

Руководство по Watcher

30 July 2018

Введение в Flussonic Watcher

Аппаратные требования

Установка Watcher Single

Установка Watcher Cluster

Обновление структуры БД

Настройки

Сброс пароля

Фейловер захвата

Прием событий о движении с камеры

Auto-login

Пользовательский интерфейс

Мобильные приложения

Кастомизация интерфейса

Брендирование email

Watcher API

Импорт пользователей по API

Импорт камер по API

Интеграция в существующую систему

Бэкэнд для авторизации пользователей

RADIUS

API для мобильных приложений

Управление камерами

Создание клиентской мозаики

Управление пользователями

Группы

Предоставление доступа к камерам

Серийные номера

Импорт и Поиск камер

Введение в Flussonic Watcher

Flussonic Watcher

— программный комплекс, устанавливаемый на сервера и камеры клиента, для трансляции видео, записи и предоставления персонализированного удаленного доступа к IP-камерам и архиву.

С его помощью вы можете за несколько часов запустить свою собственную облачную систему видеонаблюдения и получить удаленный и безопасный доступ к вашим видеокамерам из любой точки мира через обычный веб-браузер или мобильное приложение.

Также, вы можете установить Flussonic Watcher и внутри локальной сети, сделав закрытую (Private Cloud) систему видеонаблюдения.

Презентация Flussonic Watcher

Операторам — для запуска собственного облачного сервиса абонентского видеонаблюдения.

Производствам и бизнесам — для аудио-видео фиксации и удаленного контроля за производственными процессами.

Управляющим компаниям и застройщикам — для трансляции видео с социально-значимых объектов, а также для контроля за строительством.

Муниципальным и федеральным властям — для предоставления населению доступа к публичным камерам, для закрытых проектов спец. служб, для программ типа «Безопасный город», «Безопасный регион».

Серверное ПО — база данных пользователей и камер, медиа-сервер (осуществляет захват, транскодирование, запись и мультипротокольную доставку видео на разные устройства), API для интеграции с внешними системами;

Веб-интерфейс — интерфейс для пользователей и админов, работающий в любом браузере. В интерфейсе вы найдете: dashboard с камерами и архивом, избранное, группы, карту, инструменты управления пользователями и камерами, настройки, инструменты брендинга интерфейса и многое другое;

Мобильные приложения для iOS и Android — удобный удаленный доступ к камерам, архиву и другим инструментам;

Прошивка для камер (агент) — устанавливается на камеры, для обеспечения видимости из-за NAT и стабильной шифрованной доставки видео напрямую от камеры на ваш сервер.

Single — базовая версия. Подходит тем, кому не нужно брендинга интерфейса (свое лого, цвета), для проектов до 500 камер и не более 1 сервера.

Cluster — продвинутая версия. Включает в себя инструменты брендинга (смена лого, цвета, текста), возможность расширения до десятков тысяч камер, включая возможность собрать кластер с резервированием потоков, прошивка для камер. Подходит тем, кто хочет запустить свой собственный абонентский сервис (VSaaS) или большую систему видеонаблюдения.

Аппаратные требования

Flussonic Watcher работает в двух вариантах:

Single — Подходит тем, кому не нужно брендирование интерфейса (свое лого, цвета), для проектов до 500 камер, не более 1 сервера.

Cluster — Продвинутая версия. Включает в себя инструменты брендирования (смена лого, цвета, текста), возможность расширения до десятков тысяч камер, включая возможность собрать кластер из десятка серверов с резервированием потоков. Для данной конфигурации требуется минимум 2 сервера. Подходит тем, кто делает абонентский сервис или большую систему видеонаблюдения.

Управляющий сервер и он же транслирующий:

Операционная система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: Не ниже CPU Xeon E-3 1230v5 3.4 GHz, 32Gb RAM;

База Данных: SQLite или PostgreSQL;

Диски: Расчет дискового пространства следует произвести из того, что 1 мбит/с камера за сутки пишет 10Гб видео, 2 мбит/с — 20Гб. Например, 1 мбит/с камера с 7-дневным архивом займет 70Гб.

Управляющий сервер (где стоит БД и веб-интерфейс Flussonic Watcher):

Система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: виртуальный или «железный» сервер, CPU с 2-я ядрами, 8Gb RAM;

База Данных: PostgreSQL;

Диски: SSD 64Гб.

Транслирующий сервер (серверы, куда заведены потоки с камер и пишется архив):

Система: Ubuntu 14.04 и выше, Debian 7 и выше;

Железо: Не ниже CPU Xeon E-3 1230v5 3.4 GHz, 32Gb RAM;

Диски: Расчет дискового пространства следует произвести из того, что 1 мбит/с камера за сутки пишет 10Гб видео, 2 мбит/с — 20Гб. Например, 1 мбит/с камера с 7-дневным архивом займет 70Гб.

Данные конфигурации серверов приведены как пример, пригодный для 500 камер, с 1 мбит/с потоком, 500 пользователей, с выключенным фейловером захвата и без собранных мозаик.

При увеличении битрейта до 2 мбит/с, кол-

во камер на 1 сервер следует сократить на 2 (до 250). Рекомендации актуальны только с учетом того, что на серверах не будет другого запущенного ПО.

Установка Watcher Single

Watcher Single подходит для маленьких и средних проектов, где максимальное количество камер не превышает 500. В данном режиме все камеры работают с одним сервером, где установлены база данных, веб-интерфейс, потоки и архив.

Flussonic Watcher умеет работать с двумя базами данных: SQLite и PostgreSQL.

Используйте SQLite только для тестов. Данная БД подключается очень просто (одной командой в интерфейсе администратора или конфигурационном файле), но имеет ограничения в виде невысокой скорости работы и невозможности сделать резервную копию.

Для полноценной работы рекомендуем использовать PostgreSQL.

Оглавление:

Первый способ установки (для SQLite)

Второй способ установки (для SQLite и PostgreSQL)

Первый запуск Watcher Single

Первый способ установки (для SQLite)

Данный способ реализуется через интерфейс администратора. Для запуска Watcher Single:

- 1) Установите Flussonic Media Server.
- 2) Зайдите в панель администратора: `http://flussonic_server:8080`.
- 3) Перейдите в раздел IP cameras.
- 4) В "Database path" укажите путь к БД:
`sqlite:///opt/flussonic/priv/vsaas.db`

- 5) Нажмите Сохранить.

После этого по адресу `http://flussonic_server:8080` будет открываться интерфейс Flussonic Watcher.

Для того, чтобы вернуться в панель администратора Flussonic Media Server, перейдите по адресу `http://flussonic:8080/admin`.

Второй способ установки (для SQLite и PostgreSQL)

Данный способ реализуется с помощью консоли и конфигурационного файла.

Watcher с SQLite:

- 1) Установите Flussonic Media Server.
- 2) Откройте конфигурационный файл `/etc/flussonic/flussonic.conf` в текстовом редакторе.
- 3) Вставьте строки:

```
vsaas {  
    database sqlite:///opt/flussonic/priv/vsaas.db;  
}
```

4) Сохраните и перезапустите Flussonic Media Server командой:

```
service flussonic restart
```

После этого по адресу http://flussonic_server:8080 будет открываться интерфейс Flussonic Watcher.

Для того, чтобы вернуться в панель администратора Flussonic Media Server, перейдите по адресу <http://flussonic:8080/admin>.

Важно! Для корректной работы Flussonic Watcher необходимо добавить в DNS-записи hostname Watcher сервера.

Flussonic Watcher с PostgreSQL:

Для установки PostgreSQL необходимо в консоли под root-доступом исполнить команду:

```
apt install postgresql
```

В случае с PostgreSQL базу и пользователя нужно создать самостоятельно, используя следующие команды в консоли:

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

Ответ: CREATE DATABASE vsaas_production ENCODING 'UTF8' TEMPLATE template0;

```
sudo -u postgres createuser -P vsaas
```

Ответ: Enter password for new role: (введите пароль супер-админа)

Ответ: Enter it again: (повторите пароль супер-админа)

Затем укажите созданную БД в конфигурационном файле `/etc/flussonic/flussonic.conf`, открыв его в текстовом редакторе и вставив туда строки:

```
vsaas {  
    database postgresql://vsaas:vsaas@localhost/vsaas_production;  
}
```

Сохраните и перезапустите Flussonic Media Server командой:

```
service flussonic restart
```

Теперь при заходе на главную страницу Flussonic Media Server (http://flussonic_server:8080) будет открываться интерфейс Flussonic Watcher.

Для того, чтобы вернуться в панель администратора Flussonic Media Server, перейдите по адресу <http://flussonic:8080/>.

Первый запуск Watcher Single

При первом запуске Flussonic Watcher по адресу <http://flussonic:8080/> вы попадете в веб-интерфейс системы, где появится меню создания первого администратора.

Введите email и пароль будущего администратора системы видеонаблюдения.

После создания аккаунта администратора вы попадете на страницу управления камерами, где сможете добавить камеры или перейти к другим настройкам системы.

Установка Watcher Cluster

Watcher Cluster работает минимум с двумя серверами и создан для сборки кластера.

Конфигурация серверов:

1 сервер — управляющий (веб-интерфейс + PostgreSQL).

1-100 серверов — стримеры (транслирующие сервера с потоками и архивами), которые контролируются управляющим сервером.

Транслирующий сервер (стример) — это сервер с установленным Flussonic Media Server, на котором хранятся архивы и потоки с камер.

С точки зрения организации видеонаблюдения Watcher Cluster отличается от Watcher Single следующим:

Watcher Cluster подходит для средних и крупных проектов, где требуется поддержка неограниченного количества камер.

Watcher Cluster поддерживает возможность собрать серверный кластер с инструментами обеспечения отказоустойчивости захвата потоков.

Watcher Cluster поддерживает инструменты брендинга веб-интерфейса (возможность поставить свой логотип, выбрать свои цвета и т.д.).

Watcher Cluster включает в себя прошивку для камер (агент), с помощью которого становится возможной видимость камер из-за NAT. Агент делает возможным plug-and-play режим, а также повышает стабильность и реализует шифрование доставки видео напрямую с камер на стримеры (для подготовки прошивки обратитесь к нашему менеджеру, который ведет ваш проект).

Watcher Cluster работает только с базой данных PostgreSQL и сам проинициализирует БД при первом запуске.

ВАЖНО! При обновлениях Flussonic Watcher необходимо делать миграцию. Мы рекомендуем при каждом обновлении сервера делать резервное копирование базы данных.

Установка PostgreSQL

Прежде чем приступить к установке Watcher Cluster, необходимо установить Flussonic Media Server и базу данных PostgreSQL.

Для установки PostgreSQL в консоли под root-доступом выполните команду:

```
apt install postgresql
```

Базу данных и пользователя необходимо создать самостоятельно. Для этого выполните команды в консоли:

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

Ответ: CREATE DATABASE vsaas_production ENCODING 'UTF8' TEMPLATE template0;

```
sudo -u postgres createuser -P vsaas
```

Ответ: Enter password for new role: (введите пароль супер-админа)

Ответ: Enter it again: (повторите пароль супер-админа)

Затем укажите созданную БД в конфигурационном файле /etc/flussonic/flussonic.conf. Откройте его в текстовом редакторе и вставьте строки:

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

В [PASSWORD] впишите пароль, который вы указали при создании пользователя базы данных.

После внесенных изменений перезапустите Flussonic Media Server командой:

```
service flussonic restart
```

После этого по адресу http://flussonic_server:8080 будет открываться интерфейс Flussonic Watcher.

Для того, чтобы вернуться в панель администратора Flussonic Media Server, перейдите по адресу <http://flussonic:8080/admin>.

ВАЖНО! Для корректной работы Watcher необходимо добавить в DNS-записи hostname Watcher сервера.

После установки последует создание главного администратора.

Создание администратора

При первом запуске Flussonic Watcher в веб-интерфейсе появится меню создания первого администратора. На этом экране необходимо ввести e-mail и пароль будущего администратора системы видеонаблюдения.

Далее, вам нужно перейти в раздел Настройки > Стримеры > Edit > Создать кластер.

Создание кластера

После авторизации в Flussonic Watcher под администратором, необходимо перейти в Настройки > Стримеры > Edit.

Данный раздел предназначен для создания кластера (многосерверного режима). В кластерном режиме камеры не заработают пока в настройках Flussonic Watcher вы не добавите хотя бы один сервер типа «стример», на котором они будут расположены.

Перед добавлением нового стримера вам необходимо установить на отдельный сервер Flussonic Media Server. То есть помимо сервера, на котором вы установили Flussonic Watcher, должен быть «поднят» еще один сервер с внешним (публичным) IP-адресом (пользователь должен быть в состоянии увидеть эти сервера по прямому адресу), а также прописанным cluster_key (он должен совпадать с cluster_key Flussonic Watcher).

Подробнее о cluster_key.

Когда выделенный сервер (стример) для камер готов к работе, нужно добавить его в настройках Flussonic Watcher.

Перейдите в Настройки > Стримеры > Edit и заполните карточку Новый стример:

Hostname – URL стримера. Пример: <http://192.168.1.2:8080>, но лучше доменное имя.

Cluster key – cluster key (если кластерный ключ совпадает у Flussonic Watcher и стримера, то заполнять данное поле необязательно).

Путь к архиву – пример: /dvr, или множество путей (через пробел): /dvr1 /dvr2 и т.д. (важно заполнить данное поле, иначе архив не будет работать).

При добавлении нескольких стримеров вам необходимо выбрать основной, нажав на «Выбрать по умолчанию» в карточке созданного стримера. В этом случае все новые камеры будут добавляться на него автоматически.

Важно!

В разделе Стримеры не нужно добавлять хост, на котором развернут сам Flussonic Watcher!

На всех серверах должно стоять одинаковое время.

Для каждого стримера вы можете включить резервирование потоков (фейловер).

Как это сделать указано в разделе «Фейловер захвата».

Обновление структуры БД

Иногда новая версия Flussonic Watcher требует для работы изменений в структуре своей базы данных. В этом случае web-интерфейс покажет сообщение:

Изменения в структуру базы вносит скрипт:

```
/opt/flussonic/contrib/watcher_db_migrate.sh
```

Его необходимо запустить на сервере вручную.

Пример успешного выполнения скрипта:

```
# /opt/flussonic/contrib/watcher_db_migrate.sh
INFO [alembic.runtime.migration] Context impl SQLiteImpl.
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
```

После этого интерфейс Flussonic Watcher будет снова доступен для работы.

Важно! Перед запуском этого скрипта необходимо сделать резервную копию базы.

Если в ходе выполнения watcher_db_migrate.sh появится ошибка:

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

Необходимо установить предыдущую версию(до даунгрейда/обновления) и выполнить :

```
cd /opt/flussonic/apps/vsaas
/opt/flussonic/bin/python -m manage db downgrade
```

Тип базы и место её хранения описано в конфигурации Flussonic Media Server:
/etc/flussonic/flussonic.conf.

Sqlite:

```
vsaas {
    database sqlite:///opt/flussonic/priv/vsaas.db;
}
```

Скопируйте файл /opt/flussonic/priv/vsaas.db.

Postgresql:

```
vsaas {
    database postgresql://vsaas:[password]@localhost/vsaas_production;
}
```

Выполните резервное копирование с помощью утилиты pg_dump:

```
pg_dump -U vsaas vsaas > /path/to/backup/vsaas.sql
```

Настройки

Для администратора доступны следующие настройки Flussonic Watcher.

Режим работы — режим работы: многосерверный кластер или запуск плагином на одном сервере.

Стримеры — становится активным в кластерном режиме.

Archive path — путь к хранилищу DVR. Может быть Swift-хранилищем, доступно только в standalone режиме с запуском на одном сервере. В кластерном режиме путь для DVR задается в настройках стримера.

API ключ — токен использующийся для мобильного доступа. Необходимо использовать в личном кабинете на сайте «Эрливидео» для активации доступа к мобильным приложениям.

Доступ для гостей — разрешать ли доступ незалогиненным пользователям.

Самостоятельная регистрация пользователей — включить возможность самостоятельной регистрации пользователей.

Главная страницы — какую страницу показывать автоизованным пользователям: карту или дашбоард.

Главная страница для гостей — какую страницу показывать гостям: карту или дашбоард.

Тип панели по-умолчанию — какую страницу показывать на дашборде: камеры или группы.

External Authentication — внешняя авторизация. Указывается http адрес или адрес RADIUS сервера.

Показывать карту в гл. меню — можно добавить или убрать карту из бокового меню.

Центр карты — по каким координатам будет отцентрирована карта.

Язык карты — язык карты.

Движок карт — выбор поставщика карт: Google Maps, OpenStreetMaps или OpenStreetMaps Offline.

Управление серийными номерами — управление серийными номерами камер в Flussonic Watcher. Нужно в случае, когда вы используете агента.

Дополнительный текст в заголовке — подробнее в разделе «Кастомизация интерфейса»

Свой заголовок страницы — подробнее в разделе «Кастомизация интерфейса»

Свой логотип — подробнее в разделе «Кастомизация интерфейса»

Свой логотип для страницы входа — подробнее в разделе «Кастомизация интерфейса»

Своя цветовая схема — подробнее в разделе «Кастомизация интерфейса»

Сброс пароля

Восстановление забытого пароля возможно с помощью отправки специального письма на почтовый адрес пользователя. Письмо содержит ссылку для восстановления пароля.

Чтобы этот механизм заработал, в Flussonic Watcher должны быть настроен SMTP сервер для отправки писем.

Также пароль можно поменять с помощью специальной утилиты, которая входит в поставку Flussonic Media Server:

```
/opt/flussonic/contrib/reset_password email.address@example.com pwd123xyz
```

Первым аргументом укажите e-mail, вторым — новый пароль.

В ответ программа должна вывести на экран:

```
Changing password for email.address@example.com
```

Фейловер захвата

Функция фейловер захвата разработана, чтобы при падении одного из стримеров камеры (стримы) равномерно распределялись по работающим серверам кластера.

Доступ к архиву на упавшем сервере пропадает, начинается запись нового архива (на донорском сервере).

При восстановлении работы основного сервера камеры автоматически возвращаются обратно, возобновляя доступ к старому архиву (если он не был поврежден), но теряя архив, записанный на резервном сервере.

Для включения фейловера, необходимо нажать на соответствующую кнопку в карточке каждого стримера, в разделе «Стримеры» и поменять режим работы в файле `/etc/flussonic/flussonic.conf` на сервере Watcher:

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

Где 30 — частота проверки стримеров в секундах. После изменения настройки нужно перезапустить Watcher:

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

Прием событий о движении с камеры

Прием событий о движении с камеры

Flussonic Watcher умеет принимать события по email, для этого у него есть специальный smtp-сервер. Настройка состоит из двух этапов:

Настройка приема писем в Flussonic Watcher

Настройка камеры на отправку писем через smtp-сервер Flussonic Watcher.

Как работает запись по движению:

Flussonic Watcher ведет постоянную запись с указанной глубиной записи как обычно, но когда прилетает e-mail с движением, он отмечает этот интервал у себя в базе данных, чтобы отразить метки в архивном плеере и, что самое важное, защищает эту запись от удаления.

Это значит, что вам достаточно установить глубину архива: например, 6 часов. И затем включить прием событий. В итоге вы получите 6 часов непрерывного архива + записи по датчику будут храниться до тех пор, пока на диске есть свободное место. Запись новых событий будет удалять старые.

Расчитав необходимый объем диска исходя из битрейта камер и частоты движений, можно сэкономить до 50-90% дискового пространства по сравнению с обычной записью без событий.

Включаем прием эвентов в Watcher

Для включения поддержки захвата событий о движении с камеры необходимо открыть конфигурационный файл /etc/flussonic/flussonic.conf и прописать camera_alarm плагин:

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

Параметр catch определяет слово, которое Flussonic Watcher будет искать в теме письма.

Большинство камер по умолчанию присылают письма примерно с таким заголовком: "Camera 123 Motion Detected at 14:21 27-10-2017".

Если ваша камера отправляет письмо совсем с другой темой или имеет возможно задать свой, то настройте catch по своему усмотрению.

Параметр listen определяет интерфейс и порт для встроенного smtp-сервера. Тут же можно указать логин и пароль: listen smtp://username:password@0.0.0.0:1025;

Перезапустите Flussonic Media Server. На странице редактирования камеры появится возможность включить сбор событий с камеры.

На странице управления камерами появится ссылка на список событий.

Настройка камеры

Настройка камеры сводится к указанию адреса smtp-сервера, имени отправителя и получателя.

Очень важно указать правильно имя отправителя и получателя. Это должно быть полное имя камеры, например: cam1-abcdefg@example.com. Полное имя камеры можно найти в

интерфейсе Flussonic Watcher. Будьте внимательны, если вы назвали камеру cam1, то Flussonic Watcher присвоит ей ID и получится полное имя — cam1-abcdefg.

И в интерфейсе архива камеры будут появляться отметки о движении.

Auto-login

В Flussonic Watcher есть возможность входа пользователя без ввода пароля по специальной ссылке. Это может быть необходимо, если вы хотите упростить вход в Flussonic Watcher для своих пользователей или предотвратить передачу паролей третьим лицам. Предполагается, что ссылка для автологина выдается уже авторизованному клиенту.

Для того чтобы сгенерировать URL адрес для автологина, нужно предварительно запросить у Flussonic Watcher такой токей ссылке `/vsaas/api/generate-autologin-token`.

После этого пользователя можно авторизовать через POST на `/vsaas/autologin`.

1) Предварительно необходимо сгенерировать админский токен:

Нам понадобится API KEY со страницы настроек административного интерфейса Flussonic Watcher, для того чтобы генерировать админский токен.

```
sign = md5(salt + ":" + utc + ":" + api_key)
```

```
admin_token = salt + ":" + utc + ":" + sign
```

где:

`salt` — любая случайная строка;

`utc` — текущее UTC время в секундах;

`api_key` — ключ API со страницы настроек админки Flussonic Watcher.

Сгенерированный токен имеет конечное время жизни и привязывается к указанному времени `utc`.

Например, если

```
salt = "20a666"
```

```
utc = "1487258700"
```

```
api_key = "HELLO"
```

тогда итоговый `admin_token` будет: `"20a666:1487258700:4b60f36de708e5b3472155db2fea990a"`

2) Итак, админский токен уже готов, теперь нужно получить токен для автологина конкретного пользователя. Для этого нужно сделать POST запрос в json-формате:

```
curl --header "X-Vsaas-API-Key: ADMIN_TOKEN" --header "Content-Type: application/json" --request POST --data '{"login": LOGIN, "valid_till": VALID_TILL, "lifetime": LIFETIME}' "http://watcher.com/vsaas/api/generate-autologin-token"
```

`ADMIN_TOKEN` — токен сгенерированный в шаге 1. Этот токен передаётся в HTTP-заголовке `X-Vsaas-API-Key`.

`LOGIN` — login (он же email) пользователя к которому нужно дать доступ. Строка. Обязательный параметр.

`VALID_TILL` — время UTC в секундах до сколько этот токен будет действителен для автологина. Целое число. Необязательный параметр.

`LIFETIME` — длительность сессии открываемой через автологин в секундах. Целое число. Необязательный параметр.

Ответ будет в JSON формате:

```
{
```

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

Нам нужен этот autologin_token.

3) autologin_token полученный на шаге 2 используется для POST-запроса в Flussonic Watcher. Например, таким образом:

По клику пользователя должно перекинуть в интерфейс Flussonic Watcher и залогинить.

Пользовательский интерфейс

После добавления пользователя в систему, он может зайти в интерфейс Flussonic Watcher под своим e-mail и паролем.

После этого он сможет посмотреть доступные камеры, которые ему разрешил смотреть администратор: публичные, только для авторизованных и приватные.

Пользователю доступны следующие меню:

- Дашбоард «Мои камеры».

- Карта.

- Избранное.

- Камеры.

- Пользователи и группы. (Для Администратора)

- Настройки. (Для Администратора)

Дашбоард «Мои камеры»

Дашбоард «Мои камеры» — основная страница доступа к камерам, которые доступны пользователю.

Доступны фильтры по доступу:

- Любой.

- Публичный.

- Для клиентов.

- Частный.

Режимы отображения камер:

- Средняя сетка.

- Крупная сетка.

- Список.

Также можно включить Компактный режим и Скрыть отключенные камеры.

В этом списке показываются скриншоты с камер, которые периодически обновляются. Чтобы просмотреть какую-либо камеру или посмотреть архив, достаточно кликнуть на неё или на кнопку Play.

Нажав на название камеры, можно открыть ее на просмотр в большом окне или перейти на отдельную страницу просмотра архива:

Мозаика — это возможность собрать на отдельной странице до 8 любых камер по выбору пользователя. Чтобы пользователю были доступны мозаики, их необходимо создать в разделе Камеры.

Карта

На карте показаны камеры, для которых в настройках указаны координаты.

Избранное

В «Избранное» добавляются камеры, которые пользователь отметил «звёздочкой».

Камеры

В меню «Камеры» производятся все операции с камерами:

- Добавление камеры.

- Поиск камеры по протоколу ONVIF.

- Импорт из Flussonic.

Также есть возможности:

- Экспорт списка файлов в формате CSV.

- Просмотр списка серийных номеров камер.

- Создание мозаики.

Подробнее про добавление камеры в разделе «Управление камерами».

Пользователи и группы

В этом меню производится добавление и изменение пользователей Flussonic Watcher.

Подробнее в разделе «Управление пользователями»

С помощью групп вы сможете упростить навигацию по камерам и пользователям, делая логическую группировку (например, по этажам, отделам, регионам, микрорайонам и т.д.), а также облегчить процесс распределения прав видимости камер. Подробнее в разделе «Группы»

Настройки

В этом меню производится настройка Flussonic Watcher. Подробнее в разделе «Настройки»

Мобильные приложения

ВАЖНО!

Для того, чтобы мобильные приложения знали с какого сервера забирать видео, необходимо получить

Оператор ID

, куда будет зашит адрес вашего Flussonic Watcher.

По умолчанию, приложение обращается к серверу «Эрливидео», поэтому без создания личного Оператор ID авторизация в мобильном приложении работать не будет.

Для получения Оператор ID необходимо залогиниться в личном кабинете, кликнуть на ключ инсталляции на главном экране ЛК и в поле «Watcher URL» указать публичный URL вашего Flussonic Watcher в виде: `http://APIKEY@watcher-host`.

Адрес Flussonic Watcher должен быть виден из интернета и, желательно, быть в виде доменного имени, а не IP-адреса.

Для использования мобильного приложения вам необходимо скачать его в Apple Store или Google Play, а далее авторизоваться под созданным в вашем Watcher логином/паролем и полученным в `manage.erlyvideo.org` Оператор ID.

APIKEY — идентификационный ключ, указан в интерфейсе настроек Flussonic Watcher.

Watcher-Hostname — публично видимый путь к Flussonic Watcher.

ВАЖНО! Для работы мобильного приложения необходимо, чтобы ваш сервер был виден из сети интернет и обладал фиксированным белым IP-адресом. Если ваш сервер стоит за NAT или защищен Firewall, вам необходимо сделать проброс порта 1935 до сервера (протокол RTMP) для корректной работы приложения.

Кастомизация интерфейса

В

Watcher Cluster

есть инструментарий для кастомизации интерфейса. Для этого найдите в разделе

Настройки

:

Дополнительный текст в заголовке. Добавление произвольного текста в заголовок интерфейса.

Свой заголовок страницы. Изменение заголовка.

Свой логотип. Установка логотипа в формате PNG. Для корректного отображения логотип рекомендуется загружать вариант в белом/сером цвете на прозрачном фоне, высота — не более 50 пикселей.

Свой логотип для страницы входа.

Своя цветовая схема. Изменение цветовой схемы. Есть четыре готовых темы и возможность самостоятельного редактирования CSS. Свои CSS-правила следует сохранить в файле:

/vsaas/static/css/user.css. Файл user.css.example может быть использован в качестве примера.

Также вы можете брендировать email-шаблон восстановления пароля. Подробнее в статье [Брендирование email](#).

Брендинг email

Настройка SMTP сервера

Для настройки почты и отправки писем восстановления паролей в конфигурации необходимо указать параметры сервера исходящих сообщений.

Пример. Используем параметры:

Почтовый адрес: email.address@example.com;

Пароль: xyz123;

Адрес почтового сервера: smtp.example.com;

Порт: 465;

Защита соединения: SSL

Пример конфигурации:

```
vsaas {  
  database sqlite://opt/flussonic/priv/vsaas.db;  
  smtp_server smtp.example.com:465;  
  smtp_login email.address:xyz123;  
  email_from "Flussonic Watcher ";  
  smtp_opts ssl;  
}
```

Смена шаблона

В рамках опции брендинга Flussonic Watcher есть возможность поменять email-шаблон восстановления пароля. Для того чтобы использовать свой собственный шаблон письма сброса пароля, либо html-шаблон необходимо сделать следующее:

Зайти в директорию /opt/flussonic/apps/vsaas/watcher/templates.

Создать свои собственные шаблоны писем Можно использовать файлы password_reset_request.email и password_changed.email, как пример.

Сохранить свои шаблоны используя приставку custom_. Таким образом на сервере должны появиться два файла custom_password_reset_request.email и custom_password_changed.email.

Для того, чтобы использовать ещё и html-шаблоны, нужно в той же директории создать ещё два файла по такому же принципу: custom_password_reset_request.email.html и custom_password_changed.email.html.

Шаблон делится на две части:

Заголовок и непосредственно тело шаблона.

На данный момент в заголовке можно указывать тему письма.

Кроме того, в теле можно использовать две переменных:

{{data.base_url}}

{{data.token}}

Примеры шаблонов:

custom_password_reset_request.email:

subject: "Это тема письма"

А это тело письма, вот ссылка: `{{data.base_url}}/vsaas/forgot-password/{{data.token}}`

erlang
custom_password_reset_request.email.html:

subject: "Это тема письма"

Тело письма, ссылка

Watcher API

С помощью Watcher API вы сможете интегрировать сервис видеонаблюдения в вашу экосистему или сделать свои мобильные приложения.

Для Flussonic Watcher разработан API, с помощью которого можно импортировать и экспортировать пользователей и камеры, включая массовый импорт-экспорт. API имеет развитый инструментарий интеграции с биллингом, позволяющий управлять доступностью камер, частных архивов и отдельных пользователей из биллинга, в том числе менять пароли, собирать статусы камер и решать прочие задачи, доступные в интерфейсе.

Также, API позволяет настроить авторизацию через RADIUS-сервер или использовать авторизационный бэкэнд.

Импорт пользователей по API Импорт камер по API Интеграция с биллингом Смена пароля Бэкэнд для авторизации пользователей RADIUS

Импорт пользователей по API

В Flussonic Watcher есть API, которое позволяет разом заливать большое количество уже созданных пользователей и их настройки из внешней системы, базы данных или таблицы. Запросы к API должны быть авторизованы с помощью API-ключа. Для этого вы должны отправить API-ключ в http заголовке x-vsaa-api-key.

Вы можете найти свой APIKEY в разделе Настройки Watcher-a.

Для импорта списка пользователей надо сформировать CSV с полями и отправить на `http(s)://URL_ВАШЕГО_ВОТЧЕРА/vsaas/api/import/users`:

email — E-mail используется для идентификации пользователей и восстановления пароля

password — пароль открытым текстом

is_active 1: активный пользователь, 0: пользователь заблокирован

is_admin 1: пользователь является администратором, 0 - обычный пользователь

groups — список групп, к которым принадлежит пользователь. Разделитель - ";"

note — комментарий к пользователю

Импорт через команду в консоли:

```
curl --data-binary @mydata.csv -H 'Content-type:text/csv' -H 'x-vsaa-api-key: API_KEY_ВАШЕГО_ВОТЧЕРА' http://WATCHER-HOSTNAME/vsaas/api/import/users
```

Пример файла mydata.csv:

```
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
```

Пример одной строкой:

```
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 --data-binary @-
-H 'Content-type:text/csv' -H 'x-vsaa-api-key: 3a7d9386-6c3a-440d-a75d-e6b3fdc3368e'
```

Ответ:

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

Импорт камер по API

```
dd { font-style: italic; /* Курсивное начертание текста <em>/> } dt { margin-top: 1em; /</em>
Отступ сверху */ }
```

Импорт камер выполняется через POST запрос на URL

`http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/interchange/usercameras`

Импорт через команду в консоли:

```
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: '
```

Пример файла mydata.csv:

```
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
```

Пример одной строкой:

```
echo -
e "stream_url,substream_url,thumbnails,onvif_url,onvif_profile,ptz,dvr_depth,dvr_path,enabled,access,title\nr
tsp://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/us
ercameras --data-binary @- -H 'Content-type:text/csv' -H 'x-vsaas-api-key: 3a7d9386-6c3a-440d-a75d-
e6b3fdc3368e'
```

Ответ:

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

Надо передать CSV или JSON со следующими полями:

title: Название камеры.

name: Название потока. По умолчанию — title плюс случайный суффикс.

static: 1 — постоянно работающий поток (static), 0 — поток по запросу (ondemand).

stream_url: RTSP URL основного потока.

substream_url: RTSP URL вторичного видеопотока (для мультибитрейта). По умолчанию нет.

thumbnails (строка): URL снепшотов с камеры. Отправив 0 вместо URL вы можете сбросить адрес thumbnails. Если вы не знаете URL снепшотов с камеры, передайте в строке значение 1 и их включит Watcher (в данном варианте будет 10% увеличение нагрузки на сервер, рекомендуем использовать URL, для общения с камерой напрямую).

onvif_url: URL по которому камера будет отвечать по onvif протоколу. По умолчанию нет.

onvif_profile: ONVIF profile

ptz: (0 или 1) — вкл/выкл PTZ (если камера поддерживает)

access: Тип доступа к камере: (private/public/authorized) — видимость камеры, публичная она будет (для всех абонентов) или приватная (только для owner'a камеры). По умолчанию — private.

owner: (логин привязанного абонента / владельца камеры)

enabled: (0 или 1) — вкл/выкл камеры.

dvr_path: Путь для сохранения архива. По умолчанию — архива нет.

dvr_depth: (целое число в днях) — количество дней записи архива для камеры. 0 — отключает архив.

coordinates: Координаты.

postal_address: Адрес.

comment: Комментарий.

agent_model (строка): модель камеры.

agent_serial (строка): серийный номер камеры.

agent_id (строка): уникальный номер агента на камере.

agent_key (строка): специальное поле, используемое для авторизации камеры в Flussonic Watcher.

agent_pin (строка): специальное поле, используемое для авторизации камеры в Flussonic Watcher.

Интеграция в существующую систему

В этой статье описан типовой сценарий внедрения Flussonic Watcher с контролируемой продажей камер по подписке и учетом абонентов и их услуг в сторонней системе. Далее будут использоваться термины:

клиент — клиент компании Эрливидео, владелец сервиса

абонент — абонент у клиента, пользователь сервиса

биллинг — система внешняя к Watcher, в ней ведется тарификация услуг клиента абонентам и взимание денег

Сценарий следующий:

На партию камер заливается модифицированная прошивка с Flussonic Agent

В этой прошивке зашита информация о том, к какому Flussonic Watcher надо привязать эту камеру

Камеру с серийником клиент заносит в систему инвентаризации биллинга, пока она ещё лежит на складе

При продаже абоненту сотрудник клиента связывает в биллинге серийный номер камеры с идентификатором абонента

При первом включении камера от нашего сервера активации получает данные для авторизации в Flussonic Watcher.

Эти данные никак не связаны с идентификатором абонента, это авторизация камеры

Активированная камера немедленно начинает попытки соединения с Flussonic Watcher

Сервер активации посылает данные о камере напрямую в Watcher или в биллинг

Биллинг получает информацию о свежесозданной камере, добавляет к ней идентификатор абонента, занесенный ранее в систему инвентаризации

Биллинг отправляет информацию о свежесозданной камере в Watcher и камера там заводится. Этот и предыдущий пункты необходимо реализовать в рамках интеграции на стороне клиента

Теперь камера может подключиться к Watcher и начать отдавать видео на Flussonic

Важно то, что при такой организации процесса не требуется никакой настройки роутеров, камер и прочих сетевых устройств у абонента. После включения камеры в сеть она автоматически появится в личном кабинете.

Агент и модификация прошивки не являются обязательным требованием, всё может работать и со стандартными, неизмененными камерами, ниже будут описаны детали.

Подробнее об агенте в подробной статье.

API биллинга

Со стороны клиента надо реализовать в биллинге API для приема данных о свежесозданных камерах и отправку этих данных в Watcher. Такая схема с проксированием данных нужна для добавления информации о владельце камеры и услугах, которые доступны на этой камере.

Вся концепция подобного использования биллинга подразумевает, что именно он является центральным местом хранения данных в системе, а не Flussonic Watcher. Такая практика

является стандартной и позволяет централизованно управлять услугами в разных системах, связывая, например, умный дом и видеонаблюдение в едином проекте.

Сервер активации, обслуживаемый Эрливидео, присылает запрос на сконфигурированный урл (по вопросу конфигурации необходимо обратиться к техподдержке) с CSV или списком JSON объектов.

Все данные, которые присылает сервер активации надо переслать в Flussonic Watcher без изменений, если только нет задачи по какой-либо причине их поменять. Так, например, может прийти флаг ptz=1, его можно выставить в 0, если не хочется вообще управлять этой камерой в Watcher.

Поля, передаваемые от сервера активации (Эрливидео) в биллинг:

agent_model (строка) — модель камеры

agent_serial (строка) — серийный номер камеры

agent_id (строка) — уникальный номер агента на камере

agent_key (строка) — специальное поле, используемое для авторизации камеры Watcher'ом

stream_url (строка) — основной RTSP-URL потока

substream_url (строка) — вторичный RTSP-URL потока

thumbnails (строка) — URL снэпшотов с камеры

onvif_url (строка) — URL по которому камера будет отвечать по onvif протоколу

onvif_profile (строка) — служебное поле

ptz (0 или 1) — вкл/выкл PTZ

Пришедшая от сервера активации информация должна создавать в биллинге новые записи для камер, либо «склеивать» эту информацию с уже существующими в биллинге или другой системе учета камер, камеры по параметру agent_serial (серийный номер камеры). Важно понимать, что agent_id может поменяться в случае, если камеру сбросили или передали другому абоненту. Серийный номер у камеры меняться не должен.

Т.е. если в биллинге существует система инвентаризации в которой камера привязывается к абоненту до первого включения, то новая запись появляться не будет, вместо этого надо заполнить пропущенные поля в существующей строчке в БД.

На стороне биллинга можно быть реализована возможность дозаполнять атрибуты к камерам, например привязка к владельцу или детали по управлению услугами для формирования тарифных планов.

Управление камерами в Watcher

Биллинг может управлять камерами, которые существуют в Watcher. В частности при получении данных от сервера активации он может дополнить данные своими полями и отправить информацию о камерах в Watcher.

Атрибуты заполняемые на стороне биллинга:

owner — (логин привязанного абонента).

dvr_depth (целое число в днях) - кол-во дней записи архива для камеры. 0 — отключает архив.

enabled (0 или 1) - вкл/выкл камеры.

access (private/public) - видимость камеры, публичная она будет (для всех абонентов) или приватная (только для owner'a камеры).

Разработка механизма создания тарифных планов и логики, которая будет влиять на доступ,

глубину архива и прочие настройки, находится на стороне клиента.

После добавления полей индивидуальной настройки (owner, dvr_depth, enabled, access) необходимо отправить расширенный список камер в формате CSV в Watcher, по ссылке: [http\(s\)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v2/cameras/import](http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v2/cameras/import).

```
curl -v http://localhost:8080/vsaas/api/v2/cameras/import \
-d \
'[{
  "name":"cam1","stream_url":"fake://clock","access":"private",
  "enabled":true,"dvr_depth":3,"agent_id":"123098456","agent_serial":"mJ0ODnktZFc",
  "agent_key":"salt:secretkey","owner":"alex@smith.com","dvr_path":"movies"
}]' \
-H "X-Vsaas-API-Key: 7ab056b1-5bb1-4501-b528-d69538392842" \
-H "Content-Type: application/json"
....
{
  "deleted": 0,
  "updated": 0,
  "errors": {},
  "created": 1
}
```

Здесь X-Vsaas-API-Key: 7ab056b1-5bb1-4501-b528-d69538392842 это API key из настроек Watcher.

В случае если биллинг отправляет в Watcher камеру с идентификатором несуществующего пользователя (поле owner), Watcher создает нового пользователя. Его пароль при этом не передается. Мы подразумеваем, что клиент не хранит пароли абонентов в открытом виде и никак не может передать. Для того, что бы абонент мог залогиниться в Watcher, необходимо настроить внешний авторизационный бекенд в Watcher.

Управление пользователями (абонентами)

Параллельно с информацией по камерам можно отправлять в Watcher информацию об абонентах.

Вызов [http\(s\)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v1/users](http(s)://URL_ВАШЕГО_БОТЧЕРА/vsaas/api/v1/users)

Пример:

```
curl -v http://localhost:8080/vsaas/api/v2/users \
-d \
'[{
  "login":"user@domain.com","dvr_allowed":true,"external_id":"12346780"
}]' \
-H "X-Vsaas-API-Key: 7ab056b1-5bb1-4501-b528-d69538392842" \
-H "Content-Type: application/json"
...
[
  {
    "authorized_ip": null,
    "enabled": null,
    "id": null,
    "note": null,
```

```
"is_admin": null,  
"dvr_allowed": true,  
"notification_email": null,  
"external_id": "12346780",  
"login": "user@domain.com"  
}  
]
```

Полный список полей, которые можно отправлять по ссылке на полную документацию

Бэкэнд для авторизации пользователей

Схема работы

Оператор реализует обработчик http-запросов, в котором делает необходимую ему логику по аутентификации абонентов (пример в комментарии ниже).

Оператор указывает в настройках Flussonic Watcher http url к аутентификационному бекенду (settings/authentication backend).

Абонент логинится в Flussonic Watcher (отправляет логин/пароль со страницы).

Flussonic Watcher передает их бекенду оператора в параметрах запроса.

Бекенд на основе собственных данных принимает решение о том пускать абонента или нет.

В случае, когда абоненту разрешен вход, бекенд возвращает http код 200.

В противном случае, бекенд возвращает http код 403.

Если абонент не был найден, ожидается код 404.

Вместе с разрешением на вход, бекенд может вернуть в теле ответа json с указанием актуальных групп этого абонента.

Получив ответ, Flussonic Watcher актуализирует информацию о пользователе в собственной базе.

При полученном разрешении обновляется хеш пароля абонента и список групп, в которые этот абонент входит.

При отказе пароль абонента в Flussonic Watcher аннулируется.

В случае, если аутентификационный бекенд был недоступен или не успел ответить за 2 секунды, проверка абонента происходит по внутренней базе Flussonic Watcher.

```
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)
```

Примеры запросов

Абонента пускаем, ему доступны группы а и 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: */*
>
```

Абонента пускаем, групп нет:

```
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: */*
>
```

Абонента не пускаем:

```
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: */*
>
```

Абонент не найден:

```
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: */*
>
```

RADIUS

Для авторизации пользователей в вотчере может использоваться RADIUS сервер. Это может быть удобным при большом количестве пользователей.

Включается она вручную. Откройте консоль БД:

```
sqlite3 /opt/fluxsonic/priv/vsaas.db
```

или

```
psql -U vsaas -d vsaas_production
```

(в вашем случае параметры доступа к БД могут отличаться).

Добавьте опцию:

```
insert into config (grp,name,value) values ('global', 'radius', '127.0.0.1:1812/secret');
```

Строка '127.0.0.1:1812/secret' состоит из трёх частей: хост, порт и секрет. Измените её в соответствии с настройками вашего RADIUS сервера.

Теперь при логине пользователей вотчер будет обращаться по указанному адресу по протоколу RADIUS. В Access-Request запросе вотчер посылает логин и пароль:

User-Name

User-Password

В ответ RADIUS сервер в случае успешной авторизации может послать список групп пользователя. Для передачи используется атрибут Filter-Id(11). Каждая группа передаётся в отдельном атрибуте.

Вотчер обращается к RADIUS каждый раз при входе пользователя.

Если RADIUS отвечает Access-Accept, вотчер пускает пользователя, сохраняет в БД хэш пароля и принадлежность к группам.

Если RADIUS отвечает Access-Reject, пользователь блокируется в БД.

Если RADIUS не ответил, вотчер ищет пользователя в БД.

Необходимо учитывать, что RADIUS должен знать про всех пользователей, в том числе администраторов. Признак, является ли пользователь администратором, не может быть передан в ответе RADIUS и назначается только через вотчер.

API для мобильных приложений

API для разработки своих собственных мобильных приложений.

POST: /vsaas/api/login

Пример:

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

Параметры:

HTTP request payload

```
{"email": , "password": }
```

Ответ запроса: JSON вида

```
{  
  success: boolean,  
  session:  
}
```

Добавление камеры в Избранное

POST: /vsaas/my/fav/cameras

Параметры:

HTTP request payload

```
{"id": } // идентификатор камеры
```

Ответ запроса: JSON вида

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

Удаление камеры из Избранного

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

Параметры:

cam_id идентификатор камеры

Ответ запроса: JSON вида

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

Все доступные камеры

GET: /vsaas/api/my/cameras

Параметры:

HTTP request headers

x-vsaas-session: <session_token>

Список избранных камер

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

Параметры:

HTTP request headers

x-vsaas-session: <session_token>

Ответ запроса: JSON вида

```
{
  "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 // глубина архива в днях
  "permissions": { // доступные пользователю права
    "dvr": boolean // просмотр архива
    "ptz": boolean // управление PTZ
    "view": boolean // просмотр live
  }
  "ptz": boolean // доступно ли PTZ управление камеры
  "server": string // demo-watcher.flussonic.com
  "static": boolean // тип потока: статический или по запросу (on-demand)
  "status": { // информация о состоянии потока
    "source_error": string
    "alive": boolean
  }
  "thumbnails": boolean // включена ли генерация снэпшотов с камеры
  "urls": { // ссылки на различные протоколы получения видеопотока
    "hls": string
    "hds": string
    "jpg": string
    "rtmp": string
    "rtsp": string
    "media_info": string // URL информации о потоке
    "recording_status": string // URL состояния архива
  }
}
```

Управление камерами

Сразу после установки модуля в системе ещё нет камер, даже если они настроены на Flussonic Media Server:

Для добавления камеры в систему нажмите «Добавить» :

Название. Должно быть заполнено английскими буквами и цифрами, поскольку будет использоваться в URL-адресах.

URL. Нужно указать адрес камеры. Например: `rtsp://mycam.local/stream0`

Дополнительно. Дополнительный адрес. Например: `rtsp://mycam.local/stream1`;
`rtsp://mycam.local/stream2`

Дополнительно. Дополнительный адрес. Например: `rtsp://mycam.local/stream1`;
`rtsp://mycam.local/stream2`

URL скриншотов. Адрес снятия скриншотов с камеры.

Расположение. Расположение камеры. Также можно указать его на карте.

Адрес. Почтовый адрес камеры.

Владелец. Выбор владельца камеры. Владелец настраиваются в меню Пользователи и Группы. Владелецу камера будет доступна вне зависимости настроек доступа.

Доступ. Выбор доступа к камере: публичный, для клиентов и частный:

Публичный (public) — доступна всем пользователям Flussonic Watcher, вне зависимости от настроек групп и пользователей. При установке значения public источник появится в списке камер у всех пользователей вашего сервера. Также без ограничений по доступу камеру можно опубликовать на web-странице с постоянным доступом без авторизации.

Для клиентов (authorized) — также доступна для всех пользователей, но требуется авторизация на вашем Flussonic Watcher.

Частная (private) — доступна пользователю только при добавлении доступа к камере в разделе Пользователи и Группы.

Публичный (public) — доступна всем пользователям Flussonic Watcher, вне зависимости от настроек групп и пользователей. При установке значения public источник появится в списке камер у всех пользователей вашего сервера. Также без ограничений по доступу камеру можно опубликовать на web-странице с постоянным доступом без авторизации.

Для клиентов (authorized) — также доступна для всех пользователей, но требуется авторизация на вашем Flussonic Watcher.

Частная (private) — доступна пользователю только при добавлении доступа к камере в разделе Пользователи и Группы.

Сервер. Выбор сервера Flussonic Media Server.

Архив. глубина записи. Для того, что бы активировать эту опцию, надо настроить хранилище в настройках системы или настройках стримера;

Параметры.

Включение и отключение камеры.

Постоянно работает камера или по запросу.

Доступно изображение для предпросмотра или нет.

Ограничение доступа к архиву камеры.

Включение и отключение камеры.

Постоянно работает камера или по запросу.

Доступно изображение для предпросмотра или нет.

Ограничение доступа к архиву камеры.

Примечание. Текстовый комментарий. Например, это может быть описание того, куда камера смотрит.

Нажмите «Сохранить». После этого камера появится на странице управления и в дашборде:

Создание клиентской мозаики

Мозаика

— это возможность собрать на отдельной странице до 16 любых камер по выбору пользователя.

Создание мозаики

Чтобы создать мозаику, зайдите в Flussonic Watcher с правами администратора. Затем:

- 1) Перейдите в раздел «Камеры».
- 2) Нажмите кнопку «Мозаика».
- 3) Нажмите кнопку «Добавить».
- 4) В открывшемся окне необходимо:
 - Написать название мозаики.
 - Отметить чекбокс «включена».
 - Выбрать количество камер в мозаике.
 - Перетащить нужные камеры в сетку. Если камер много, то можно воспользоваться поиском.
- 5) Нажмите «Сохранить».

Отредактировать мозаику или удалить её можно в разделе «Камеры» > «Мозаики».

Чтобы временно отключить доступ пользователей к мозаике, уберите чекбокс «включена» в настройках.

Все созданные мозаики доступны в «Мои камеры» > «Мозаика».

Управление пользователями

После инсталляции модуля на странице управления пользователями будет только один пользователь-администратор:

Добавление пользователя

Для добавления нового пользователя необходимо ввести:

Логин

Пароль

Внутренний серый ip-адрес для автологина

E-mail

Чекбоксами отметить:

Активен. Отметить, чтобы аккаунт был активен.

Администратор. Отметить, если пользователю необходимы права администратора.

Доступ к архиву. Отметить, если пользователю необходим доступ к архиву записей.

Также для пользователя можно оставить комментарий.

Добавление камер

Чтобы добавить пользователю камеры, нажмите на тег «0 камер», затем «Добавить» и в открывшемся списке выделите те камеры, которые доступны пользователю.

Нажмите «Удалить», если хотите закрыть пользователю доступ к камере.

Кликнув на тег «0 групп», вы можете добавить пользователю группу камер. Подробнее в статье Группы.

После добавления камер они будут доступны пользователю в дашбоарде «Мои камеры».

Группы

С помощью групп вы сможете упростить навигацию по камерам и пользователям, делая логическую группировку (например, по этажам, отделам, регионам, микрорайонам и т.д.), а также облегчить процесс распределения прав видимости камер. Достаточно создать группу с камерами, а затем добавить в нее пользователя, которому в несколько кликов станут доступными все приватные камеры данной группы.

Чтобы создать группу камер и пользователей перейдите в меню «Пользователи и группы» > «Группы» > «Добавить».

Затем укажите название новой группы, описание (при необходимости) и нажмите «Сохранить». В списке групп вы увидите созданную группу.

Кликнув на тег «0 камер», вы сможете добавить в группу камеры из списка, а, кликнув на тег «0 пользователей» > «Добавить».

Тоже самое можно сделать, кликнув на группу > «Камеры» > «Добавить».

В списке добавляемых пользователей вы сможете поставить флаг на видимость архива (DVR) для каждого юзера, а также возможность управлять камерой PTZ (не для всех камер).

Важно! Чтобы дать доступ к камере ограниченному количеству пользователей с помощью групп, необходимо присвоить камере статус Private. Статус Public делает камеру публичной и ее видно всем пользователям, независимо от того, в группе она или нет.

Предоставление доступа к камерам

С помощью инструментов предоставления доступа к камерам у администратора есть возможность выдавать доступ к приватным камерам конкретным пользователям Flussonic Watcher, не создавая группу.

Для этого зайдите в раздел «Пользователи и группы» и возле конкретного пользователя нажмите на «0 расшаренных камер». Затем нажмите «Добавить».

В списке доступных камер отметьте чекбоксами нужные и нажмите «Добавить». После этого выбранный пользователь получит доступ к приватным камерам, не будучи их владельцем.

Тоже самое можно сделать через карточку пользователя. Нажмите в верхней части на «0 камер» и у вас появится возможность добавить одну или несколько камер, которые станут доступными для данного пользователя.

Предоставлять доступ к камерам могут только администраторы, у пользователей такой возможности пока нет.

Серийные номера

Для Flussonic Watcher с камерными агентами необходимо настроить автопривязку пользователя к проданной камере.

Для этого:

- 1) Заведите пользователя в разделе Пользователи и Группы.
- 2) Зайдите в раздел Камеры > Серийные номера.
- 3) Добавьте серийный номер с коробки камеры и укажите пользователя, которому она принадлежит.

При первом включении камеры произойдет ее активация, провижнинг на сервер, а далее привязка пользователя к камере по серийному номеру. Также применится предустановленное значение глубины архива.

Для пользователя весь процесс первого включения будет выглядеть так:

Включил камеру в сеть.

Подключил камеру к интернету.

Зашел в Flussonic Watcher по выданному логин/паролю.

Камера уже доступна и ее можно смотреть.

Импорт и Поиск камер

Импорт из Flussonic Media Server

В простом случае запуска Flussonic Watcher с камерами доступными по локальной сети, можно воспользоваться как поиском по Onvif, так и импортом потоков из Flussonic Media Server.

Можно импортировать в Flussonic Watcher камеру, уже добавленную в Flussonic Media Server.

Поиск камер

Flussonic Watcher может искать Onvif-совместимые камеры с помощью механизма WS-Discovery. Это протокол обнаружения сервисов, который работает в камерах Ubiquiti, Samsung и т.д.

Некоторые камеры не смогут определиться без логина/пароля, поэтому, если не получается найти нужную камеру, то надо ввести логин/пароль и ещё раз запустить поиск.

У многих камер доступны два и более различных потоков, вы сможете добавить любой из них (H264):

После выбора профиля камеры, в поле RTSP URL установится правильное значение. Для добавления камеры останется только ввести для нее имя.

Также необходимо настроить путь для DVR в настройках системы.