attribute for previously added cameras.

3) When you have the camera information, it will be updated automatically.

Backend for user authorization

How it works:

An operator needs to realize the HTTP request handler, that has the logic to authenticate users.

The operator enters the path to the Flussonic Watcher authentication backend (Settings — Authentication backend).

A user logs into Flussonic Watcher using login/password.

Watcher transfers this data to the backend operator in a request payload.

Backend checks the incoming data.

If the user can be authenticated, the system returns the 200 HTTP code.

If the authentication data is incorrect, it returns the 403 HTTP code.

If the user is not found, the system gives 404.

Along with the login permissions, the backend can return the list of groups in the JSON body.

Flussonic Watcher checks the information about the groups in the database.

If the permission is granted, the password is passed as ancoded hash string and the group belongings are checked.

In case of decline, the user password is reset.

If the authentication backed was out of reach or send the response less than 2 seconds. The user check if being done in the Watcher’s database.

```
import falcon, json

class AuthResource:
  def on_get(self, req, resp):
    print "GET %r\n%r" % (req.uri, req.params)
    login = req.params.get('login', None)
    password = req.params.get('password', None)
    if not login or not password:
      print 'incorrect request login: %r, pass: %r' % (login, password)
      resp.status = falcon.HTTP_400
      return

    if login == 'user0':
      if password == 'letmein':
        return
      resp.status = falcon.HTTP_403
      return

    if login == 'user1':
      if password == 'letmein':
        resp.body = json.dumps(dict(groups=['a', 'b']))
        return
      resp.status = falcon.HTTP_403
      return
```

```
    resp.status = falcon.HTTP_404

app = falcon.API()
ad = AuthResource()

app.add_route('/auth', ad)
```

Examples

A user can get through with groups A and B:

```
curl -vvv http://localhost:8001/auth\?login\=user1\&password\=letmein

*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user1&password=letmein HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: gunicorn/19.7.0
< Date: Mon, 20 Mar 2017 10:16:12 GMT
< Connection: close
< content-length: 22
< content-type: application/json; charset=UTF-8
<
* Closing connection 0
{"groups": ["a", "b"]}
```

A user can get through without groups:

```
curl -vvv http://localhost:8001/auth\?login\=user0\&password\=letmein

*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user0&password=letmein HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Server: gunicorn/19.7.0
< Date: Mon, 20 Mar 2017 10:16:21 GMT
< Connection: close
< content-length: 0
< content-type: application/json; charset=UTF-8
<
* Closing connection 0
```

A user can’t get through:

```
curl -vvv http://localhost:8001/auth\?login\=user0\&password\=wrong

*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user0&password=wrong HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 403 Forbidden
< Server: gunicorn/19.7.0
< Date: Mon, 20 Mar 2017 10:16:27 GMT
< Connection: close
< content-length: 0
< content-type: application/json; charset=UTF-8
<
* Closing connection 0
```

A user is not found:

```
curl -vvv http://localhost:8001/auth\?login\=user10\&password\=wrong

*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 8001 (#0)
> GET /auth?login=user10&password=wrong HTTP/1.1
> Host: localhost:8001
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 404 Not Found
< Server: gunicorn/19.7.0
< Date: Mon, 20 Mar 2017 10:20:04 GMT
< Connection: close
< content-length: 0
< content-type: application/json; charset=UTF-8
<
* Closing connection 0
```

RADIUS authentication

RADIUS server can be used to authenticate Watcher users. It is especially useful, if you have a large number of users.

The setting can be enabled via admin interface:

The address 'radius://ldap.example.com:1812/secret' consists of 3 parts: host, port and secret. Change it according to your RADIUS server settings. Now, when a user try to login, Watcher redirects to the server via RADIUS protocol. Watcher sends User-Name and User-Password in the Access-Request query. The RADIUS server responds by giving the group list for this user. An attribute Filter-Id(11) is used here. Each group is stored in a separate attribute.

- Watcher redirects to RADIUS on every user log in.
- If the RADIUS answers Access-Accept, Watcher logs user in and saves the HEX password and group belongings to the database.
- If the RADIUS answers Access-Reject, the user becomes locked in the database.
- If the RADIUS did not answer, Watcher searches a user in the database.

It is necessary to bear in mind that RADIUS should know about all users, including administrators. The administrator user attribute can not be transferred to the RADIUS response and it can be assigned through Watcher only.

API for mobile applications

POST: /vsaas/api/login

Example:

`curl -H 'Content-Type: application/json' -
d '{"email": "email@example.com", "password": "passwwd"}' http://localhost:8080/vsaas/api/login`

Parameters:

HTTP request payload

{"email": <login>, "password": <password>}

Reply: JSON

```
{
   success: boolean,
   session: <session_token>
}
```

Adding a camera to Favorites

POST: /vsaas/my/fav/cameras

Parameters:

HTTP request payload

{"id": <string>} // camera id

Reply: JSON

{"success": true}

Removing a camera from Favorites

DELETE: /vsaas/my/fav/cameras/{cam_id}

Parameters:

cam_id camera id

Reply: JSON

{"success": true}

All the available cameras

GET: /vsaas/api/my/cameras

Parameters:

HTTP request headers

x-vsaas-session: <session_token>

A list of favourites cameras

GET: /vsaas/api/my/cameras?filter=favorite

Parameters:

HTTP request headers

x-vsaas-session: <session_token>

Reply: JSON

```
{
   "id": string
   "title": string
   "access": string // [“private”, “public”, “authorized”]
   "comment": string
   "postal_address": string
   "coordinates": string // "55.7512 37.6184"
   "streamer_url": string // "http://demo-watcher.flussonic.com"
   "dvr_depth": int // the depth of the archive (per days)
   "permissions": { // the user rights status
      "dvr": boolean // archive status
      "ptz": boolean // PTZ control
      "view": boolean // live-view
   }
   "ptz": boolean // PTZ control status
   "server": string // demo-watcher.flussonic.com
   "static": boolean // stream type: static or on-demand
   "status": { // information about the stream condition
      "source_error": string
      "alive": boolean
   }
   "thumbnails": boolean // the camera's snapshots status
   "urls": { // the links of any protocols receiving the stream
      "hls": string
      "hds": string
      "jpg": string
      "rtmp": string
      "rtsp": string
      "media_info": string // URL information of the stream
      "recording_status": string // URL archive status
   }
}
```

Integration of Flussonic Watcher SDK into Apps for Android

This guide describes how to use the Flussonic Watcher SDK to create custom Android apps for work with Watcher.

Our customers who use Flussonic Watcher develop their own mobile applications for Android. Also, they might want to expand the functionality of their applications by adding features working with IP cameras connected to Watcher.

We provide a set of developer tools for quickly integrating the features of a Flussonic Watcher mobile app into your own apllications.

The set of developer's tools includes:

The SDK for Android apps

With this SDK, you can integrate your apps with Flussonic Watcher player and its controls, as well as the block of images received from IP cameras.

Source code for a demo app for iOS with an example of SDK usage

The source code of a demo app with explanatory comments.

The SDK documentation that descibes what you can do with this SDK.

The documentation on the Flussonic Watcher SDK for Android includes:

Configuring gradle scripts

Configuring the manifest

About the Flussonic Watcher SDK

The demo application

API reference

Initializing SDK components

The React Native module of the SDK

Integration of Flussonic Watcher SDK into Apps for iOS

This guide describes how to use the Flussonic Watcher SDK to create custom iOS apps for work with Watcher.

Our customers who use Flussonic Watcher develop their own mobile applications for iOS. Also, they might want to expand the functionality of their applications by adding features working with IP cameras connected to Watcher.

We provide a set of developer tools for quickly integrating the features of a Flussonic Watcher mobile app into your own apllications.

The set of developer's tools includes:

The SDK for iOS apps

With this SDK, you can integrate your apps with Flussonic Watcher player and its controls, as well as the block of images received from IP cameras.

Source code for a demo app for iOS with an example of SDK usage

The source code of a demo app with explanatory comments.

The SDK documentation that descibes what you can do with this SDK.

The documentation on the Flussonic Watcher SDK for iOS includes:

About the Flussonic Watcher SDK

How to start using the SDK

The demo application

API reference