FindFace

NtechLab

Apr 24, 2024

CONTENTS

1 System Administrator's Guide **3**
- 1.1 Architecture . . . 3
- 1.2 System Requirements . . . 7
- 1.3 Requirements for CCTV Cameras . . . 8
- 1.4 Licensing Info . . . 11
- 1.5 Deploy FindFace . . . 12
- 1.6 Maintenance and Troubleshooting . . . 44
- 1.7 Appendices . . . 69

2 User's Guide **125**
- 2.1 First Steps after Deployment . . . 125
- 2.2 Work with FindFace . . . 129
- 2.3 Advanced Functionality . . . 197

3 Integrations **229**
- 3.1 HTTP API . . . 229
- 3.2 Webhooks . . . 230
- 3.3 Partner Integrations . . . 253
- 3.4 Custom Plugins . . . 262

Python Module Index **279**

Index **281**

FindFace is a multifunctional biometric system, based on FindFace Enterprise Server, a cutting-edge AI facial recognition technology. FindFace is a turnkey solution that you can harness in such areas as retail, banking, social networking, entertainment, sports, event management, dating services, video surveillance, public safety, homeland security, and others.

FindFace detects and identifies human faces, and notifies responsible officials about their appearance. It can also recognize such facial attributes as gender, age, emotions, glasses, face mask, beard, and many others, and display this information in a face recognition event.

The integrated 2D anti-spoofing system ensures that it is a live person in front of a camera and eliminates the possibility of fraud using images on paper or mobile device screens.

FindFace supports the integration of third-party solutions via *HTTP API*, *webhooks*, and *plugins*, so you can enhance your current system or application with face recognition functionality.

Core features

- AI-based platform.
- Fast and robust real-time biometric identification against dossier databases.
- Support for live video and archives, most video formats and codecs that FFmpeg can decode.
- Face verification.
- AI recognition of gender, age, emotions, glasses, beard, face mask, and other face attributes.
- AI face liveness detector.
- AI recognition of a person.
- Database search.
- Possibility of counting faces and silhouettes on connected cameras.
- Video surveillance.

Environment

- Developer-friendly installer and user-friendly interface.
- Single- and multi-host deployment.
- Increased performance and fault-tolerance in high load systems with numerous cameras and clients.
- Possibility of distributing dossier database among several hosts with synchronization and replication.
- Network or on-premise licensing.
- CPU- and GPU-based acceleration for your choice.
- Mobile app.

Security

- Advanced user management.
- Authentication based on a password, certificate, and face recognition for guaranteed system protection.
- Dossier security.
- Comprehensive, friendly, searchable audit logs.
- Backup and restore utilities.

Make the most of your system

- Social interaction analysis.
- Know your customer analytics.
- Detailed reports on face recognition events, episodes, search events, persons, counters, cameras, dossiers, and KYC analytics.
- Face liveness detector as a standalone service.

Useful little things

- Quick dossier database creation.
- Complete dossier customization.
- Deduplication support for events and dossiers.
- Personal data protection (GDPR and similar laws).
- Extended set of search filters.
- Scheduled database cleanup.

Integration

- Integration via HTTP API, webhooks, and python plugins.
- Integrations with favored vendors.

CHAPTER

ONE

SYSTEM ADMINISTRATOR'S GUIDE

This chapter is all about FindFace deployment and further updates and maintenance during exploitation.

1.1 Architecture

Though you mostly interact with FindFace through its web interface, be sure to take a minute to learn the FindFace architecture. This knowledge is essential for the FindFace deployment, integration, maintenance, and troubleshooting.

In this chapter:

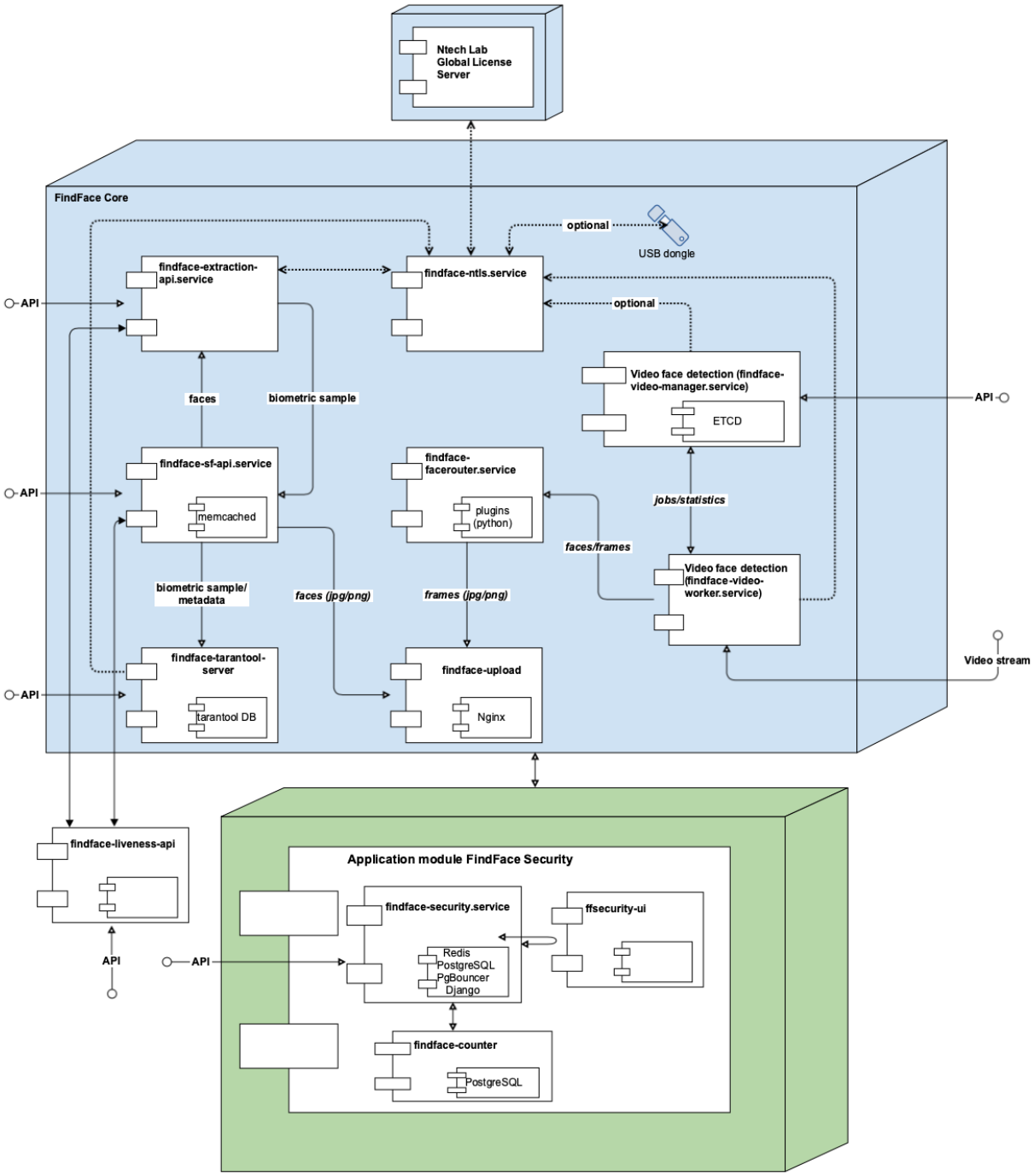
- *Architectural Elements*
  - *Architecture scheme*
  - *FindFace Core*
  - *FindFace Application Module*
- *Single- and Multi-Host Deployment*
- *CPU- and GPU-acceleration*

1.1.1 Architectural Elements

FindFace consists of the following fundamental architectural elements:

- FindFace core, a cutting-edge AI-based face recognition technology that can be used as a separate product FindFace Enterprise Server.
- FindFace, which is a turnkey application module for FindFace Enterprise Server.

Architecture scheme



FindFace Core

The FindFace core includes the following components:

Component	Ports in use	Description	Vendor
findface-extraction-api	18666	Service that uses neural networks to detect a face in an image and extract a face biometric sample (feature vector). It also performs recognition of face attributes such as gender, age, emotions, beard, glasses, face mask, and others, and silhouette recognition (if configured). CPU- or GPU-acceleration.	NtechLab own deployment
findface-sf-api	18411	Service that implements HTTP API for face detection and face recognition.	
findface-tarantool-server	32001, shard ports (default 330xx, 81xx)	Service that provides interaction between the `findface-sf-api` service and the biometric database (database that stores face biometric samples) powered by Tarantool.	
findface-upload	3333	NginX-based web server used as a storage for original images, thumbnails and normalized face images.	
findface-facerouter	18820	Service used to define processing directives for detected faces. In FindFace Security, its functions are performed by `findface-security` (see *FindFace Application Module*). If necessary, you can still deploy and enable this component for integration purposes (see *Custom Plugins*).	
findface-video-manager	18810, 18811	Service, part of the video face detection module, that is used for managing the video face detection functionality, configuring the video face detector settings and specifying the list of to-be-processed video streams.	
findface-video-worker	18999	Service, part of the video face detection module, that recognizes a face in the video and posts its normalized image, full frame and metadata (such as the camera ID and detection time) to the `findface-facerouter` service for further processing according to given directives. Provides *face liveness detection* if enabled. CPU- or GPU-acceleration.	
findface-ntls	443 (TCP), 3133, 3185	License server which interfaces with the NtechLab Global License Server or a USB dongle to verify the *license* of your FindFace instance.	
Tarantool	Shard ports (default 330xx, 81xx)	Third-party software which implements the biometric database that stores extracted biometric samples (feature vectors) and face identification events. The system data, dossiers, user accounts, and camera settings are stored in PostgreSQL (part of the FindFace application module).	Tarantool
etcd	2379	Third-party software that implements a distributed key-value store for `findface-video-manager`. Used as a coordination service in the distributed system, providing the video face detector with fault tolerance.	etcd
NginX	80; SSL: 8002, 8003, 443, 80	Third-party software which implements the system web interfaces.	nginx
memcached	11211	Third-party software which implements a distributed memory caching system. Used by `findface-extraction-api` as a temporary storage for extracted face biometric samples before they are written to the biometric database powered by Tarantool.	memcached

FindFace Application Module

The FindFace application module includes the following components:

Component	Ports in use	Description	Vendor
findface-security	Configurable	Component that serves as a gateway to the FindFace core. Provides interaction between the FindFace Core and the web interface, the system functioning as a whole, HTTP and web socket, biometric monitoring, event notifications, episodes, webhooks, and counters. Includes the following internal services: NTLS checker, Counter manager, Webhooks manager, Persons clusterizator, Event episodes manager, and Video archive queue manager. The last four can be enabled and disabled via the `/etc/findface-security/config.py` configuration file.	NtechLab own deployment
findface-security-ui	Configurable	Main web interface that is used to interact with FindFace. Allows you to work with face identification events, search for faces, manage cameras, users, dossiers, and watch lists, collect real-time statistics, and many more.	
findface-counter	18300	Service used for face deduplication.	
findface-liveness-api	18301	Besides the embedded functionality provided by `findface-video-worker`, face liveness detection can also be harnessed as a standalone service `findface-liveness-api`. The service takes a specific number of frames from a video fragment and returns the best quality face, and decimal liveness result averaged across the taken frames. The service is also used for authentication based on facial recognition. See *Liveness Detection as Standalone Service* and *Authentication and Session Monitoring*.	
PostgreSQL	5432	Third-party software which implements the main system database that stores detailed and categorized dossiers on particular persons, as well as data for internal use such as user accounts and camera settings. The face biometric data and face identification events are stored in Tarantool (part of the FindFace core).	PostgreSQL
Pgbouncer	5439	Third-party software, a lightweight connection pooler for PostgreSQL. Optional, used to increase the database performance under high load.	PgBouncer
Redis	6379	Third-party software which implements a message broker inside `findface-security`.	Redis
Django	5439	Third-party software which implements a web framework for the FindFace web interface.	Django

See also:

Components in Depth

1.1.2 Single- and Multi-Host Deployment

You can deploy FindFace on a single host or in a cluster environment. If you opt for the latter, we offer you one of the following deployment schemes:

- Deploy FindFace standalone and distribute additional `findface-video-worker` components across multiple hosts.
- Distribute the FindFace components across multiple hosts. If necessary, set up load balancing.

See *Guide to Typical Cluster Installation* for details.

1.1.3 CPU- and GPU-acceleration

The `findface-extraction-api` and `findface-video-worker` services can be either CPU- or GPU-based. During installation from the developer-friendly *installer*, you will have an opportunity to choose the acceleration type you need.

If you opt to install FindFace from the *repository package*, deploy the `findface-extraction-api` and `findface-video-worker-cpu` packages on a CPU-based server, and the `findface-extraction-api-gpu` and/or `findface-video-worker-gpu` packages on a GPU-based server.

Important: Refer to *System Requirements* when choosing hardware configuration.

Important: If the resolution of a camera(s) in use is more than 1280x720px, it is strongly recommended to use the GPU-accelerated package `findface-video-worker-gpu`.

Note: The *liveness detector* is much slower on CPU than on GPU.

1.2 System Requirements

To calculate the FindFace host(s) characteristics, use the requirements provided in this chapter.

Tip: Be sure to learn about the FindFace *architecture* first.

In this chapter:

- *Basic Configuration*
- *Required Administrator Skills*

1.2.1 Basic Configuration

Important: If the resolution of a camera(s) in use is more than 1280x720px, it is strongly recommended to use the GPU-accelerated package `findface-video-worker-gpu`.

Note: In the case of a high-load system (~> 15 events per second), we recommend using an SSD.

Note: You can also use an Intel-based VM if there is AVX2 support, and eight physical cores are allocated exclusively to the VM.

Tip: For more accurate hardware selection, contact our support team by support@ntechlab.com.

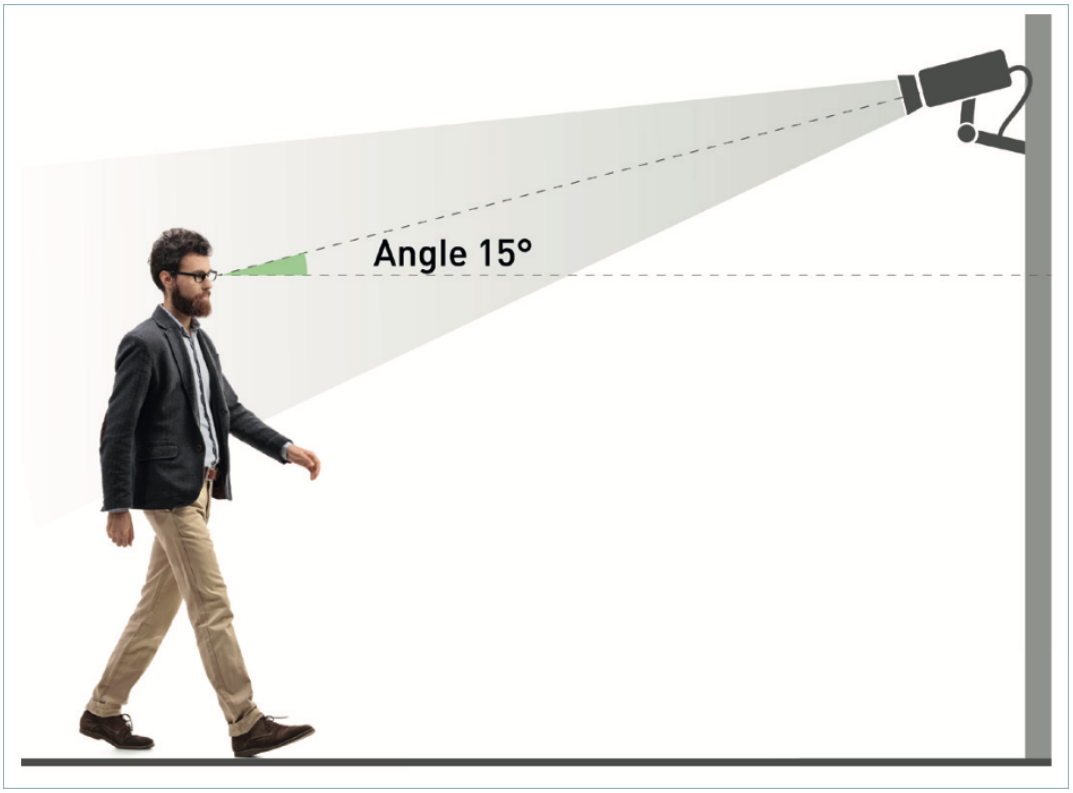
1.2.2 Required Administrator Skills

A FindFace administrator must know and understand OS Ubuntu at the level of an advanced user.

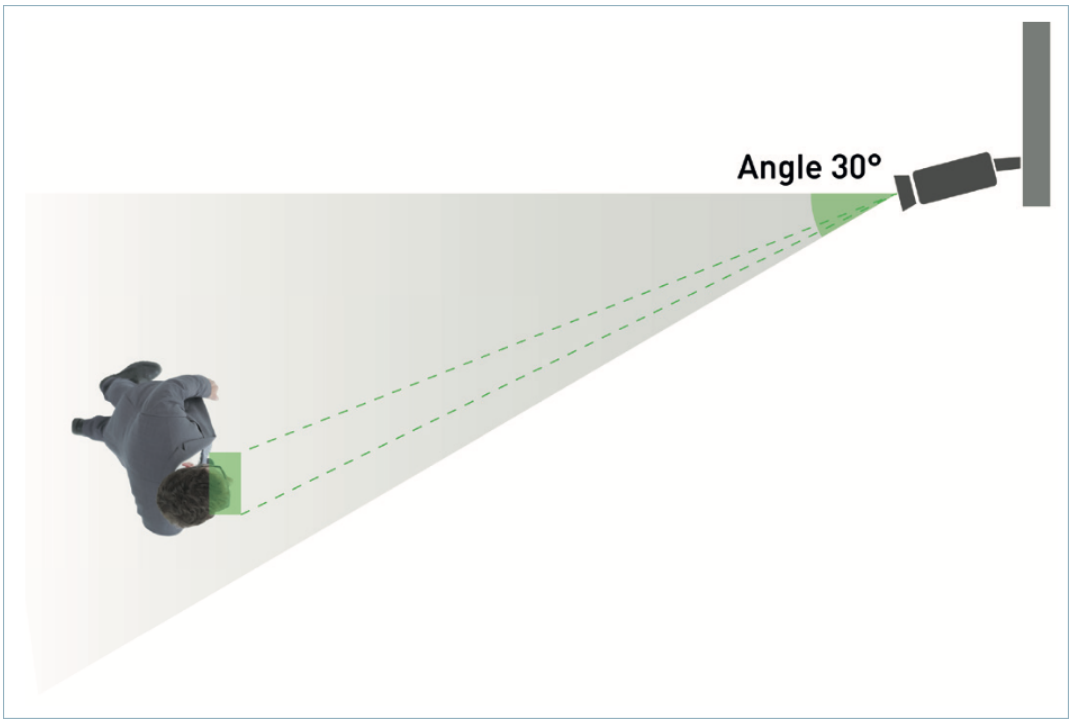
1.3 Requirements for CCTV Cameras

The primary requirements for installation and characteristics of CCTV cameras in your FindFace-based face recognition system are the following:

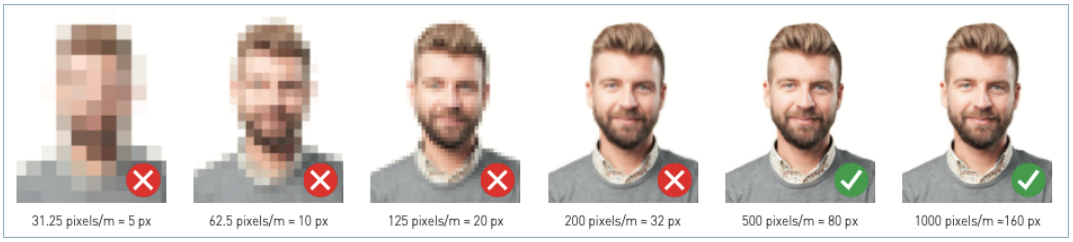
1. For correct face detection in a video stream, mount the camera so that the face of each individual entering the monitored area surely appears in the camera field of view.
2. The vertical tilt angle of the camera should not exceed 15°. The vertical tilt is a deviation of the camera's optical axis from the horizontal plane, positioned at the face center's level for an average height person (160 cm).



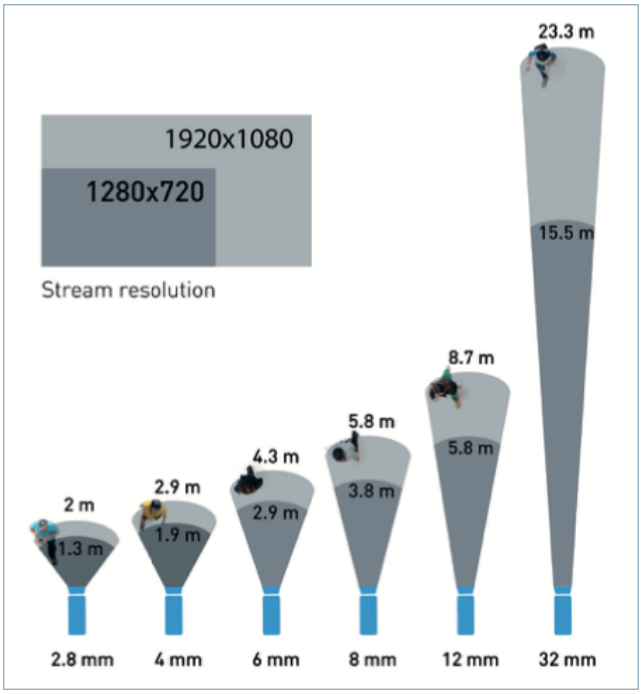
3. The horizontal deflection angle should not exceed 30°. The horizontal deflection is a deviation of the camera's optical axis from the motion vector of the main flow of objects subject to recognition.



4. The minimum pixel density required for identification is 500 pixels/m (roughly corresponds to a face width of 80 pixels).



5. Select such a focal length of the camera's lenses that provides the required pixel density at a predetermined distance to the recognition objects. The picture below demonstrates how to calculate the focal length subject to the distance between the camera and recognition objects. Estimating the focal length for a particular camera requires either calculators or a methodology provided by the camera manufacturer.



6. The exposure must be adjusted so that the face images are sharp ("in focus"), non-blurred, and evenly lit (not overlit or too dark).

7. For imperfect lighting conditions such as flare, too bright or too dim illumination, choose cameras with WDR hardware (Wide Dynamic Range) or other technologies that provide compensation for backlight and low illumination. Consider BLC, HLC, DNR, high optical sensitivity, Smart infrared backlight, AGC, and such.

8. Video compression: most video formats and codecs that FFmpeg can decode.
9. Video stream delivery protocols: RTSP, HTTP.

Tip: To calculate the precise hardware configuration tailored to your purposes, contact our experts by support@ntechlab.com.

1.4 Licensing Info

In this chapter:

- *Licensing Principles*
- *View and Update License*

1.4.1 Licensing Principles

The following criteria apply to FindFace licensing:

1. The number of extracted biometric samples. The biometric samples are extracted from faces detected in the video and from dossier photos, as well as when building so-called *person* centroids.

   The licensing scheme is the following:

   - Events: 1 event of video face detection = 1 face in a license.
   - Dossier: 1 photo in a dossier = 1 face in a license.
   - Persons: 1 person = 1 face in a license.
2. The number of cameras in use.
3. The number of model instances in use in the `findface-extraction-api` component.
4. Face features recognition: gender/age/emotions/glasses/beard/face mask.
5. Face liveness detection.
6. Integration with partners.

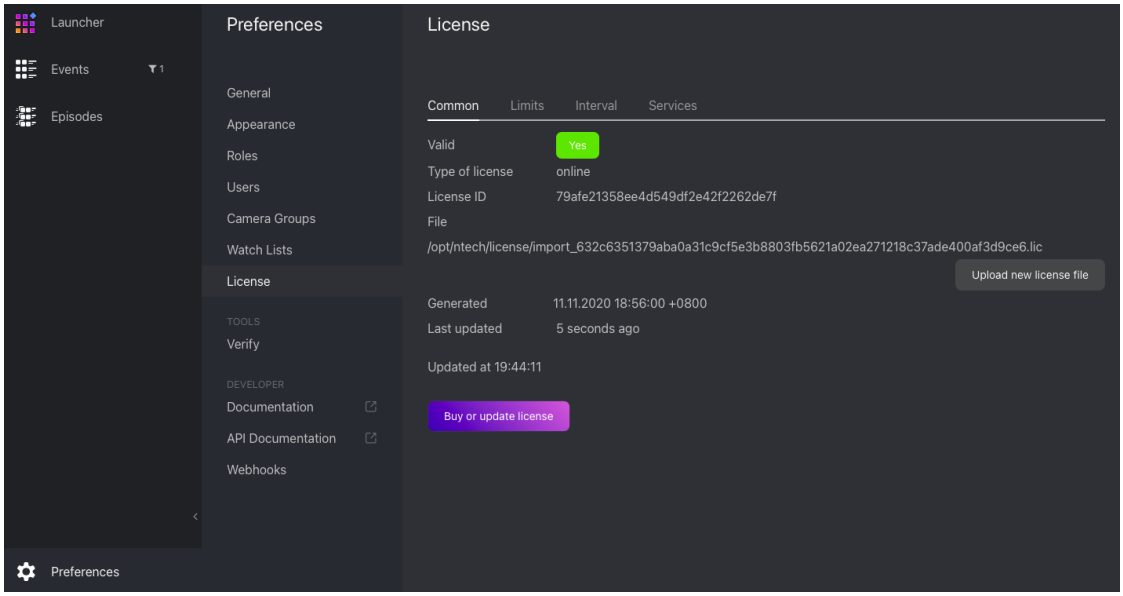
You can choose between the online and on-premise (aka offline) licensing:

- The online licensing is provided by interaction with the NtechLab Global License Manager `license.ntechlab.com` and requires a stable internet connection, DNS, and open port 443 TCP. Upon being disconnected from the internet, the system will continue working off-grid for 4 hours. It is possible to prolongate this period for up to 2 days (inform your manager if you need it).
- The on-premise (offline) licensing requires a USB port on the physical server with the `findface-ntls` component (license server in the *FindFace core*), that you will use to plug in a provided USB dongle.

For the system to function, a single instance of `findface-ntls` should be enough. If your system requires more license servers, contact your NtechLab manager beforehand to prevent your system from being blocked.

1.4.2 View and Update License

After installing FindFace, upload the license file you obtained from the manager into the system. To do so, navigate to *Preferences -> License*.



Use the same tab to consult current licensing information and upgrade your license.

See also:

Troubleshoot Licensing and findface-ntls

1.5 Deploy FindFace

FindFace provides the following deployment options:

- from a console installer
- step-by-step from an APT repository

Important: Starting services `findface-extraction-api-gpu` and `findface-video-worker-gpu` for the first time after deployment may take up a considerable amount of time, up to 45 minutes, due to the caching process.

Important: Although FindFace provides *tools* to ensure its protection from unauthorized access, they are not replacing a properly configured firewall. Be sure to use a firewall to heighten the FindFace network protection.

1.5.1 Deploy from Console Installer

To deploy FindFace, use a developer-friendly console installer.

Tip: Before deployment, be sure to consult the *system requirements*.

Important: The FindFace host must have a static IP address in order to be running successfully. To make the IP address static, open the `/etc/network/interfaces` file and modify the current primary network interface entry as shown in the case study below. Be sure to substitute the suggested addresses with the actual ones, subject to your network specification.

```
sudo vi /etc/network/interfaces

iface eth0 inet static
address 192.168.112.144
netmask 255.255.255.0
gateway 192.168.112.254
dns-nameservers 192.168.112.254
```

Restart networking.

```
sudo service networking restart
```

Be sure to edit the `etc/network/interfaces` file with extreme care. Please refer to the Ubuntu guide on networking before proceeding.

To deploy FindFace from the console installer, do the following:

1. Download the installer file `findface-security-and-server-4.5.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

   ```
   chmod +x findface-security-and-server-4.5.run
   ```

4. Execute the `.run` file.

   ```
   sudo ./findface-security-and-server-4.5.run
   ```

   The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. Fill out the prompts appropriately once requested. The questions are the following:

   1. Product to install: FindFace.
   2. Installation type:
      - `1`: install FindFace standalone.
      - `2`: install FindFace and configure it to interact with additional remote `findface-video-worker` instances.

      Tip: To install only `findface-video-worker` on a host, refer to *Additional findface-video-worker deployment on remote hosts*.

- 3: install only the apt repository that can be further used for the *step-by-step deployment*.

  Important: This installation type doesn't provide installation of neural network models essential for the `findface-extraction-api` functioning. Be sure to *manually install* them on the host(s) with `findface-extraction-api`.

- 4: fully customized installation.

  Important: Be sure to *manually install* neural network models on the host(s) with `findface-extraction-api`.

3. Type of `findface-video-worker` package: CPU or GPU.
4. Type of `findface-extraction-api` package: CPU or GPU.

Once all the questions answered, the answers will be saved to a file `/tmp/<findface-installer-*>.json`. You can edit this file and use it to install FindFace on other hosts without having to answer the questions again.

Should you choose to install FindFace standalone, its components will be automatically installed, configured and/or started in the following configuration:

Important: In the case of a clean install, the installer will automatically configure `findface-extraction-api` to use the `jackfruit_480` neural network. Otherwise, you will be able to choose between `jackfruit_480` and the previous model. It is strictly not recommended to use the installer to update the system.

After the installation is complete, the following output is shown on the console:

Tip: Be sure to save this data: you will need it later.

```
##########################################################################
#                        Installation is complete                        #
##########################################################################
- upload your license to http://172.20.77.17/#/license/
- user interface: http://172.20.77.17/
  superuser:      admin
  password:       admin
  documentation:  http://172.20.77.17/doc/
```

5. Upload the FindFace license file via the main web interface `http://<Host_IP_address>/#/license`. To access the web interface, use the provided `admin` credentials.

Note: The host IP address is shown in the links to FindFace web services in the following way: as an external IP address if the host belongs to a network, or `127.0.0.1` otherwise.

Important: Do not disclose the superuser (Super Administrator) credentials to others. To administer the system, create a new user with administrator privileges. Whatever the role, the Super Administrator cannot be deprived of its rights.

6. To automatically install FindFace on another host without answering the installation questions, use the `/tmp/<findface-installer-*>.json` file. Execute:

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

Tip: You can find an example of the installation file in *Installation File*.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

1.5.2 Deploy Step-by-Step from Repository

This section will guide you through the FindFace step-by-step deployment process. Follow the instructions below minding the sequence.

In this section:

- *Install APT Repository*
- *Prerequisites*
- *Provide Licensing*
- *Deploy Main Database*
- *Deploy FindFace Core*
- *Deploy FindFace Application Module and Biometric Database*

Install APT Repository

First of all, install the FindFace apt repository as follows:

1. Download the installer file `findface-security-and-server-4.5.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

```
chmod +x findface-security-and-server-4.5.run
```

4. Execute the `.run` file.

```
sudo ./findface-security-and-server-4.5.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. Fill out the prompts appropriately once requested. The questions are the following:

1. Product to install: `FindFace`.
2. Installation type: `repo: Don't install anything, just set up the APT repository`.
3. Neural network models to install if necessary. To select a model(s), deselect all those on the list by entering `-*` in the command line first, then select the required model by entering its sequence number (keyword): for example, `1 3`. Enter done to save your selection and proceed to another step.

   Important: At least one model for face biometry has to be installed.

After that, the FindFace apt repository will be automatically installed.

Prerequisites

FindFace requires such third-party software as PostgreSQL, Pgbouncer, Redis, etcd, and memcached. Do the following:

1. Install the prerequisite packages as such:

```
sudo apt update
sudo apt install -y postgresql-10 redis-server etcd memcached pgbouncer
```

2. Open the `/etc/memcached.conf` configuration file. Set the maximum memory to use for items in megabytes: `-m 1024`. Set the max item size: `-I 16m`. If one or both of these parameters are absent, simply add them in the file.

```
sudo vi /etc/memcached.conf

-m 1024
-I 16m
```

3. Give a strong password to the `ntech` user (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` in the example below). Output the credentials to the `pgbouncer` user list.

```
echo '"ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3"' | sudo tee -a /etc/pgbouncer/
↪userlist.txt
```

4. Configure `pgbouncer`. In `/etc/pgbouncer/pgbouncer.ini`, add the `ffsecurity` database to the `databases` section. Configure named parameters, as shown in the example below. Parameters other than those must be commented out.

```
sudo vi /etc/pgbouncer/pgbouncer.ini

[databases]
ffsecurity = dbname=ffsecurity host=localhost port=5432 user=ntech
[pgbouncer]
pidfile = /var/run/postgresql/pgbouncer.pid
listen_addr = 127.0.0.1
listen_port = 5439
unix_socket_dir = /var/run/postgresql
```

(continues on next page)

(continued from previous page)

```
auth_type = plain
auth_file = /etc/pgbouncer/userlist.txt
pool_mode = transaction
server_reset_query = DISCARD ALL
max_client_conn = 16384
default_pool_size = 20
syslog = 1
```

5. Enable the prerequisite services autostart and launch the services:

```
sudo systemctl enable postgresql@10-main.service redis-server etcd.service␣
↪memcached.service pgbouncer.service
sudo systemctl restart postgresql@10-main.service redis-server etcd.service␣
↪memcached.service pgbouncer.service
```

Provide Licensing

See also:

Licensing Info

You receive a license file from your NTechLab manager. If you opt for the on-premise licensing, we will also send you a USB dongle.

The FindFace licensing is provided as follows:

1. Deploy `findface-ntls`, license server in the FindFace core.

   Important: There must be only one `findface-ntls` instance in each `FindFace` installation.

   Tip: In the `/etc/findface-ntls.cfg` configuration file, you can change the license folder and specify your proxy server IP address if necessary. You can also change the `findface-ntls` web interface remote access settings. See *findface-ntls* for details.

```
sudo apt update
sudo apt install -y findface-ntls
sudo systemctl enable findface-ntls.service && sudo systemctl start findface-ntls.
↪service
```

2. Upload the license file via the `findface-ntls` web interface in one of the following ways:

   - Navigate to the `findface-ntls` web interface `http://<NTLS_IP_address>:3185/#/`. Upload the license file.

     Tip: Later on, use the FindFace main web interface to consult your license information, and upgrade or extend your license (*Settings -> License*).

   - Directly put the license file into the license folder (by default, `/opt/ntech/license`, can be changed in the `/etc/findface-ntls.cfg` configuration file).

3. For the on-premise licensing, insert the USB dongle into a USB port.

4. If the licensable components are installed on remote hosts, specify the IP address of the `findface-ntls` host in their configuration files. See *findface-extraction-api*, *findface-tarantool-server*, *Video face detection: findface-video-manager and findface-video-worker* for details.

See also:

Licensing Info

Deploy Main Database

In FindFace, the main system database is based on PostgreSQL. To deploy the main database, do the following:

1. Open the `pgbouncer` list of users `/etc/pgbouncer/userlist.txt`. Copy the `ntech` user's password (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` in the example below).

```
sudo cat /etc/pgbouncer/userlist.txt

"ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3"
```

2. Using the **PostgreSQL** console, create a new user `ntech` with the copied password, and databases `ffsecurity` and `ffcounter` in PostgreSQL.

```
sudo -u postgres psql

postgres=# CREATE ROLE ntech WITH LOGIN PASSWORD '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3';

postgres=# CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE=
↪'en_US.UTF-8' LC_CTYPE='en_US.UTF-8' TEMPLATE template0;

postgres=# CREATE DATABASE ffcounter WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE='C.
↪UTF-8' LC_CTYPE='C.UTF-8' TEMPLATE template0;
```

Tip: To quit from the **PostgreSQL** console, type `\q` and press `Enter`.

3. Allow authentication by UID of a socket client in **PostgreSQL**. Restart **PostgreSQL**.

```
echo 'local all ntech peer' | sudo tee -a /etc/postgresql/10/main/pg_hba.conf

sudo systemctl restart postgresql@10-main.service
```

Deploy FindFace Core

To deploy the FindFace core, do the following:

Tip: You can find the description of the FindFace core components and their configuration parameters in *Architecture* and *Components in Depth*.

1. For FindFace on GPU, *install NVIDIA drivers*.
2. Install the FindFace core components:

```
sudo apt update
sudo apt install -y findface-tarantool-server findface-extraction-api findface-sf-
↪api findface-upload findface-video-manager findface-video-worker-cpu findface-
↪liveness-api
```

Note: To install the GPU-accelerated `findface-extraction-api` component, use `findface-extraction-api-gpu` instead of `findface-extraction-api` in the command.

Note: To install the GPU-accelerated `findface-video-worker` component, use `findface-video-worker-gpu` instead of `findface-video-worker-cpu` in the command. If you have several video cards on your server, see *Multiple Video Cards Usage*.

Important: Be sure to *manually install* neural network models on the host(s) with `findface-extraction-api`.

3. In the `/etc/findface-sf-api.ini` configuration file, enable the `allow-return-facen` parameter.

```
sudo vi /etc/findface-sf-api.ini

...
limits:
  ...
  allow-return-facen: true
...
```

4. In the `/etc/findface-extraction-api.ini` configuration file, enable recognition models for face features such as gender, age, emotions, glasses, beard, and face mask, subject to your needs. Be sure to choose the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models. See *Face Features Recognition* for details.

```
sudo vi /etc/findface-extraction-api.ini

models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

The following models are available:

Face feature	Acceleration	Configuration file parameter
face (biometry)	CPU	`face: face/ifruit_320.cpu.fnk` `face: face/jackfruit_160.cpu.fnk` `face: face/jackfruit_320.cpu.fnk` `face: face/jackfruit_480.cpu.fnk`
	GPU	`face: face/ifruit_320.gpu.fnk` `face: face/jackfruit_160.gpu.fnk` `face: face/jackfruit_320.gpu.fnk` `face: face/jackfruit_480.gpu.fnk`
age	CPU	`age: faceattr/age.v1.cpu.fnk`
	GPU	`age: faceattr/age.v1.gpu.fnk`
gender	CPU	`gender: faceattr/gender.v2.cpu.fnk`
	GPU	`gender: faceattr/gender.v2.gpu.fnk`
emotions	CPU	`emotions: faceattr/emotions.v1.cpu.fnk`
	GPU	`emotions: faceattr/emotions.v1.gpu.fnk`
glasses	CPU	`glasses3: faceattr/glasses3.v0.cpu.fnk`
	GPU	`glasses3: faceattr/glasses3.v0.gpu.fnk`
beard	CPU	`beard: faceattr/beard.v0.cpu.fnk`
	GPU	`beard: faceattr/beard.v0.gpu.fnk`
face mask	CPU	`medmask3: faceattr/medmask3.v2.cpu.fnk`
	GPU	`medmask3: faceattr/medmask3.v2.gpu.fnk`

Tip: To disable a recognition model, simply pass an empty value to a relevant parameter. Do not remove the parameter itself as in this case the system will be searching for the default model.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

5. Open the `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) configuration file. In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list. In the `capacity` parameter, specify the maximum number of video streams to be processed by `findface-video-worker`. In the `streamer` section, specify the IP address and port to access the *video wall*. The streamer `port` must be set to `18999`.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini

mgr-static=127.0.0.1:18811

capacity=10

[streamer]
#-----------------------------
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999
## streamer url - how to access this worker on streamer_port
## type:string env:CFG_STREAMER_URL longopt:--streamer-url
url = 127.0.0.1:18999
```

6. Enable the FindFace core services autostart and launch the services.

```
sudo systemctl enable findface-extraction-api findface-sf-api findface-video-
↪manager findface-video-worker-cpu findface-liveness-api
sudo systemctl start findface-extraction-api findface-sf-api findface-video-manager␣
↪findface-video-worker-cpu findface-liveness-api
```

Deploy FindFace Application Module and Biometric Database

To deploy the FindFace application module, do the following:

1. Install the `findface-security`, `findface-security-ui`, and `findface-counter` components. Enable the `findface-counter` autostart and launch the service.

```
sudo apt update
sudo apt install -y findface-security findface-security-ui findface-counter
sudo systemctl enable findface-counter && sudo systemctl start findface-counter
```

2. Migrate the database architecture from FindFace to **PostgreSQL**, create user groups with *predefined* rights and the first user with administrator rights (a.k.a. Super Administrator).

Important: Super Administrator cannot be deprived of its rights, whatever the role.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

3. Create a structure of the Tarantool-based biometric database.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

4. Open the `/etc/tarantool/instances.available/FindFace.lua` configuration file. Check whether it contains the `dofile` command, `meta_indexes` and `meta_scheme` definitions, as in the example below.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua

dofile("/etc/findface-security/tnt_schema.lua")
-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    meta_indexes=meta_indexes,
    meta_scheme = meta_scheme
})
```

Important: The IP address and port number specified in the `shards` section of the `/etc/findface-sf-api.ini` configuration file must be identical to those in the `FindFace.start` section.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua
```

(continues on next page)

(continued from previous page)

```
...

FindFace.start("127.0.0.1", 8101...)
```

```
sudo vi /etc/findface-sf-api.ini

storage-api:
  ...
  shards:
  - master: http://127.0.0.1:8101/v2/
  ...
```

5. Enable the `findface-tarantool-server` service autostart and launch the service.

```
sudo systemctl enable tarantool@FindFace.service && sudo systemctl start␣
↪tarantool@FindFace.service
```

6. Open the `/etc/findface-security/config.py` configuration file. Specify the following parameters:

   - `SERVICE_EXTERNAL_ADDRESS`: FindFace IP address or URL prioritized for the Genetec integration and webhooks. Once this parameter not specified, the system uses `EXTERNAL_ADDRESS` for these purposes. To use Genetec and webhooks, be sure to specify at least one of those parameters: `SERVICE_EXTERNAL_ADDRESS`, `EXTERNAL_ADDRESS`.
   - `EXTERNAL_ADDRESS`: (Optional) IP address or URL that can be used to access the FindFace web interface. Once this parameter not specified, the system auto-detects it as the external IP address. To access FindFace, you can use both the auto-detected and specified IP addresses.
   - `VIDEO_DETECTOR_TOKEN`: to authorize the video face detection module, come up with a token and specify it here.
   - `VIDEO_MANAGER_ADDRESS`: IP address of the `findface-video-manager` host.
   - `NTLS_HTTP_URL`: IP address of the `findface-ntls` host.
   - `ROUTER_URL`: IP address of the `findface-security` host that will receive detected faces from the `findface-video-worker` instance(s). Specify either external or internal IP address, subject to the network through which `findface-video-worker` interacts with `findface-security`. Change the default port, subject to the *redirect settings* from HTTP to HTTPS, or omit it leaving only the IP address.
   - `SF_API_ADDRESS`: IP address of the `findface-sf-api` host.
   - `DATABASES` (section): fill it in as such: `'PORT': 5439, 'USER': 'ntech', 'PASSWORD': '<password from /etc/pgbouncer/userlist.txt>'` (see *Prerequisites*).

Tip: If necessary, ensure data security by enabling *SSL*.

Tip: If necessary, set `'IGNORE_UNMATCHED': True` to disable logging events for faces which have no match in the dossiers (negative verification result). Enable this option if the system has to process a large number of faces. The face similarity threshold for verification is defined by the `CONFIDENCE_THRESHOLD` parameter.

Tip: It is NOT recommended to change the `MINIMUM_DOSSIER_QUALITY` default value. This parameter deter-

mines the minimum quality of a face in a dossier photo. Photos containing faces of worse quality will be rejected when uploading to a dossier. Upright faces in frontal position are considered the best quality.

Important: If you enabled recognition models in the `/etc/findface-extraction-api.ini` configuration file, add the line `EVENTS_FEATURES` to the `FFSECURITY` section, subject to the list of enabled models. This line must be placed between `SF_API_ADDRESS` and `LIVENESS_THRESHOLD` as shown in the example below. See *Face Features Recognition* for details.

```
'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask'],
```

```
sudo vi /etc/findface-security/config.py

# ==============================================================================
# FindFace configuration file
# ==============================================================================
#
# This config file is written in Python's syntax and interpreted at FindFace
# service startup. You have to restart the service in order to apply changes.
#
# If you have any questions or suggestions, please contact us at support@ntechlab.
↪com


# ==============================================================================
# GENERAL SETTINGS
# ==============================================================================

# enables additional logs
DEBUG = False

# media files directory
MEDIA_ROOT = "/var/lib/findface-security/uploads"

# static files directory
STATIC_ROOT = "/var/lib/findface-security/static"

# language code
LANGUAGE_CODE = 'en-us'

# time zone
TIME_ZONE = 'UTC'

# Database is used by FindFace to store cameras,
# camera groups, watchlists and so on. Only PostgreSQL is supported.
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'DISABLE_SERVER_SIDE_CURSORS': True,
        'NAME': 'ffsecurity',
        'PORT': 5439, 'USER': 'ntech', 'PASSWORD': 'ZnUqeWKMzT6T2Bj2G4gbFn7cxRSMcxS7
↪'
```

(continues on next page)

(continued from previous page)

```
    }
}

# Signature key for session encryption
# Use pwgen -sncy 50 1|tr "" "." to generate your own unique key
SECRET_KEY = '8b26839acde2633362bdb176e741a650'

# ==============================================================================
# FindFace SETTINGS
# ==============================================================================

# SERVICE_EXTERNAL_ADDRESS is prioritized for FFSecurity webhooks and Genetec␣
↪plugin.
# EXTERNAL_ADDRESS is used instead if SERVICE_EXTERNAL_ADDRESS is not provided.
# You must provide either SERVICE_EXTERNAL_ADDRESS or EXTERNAL_ADDRESS in order
# to be able to work with FFSecurity webhooks and Genetec plugin.
SERVICE_EXTERNAL_ADDRESS = 'http://172.20.77.120'

# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via external␣
↪links.
EXTERNAL_ADDRESS = ''

# - Base FFSecurity settings -

# enable permissions system
ENABLE_ACL = True

FFSECURITY = {
    # findface-video-worker authorization token
    'VIDEO_DETECTOR_TOKEN': '3243a92b03c3411d4faa3cdd72f967b6',

    # base face matching confidence threshold
    'CONFIDENCE_THRESHOLD': 0.745,

    # episodes specific matching threshold that is used to join faces in an episode
    'EPISODES_THRESHOLD': 0.689,

    # minimum face quality sufficient to add it to a dossier
    'MINIMUM_DOSSIER_QUALITY': 0.45,

    # do not save unmatched events (GDPR support)
    'IGNORE_UNMATCHED': False,

    # blur all unmatched faces on the full frame of the matched event (GDPR support)
    'BLUR_UNMATCHED_FACES': False,

    # full frame jpeg quality when `BLUR_UNMATCHED_FACES` is enabled
    'BLURRED_FULLFRAME_JPEG_QUALITY': 85,

    # matched events older than EVENTS_MAX_MATCHED_AGE will be automatically
    # deleted (every night at 1:17 am by default)
    'EVENTS_MAX_MATCHED_AGE': 30,
```

(continues on next page)

(continued from previous page)

```
    # same as above but for unmatched events
    'EVENTS_MAX_UNMATCHED_AGE': 30,

    # same as EVENTS_MAX_MATCHED_AGE but for matched full frame images only␣
↪(thumbnails won't be deleted)
    'EVENTS_MAX_FULLFRAME_UNMATCHED_AGE': 30,

    # same as above but for unmatched full frame images only (thumbnails won't be␣
↪deleted)
    'EVENTS_MAX_FULLFRAME_MATCHED_AGE': 30,

    # same as above but for counter records
    'COUNTER_RECORDS_MAX_AGE': 30,

    # same as above but for person events (if no person events left in person, it␣
↪is deleted too)
    'PERSON_EVENTS_MAX_AGE': 90,

    # when closing episode, delete all events except the best episode event
    'EPISODE_KEEP_ONLY_BEST_EVENT': False,

    # NTLS licence server url
    'NTLS_HTTP_URL': 'http://127.0.0.1:3185',

    # findface-video-worker face posting address,
    # it must be set to either FFSecurity EXTERNAL_ADDRESS (by default)
    # or findface-facerouter url (in some specific cases)
    'ROUTER_URL': 'http://127.0.0.1',

    # send serialized dossiers, dossier-lists, camera and camera groups in webhooks
    'VERBOSE_WEBHOOKS': False,

    # jpeg quality used when saving thumbnails
    'THUMBNAIL_JPEG_QUALITY': 75,

    # FFServer services urls
    'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
    'SF_API_ADDRESS': 'http://127.0.0.1:18411',
    'FFCOUNTER_ADDRESS': 'http://127.0.0.1:18300',
    'LIVENESS_API_ADDRESS': 'http://127.0.0.1:18301',

    # upload video archives to this path, it differs from media root and
    # you have to change alias in corresponding nginx location also (/videos/)
    'VIDEO_ARCHIVE_UPLOAD_PATH': '/var/lib/findface-security/uploads/videos/',

    # additional events features.
    # make sure that corresponding extractors
    # are licensed and enabled at findface-extraction-api config file.
    # available features are: gender, age, emotions, beard, glasses, medmask.
    'EVENTS_FEATURES': [],
```

(continues on next page)

(continued from previous page)

```
    # feature specific confidence thresholds
    'LIVENESS_THRESHOLD': 0.85,
    'EMOTIONS_THRESHOLD': 0.25,
    'BEARD_THRESHOLD': 0.7,

    # counters full frame saving options:
    # `always` - save always
    # `detect` - save only if faces or silhouettes have been detected
    # `never` - never save full frames
    'COUNTERS_SAVE_FULLFRAME': 'always',
    'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
    'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,

    # max camera frames_dropped percent
    'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
    # max camera faces_failed percent
    'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},

    # -- Persons configuration --

    # rrule (recurrence rule) for scheduling persons clusterization
    # WARNING: all scheduling works with UTC time and NOT aware of any timezone
    'PERSONS_CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;
↪BYHOUR=0;BYMINUTE=0',

    # face to person matching confidence threshold
    'PERSONS_CONFIDENCE_THRESHOLD': 0.745,

    # minimum required face quality for person creation
    'PERSON_EVENT_MIN_QUALITY': 0.45,
    # minimum required number events in episode for person creation
    'PERSON_EVENT_MIN_EPISODE_EVENTS': 1,

    # maximum concurrent video manager jobs for video archives processing
    'MAX_VIDEO_ARCHIVE_JOBS': 3,

    # reports image saving options
    'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
    'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
    'REPORT_FULLFRAME_JPEG_QUALITY': 75,
    'REPORT_FULLFRAME_MAX_HEIGHT': 250,


    # -- Startup tests --

    # required services availability test
    'SERVICES_AVAILABILITY_TEST': True,

    # enable saving audit logs to PostgresSQL
    'ENABLE_AUDIT_LOGS': True,

    # -- FFSecurity Onvif --
```

(continues on next page)

(continued from previous page)

```
    # auth credentials for ffsecurity_onvif
    # ONVIF_CREDENTIALS = [
    #     {
    #         "hostnames": ["192.168.1.64", "2a00:1370:8117:ab87:a614:37ff:fe49:2683
↪"],
    #         "login": "admin",
    #         "password": "admin123"
    #     }
    # ],
    'ONVIF_CREDENTIALS': {},
    # list of all hostnames that will be ignored during Onvif service discovery
    # ONVIF_IGNORE_LIST = ["192.168.1.217"],
    'ONVIF_IGNORE_LIST': [],

    # -- Optional parameters --

    # Edit CUSTOM_FIELDS->dossier_meta section to customize dossier content.
    # Below is an example for integration FindFace with Sigur.

    # Edit CUSTOM_FIELDS->dossier_face section to customize dossier face content.
    # Below is an example with every field type possible.

    # 'CUSTOM_FIELDS': {
    #     'dossier_meta': {
    #         'items': [
    #             {
    #                 'name': 'personid',
    #                 'default': '',
    #                 'label': 'PersonID',
    #                 'display': ['list', 'form'],
    #                 'description': 'Sigur person ID'
    #             },
    #             {
    #                 'name': 'firstname',
    #                 'default': '',
    #                 'label': 'First Name',
    #                 'display': ['list', 'form'],
    #                 'description': 'Sigur first name'
    #             },
    #             {
    #                 'name': 'lastname',
    #                 'default': '',
    #                 'label': 'Last Name',
    #                 'display': ['list', 'form'],
    #                 'description': 'Sigur last name'
    #             },
    #             {
    #                 'name': 'version',
    #                 'default': '',
    #                 'label': 'Version',
    #                 'display': ['list', 'form'],
```

(continues on next page)

(continued from previous page)

```
    #                   'description': 'Sigur photo version'
    #               }
    #           ],
    #           'filters': [
    #               {
    #                   'name': 'personid',
    #                   'label': 'Sigur person ID filter',
    #                   'field': 'personid'
    #               }
    #           ]
    #       },
    #       'dossier_face': {
    #           'items': [
    #               {
    #                   "field_name": "tag_name_1",
    #                   "type": "string",
    #                   "default": "change_me"
    #               },
    #               {
    #                   "field_name": "tag_name_2",
    #                   "type": "uint",
    #                   "default": 123
    #               },
    #               {
    #                   "field_name": "tag_name_3",
    #                   "type": "bool",
    #                   "default": True
    #               },
    #           ]
    #       }
    # },

    # maximum event age in seconds than could be added to an episode.
    # 'EPISODE_SEARCH_INTERVAL': 60,
    # If none of these events matched, new episode is created.

    # maximum episode duration (episode is closed after)
    # 'EPISODE_MAX_DURATION': 300,

    # if no new event added to an episode during this timeout, episode will be␣
↪closed.
    # 'EPISODE_EVENT_TIMEOUT': 30,

    # maximum created thumbnail width
    # 'THUMBNAIL_MAX_WIDTH': 320,

    # url of the backend which is used for social network search.
    # contact support for additional information.
    # 'SOCIAL_BACKEND': None,

    # additional social backend headers.
    # 'SOCIAL_HEADERS': {},
```

(continues on next page)

(continued from previous page)

```
    # unacknowledged events notification interval
    # 'UNACKNOWLEDGED_NOTIFY_INTERVAL': 1,

    # set to True to run all media requests (photos, attachments) through the
    # django application for acl checks.
    # enabling this setting slightly increases security but
    # has severe negative effects on performance.
    # you will also have to mark /uploads/ location as 'internal' in nginx config
    #
    # 'OVERPROTECT_MEDIA': False,
}

# - FindFace authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
    # available options: face, password, face_and_password, face_or_password
    'AUTH_TYPE': 'face_or_password',
    'FACE_AUTH_CONFIDENCE': 0.745,
    # 180 days by default
    'MAXIMUM_SESSION_LENGTH': 15552000,
    # session renew works only with face or face_or_password authorization type
    'NEED_SESSION_RENEW': False,
    'RENEW_SESSION_INTERVAL': 0,
    'MAXIMUM_RENEW_ATTEMPTS': 2,
}

# - FindFace user interface configuration dictionary -

FFSECURITY_UI_CONFIG = {
    "event": {
        "features": {
            "f_gender_class": ["male", "female"],
            "age": {
                "f_age_gte": "",
                "f_age_lte": ""
            },
            "f_emotions_class": ["angry", "disgust", "fear", "happy", "sad",
↪"surprise", "neutral"],
            "f_glasses_class": ["none", "eye", "sun"],
            "f_beard_class": ["none", "beard"],
            "f_liveness_class": ["real", "fake"],
            "f_medmask_class": ["none", "incorrect", "correct"],
        }
    },

    # Adjustable confidence threshold presets for face matching.
    # Please consult with our support before changing.
    "confidence_display": [
        {"confidence": 0.00, "color": "#000000", "label": {"ru": " ", "en": "Very␣
↪Low"}},
        {"confidence": 0.65, "color": "#FF0300", "label": {"ru": "", "en": "Low"}},
```

(continues on next page)

(continued from previous page)

```
        {"confidence": 0.70, "color": "#FFB700", "label": {"ru": "", "en":
↪"Slightly Low"}},
        {"confidence": 0.72, "color": "#B8FA00", "label": {"ru": "", "en": "Moderate
↪"}},
        {"confidence": 0.75, "color": "#7EFF2B", "label": {"ru": "", "en":
↪"Slightly High"}},
        {"confidence": 0.80, "color": "#4DFF60", "label": {"ru": "", "en": "High"}},
        {"confidence": 0.85, "color": "#1DFF96", "label": {"ru": " ", "en": "Very␣
↪High"}},
    ]
}

# -- ASGI-server configuration --
# consult support before changing these settings.

# per worker thread pool size.
ASGI_THREADS = 32

UVICORN_SETTINGS = {
    # worker processes count, 'auto' sets it to logical cpu count
    'workers': 'auto',
    'host': 'localhost',
    'port': 8002,
    # websocket worker processes count,
    # 'auto' sets it to logical cpu count, but not more than 8.
    'ws-workers': 'auto',
    'ws-host': 'localhost',
    'ws-port': 8003,
}

# disable unused services to increase
# overall system performance in some cases.
SERVICES = {
    "ffsecurity": {
        "episodes": True,
        "webhooks": True,
        # use queue manager to prevent drops of video archive events
        "video_archive_events_manager": True,
        "persons": False,
    }
}

# -- Other settings --

# The number of threads in the night clusterization.
# Not recommended values are greater than the number of cores in the processor.
# Consult with support before changing this value.
NUMPY_OMP_NUM_THREADS = 'auto'


# ==============================================================================
# FindFace PLUGINS
```

(continues on next page)

(continued from previous page)

```
# ==============================================================================
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.

# =============== Axxon ================
# INSTALLED_APPS.append('ffsecurity_axxon')

# AXXON = [
#     {
#         'name': 'server_name',
#         'api': 'http://example.com/',
#         'rtsp': 'rtsp://example.com:554/',
#         'user': 'user',
#         'password': 'password',
#     }
# ]

# FFSECURITY_UI_CONFIG['dossier'] = {
#    'video': True,
# }


# =============== Genetec ================
# INSTALLED_APPS.append('ffsecurity_genetec')


# ================ Sova ==================
# INSTALLED_APPS.append('ffsecurity_sova')


# ================ Sigur =================
# keep in mind, that SIGUR plugin also uses CUSTOM_FIELDS and THUMBNAIL_MAX_WIDTH␣
↪settings
# INSTALLED_APPS.append('ffsecurity_sigur')
# SIGUR = {
#     'LOGIN': 'admin',
#     'PASSWORD': 'admin',
#     'MF_SELECTOR': 'biggest',  # what to do with several faces in sigur person photo;
↪ allowed ['biggest', 'reject']
#     'ONLY_RT_EVENTS': True,  # only events with bs_type == realtime,
#     'EVENT_DELAY': 0.004 # minimum time between two events of same person in␣
↪seconds. If interval between two events with same person is less, than this value,
↪ second event will be dropped
# }


# ======= CryptoPRO authentication =======
# INSTALLED_APPS.append('ffsecurity_cproauth')
# REST_FRAMEWORK['DEFAULT_AUTHENTICATION_CLASSES'] = [
#     'ffsecurity.auth.TokenAuthentication',
#     'ffsecurity_cproauth.auth.CryptoProOrTokenAuthentication'
# ]
```

(continues on next page)

(continued from previous page)

```
# ========== DossierLists sync ===========
# INSTALLED_APPS.append('ffsecurity_sync')

# token must be identical on master and slave
# use pwgen -s 64 1
# SYNC_TOKEN = 'change_me'
# rrule that defines sync schedule
# SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=4;BYMINUTE=0'
# if True synchronization will occur on FindFace startup and restart
# SYNC_AT_STARTUP = False
# if True synchronization will occur immediately after creating synchronization for␣
↪dossier list
# SYNC_AT_CREATION = False


# ============== Puppeteer ===============
# INSTALLED_APPS.append('ffsecurity_puppeteer')

# PUPPETEER_CONFIG = {
#     'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600,        # maximum lifetime of search␣
↪results not saved involuntarily
#     'REMOTE_MONITORING_SYNC_INTERVAL': 600,          # monitoring data␣
↪synchronization interval, seconds
#     'ENABLE_DAILY_SEARCH': True,                     # daily search activation␣
↪(default False)
#     'DAILY_SEARCH_PUSH_HOUR': 2,                     # daily search dossiers␣
↪synchronization hour
#     'DAILY_SEARCH_PULL_HOUR': 6,                     # hour in which results of␣
↪daily search will be obtained
#     'puppets': [
#         {
#             'id': 'first_puppet',                    # puppet ID
#             'url': 'http://1.1.1.1:8010/',           # puppet URL
#             'token': 'first_puppet_token',           # use pwgen -s 64 1 (should␣
↪match the token in puppet)
#             'facen_model': 'jackfruit_480'           # face model in puppet
#         },
#         {
#             'id': 'second_puppet',
#             'url': 'http://1.1.1.1:8010/',
#             'token': 'second_puppet_token',
#
#              # if remote installation has a different face model than the one␣
↪used in FFSecurity -
#              # you need to specify its name and ExtractionAPI URL where the␣
↪corresponding face model is specified
#             'facen_model': 'grapefruit_480',
#             'extractor': 'http://127.0.0.1:18667',
#         },
#     ]
```

(continues on next page)

(continued from previous page)

```
# }
#
# ================= Vns ==================
# A plugin for using FindFace as a puppeteer server
# INSTALLED_APPS.append('ffsecurity_vns')

# VNS_CONFIG = {
#     'USERS': {
#         'user1': 'token1',
#         'user2': 'token2'
#     },
#     'MONITORING_THRESHOLD': 0.75,
#     'DAILY': {
#         'ENABLED': False,
#         'THRESHOLD': 0.75,
#         'START_TIME': "00:00:00"
#     }
# }
```

7. Generate a signature key for the session encryption (used by Django) by executing the command below. Specify this key as `SECRET_KEY`.

```
pwgen -sncy 50 1|tr "'" "."
```

8. Start the services.

```
sudo systemctl enable findface-security
sudo systemctl start findface-security
```

9. Disable the default nginx server and add the `findface-security` server to the list of enabled servers. Restart nginx.

```
sudo rm /etc/nginx/sites-enabled/default

sudo ln -s /etc/nginx/sites-available/ffsecurity-nginx.conf /etc/nginx/sites-
↪enabled/

sudo nginx -s reload
```

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

1.5.3 Additional `findface-video-worker` deployment on remote hosts

To install only the `findface-video-worker` service, do the following:

Tip: Before deployment, be sure to consult the *system requirements*.

Tip: If you have several video cards on your server, see *Multiple Video Cards Usage* before deploying `findface-video-worker-gpu`.

1. On the FindFace server, open the `/etc/findface-security/config.py` configuration file and make sure that the `ROUTER_URL` parameter contains the external IP address of the FindFace server and not the localhost. The `findface-video-worker` instances on the remote hosts will be using this address for posting faces.

```
sudo vi /etc/findface-security/config.py

...

'ROUTER_URL': 'http://192.168.0.12',

...
```

2. Download the installer file `findface-security-and-server-4.5.run`.
3. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
4. From this directory, make the `.run` file executable.

```
chmod +x findface-security-and-server-4.5.run
```

5. Execute the `.run` file.

```
sudo ./findface-security-and-server-4.5.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. Fill out the prompts appropriately once requested. The questions are the following:

1. Product to install: FindFace Video Worker.
2. Type of `findface-video-worker` package: CPU or GPU.
3. IP address of the `findface-security` host.

After that, the installation process will automatically begin.

Note: The answers will be saved to a file `/tmp/<findface-installer-*>.json`. You can edit this file and use it to install FindFace on other hosts without having to answer the questions again.

Note: If you chose to install `findface-ntls` and/or `findface-video-manager` on different hosts than that with `findface-security`, specify their IP addresses in the `/etc/findface-video-worker-cpu.ini` (/etc/findface-video-worker-gpu.ini) configuration file after the installation.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

In the `ntls-addr` parameter, specify the `findface-ntls` host IP address.

```
ntls-addr=127.0.0.1:3133
```

In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list.

```
mgr-static=127.0.0.1:18811
```

Tip: To automatically install `findface-video-worker` on another host without answering the installation questions, use the `/tmp/<findface-installer-*>.json` file. Execute:

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

You can find an example of the installation file in *Installation File*.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

1.5.4 Neural Network Models Installation

To detect and identify faces and face features (gender, age, emotions, beard, glasses, face mask, and others), `findface-extraction-api` uses neural networks.

If you have to manually initiate the models installation, use the console installer as follows:

1. Execute the prepared `findface-security-and-server-4.5.run` file.

   ```
   sudo ./findface-security-and-server-4.5.run
   ```

2. Select the installation type: `Fully customized installation`.
3. Select a FindFace component to install: `findface-data`. To do so, first deselect all the listed components by entering `-*` in the command line, then select the required component by entering its sequence number (keyword): `1`. Enter `done` to save your selection and proceed to another step.
4. In the same manner, select models to install. After that, the installation process will automatically begin.

Note: You can find installed face recognition models at `/usr/share/findface-data/models/face/`, face features recognition models at `/usr/share/findface-data/models/faceattr/`.

```
ls /usr/share/findface-data/models/face/
ifruit_320.cpu.fnk  ifruit_320.gpu.fnk  jackfruit_160.cpu.fnk  jackfruit_160.gpu.fnk ␣
↪jackfruit_320.cpu.fnk  jackfruit_320.gpu.fnk jackfruit_480.cpu.fnk  jackfruit_480.gpu.
↪fnk

ls /usr/share/findface-data/models/faceattr/
age.v1.cpu.fnk  beard.v0.cpu.fnk  emotions.v1.cpu.fnk  gender.v2.cpu.fnk  glasses3.v0.
↪cpu.fnk  liveness.alleyn.v2.cpu.fnk  medmask3.v2.cpu.fnk  quality.v1.cpu.fnk
age.v1.gpu.fnk  beard.v0.gpu.fnk  emotions.v1.gpu.fnk  gender.v2.gpu.fnk  glasses3.v0.
↪gpu.fnk  liveness.alleyn.v2.gpu.fnk  medmask3.v2.gpu.fnk  quality.v1.gpu.fnk
```

1.5.5 Fully Customized Installation

The FindFace developer-friendly installer provides you with quite a few installation options, including the fully customized installation. This option is mostly used when deploying FindFace in a highly distributed environment.

To initiate the fully customized installation, answer the installer questions as follows:

- Product to install: `FindFace`.
- Installation type: `Fully customized installation`.
- FindFace components to install: whenever you have to make a selection, first deselect all the listed components by entering `-*` in the command line, then select required components by entering their sequence number (keyword), for example: `1 7` (`findface-data`, `findface-extraction-api`), `13` (`findface-tarantool-server`), or `9` (findface-upload). Enter `done` to save your selection and proceed to another step.
- Related questions such as about the acceleration type: CPU or GPU.

1.5.6 Guide to Typical Cluster Installation

This section is all about deploying FindFace in a cluster environment.

Tip: If after having read this section, you still have questions, do not hesitate to contact our experts by support@ntechlab.com.

The reasons for deploying FindFace in a cluster are the following:

- Necessity to distribute the video processing high load.
- Necessity to process video streams from a group of cameras in the place of their physical location.

  Note: The most common use cases where such need comes to the fore are hotel chains, chain stores, several security checkpoints in the same building, etc.

  See also:

  Allocate findface-video-worker to Camera Group

- Necessity to distribute the biometric sample extraction high load.
- Large number of faces to search through, that requires implementation of a distributed face database.

Before you start the deployment, outline your system architecture, depending on its load and allotted resources (see *System Requirements*). The most common distributed scheme is as follows:

- One principal server with the following components: `findface-ntls`, `findface-security`, `findface-sf-api`, `findface-video-manager`, `findface-upload`, `findface-video-worker`, `findface-extraction-api`, `findface-tarantool-server`, and third-parties.
- Several additional video processing servers with installed `findface-video-worker`.
- (If needed) Several additional biometric servers with installed `findface-extraction-api`.
- (If needed) Additional database servers with multiple Tarantool shards.

This section describes the most common distributed deployment. In high load systems, it may also be necessary to distribute the API processing (`findface-sf-api` and `findface-video-manager`) across several additional servers. In this case, refer to *Fully Customized Installation*.

To deploy FindFace in a cluster environment, follow the steps below:

- *Deploy Principal Server*
- *Deploy Video Processing Servers*
- *Deploy Biometric Servers*
- *Distribute Load across Biometric Servers*
- *Distribute Database*
- *Configure Network*

Deploy Principal Server

To deploy the principal server as part of a distributed architecture, do the following:

1. On the designated physical server, *install* FindFace from installer as follows:
   - Product to install: `FindFace`.
   - Installation type: `Single server, multiple video workers`. In this case, FindFace will be installed and configured to interact with additional remote `findface-video-worker` instances.
   - Type of the `findface-video-worker` acceleration (on the principal server): CPU or GPU, subject to your hardware configuration.
   - Type of the `findface-extraction-api` acceleration (on the principal server): CPU or GPU, subject to your hardware configuration.

   After the installation is complete, the following output will be shown on the console:

```
##########################################################################
#                        Installation is complete                        #
##########################################################################
- upload your license to http://172.20.77.17/#/license/
- user interface: http://172.20.77.17/
  superuser:      admin
  password:       admin
  documentation:  http://172.20.77.17/doc/
```

2. Upload the FindFace license file via the main web interface `http://<Host_IP_address>/#/license`. To access the web interface, use the provided `admin` credentials.

   Note: The host IP address is shown in the links to FindFace web services in the following way: as an external IP address if the host belongs to a network, or `127.0.0.1` otherwise.

   Important: Do not disclose the superuser (Super Administrator) credentials to others. To administer the system, create a new user with the administrator privileges. Whatever the role, Super Administrator cannot be deprived of its rights.

3. Allow the licensable services to access the `findface-ntls` license server from any IP address, To do so, open the `/etc/findface-ntls.cfg` configuration file and set `listen = 0.0.0.0:3133`.

```
sudo vi /etc/findface-ntls.cfg

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.1:3133
```

Deploy Video Processing Servers

On an additional video processing server, install only a `findface-video-worker` instance following the *step-by-step instructions*. Answer the installer questions as follows:

- Product to install: FindFace Video Worker.
- Type of the `findface-video-worker` acceleration: CPU or GPU, subject to your hardware configuration.
- FindFace IP address: IP address of the principal server.

After that, the installation process will automatically begin. The answers will be saved to a file `/tmp/<findface-installer-*>.json`. Use this file to install `FindFace Video Worker` on other hosts without having to answer the questions again, by executing:

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

Note: If `findface-ntls` and/or `findface-video-manager` are installed on a different host than that with `findface-security`, specify their IP addresses in the `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`) configuration file after the installation.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

In the `ntls-addr` parameter, specify the `findface-ntls` host IP address.

```
ntls-addr=127.0.0.1:3133
```

In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list.

```
mgr-static=127.0.0.1:18811
```

Deploy Biometric Servers

On an additional biometric server, install only a `findface-extraction-api` instance from the console installer. Answer the installer questions as follows:

- Product to install: `FindFace`.
- Installation type: `Fully customized installation`.
- FindFace components to install: `findface-extraction-api` and `findface-data`. To make a selection, first deselect all the listed components by entering `-*` in the command line, then select `findface-extraction-api` and `findface-data` by entering their sequence number (keyword): `1 7`. Enter `done` to save your selection and proceed to another step.
- Type of `findface-extraction-api` acceleration: CPU or GPU.
- Modification of the `/etc/findface-extraction-api.ini` configuration file: specify the IP address of the `findface-ntls` server.
- Neural network models to install: CPU or GPU model for face biometrics (mandatory), and (optional) CPU/GPU models for gender, age, emotions, glasses, beard, and face mask recognition. To make a selection, first deselect all the listed models by entering `-*` in the command line, then select required models by entering their sequence number (keyword), for example, `8 2` to select the GPU-models for biometric sample extraction and age recognition. Enter `done` to save your selection and proceed to another step. Be sure to choose the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models. See *Face Features Recognition* for details.

  The following models are available:

Face feature	Acceleration	Package
face (biometry)	CPU	findface-data-ifruit-320-cpu_3.0.0_all.deb findface-data-jackfruit-160-cpu_3.0.0_all.deb findface-data-jackfruit-320-cpu_3.0.0_all.deb findface-data-jackfruit-480-cpu_3.0.0_all.deb
	GPU	findface-data-ifruit-320-gpu_3.0.0_all.deb findface-data-jackfruit-160-gpu_3.0.0_all.deb findface-data-jackfruit-320-gpu_3.0.0_all.deb findface-data-jackfruit-480-gpu_3.0.0_all.deb
age	CPU	findface-data-age.v1-cpu_3.0.0_all.deb
	GPU	findface-data-age.v1-gpu_3.0.0_all.deb
gender	CPU	findface-data-gender.v2-cpu_3.0.0_all.deb
	GPU	findface-data-gender.v2-gpu_3.0.0_all.deb
emotions	CPU	findface-data-emotions.v1-cpu_3.0.0_all.deb
	GPU	findface-data-emotions.v1-gpu_3.0.0_all.deb
glasses3	CPU	findface-data-glasses3.v0-cpu_3.0.0_all.deb
	GPU	findface-data-glasses3.v0-gpu_3.0.0_all.deb
beard	CPU	findface-data-beard.v0-cpu_3.0.0_all.deb
	GPU	findface-data-beard.v0-gpu_3.0.0_all.deb
face mask	CPU	findface-data-medmask3.v2-cpu_3.0.0_all.deb
	GPU	findface-data-medmask3.v2-gpu_3.0.0_all.deb

After that, the installation process will automatically begin. The answers will be saved to a file `/tmp/<findface-installer-*>.json`. Use this file to install `findface-extraction-api` on other hosts without having to answer the questions again.

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

After all the biometric servers are deployed, distribute load across them by using a *load balancer*.

Distribute Load across Biometric Servers

To distribute load across several biometric servers, you need to set up load balancing. The following step-by-step instructions demonstrate how to set up `nginx` load balancing in a round-robin fashion for 3 `findface-extraction-api` instances located on different physical hosts: one on the FindFace principal server (`172.168.1.9`), and 2 on additional remote servers (`172.168.1.10`, `172.168.1.11`). Should you have more biometric servers in your system, load-balance them by analogy.

Tip: You can use any load balancer according to your preference. Please refer to the relevant official documentation for guidance.

To set up load balancing, do the following:

1. Designate the FindFace principal server (recommended) or any other server with nginx as a gateway to all the biometric servers.

   Important: You will have to specify the gateway server IP address when configuring the FindFace *network*.

   Tip: You can install nginx as such:

   ```
   sudo apt update
   sudo apt install nginx
   ```

2. On the gateway server, create a new nginx configuration file.

   ```
   sudo vi /etc/nginx/sites-available/extapi
   ```

3. Insert the following entry into the just created configuration file. In the `upstream` directive (`upstream extapibackends`), substitute the exemplary IP addresses with the actual IP addresses of the biometric servers. In the `server` directive, specify the gateway server listening port as `listen`. You will have to enter this port when configuring the FindFace *network*.

   ```
   upstream extapibackends {
           server 172.168.1.9:18666; ## ``findface-extraction-api`` on principal server
           server 172.168.1.10:18666; ## 1st additional extraction server
           server 127.168.1.11:18666; ## 2nd additional extraction server
   }
   server {
           listen 18667;
           server_name extapi;
           client_max_body_size 64m;
           location / {
                   proxy_pass http://extapibackends;
                   proxy_next_upstream error;
           }
   ```

(continues on next page)

(continued from previous page)

```
        access_log /var/log/nginx/extapi.access_log;
        error_log /var/log/nginx/extapi.error_log;
}
```

4. Enable the load balancer in nginx.

```
sudo ln -s /etc/nginx/sites-available/extapi /etc/nginx/sites-enabled/
```

5. Restart nginx.

```
sudo service nginx restart
```

6. On the principal server and each additional biometric server, open the `/etc/findface-extraction-api.ini` configuration file. Substitute localhost in the `listen` parameter with the relevant server address that you have specified in `upstream extapibackends` (`/etc/nginx/sites-available/extapi`) before. In our example, the address of the 1st additional extraction server has to be substituted as such:

```
sudo vi /etc/findface-extraction-api.ini

listen: 172.168.1.10:18666
```

7. Restart the `findface-extraction-api` on the principal server and each additional biometric server.

```
sudo systemctl restart findface-extraction-api.service
```

The load balancing is now successfully set up. Be sure to specify the actual gateway server IP address and listening port, when configuring the FindFace *network*.

Distribute Database

The `findface-tarantool-server` component connects the Tarantool database and the `findface-sf-api` component, transferring search results from the database to `findface-sf-api` for further processing. To increase search speed, multiple `findface-tarantool-server` shards can be created on each Tarantool host. Their running concurrently leads to a remarkable increase in performance. Each shard can handle up to approximately 10,000,000 faces. When deploying `findface-tarantool-server` from installer, shards are created automatically given the server hardware.

To distribute the face database, install only a `findface-tarantool-server` instance on each additional database server. Answer the installer questions as follows:

- Product to install: `FindFace`.
- Installation type: `Fully customized installation`.
- FindFace components to install: `findface-tarantool-server`. To make a selection, first deselect all the listed components by entering `-*` in the command line, then select `findface-tarantool-server` by entering its sequence number (keyword): `13`. Enter `done` to save your selection and proceed to another step.

After that, the installation process will automatically begin. The answers will be saved to a file `/tmp/<findface-installer-*>.json`. Use this file to install `findface-tarantool-server` on other hosts without having to answer the questions again.

```
sudo ./findface-security-and-server-4.5.run -f /tmp/<findface-installer-*>.json
```

As a result of the installation, `findface-tarantool-server` shards will be automatically installed in the amount of `N = min(max(min(mem_mb // 2000, cpu_cores), 1), 16 * cpu_cores)`. I.e., it is equal to the RAM size in

MB divided by 2000, or the number of CPU physical cores (but at least one shard), or the number of CPU physical cores multiplied by 16, should the first obtained value be greater.

Be sure to specify the shards IP addresses and ports, when configuring the FindFace *network*. To learn the port numbers, execute on each database server:

```
sudo cat /etc/tarantool/instances.enabled/*shard* | grep -E ".start|(listen =)"`
```

You will get the following result:

```
    listen = '127.0.0.1:33001',
FindFace.start("127.0.0.1", 8101, {
    listen = '127.0.0.1:33002',
FindFace.start("127.0.0.1", 8102, {
```

You can find the port number of a shard in the `FindFace.start` section, for example, `8101`, `8102`, etc.

Configure Network

After all the FindFace components are deployed, configure their interaction over the network. Do the following:

1. Open the `/etc/findface-sf-api.ini` configuration file:

   ```
   sudo vi /etc/findface-sf-api.ini
   ```

   Specify the following parameters:

Parameter	Description
`extraction-api -> extraction-api`	IP address and listening port of the *gateway biometric server* with set up load balancing.
`storage-api -> shards -> master`	IP address and port of the `findface-tarantool-server` master shard. Specify each shard by analogy.
`upload_url`	WebDAV NginX path to send original images, thumbnails and normalized face images to the `findface-upload` service.

   ```
   ...
   extraction-api:
     extraction-api: http://172.168.1.9:18667

   ...
   webdav:
     upload-url: http://127.0.0.1:3333/uploads/

   ...
   storage-api:
     ...
     shards:
     - master: http://172.168.1.9:8101/v2/
       slave: ''
     - master: http://172.168.1.9:8102/v2/
       slave: ''
     - master: http://172.168.1.12:8101/v2/
       slave: ''
   ```

(continues on next page)

(continued from previous page)

```
  - master: http://172.168.1.12:8102/v2/
    slave: ''
  - master: http://172.168.1.13:8102/v2/
    slave: ''
  - master: http://172.168.1.13:8102/v2/
    slave: ''
```

2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

Specify the following parameters:

```
sudo vi /etc/findface-security/config.py

...
# SERVICE_EXTERNAL_ADDRESS prioritized for webhooks and genetec
SERVICE_EXTERNAL_ADDRESS = 'http://localhost'
EXTERNAL_ADDRESS = 'http://127.0.0.1'



...
FFSECURITY = {
    'VIDEO_DETECTOR_TOKEN': '7ce2679adfc4d74edcf508bea4d67208',
    ...
    'EXTRACTION_API': 'http://172.168.1.9:18667/',
    'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
    ...
    'NTLS_HTTP_URL': 'http://127.0.0.1:3185',
    'ROUTER_URL': 'http://172.168.1.9',
    ...
    'SF_API_ADDRESS': 'http://127.0.0.1:18411',
    ...
}
```

The FindFace components interaction is now set up.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

1.5.7 Add NVIDIA Repository and Install Drivers (GPU only)

FindFace on GPU requires the prior installation of NVIDIA drivers.

To add the NVIDIA repository and install the drivers, do the following:

1. Download the installer file `findface-security-and-server-4.5.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

```
chmod +x findface-security-and-server-4.5.run
```

4. Execute the `.run` file.

```
sudo ./findface-security-and-server-4.5.run
```

5. Choose the product to install: `NVIDIA CUDA drivers`.

1.6 Maintenance and Troubleshooting

1.6.1 Back Up and Recover Data Storages

This section is all about the backup and recovery of the FindFace data storages, which are the following:

- Tarantool-based biometric database that stores biometric samples (feature vectors) and face identification events.
- Main system database `ffsecurity` based on PostgreSQL, that stores internal system data, dossiers, user accounts, and camera settings.
- Directory `/var/lib/findface-security/uploads` that stores uploaded dossier photos, video files, and such event artifacts as full frames, face thumbnails, and normalized face images.
- Directory `/var/lib/ffupload/` that stores only such event artifacts as face thumbnails.

In this section:

- *Biometric Database Backup and Recovery*
  - *Utilities*
  - *Back Up Database*
  - *Recover Database*
- *Main Database Backup and Recovery*
- *Artifacts Backup and Recovery*

Biometric Database Backup and Recovery

There are 3 galleries in the Tarantool-based biometric database:

- `ffsec_dossier_face`: biometric samples extracted from dossier photos.
- `ffsec_events`: biometric samples extracted from faces detected in the video.
- `ffsec_persons`: centroids of persons (virtual biometric samples averaged across all person's faces) and metadata.

The database backup/recovery functionality allows you to fully restore all the galleries when needed.

To avoid data loss, we recommend you back up a biometric database at least once a week. Overall, the backups' frequency depends on the number of dossiers and face recognition events, and available disk space.

Be sure to back up the database before *migrating* your system to another biometric model.

Utilities

To back up and recover the FindFace biometric database, the following utilities are needed:

1. backup: `findface-storage-api-dump`,
2. recovery: `findface-storage-api-restore`.

These utilities are automatically installed along with `findface-sf-api`.

Back Up Database

To back up the biometric database, use the `findface-storage-api-dump` utility as follows:

Important: The following services must be active: `findface-tarantool-server`, `findface-sf-api`.

Note: The backup functionality can be applied to a distributed database. In this case, the `findface-storage-api-dump` utility will back up galleries on all the shards specified in `/etc/findface-sf-api.ini`.

1. On the server with `findface-sf-api`, create a directory to store the backup files (`/etc/findface_dump` in the example below).
2. Launch the `findface-storage-api-dump` utility by executing:

```
sudo findface-storage-api-dump -output-dir=/etc/findface_dump -config /etc/findface-
↪sf-api.ini
```

   The utility will back up at once all the galleries into the files with corresponding names `ffsec_dossier_face.json`, `ffsec_events.json`, `ffsec_persons.json`, and save them into the directory. These files contain all the data needed to restore the entire database.

Recover Database

To recover the biometric database from the backup, launch the `findface-storage-api-restore` utility for all the files in the backup folder:

```
sudo findface-storage-api-restore -config /etc/findface-sf-api.ini /etc/findface_dump/*.
↪json
```

The recovery process can be interrupted and resumed whenever necessary. To resume the process after the interruption, launch the `findface-storage-api-restore` utility again.

See also:

- *Backup Options*
- *Restore Options*

Main Database Backup and Recovery

To back up the main database `ffsecurity` based on PostgreSQL, execute:

```
sudo -u postgres pg_dump ffsecurity > ffsecurity_postgres_backup.sql
```

To recover the main database, do the following:

1. Stop the `findface-security` service.

   ```
   sudo systemctl stop findface-security.service
   ```

2. Stop the `pgbouncer` service to delete its active sessions with the `ffsecurity` database.

   ```
   sudo systemctl stop pgbouncer.service
   ```

3. Open the PostgreSQL interactive terminal.

   ```
   sudo -u postgres psql
   ```

4. Remove the old `ffsecurity` database.

   ```
   DROP DATABASE ffsecurity;
   ```

5. Create a new `ffsecurity` database. Leave the PostgreSQL interactive terminal.

   ```
   CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE='C.UTF-8'␣
   ↪LC_CTYPE='C.UTF-8' TEMPLATE template0;
   ```

6. Start the `pgbouncer` service.

   ```
   sudo systemctl start pgbouncer.service
   ```

7. Recover the database content from the backup.

   ```
   sudo -u postgres psql -d ffsecurity -f ffsecurity_postgres_backup.sql
   ```

8. Migrate the database architecture from FindFace to **`PostgreSQL`**, re-create user groups with *predefined* rights and the first user with administrator rights.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

9. Start the `findface-security` service.

```
sudo systemctl start findface-security.service
```

Artifacts Backup and Recovery

The FindFace artifacts, such as uploaded dossier photos, video files, and such event artifacts as full frames, face thumbnails, and normalized face images, are stored in the following directories:

- `/var/lib/findface-security/uploads`
- `/var/lib/ffupload/`

To back up the artifacts, execute:

```
sudo tar -cvzf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz /var/lib/findface-
↪security/uploads/
sudo tar -cvzf /home/some_directory/var_lib_ffupload.tar.gz /var/lib/ffupload/
```

To recover the artifacts, execute the following commands from the root directory:

```
cd /
sudo tar -xvf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz
sudo tar -xvf /home/some_directory/var_lib_ffupload.tar.gz
```

1.6.2 Migrate Face Data to Different Neural Network Model

Tip: Do not hesitate to contact our experts on migration by support@ntechlab.com.

Sometimes you have to migrate the face biometric data to another neural network model, such as when you decide to update to the latest version of the product that uses a different set of neural networks.

To migrate to a different neural network model, do the following:

1. Create a backup of the Tarantool-based biometric database in any directory of your choice, for example, `/etc/findface_dump`.

   Tip: See *Back Up and Recover Data Storages* for details.

   ```
   mkdir -p /etc/findface_dump
   cd /etc/findface_dump
   sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
   ```

2. Stop the `findface-sf-api` service.

   ```
   sudo systemctl stop findface-sf-api.service
   ```

3. Create new shards that will host regenerated biometric samples.

   1. Open the `/etc/tarantool/instances.available/` directory and find out the number of shards by counting the number of configuration files `shard-*.lua`.

      Note: There are four shards in the example below.

```
cd /etc/tarantool/instances.available/

ls -l

shard-001.lua
shard-002.lua
shard-003.lua
shard-004.lua
```

   2. Create the same number of new shards by copying the configuration files `shard-*.lua`.

      Note: For convenience, the second digit in the new names is `1`: `shard-01*.lua`.

```
sudo cp shard-001.lua shard-011.lua
sudo cp shard-002.lua shard-012.lua
sudo cp shard-003.lua shard-013.lua
sudo cp shard-004.lua shard-014.lua
```

   3. Modify the following lines in each new shard's configuration file, depending on its name (`shard-011`, `shard-012`, etc., in our example):

Old value	New value
listen = '127.0.0.1:32001'	Listen = '127.0.0.1:32011'
vinyl_dir = '/opt/ntech/var/lib/tarantool/shard-001'	vinyl_dir = '/opt/ntech/var/lib/tarantool/shard-011'
work_dir = '/opt/ntech/var/lib/tarantool/shard-001'	work_dir = '/opt/ntech/var/lib/tarantool/shard-011'
memtx_dir = '/opt/ntech/var/lib/tarantool/shard-001/snapshots'	memtx_dir = '/opt/ntech/var/lib/tarantool/shard-011/snapshots'
wal_dir = '/opt/ntech/var/lib/tarantool/shard-001/xlogs'	wal_dir = '/opt/ntech/var/lib/tarantool/shard-011/xlogs'
FindFace.start("127.0.0.1", 8101, {	FindFace.start("127.0.0.1", 8111, {

   4. Create symbolic links to the new shards.

```
cd /etc/tarantool/instances.enabled/

sudo ln -s /etc/tarantool/instances.available/shard-01*.lua /etc/tarantool/
↪instances.enabled/
```

   5. Create directories that will host files of the new shards. Assign permissions for the created directories.

```
cd /opt/ntech/var/lib/tarantool/

mkdir -p shard-01{1..4}/{index,snapshots,xlogs}

chown tarantool:tarantool shard-01* shard-01*/*
```

4. Open the `/etc/findface-extraction-api.ini` configuration file and replace the old neural network model with the new one (`jackfruit_480.cpu.fnk` in the example).

```
sudo vi /etc/findface-extraction-api.ini

face: face/ifruit_320.cpu.fnk -> face: face/jackfruit_480.cpu.fnk
```

Restart the `findface-extraction-api` service.

```
sudo systemctl restart findface-extraction-api.service
```

5. Start the new shards.

```
for i in {11..14}; do sudo systemctl start tarantool@shard-0$i; done
```

6. Create a configuration file with migration settings <`migration.ini`> based on the example below.

```
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 0s
  extraction-api: http://127.0.0.1:18666
storage-api-from: # current location of the gallery
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  shards:
    - master: http://127.0.0.1:8101/v2/
      slave: ""
    - master: http://127.0.0.1:8102/v2/
      slave: ""
    - master: http://127.0.0.1:8103/v2/
      slave: ""
    - master: http://127.0.0.1:8104/v2/
      slave: ""
storage-api-to:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
```

(continues on next page)

(continued from previous page)

```
  shards:
    - master: http://127.0.0.1:8111/v2/
      slave: ""
    - master: http://127.0.0.1:8112/v2/
      slave: ""
    - master: http://127.0.0.1:8113/v2/
      slave: ""
    - master: http://127.0.0.1:8114/v2/
      slave: ""
workers_num: 3
faces_limit: 100
extraction_batch_size: 8
normalized_storage:
  type: webdav
  enabled: True
  webdav:
    upload-url: http://127.0.0.1:3333/uploads/
  s3:
    endpoint: ""
    bucket-name: ""
    access-key: ""
    secret-access-key: ""
    secure: False
    region: ""
    public-url: ""
    operation-timeout: 30
```

In the `storage-api-from` section, specify the old shards to migrate the data from.

```
storage-api-from: # current location of the gallery
  ...
  shards:
    - master: http://127.0.0.1:8101/v2/
      slave: ""
    - master: http://127.0.0.1:8102/v2/
      slave: ""
    - master: http://127.0.0.1:8103/v2/
      slave: ""
     - master: http://127.0.0.1:8104/v2/
  ...
```

In the `storage-api-to` section, specify the new shards that will host migrated data.

```
storage-api-to:
  ...
  shards:
    - master: http://127.0.0.1:8111/v2/
      slave: ""
    - master: http://127.0.0.1:8112/v2/
      slave: ""
    - master: http://127.0.0.1:8113/v2/
      slave: ""
```

(continues on next page)

(continued from previous page)

```
    - master: http://127.0.0.1:8114/v2/
      slave: ""
  ...
```

7. Launch the `findface-sf-api-migrate` utility with the `-config` option and provide the `<migration.ini>` configuration file.

```
findface-sf-api-migrate -config migration.ini
```

Note: The migration process can take up a significant amount of time if there are many events and dossiers in the system.

8. After the migration is complete, stop the old shards and disable their autostart in OS (do not remove them).

```
for i in {01..04}; do sudo systemctl stop tarantool@shard-0$i.service ; done

for i in {01..04}; do sudo systemctl disable tarantool@shard-0$i.service ; done
```

9. Open the `/etc/findface-sf-api.ini` configuration file and adjust the shards ports, subject to the new shards settings. Restart the `findface-sf-api` service.

```
sudo vi /etc/findface-sf-api.ini

shards:
  - master: http://127.0.0.1:8111/v2/
    slave: ""
  - master: http://127.0.0.1:8112/v2/
    slave: ""
  - master: http://127.0.0.1:8113/v2/
    slave: ""
  - master: http://127.0.0.1:8114/v2/
    slave: ""

sudo systemctl start findface-sf-api.service
```

10. Import the new database structure from the `tnt_schema.lua` file.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

See also:

Modify Biometric Database Structure.

11. Migrate the main database architecture from FindFace to **PostgreSQL**, re-create user groups with *predefined* rights, and the first user with administrator rights.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

12. Restart the services.

```
sudo systemctl restart findface-security.service
sudo systemctl restart findface-extraction-api findface-video-worker* findface-
↪video-manager findface-sf-api
```

1.6.3 Modify Biometric Database Structure

Sometimes it may be necessary to apply a new structural schema to your Tarantool-based biometric database, for example, when updating to the latest version of the product, or when you want to enhance the default database structure with additional parameters, advanced face metadata, and so on.

In this section:

- *About Database Structure*
- *Structure Modification*

About Database Structure

In FindFace, the database structure is set via the `/etc/findface-security/tnt_schema.lua` file.

The structure is created as a set of fields. Each field is described with the following parameters:

- id: field id;
- name: field name, must be the same as the name of a relevant face parameter;
- field_type: data type;
- default: field default value. If a default value exceeds '1e14 – 1', use a string data type to specify it, for example, `"123123.."` instead of `123123...`

You can find the default `tnt_schema.lua` file below:

```
meta_scheme = {
    -- internal.normalized_id:
    {
        default = '',
        field_type = 'string',
        id = 1,
        name = 'normalized_id',
    },
    -- internal.feat:
    {
        default = '',
        field_type = 'string',
        id = 2,
        name = 'feat',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged:
    {
        default = 0,
        field_type = 'unsigned',
```

(continues on next page)

(continued from previous page)

```
        id = 3,
        name = 'm:acknowledged',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_by:
    {
        default = 0,
        field_type = 'unsigned',
        id = 4,
        name = 'm:acknowledged_by',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_date:
    {
        default = 0,
        field_type = 'unsigned',
        id = 5,
        name = 'm:acknowledged_date',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:acknowledged_reaction:
    {
        default = '',
        field_type = 'string',
        id = 6,
        name = 'm:acknowledged_reaction',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:camera:
    {
        default = 0,
        field_type = 'unsigned',
        id = 7,
        name = 'm:camera',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:camera_group:
    {
        default = 0,
        field_type = 'unsigned',
        id = 8,
        name = 'm:camera_group',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:confidence:
    {
        default = "1000000000000000000",
        field_type = 'unsigned',
        id = 9,
        name = 'm:confidence',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:created_date:
    {
        default = 0,
        field_type = 'unsigned',
        id = 10,
        name = 'm:created_date',
    },
```

(continues on next page)

(continued from previous page)

```
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:episode:
    {
        default = 0,
        field_type = 'unsigned',
        id = 11,
        name = 'm:episode',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:episode_open:
    {
        default = 0,
        field_type = 'unsigned',
        id = 12,
        name = 'm:episode_open',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_age:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 13,
        name = 'm:f_age',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_beard_class:
    {
        default = '',
        field_type = 'string',
        id = 14,
        name = 'm:f_beard_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_beard_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 15,
        name = 'm:f_beard_score',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_emotions_class:
    {
        default = '',
        field_type = 'string',
        id = 16,
        name = 'm:f_emotions_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_emotions_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 17,
        name = 'm:f_emotions_score',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_gender_class:
    {
        default = '',
```

(continues on next page)

(continued from previous page)

```
        field_type = 'string',
        id = 18,
        name = 'm:f_gender_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_gender_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 19,
        name = 'm:f_gender_score',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_glasses_class:
    {
        default = '',
        field_type = 'string',
        id = 20,
        name = 'm:f_glasses_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_glasses_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 21,
        name = 'm:f_glasses_score',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_liveness_class:
    {
        default = '',
        field_type = 'string',
        id = 22,
        name = 'm:f_liveness_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_liveness_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 23,
        name = 'm:f_liveness_score',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_medmask_class:
    {
        default = '',
        field_type = 'string',
        id = 24,
        name = 'm:f_medmask_class',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:f_medmask_score:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 25,
        name = 'm:f_medmask_score',
```

(continues on next page)

(continued from previous page)

```
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:thumbnail:
    {
        default = '',
        field_type = 'string',
        id = 28,
        name = 'm:thumbnail',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame:
    {
        default = '',
        field_type = 'string',
        id = 29,
        name = 'm:frame',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_bottom:
    {
        default = 0,
        field_type = 'unsigned',
        id = 30,
        name = 'm:frame_coords_bottom',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_left:
    {
        default = 0,
        field_type = 'unsigned',
        id = 31,
        name = 'm:frame_coords_left',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_right:
    {
        default = 0,
        field_type = 'unsigned',
        id = 32,
        name = 'm:frame_coords_right',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:frame_coords_top:
    {
        default = 0,
        field_type = 'unsigned',
        id = 33,
        name = 'm:frame_coords_top',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:pk:
    {
        default = 0,
        field_type = 'unsigned',
        id = 34,
        name = 'm:pk',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched:
    {
```

(continues on next page)

(continued from previous page)

```
        default = 0,
        field_type = 'unsigned',
        id = 35,
        name = 'm:matched',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_dossier:
    {
        default = 0,
        field_type = 'unsigned',
        id = 36,
        name = 'm:matched_dossier',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_face:
    {
        default = 0,
        field_type = 'unsigned',
        id = 37,
        name = 'm:matched_face',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:matched_lists:
    {
        default = {},
        field_type = 'set[unsigned]',
        id = 38,
        name = 'm:matched_lists',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:normalized_photo:
    {
        default = '',
        field_type = 'string',
        id = 39,
        name = 'm:normalized_photo',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:quality:
    {
        default = "10000000000000000000",
        field_type = 'unsigned',
        id = 40,
        name = 'm:quality',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:scores:
    {
        default = '',
        field_type = 'string',
        id = 41,
        name = 'm:scores',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:temperature:
    {
        default = "0",
        field_type = 'unsigned',
        id = 42,
```

(continues on next page)

(continued from previous page)

```
        name = 'm:temperature',
    },
    -- <class 'ffsecurity.entities_tnt.event.models.ListEvent'>.m:video_source:
    {
        default = 0,
        field_type = 'unsigned',
        id = 43,
        name = 'm:video_source',
    },
    -- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:dossier:
    {
        default = 0,
        field_type = 'unsigned',
        id = 44,
        name = 'm:dossier',
    },
    -- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:modified_date:
    {
        default = 0,
        field_type = 'unsigned',
        id = 45,
        name = 'm:modified_date',
    },
    -- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:source_photo:
    {
        default = '',
        field_type = 'string',
        id = 46,
        name = 'm:source_photo',
    },
    -- <class 'ffsecurity.entities_tnt.dossier_face.models.DossierFace'>.m:source_photo_
↪name:
    {
        default = '',
        field_type = 'string',
        id = 47,
        name = 'm:source_photo_name',
    },
}
-- Fields referenced by multiple models: m:thumbnail, m:frame_coords_left, m:frame_
↪coords_right, m:pk, m:frame_coords_bottom, m:created_date, m:frame_coords_top
meta_indexes = {'m:episode', 'm:episode_open', 'm:video_source'}
```

Structure Modification

To modify the database structure, do the following:

1. Stop the `findface-security` service.

```
sudo systemctl stop findface-security.service
```

2. Create a backup of the Tarantool-based biometric database in any directory of your choice, for example, `/etc/findface_dump`.

Tip: See *Back Up and Recover Data Storages* for details.

```
mkdir -p /etc/findface_dump
cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

3. Prepare the `tnt_schema.lua` file containing the new database structure.
4. Modify the database structure by applying the new `tnt_schema.lua` file.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua
```

5. Navigate to the directory with Tarantool configuration file(s) `/etc/tarantool/instances.available/`. For each shard, make sure that there is a line `dofile("/etc/findface-security/tnt_schema.lua")` before the `FindFace.start` section and `meta_scheme` and `meta_indexes` are defined in the `FindFace.start` parameters.

```
sudo vi /etc/tarantool/instances.available/<shard_00N>.lua

dofile("/etc/findface-security/tnt_schema.lua")

FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    meta_indexes=meta_indexes,
    meta_scheme = meta_scheme
})
```

6. Purge data from all the directories relevant to active shards.

```
sudo rm /opt/ntech/var/lib/tarantool/shard-*/{index,snapshots,xlogs}/*
```

7. Restore the Tarantool database from the backup.

Important: If some fields were removed from the new database structure, you have to first manually delete the corresponding data from the backup copy.

```
cd /tmp/dump
for x in *.json; do curl -X POST "http://127.0.0.1:18411/v2/galleries/${x%%.json}";␣
↪done
for x in *.json; do sudo findface-storage-api-restore -config /etc/findface-sf-api.
↪ini < "$x"; done
```

8. Start the `findface-security` service.

```
sudo systemctl start findface-security.service
```

See also:

Dossier Face Custom Metadata in Tarantool

1.6.4 Remove FindFace Instance

You can automatically remove FindFace along with the database by using the `ffsec_uninstall.sh` script. The FindFace configuration files and database will be backed up.

Do the following:

1. Download the `ffsec_uninstall.sh` script to some directory on a designated host (for example, to `/home/username/`).
2. From this directory, make the script executable.

```
chmod +x ffsec_uninstall.sh
```

3. Run the script.

```
sudo ./ffsec_uninstall.sh
```

4. Answer **all** to completely remove FindFace along with the database.

1.6.5 Check Component Status

Check the status of components once you have encountered a system problem.

Component	Command to view service status
`findface-extraction-api`	sudo systemctl status findface-extraction-api.service
`findface-sf-api`	sudo systemctl status findface-sf-api.service
`findface-tarantool-server`	sudo systemctl status tarantool.service
`findface-tarantool-server` shards	sudo systemctl status tarantool@shard-00*
`findface-video-manager`	sudo systemctl status findface-video-manager.service
`findface-video-worker`	sudo systemctl status findface-video-worker*.service
`findface-ntls`	sudo systemctl status findface-ntls
`findface-security`	sudo systemctl status findface-security.service
`findface-counter`	sudo systemctl status findface-counter.service
`findface-liveness-api`	sudo systemctl status findface-liveness-api.service
`etcd`	sudo systemctl status etcd.service
`NginX`	sudo systemctl status nginx.service
`memcached`	sudo systemctl status memcached.service
`postgresql`	sudo systemctl status postgresql*
`redis`	sudo systemctl status redis.service
`pgbouncer`	sudo systemctl status pgbouncer.service

1.6.6 Service Logs

Service logs provide a complete record of each FindFace component activity. Consulting logs is one of the first things you should do to identify a cause for any system problem.

Use the `journalctl -u <component>` command to consult a component log, for example as follows:

```
journalctl -u findface-extraction-api
```

Important: In order to enable saving `journald` logs to your hard drive, uncomment and edit the `Storage` parameter in the `/etc/systemd/journald.conf` file:

```
sudo vi /etc/systemd/journald.conf
...
[Journal]
Storage=persistent
```

If necessary, uncomment and edit the `SystemMaxUse` parameter as well. This parameter determines the maximum volume of log files on your hard drive (10% by default).

```
SystemMaxUse=15
```

See also:

Audit Logs

1.6.7 Troubleshoot Licensing and `findface-ntls`

When troubleshooting licensing and `findface-ntls` (see *Licensing Info*), the first step is to retrieve the licensing information and `findface-ntls` status. You can do so by sending an API request to `findface-ntls`. Necessary actions are then to be undertaken, subject to the response content.

Tip: Please do not hesitate to contact our experts on troubleshooting by support@ntechlab.com.

Note: The online licensing is done via the NtechLab Global License Manager `license.ntechlab.com`. Check its availability. A stable internet connection and DNS are required.

To retrieve the FindFace *licensing* information and `findface-ntls` status, execute on the `findface-ntls` host console:

```
curl http://localhost:3185/license.json -s | jq
```

The response will be given in JSON. One of the most significant parameters is `last_updated`. It indicates in seconds how long ago the local license has been checked for the last time.

Interpret the `last_updated` value as follows:

- [0, 5] — everything is alright.
- (5, 30] — there may be some problems with connection, or with the local drive where the license file is stored.
- (30; 120] — almost certainly something bad happened.

- (120; ∞) — the licensing source response has been timed out. Take action.
- `"valid": false`: connection with the licensing source was never established.

```
curl http://localhost:3185/license.json -s | jq
{
  "name": "NTLS",
  "time": 1565186356,
  "type": "online",
  "license_id": "61063ce4b86945e1b70c3bdbedea453b",
  "generated": 1514467939,
  "last_updated": 5,
  "valid": {
    "value": true,
    "description": ""
  },
  "source": "/opt/ntech/license/import_
↪b68d7b7ec9a7310d18832035318cff0c9ddf11e3a9ab0ae962fbe48645e196d1.lic",
  "limits": [
    {
      "type": "time",
      "name": "end",
      "value": 1609161621
    },
    {
      "type": "number",
      "name": "faces",
      "value": 9007199254740991,
      "current": 0
    },
    {
      "type": "number",
      "name": "cameras",
      "value": 4294967295,
      "current": 0
    },
    {
      "type": "number",
      "name": "extraction_api",
      "value": 256,
      "current": 0
    },
    {
      "type": "boolean",
      "name": "gender",
      "value": true
    },
    {
      "type": "boolean",
      "name": "age",
      "value": true
    },
    {
      "type": "boolean",
```

(continues on next page)

(continued from previous page)


```
      "name": "emotions",
      "value": true
    },
    {
      "type": "boolean",
      "name": "fast-index",
      "value": true
    },
    {
      "type": "boolean",
      "name": "sec-genetec",
      "value": false
    },
    {
      "type": "boolean",
      "name": "countries",
      "value": false
    },
    {
      "type": "boolean",
      "name": "beard",
      "value": false
    },
    {
      "type": "boolean",
      "name": "glasses",
      "value": false
    },
    {
      "type": "boolean",
      "name": "liveness",
      "value": false
    }
  ],
  "services": [
    {
      "name": "video-worker",
      "ip": "127.0.0.1:53276"
    },
    {
      "name": "FindFace-tarantool",
      "ip": "127.0.0.1:53284"
    },
    {
      "name": "FindFace-tarantool",
      "ip": "127.0.0.1:53288"
    }
  ]
}
```

1.6.8 Automatic Tarantool Recovery

If your system architecture doesn't imply uninterrupted availability of Tarantool servers, it is recommended to enable automatic database recovery. In this case, each time an error occurs while reading a snapshot or xlog file, Tarantool will skip invalid records, read as much data as possible, and re-build the file.

To enable automatic database recovery, do the following:

1. For each Tarantool shard, open the configuration file `/etc/tarantool/instances.available/shard-*.lua` and uncomment `force_recovery = true`.

```
sudo vi /etc/tarantool/instances.available/shard-*.lua


box.cfg{

    force_recovery = true,
}
```

2. Restart the shards.

```
systemctl restart tarantool@shard-*
```

1.6.9 Manually Purge Old Data from Database

Tip: To schedule automatic database cleanup, see *Automatic Event And Episode Cleanup*.

To manually remove old data from the FindFace database, use the `cleanup` utility. You can separately remove the following data:

- matched events and related episodes,
- unmatched events and related episodes,
- full frames of matched events,
- full frames of unmatched events,
- counter records,
- person events.

To invoke the `cleanup` help message, execute:

```
sudo findface-security cleanup --help
usage: findface-security cleanup [-h] [--as-configured]
                                 [--events-matched-age EVENTS_MATCHED_AGE]
                                 [--events-unmatched-age EVENTS_UNMATCHED_AGE]
                                 [--events-fullframe-matched-age EVENTS_FULLFRAME_
↪MATCHED_AGE]
                                 [--events-fullframe-unmatched-age EVENTS_FULLFRAME_
↪UNMATCHED_AGE]
                                 [--counter-records-age COUNTER_RECORDS_AGE]
                                 [--person-events-age PERSON_EVENTS_AGE]
                                 [--configuration CONFIGURATION] [--version]
```

(continues on next page)

(continued from previous page)

```
                                [-v {0,1,2,3}] [--settings SETTINGS]
                                [--pythonpath PYTHONPATH] [--traceback]
                                [--no-color]
Delete FFSecurity entities
optional arguments:
  -h, --help            show this help message and exit
  --as-configured       Apply config age options for events, counter records
                        and persons. Can't be used with other arguments.
  --events-matched-age EVENTS_MATCHED_AGE
                        Minimum age in days of matched events to clean up
  --events-unmatched-age EVENTS_UNMATCHED_AGE
                        Minimum age in days of unmatched events to clean up
  --events-fullframe-matched-age EVENTS_FULLFRAME_MATCHED_AGE
                        Minimum age in days of matched events fullframes to
                        clean up
  --events-fullframe-unmatched-age EVENTS_FULLFRAME_UNMATCHED_AGE
                        Minimum age in days of unmatched events fullframes to
                        clean up
  --counter-records-age COUNTER_RECORDS_AGE
                        Minimum age in days of counter records to clean up
  --person-events-age PERSON_EVENTS_AGE
                        Minimum age in days of person events to clean up
  --configuration CONFIGURATION
                        The name of the configuration class to load, e.g.
                        "Development". If this isn't provided, the
                        DJANGO_CONFIGURATION environment variable will be
                        used.
  --version             show program's version number and exit
  -v {0,1,2,3}, --verbosity {0,1,2,3}
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on CommandError exceptions
  --no-color            Don't colorize the command output.
```

To entirely remove events and episodes older than a given number of days, use the `--events-matched-age`/`--events-unmatched-age` options. For example, to remove unmatched events and episodes older than 5 days, execute:

```
sudo findface-security cleanup --events-unmatched-age 5
```

To remove only matched events and episodes older than 5 days, execute:

```
sudo findface-security cleanup --events-matched-age 5
```

The following commands remove only full frames of matched/unmatched events:

```
sudo findface-security cleanup --events-fullframe-matched-age 5
sudo findface-security cleanup --events-fullframe-unmatched-age 5
```

To remove only counter records, execute:

```
sudo findface-security cleanup --counter-records-age 5
```

To remove only person events, execute:

```
sudo findface-security cleanup --person-events-age 5
```

Important: You must provide at least one of the mentioned arguments.

1.6.10 Disable Services

You can disable the following FindFace services should you no longer need them:

- episodes
- video archive queue manager
- webhooks
- persons

To do so, open the `/etc/findface-security/config.py` configuration file and modify the `SERVICES` section.

```
sudo vi /etc/findface-security/config.py

# disable unused services to increase
# overall system performance in some cases.
SERVICES = {
    "ffsecurity": {
        "episodes": True,
        "webhooks": True,
        # use queue manager to prevent drops of video archive events
        "video_archive_events_manager": True,
        "persons": False,
    }
}
```

After that, the corresponding tabs will disappear from the web interface.

Note: A tab will remain if there are some entities on it (for example, webhooks on the *Webhooks* tab). However, new artifacts will cease to arrive.

1.6.11 Reset Password

To reset a user password to the FindFace web interface, execute the following command:

```
findface-security changepassword %username
```

1.6.12 Migrate Data to Another Disk

High disk load may lead to delays in event arrivals. In severe cases, it might result in complete inoperability of FindFace. One of the means for reducing the disk load is to migrate the FindFace data storages to another disk.

In this section:

- *Prepare Disk*
- *Migrate Photo Storage*
- *Migrate Main Database (PostgreSQL)*

Prepare Disk

To prepare a disk for the data migration, do the following:

1. Create a new mount point (`/mnt/ffdata` in our example).

```
sudo mkdir /mnt/ffdata
sudo chown ntech:ntech /mnt/ffdata
```

2. Create a partition.

```
sudo parted /dev/sdb
mklabel gpt
mkpart primary ext4 1MiB 100%
q
sudo mkfs.ext4 /dev/sdb1
```

3. Learn the UUID of the partition (`sdb1` in our example).

```
sudo blkid | grep sdb1
/dev/sdb1: LABEL="data" UUID="0638ebe0-853e-43ea-8f35-bfae305695d1" TYPE="ext4"␣
↪PARTUUID="8cebaacc-77d7-4757-b4c6-14147e92646c"
```

4. Add the partition to fstab to make it automatically mount on booting.

```
sudo vi /etc/fstab
------------
#DATA mount
UUID=0638ebe0-853e-43ea-8f35-bfae305695d1 /mnt/ffdata/    ext4    auto,user,rw    ␣
↪0       2
-------------
```

5. Mount all the filesystems.

```
sudo mount -a
```

Migrate Photo Storage

To migrate the FindFace photo storage, do the following:

1. Stop the `findface-security` service to prevent the data loss.

```
sudo systemctl stop findface-security
```

2. By default, the photo data are stored at `/var/lib/`. Migrate the photo storage to the *new disk*.

```
sudo cp -ax /var/lib/findface-security/ -R /mnt/ffdata/
sudo rm -r /var/lib/findface-security/
sudo cp -ax /var/lib/ffupload/ -R /mnt/ffdata/
sudo rm -r /var/lib/ffupload/
```

3. Create symbolic links for the new directories.

```
sudo ln -s /mnt/ffdata/findface-security/ /var/lib/
sudo ln -s /mnt/ffdata/ffupload/ /var/lib/
```

4. Ensure that the rights are correctly assigned.

```
sudo chown ntech:ntech /mnt/ffdata/findface-security/
```

5. Start the `findface-security` service.

```
sudo systemctl start findface-security
```

Migrate Main Database (PostgreSQL)

To migrate the PostgreSQL database, do the following:

1. Learn the current database directory.

```
postgres=# SHOW data_directory;

        data_directory
------------------------------
 /var/lib/postgresql/10/main
```

2. Stop PostgreSQL.

```
sudo systemctl stop postgresql
```

3. Create a new directory that will hold the database and assign it to the ntech user.

```
mkdir postgres_data_dir
chown ntech postgres_data_dir
```

4. Migrate the database and backup the old one.

```
sudo rsync -av /var/lib/postgresql /test/postgres_data_dir/
sudo mv /var/lib/postgresql/10/main /backups/pg_backup
```

5. Substitute the directory in the PostgreSQL configuration file <PostgreSQL directory>/postgresql.conf.

```
data_directory = '/test/postgres_data_dir/postgresql/10/main'
```

6. Start PostgreSQL.

```
sudo systemctl start postgresql
```

7. Check if the directory has successfully been changed.

```
postgres=# SHOW data_directory;

        data_directory
-------------------------------------------
/test/postgres_data_dir/postgresql/10/main
```

1.7 Appendices

1.7.1 Components in Depth

findface-extraction-api

The findface-extraction-api service uses neural networks to detect a face in an image, extract face biometric data (feature vector), and recognize gender, age, emotions, and other features.

It interfaces with the findface-sf-api service as follows:

- Gets original images with faces and normalized face images.
- Returns the coordinates of the face bounding box, and (optionally) feature vector, gender, age and emotions data, should these data be requested by findface-sf-api.

Functionality:

- face detection in an original image (with return of the bbox coordinates),
- face normalization,
- feature vector extraction from a normalized image,
- face feature recognition (gender, age, emotions, beard, glasses, face mask, etc.).

The findface-extraction-api service can be based on CPU (installed from the findface-extraction-api package) or GPU (installed from the findface-extraction-api-gpu package). For both CPU- and GPU-accelerated services, configuration is done through the /etc/findface-extraction-api.ini configuration file. Its content varies subject to the acceleration type.

CPU-service configuration file:

```
detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      aliases:
      - face
      - nnd
      model: facedet/cheetah.cpu.fnk
      options:
        min_object_size: 32
        resolutions:
        - 256x256
        - 384x384
        - 512x512
        - 768x768
        - 1024x1024
        - 1536x1536
        - 2048x2048
  quality_estimator: true
normalizers:
  max_batch_size: 8
  instances: 1
  models:
    carlicplate:
      model: ''
    crop1x:
      model: ''
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.cpu.fnk
    cropbbox:
      model: ''
    norm200:
      model: facenorm/bee.v2.cpu.fnk
extractors:
  max_batch_size: 8
  instances: 1
  models:
    age: ''
    beard: ''
    carattr_color: ''
    carattr_description: ''
    carattr_license_plate: ''
    carattr_make: ''
    carattr_trash: ''
    countries47: ''
    emotions: ''
    face: face/jackfruit_480.cpu.fnk
    gender: ''
    glasses3: ''
    headpose: ''
```

(continues on next page)

(continued from previous page)

```
    liveness: faceattr/liveness.alleyn.v2.cpu.fnk
    luminance_overexposure: ''
    luminance_underexposure: ''
    medmask3: ''
    pedattr_color: ''
    pedestrian: ''
    quality: faceattr/quality.v1.cpu.fnk
    sharpness: ''
    validity: ''
ascend_device: 0
gpu_device: 0
models_root: /usr/share/findface-data/models
cache_dir: /var/cache/findface/models_cache
listen: 127.0.0.1:18666
license_ntls_server: 127.0.0.1:3133
fetch:
  enabled: true
  size_limit: 10485760
max_dimension: 6000
allow_cors: false
ticker_interval: 5000
debug: false
prometheus:
  timing_buckets:
  - 0.001
  - 0.005
  - 0.01
  - 0.02
  - 0.03
  - 0.05
  - 0.1
  - 0.2
  - 0.3
  - 0.5
  - 0.75
  - 0.9
  - 1
  - 1.1
  - 1.3
  - 1.5
  - 1.7
  - 2
  - 3
  - 5
  - 10
  - 20
  - 30
  - 50
  resolution_buckets:
  - 10000
  - 20000
  - 40000
```

(continues on next page)

(continued from previous page)

```
  - 80000
  - 100000
  - 200000
  - 400000
  - 800000
  - 1e+06
  - 2e+06
  - 3e+06
  - 4e+06
  - 5e+06
  - 6e+06
  - 8e+06
  - 1e+07
  - 12000000.0
  - 15000000.0
  - 18000000.0
  - 2e+07
  - 3e+07
  - 5e+07
  - 1e+08
  faces_buckets:
  - 0
  - 1
  - 2
  - 5
  - 10
  - 20
  - 50
  - 75
  - 100
  - 200
  - 300
  - 400
  - 500
  - 600
  - 700
  - 800
  - 900
  - 1000
```

GPU-service configuration file:

```
detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      aliases:
      - face
      - nnd
```

(continues on next page)

(continued from previous page)

```
      model: facedet/cheetah.gpu.fnk
      options:
        min_object_size: 32
        resolutions:
        - 256x256
        - 384x384
        - 512x512
        - 768x768
        - 1024x1024
        - 1536x1536
        - 2048x2048
  quality_estimator: true
normalizers:
  max_batch_size: 8
  instances: 1
  models:
    carlicplate:
      model: ''
    crop1x:
      model: ''
    crop2x:
      model: facenorm/crop2x.v2_maxsize400.gpu.fnk
    cropbbox:
      model: ''
    norm200:
      model: facenorm/bee.v2.gpu.fnk
extractors:
  max_batch_size: 8
  instances: 1
  models:
    age: ''
    beard: ''
    carattr_color: ''
    carattr_description: ''
    carattr_license_plate: ''
    carattr_make: ''
    carattr_trash: ''
    countries47: ''
    emotions: ''
    face: face/jackfruit_480.gpu.fnk
    gender: ''
    glasses3: ''
    headpose: ''
    liveness: faceattr/liveness.alleyn.v2.gpu.fnk
    luminance_overexposure: ''
    luminance_underexposure: ''
    medmask3: ''
    pedattr_color: ''
    pedestrian: ''
    quality: faceattr/quality.v1.gpu.fnk
    sharpness: ''
    validity: ''
```

(continues on next page)

(continued from previous page)

```
ascend_device: 0
gpu_device: 0
models_root: /usr/share/findface-data/models
cache_dir: /var/cache/findface/models_cache
listen: 127.0.0.1:18666
license_ntls_server: 127.0.0.1:3133
fetch:
  enabled: true
  size_limit: 10485760
max_dimension: 6000
allow_cors: false
ticker_interval: 5000
debug: false
prometheus:
  timing_buckets:
  - 0.001
  - 0.005
  - 0.01
  - 0.02
  - 0.03
  - 0.05
  - 0.1
  - 0.2
  - 0.3
  - 0.5
  - 0.75
  - 0.9
  - 1
  - 1.1
  - 1.3
  - 1.5
  - 1.7
  - 2
  - 3
  - 5
  - 10
  - 20
  - 30
  - 50
  resolution_buckets:
  - 10000
  - 20000
  - 40000
  - 80000
  - 100000
  - 200000
  - 400000
  - 800000
  - 1e+06
  - 2e+06
  - 3e+06
  - 4e+06
```

(continues on next page)

(continued from previous page)

```
  - 5e+06
  - 6e+06
  - 8e+06
  - 1e+07
  - 12000000.0
  - 15000000.0
  - 18000000.0
  - 2e+07
  - 3e+07
  - 5e+07
  - 1e+08
  faces_buckets:
  - 0
  - 1
  - 2
  - 5
  - 10
  - 20
  - 50
  - 75
  - 100
  - 200
  - 300
  - 400
  - 500
  - 600
  - 700
  - 800
  - 900
  - 1000
```

When configuring `findface-extraction-api` (on CPU or GPU), refer to the following parameters:

Parameter	Description
`cheetah -> min_object_size`	The minimum size of a face (bbox) guaranteed to be detected. The larger the value, the less resources required for face detection.
`gpu_device`	(Only for GPU) The number of the GPU device used by `findface-extraction-api-gpu`.
`license_ntls_server`	The `ntls` license server IP address and port.

You will also have to enable recognition models for face features such as gender, age, emotions, glasses3, and/or beard, subject to your needs. Be sure to choose the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models.

```
models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
```

(continues on next page)

(continued from previous page)

```
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

The following models are available:

Face feature	Acceleration	Configuration file parameter
face (biometry)	CPU	`face: face/ifruit_320.cpu.fnk` `face: face/jackfruit_160.cpu.fnk` `face: face/jackfruit_320.cpu.fnk` `face: face/jackfruit_480.cpu.fnk`
	GPU	`face: face/ifruit_320.gpu.fnk` `face: face/jackfruit_160.gpu.fnk` `face: face/jackfruit_320.gpu.fnk` `face: face/jackfruit_480.gpu.fnk`
age	CPU	`age: faceattr/age.v1.cpu.fnk`
	GPU	`age: faceattr/age.v1.gpu.fnk`
gender	CPU	`gender: faceattr/gender.v2.cpu.fnk`
	GPU	`gender: faceattr/gender.v2.gpu.fnk`
emotions	CPU	`emotions: faceattr/emotions.v1.cpu.fnk`
	GPU	`emotions: faceattr/emotions.v1.gpu.fnk`
glasses3	CPU	`glasses3: faceattr/glasses3.v0.cpu.fnk`
	GPU	`glasses3: faceattr/glasses3.v0.gpu.fnk`
beard	CPU	`beard: faceattr/beard.v0.cpu.fnk`
	GPU	`beard: faceattr/beard.v0.gpu.fnk`
face mask	CPU	`medmask3: faceattr/medmask3.v2.cpu.fnk`
	GPU	`medmask3: faceattr/medmask3.v2.gpu.fnk`

To enable the neural network model that provides the *liveness standalone service*, specify it in the `liveness` parameter: `faceattr/liveness.alleyn.v2.cpu.fnk`/`faceattr/liveness.alleyn.v2.gpu.fnk`.

```
models:
 ...
 liveness: faceattr/liveness.alleyn.v2.cpu.fnk
 ...

models:
 ...
 liveness: faceattr/liveness.alleyn.v2.gpu.fnk
```

Tip: To disable a model, simply pass an empty value to a relevant parameter. Do not remove the parameter itself as in this case the system will be searching for the default model.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

findface-sf-api

The findface-sf-api service implements HTTP API for the FindFace core main functionality such as face detection and face recognition (the mentioned functions themselves are provided by *findface-extraction-api*`). It interfaces with the biometric database powered by Tarantool via the findface-tarantool-server service, as well as with findface-extraction-api (provides face detection and face recognition) and findface-upload (provides a storage for original images and FindFace core artifacts).

To detect a face in an image, you need to send the image in an API request to findface-sf-api. The findface-sf-api will then redirect the request to findface-extraction-api for face detection and recognition.

If there is a configured video face detection module in the system (like in FindFace Security), findface-sf-api also interfaces with the findface-facerouter service. It receives data of detected in video faces along with processing directives from findface-facerouter, and then executes the received directives, for example, saves faces into a specific database gallery.

Note: In FindFace Security, findface-facerouter functions are performed by findface-security.

Functionality:

- HTTP API implementation (face detection and face recognition methods, performed via findface-extraction-api).
- saving face data to the biometric database (performed via findface-tarantool-server),
- saving original images, face thumbnails and normalized face images to an NginX-powered web server (via findface-upload).
- provides interaction between all the FindFace core components.

The findface-sf-api configuration is done through the /etc/findface-sf-api.ini configuration file.

```
listen: 127.0.0.1:18411
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  extraction-api: http://127.0.0.1:18666
storage-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  shards:
  - master: http://127.0.0.1:8101/v2/
    slave: ''
  - master: http://127.0.0.1:8102/v2/
    slave: ''
  - master: http://127.0.0.1:8103/v2/
    slave: ''
  - master: http://127.0.0.1:8104/v2/
    slave: ''
```

(continues on next page)

(continued from previous page)

```
  - master: http://127.0.0.1:8105/v2/
    slave: ''
  - master: http://127.0.0.1:8106/v2/
    slave: ''
  read_slave_first: false
  max_slave_attempts: 2
  cooldown: 2s
limits:
  url-length: 4096
  deny-networks: 127.0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8
  body-image-length: 33554432
  allow-return-facen: true
cache:
  type: memcache
  inmemory:
    size: 16384
  memcache:
    nodes:
    - 127.0.0.1:11211
    timeout: 100ms
    dns_cache_timeout: 1m0s
  redis:
    network: tcp
    addr: localhost:6379
    password: ''
    db: 0
    timeout: 5s
normalized-storage:
  type: webdav
  enabled: true
  webdav:
    upload-url: http://127.0.0.1:3333/uploads/
    timeouts:
      connect: 5s
      response_header: 30s
      overall: 35s
      idle_connection: 10s
  s3:
    endpoint: ''
    bucket-name: ''
    access-key: ''
    secret-access-key: ''
    secure: true
    region: ''
    public-url: ''
    operation-timeout: 30
```

When configuring `findface-sf-api`, refer to the following parameters:

Parameter	Description
extraction-api -> extraction-api	IP address of the findface-extraction-api host.
limits -> body-image-length	The maximum size of an image in an API request, bytes.
normalized-storage -> webdav -> upload_url	WebDAV NginX path to send original images, thumbnails and normalized face images to the findface-upload service.
storage-api -> shards -> master	IP address of the findface-tarantool-server master shard.
storage-api -> shards -> slave	IP address of the findface-tarantool-server replica shard.

findface-tarantool-server

The findface-tarantool-server service provides interaction between the findface-sf-api service and the Tarantool-based biometric database in the following way:

Tip: See Tarantool official documentation for details.

- From findface-sf-api, findface-tarantool-server receives data, such as information of detected in video faces, to write into the biometric database.
- By request from findface-sf-api, findface-tarantool-server performs database searches and returns search results.

To increase search speed, multiple findface-tarantool-server shards can be created on each Tarantool host. Their running concurrently leads to a remarkable increase in performance (70x-100x).

Functionality:

- saving face data to the biometric database,
- database search,
- implementation of direct API requests to the database (see *Direct API Requests to Tarantool*).

The findface-tarantool-server configuration is done through the /etc/tarantool/instances.available/<shard-*>.lua configuration file. In a cluster environment, configuration has to be done for each shard.

```
--
-- Please, read the tarantool documentation at https://www.tarantool.io/en/doc/1.10/
--

box.cfg{
    -- THIS IS NOT HTTP API PORT, it's for admin operations
    listen = '127.0.0.1:32001',

    --Directory to store data
    vinyl_dir = '/opt/ntech/var/lib/tarantool/shard-001',
    work_dir = '/opt/ntech/var/lib/tarantool/shard-001',
    memtx_dir = '/opt/ntech/var/lib/tarantool/shard-001/snapshots',
    wal_dir = '/opt/ntech/var/lib/tarantool/shard-001/xlogs',

    --Maximum mem usage in bytes
```

(continues on next page)

(continued from previous page)

```
    memtx_memory = 200 * 1024 * 1024,

    checkpoint_interval = 3600*4,
    checkpoint_count = 3,

    --uncomment only if you know what you are doing!!! and don't forget box.snapshot()
    -- wal_mode = 'none',

    --if true, tarantool tries to continue if there is an error while reading a snapshot/
↪xlog files:  skips invalid records, reads as much data as possible and re-builds the␣
↪file
    -- force_recovery = true,
}

pcall(function() box.schema.user.grant('guest', 'execute,read,write', 'universe') end)


dofile("/etc/findface-security/tnt_schema.lua")


-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8101, {
    license_ntls_server="127.0.0.1:3133",
    meta_indexes=meta_indexes,
    meta_scheme=meta_scheme
})
```

When configuring `findface-tarantool-server`, refer to the following parameters:

Parameter	Description
`force_recovery`	Enables automatic database recovery. In this case, each time an error occurs while reading a snapshot or xlog file, Tarantool will skip invalid records, read as much data as possible, and re-build the file.
`license_ntls_server`	IP address and port of the `findface-ntls` license server.
`memtx_memory`	Maximum RAM that can be used by a Tarantool shard. Set in bytes, depending on the number of faces the shard handles. Consult our experts by support@ntechlab.com before setting this parameter.
`meta_scheme`	A database structure to store the face recognition results. The structure is created as a set of fields. Describe each field with the following parameters: `id`: field id; `name`: field name, must be the same as the name of a relevant face parameter; `field_type`: data type; `default`: field default value, if a default value exceeds '1e14 – 1', use a string data type to specify it, for example, `"123123.."` instead of `123123...`

Default database structure is passed from `/etc/findface-security/tnt_schema.lua` to the `meta_scheme` parameter. See *Modify Biometric Database Structure* for details.

findface-upload

The findface-upload component is an NginX-based web server used as a storage for original images, thumbnails and normalized face images which it receives from the findface-sf-api component.

By default the original images, thumbnails and normalized images are stored at /var/lib/ffupload/uploads/.

The findface-upload component is automatically configured upon installation. Custom configuration is not supported.

Video face detection: findface-video-manager and findface-video-worker

Note: The findface-video-worker is delivered in a CPU-accelerated (findface-video-worker-cpu) and a GPU-accelerated (findface-video-worker-gpu) packages.

In this section:

- *Functions of findface-video-manager*
- *Functions of findface-video-worker*
- *Configure Video Face Detection*
- *Jobs*

Functions of findface-video-manager

The findface-video-manager service is the part of the video face detection module that is used for managing the video face detection functionality.

The findface-video-manager service interfaces with findface-video-worker as follows:

- It supplies findface-video-worker with settings and the list of to-be-processed video streams. To do so, it issues a so-called *job*, a video processing task that contains configuration settings and stream data.
- In a distributed system, it distributes video streams (jobs) across vacant findface-video-worker instances.

Note: Configuration settings passed via jobs have priority over the /etc/findface-video-manager.conf configuration file.

The findface-video-manager service functioning requires ETCD, third-party software that implements a distributed key-value store for findface-video-manager. In the FindFace core, ETCD is used as a coordination service, providing the video face detector with fault tolerance.

Functionality:

- allows for configuring video face detection parameters,
- allows for managing the list of to-be-processed video streams,
- implements video face detection management.

Functions of `findface-video-worker`

The `findface-video-worker` service (on CPU/GPU) is the part of the video face detection module, that recognizes faces in the video. It can work with both live streams and files, and supports most video formats and codecs that can be decoded by FFmpeg.

The `findface-video-worker` service interfaces with the `findface-video-manager` and `findface-facerouter` services as follows:

- By request, `findface-video-worker` gets a job with settings and the list of to-be-processed video streams from `findface-video-manager`.
- The `findface-video-worker` posts extracted normalized face images, along with the full frames and meta data (such as bbox, camera ID and detection time) to the `findface-facerouter` service for further processing.

Note: In FindFace, the `findface-facerouter` functions are performed by `findface-security`.

Functionality:

- detects faces in the video,
- extracts normalized face images,
- searches for the best face snapshot,
- snapshot deduplication (only one snapshot per face detection event).

When processing video, `findface-video-worker` consequently uses the following algorithms:

- **`Motion detection`**. Used to reduce resource consumption. Only when the motion detector recognizes the motion of certain intensity that the face tracker can be triggered.
- **`Face tracking`**. The face tracker traces, detects and captures faces in the video. It can simultaneously be working with several faces. It also searches for the best face snapshot, using an embedded neural network. After the best face snapshot is found, it is posted to `findface-facerouter`.

The best face snapshot can be found in one of the following modes:

- Real-time
- Offline

Real-Time Mode

In the real-time mode, `findface-video-worker` posts a face on-the-fly after it appears in the camera field of view. The following posting options are available (see `/etc/findface-video-manager.conf` or *Add Camera*):

- If `realtime_post_every_interval: true`, the face tracker searches for the best face snapshot within each time period equal to `realtime_post_interval` and posts it to `findface-facerouter`.
- If `realtime_post_every_interval: false`, the face tracker searches for the best face snapshot dynamically:
  1. First, the face tracker estimates whether the quality of a face snapshot exceeds a pre-defined internal threshold. If so, the snapshot is posted to `findface-facerouter`.
  2. The threshold value increases after each post. Each time the face tracker gets a higher quality snapshot of the same face, it is posted.
  3. When the face disappears from the camera field of view, the threshold value resets to default.

- If `realtime_post_first_immediately: true`, the face tracker doesn't wait for the first `realtime_post_interval` to complete and posts the first face from a track immediately after it passes through the quality, size, and ROI filters. The way the subsequent postings are sent depends on the `realtime_post_every_interval` value. If `realtime_post_first_immediately: false`, the face tracker posts the first face after the first `realtime_post_interval` completes.

Offline Mode

The offline mode is less storage intensive than the real-time one as in this mode `findface-video-worker` posts only one snapshot per track, but of the highest quality. In this mode, the face tracker buffers a video stream with a face in it until the face disappears from the camera field of view. Then the face tracker picks up the best face snapshot from the buffered video and posts it to `findface-facerouter`.

By default, the offline mode is enabled through the `overall_only` parameter (see `/etc/findface-video-manager.conf` or *Add Camera*).

Configure Video Face Detection

The video face detector configuration is done through the following configuration files:

1. The `findface-video-manager` configuration file `/etc/findface-video-manager.conf`:

```
listen: 127.0.0.1:18810
etcd:
  endpoints: 127.0.0.1:2379
  dial_timeout: 3s
kafka:
  enabled: false
  endpoints: 127.0.0.1:9092
master:
  lease_ttl: 10
  self_url: 127.0.0.1:18811
  self_url_http: 127.0.0.1:18810
rpc:
  listen: 127.0.0.1:18811
  heart_beat_timeout: 4s
router_url: http://127.0.0.1:18820/v0/frame
exp_backoff:
  enabled: false
  min_delay: 1s
  max_delay: 1m0s
  factor: 2
  flush_interval: 2m0s
ntls:
  enabled: false
  url: http://127.0.0.1:3185/
  update_interval: 1m0s
prometheus:
  jobs_processed_duration_buckets:
  - 1
  - 30
  - 60
  - 500
```

(continues on next page)

(continued from previous page)

```
  - 1800
  - 3600
  - 21600
  - .inf
job_scheduler_script: ''
stream_settings:
  ffmpeg_params: []
  md_threshold: 0.002
  md_scale: 0.3
  fd_frame_height: -1
  uc_max_time_diff: 30
  uc_max_dup: 3
  uc_max_avg_shift: 10
  det_period: 8
  realtime: false
  npersons: 4
  disable_drops: false
  tracker_threads: 4
  parse_sei: false
  image_arg: photo
  additional_headers: []
  additional_body: []
  api_timeout: 15000
  api_ssl_verify: true
  post_uniq: true
  min_score: -2
  min_d_score: -1000
  realtime_dly: 500
  realtime_post_perm: false
  rot: ''
  roi: ''
  draw_track: false
  send_track: 0
  min_face_size: 0
  max_face_size: 0
  overall: true
  only_norm: false
  max_candidates: 0
  jpeg_quality: 95
  ffmpeg_format: ''
stream_settings_gpu:
  play_speed: -1
  filter_min_quality: 0.45
  filter_min_face_size: 1
  filter_max_face_size: 8192
  normalized_only: false
  jpeg_quality: 95
  overall_only: false
  use_stream_timestamp: false
  ffmpeg_params: []
  router_timeout_ms: 15000
  router_verify_ssl: true
```

(continues on next page)

(continued from previous page)

```
  router_headers: []
  router_body: []
  start_stream_timestamp: 0
  imotion_threshold: 0
  rot: ''
  roi: ''
  realtime_post_interval: 1
  realtime_post_every_interval: false
  ffmpeg_format: ''
  disable_drops: false
  router_full_frame_png: false
  router_disable_normalized: false
  crop_fullframe_rot: false
  realtime_post_first_immediately: false
  post_first_track_frame: false
  post_last_track_frame: false
  track_max_duration_frames: 0
  send_track_history: false
  stream_data_filter: ''
```

When configuring `findface-video-manager`, refer to the following parameters:

Option	Description
`etcd -> endpoints`	IP address and port of the `etcd` service. Default value: `127.0.0.1:2379`.
`ntls -> enabled`	If true, `findface-video-manager` will send a job to `findface-video-worker` only if the total number of processed cameras does not exceed the allowed number of cameras from the license. Default value: false.
`ntls -> url`	IP address and port of the `findface-ntls` host. Default value: `http://127.0.0.1:3185/`.
`router_url`	IP address and port of the `findface-facerouter` host to receive detected faces from `findface-video-worker`. In FindFace, `findface-facerouter` functions are performed by `findface-security`. Default value: `http://127.0.0.1:18820/v0/frame`.

You can also configure the following parameters:

Note: In the `stream_settings-gpu` section of the file, you will find general settings for all video streams. The settings in this section work for both CPU and GPU configuration. Settings of a particular stream, passed in a job, have priority over those in the configuration file (see *Jobs*).

Note: The `stream_settings` section of the file is deprecated and necessary only for backward compatibility.

Option	Description
crop_fullframe	Crop posted full frames by ROT. Default value: false.
disable_drops	Enables posting all appropriate faces without drops. By default, if `findface-video-worker` does not have enough resources to process all frames with faces, it drops some of them. If this option is active, `findface-video-worker` puts odd frames on the waiting list to process them later. Default value: false.
ffmpeg_format	Pass FFMPEG format (mxg, flv, etc.) if it cannot be detected automatically.
ffmpeg_params	List of a video stream ffmpeg options with their values as a key=value array: ["rtsp_transpotr=tcp", .., "ss=00:20:00"]. Check out the FFmpeg web site for the full list of options. Default value: options not specified.
filter_max_face_size	Maximum size of a face in pixels. Oversized faces are not posted. Default value: 8192.
filter_min_face_size	Minimum size of a face in pixels. Undersized faces are not posted. Default value: 1.
filter_min_quality	Minimum threshold value for a face image quality. Default value: 0.45. Do not change the default value without consulting with our technical experts (support@ntechlab.com).
imotion_threshold	Minimum motion intensity to be detected by the motion detector. The threshold value is to be fitted empirically. Empirical units: zero and positive rational numbers. Milestones: 0 = detector disabled, 0.002 = default value, 0.05 = minimum intensity is too high to detect motion.
jpeg_quality	Quality of an original frame JPEG compression, in percents. Default value: 95%.
normalized_only	Enable posting only normalized face images without full frames. Default value: false.
overall_only	Enables the offline mode for the best face search. Default value: true.
play_speed	If less than zero, the speed is not limited. In other cases, the stream is read with the given `play_speed`. Not applicable for live streams.
post_first_track_frame	Post first frame of a track. Default value: false.
post_last_track_frame	Post the last frame of a track. Default value: false.
realtime_post_only_for_interval	Only for the real-time mode. Post best snapshots obtained within each `realtime_post_interval` time period. If false, search for the best snapshot dynamically and send snapshots in order of increasing quality. Default value: false.
realtime_post_first_time_detected	Enable posting a face right after it appears in a camera field of view (real-time mode). Default value: false.
realtime_post_interval	Only for the real-time mode. Defines the time period in seconds within which the face tracker picks up the best snapshot and posts it to `findface-facerouter`. Default value: 1.
router_body	Additional body fields in a request body when posting a face: ["key = value"]. Default value: body fields not specified.
roi	Enable posting faces detected only inside a region of interest WxH+X+Y. Default value: region not specified.
rot	Enables detecting and tracking faces only inside a clipping rectangle WxH+X+Y. You can use this option to reduce `findface-video-worker` load. Default value: rectangle not specified.
router_disabled_normalized	Send only full frames without normalized images. Do not enable this parameter without supervision from our team as it can affect the entire system functioning. Default value: false (send full frames and normalized images).
router_full_send_in_png	Send full frames in PNG and not in JPEG as set by default. Do not enable this parameter without supervision from our team as it can affect the entire system functioning. Default value: false (send in JPEG).
router_headers	Additional header fields in a request when posting a face: ["key = value"]. Default value: headers not specified.
router_timeout_ms	Timeout for a `findface-facerouter` (or `findface-security` in the standard FindFace configuration) response to a `findface-video-worker` API request, in milliseconds. If the timeout has expired, the system will log an error. Default value: 15000.
router_verify_ssl	Enables a https certificate verification when `findface-video-worker` and `findface-facerouter` (or `findface-security` in the standard FindFace configuration) interact over https. Default value: true. If false, a self-signed certificate can be accepted.
send_track_history	Send track history. Default value: false.
start_stream_time_stamp	Add the specified number of seconds to timestamps from a stream.
use_stream_time_stamp	Use to retrieve and post timestamps from a video stream. If false, post the actual date

1. If you opt for the CPU-accelerated package `findface-video-worker-cpu`, use the `/etc/findface-video-worker-cpu.ini` configuration file:

```
## read streams from file, do not use VideoManager
## type:string env:CFG_INPUT longopt:--input
input =

## exit on first finished job, only when --input specified
## type:bool env:CFG_EXIT_ON_FIRST_FINISHED longopt:--exit-on-first-finished
exit_on_first_finished = false

## batch size
## type:number env:CFG_BATCH_SIZE longopt:--batch-size
batch_size = 4

## http server port for metrics, 0=do not start server
## type:number env:CFG_METRICS_PORT longopt:--metrics-port
metrics_port = 0

## resize scale, 1=do not resize
## type:double env:CFG_RESIZE_SCALE longopt:--resize-scale
resize_scale = 1.000000

## maximum number of streams
## type:number env:CFG_CAPACITY longopt:--capacity
capacity = 10

## command to obtain videomanager's grpc ip:port
## type:string env:CFG_MGR_CMD longopt:--mgr-cmd
mgr_cmd =

## videomanager grpc ip:port
## type:string env:CFG_MGR_STATIC longopt:--mgr-static
mgr_static = 127.0.0.1:18811

## ntls server ip:port
## type:string env:CFG_NTLS_ADDR longopt:--ntls-addr
ntls_addr = 127.0.0.1:3133

## debug: save faces to dir
## type:string env:CFG_SAVE_DIR longopt:--save-dir
save_dir =

## minimum face size
## type:number env:CFG_MIN_FACE_SIZE longopt:--min-face-size
min_face_size = 60

## preinit detector for specified resolutions: "640x480;1920x1080"
## type:string env:CFG_RESOLUTIONS longopt:--resolutions
resolutions =

## use `resolutions` as only possible values, others will rescale
## type:bool env:CFG_STRICT_RESOLUTIONS longopt:--strict-resolutions
```

(continues on next page)

(continued from previous page)

```
strict_resolutions = false

## worker labels: labels = k=v;group=enter
## type:string env:CFG_LABELS longopt:--labels
labels =

## use timestamps from SEI packet
## type:bool env:CFG_USE_TIME_FROM_SEI longopt:--use-time-from-sei
use_time_from_sei = false

## reader frame buffer size
## type:number env:CFG_FRAME_BUFFER_SIZE longopt:--frame-buffer-size
frame_buffer_size = 128

## skip count
## type:number env:CFG_SKIP_COUNT longopt:--skip-count
skip_count = 2

#------------------------------
[streamer]
#------------------------------
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999

## streamer url - how to access this worker on streamer_port
## type:string env:CFG_STREAMER_URL longopt:--streamer-url
url = 127.0.0.1:18999

## use tracks instead detects for streamer
## type:bool env:CFG_STREAMER_TRACKS longopt:--streamer-tracks
tracks = false

## use tracks with lastFrameId=currentFrameId (.tracks must be true)
## type:bool env:CFG_STREAMER_TRACKS_LAST longopt:--streamer-tracks-last
tracks_last = false

#------------------------------
[liveness]
#------------------------------
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk =

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm =

## liveness internal algo param
## type:double env:CFG_LIVENESS_INTERVAL longopt:--liveness-interval
interval = 1.000000
```

(continues on next page)

(continued from previous page)

```
## liveness internal algo param
## type:number env:CFG_LIVENESS_STDEV_CNT longopt:--liveness-stdev-cnt
stdev_cnt = 1

#------------------------------
[imotion]
#------------------------------
## use shared decoder for imotion (experimental)
## type:bool env:CFG_IMOTION_SHARED_DECODER longopt:--imotion-shared-decoder
shared_decoder = false

#------------------------------
[send]
#------------------------------
## posting faces threads
## type:number env:CFG_SEND_THREADS longopt:--send-threads
threads = 8

## posting faces maximum queue size
## type:number env:CFG_SEND_QUEUE_LIMIT longopt:--send-queue-limit
queue_limit = 256

#------------------------------
[tracker]
#------------------------------
## interpolate undetected bboxes in track
## type:bool env:CFG_TRACKER_INTERPOLATE_BBOXES longopt:--tracker-interpolate-bboxes
interpolate_bboxes = true

## max face miss duration, sec
## type:double env:CFG_TRACKER_MISS_INTERVAL longopt:--tracker-miss-interval
miss_interval = 1.000000

## overlap threshold
## type:double env:CFG_TRACKER_OVERLAP_THRESHOLD longopt:--tracker-overlap-threshold
overlap_threshold = 0.250000

#------------------------------
[models]
#------------------------------
## path to pedestrian detector fnk
## type:string env:CFG_MODELS_BODY_DETECTOR longopt:--models-body-detector
body_detector =

## path to detector fnk
## type:string env:CFG_MODELS_DETECTOR longopt:--models-detector
detector = /usr/share/findface-data/models/facedet/cheetah_fast.cpu.fnk

## path to quality fnk
## type:string env:CFG_MODELS_QUALITY longopt:--models-quality
quality = /usr/share/findface-data/models/faceattr/quality.v1.cpu.fnk
```

(continues on next page)

(continued from previous page)

```
## path to norm for quality fnk
## type:string env:CFG_MODELS_NORM_QUALITY longopt:--models-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/bee_fast.cpu.fnk

## path to norm_crop2x fnk, for face send
## type:string env:CFG_MODELS_NORM_CROP2X longopt:--models-norm-crop2x
norm_crop2x = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.cpu.fnk

## path to cache directory
## type:string env:CFG_MODELS_CACHE_DIR longopt:--models-cache-dir
cache_dir = /var/cache/findface/models_cache
```

If you opt for the GPU-accelerated package `findface-video-worker-gpu`, use the `/etc/findface-video-worker-gpu.ini` configuration file.

```
## cuda device number
## type:number env:CFG_DEVICE_NUMBER longopt:--device-number
device_number = 0

## read streams from file, do not use VideoManager
## type:string env:CFG_INPUT longopt:--input
input =

## exit on first finished job, only when --input specified
## type:bool env:CFG_EXIT_ON_FIRST_FINISHED longopt:--exit-on-first-finished
exit_on_first_finished = false

## batch size
## type:number env:CFG_BATCH_SIZE longopt:--batch-size
batch_size = 8

## http server port for metrics, 0=do not start server
## type:number env:CFG_METRICS_PORT longopt:--metrics-port
metrics_port = 0

## resize scale, 1=do not resize
## type:double env:CFG_RESIZE_SCALE longopt:--resize-scale
resize_scale = 1.000000

## maximum number of streams
## type:number env:CFG_CAPACITY longopt:--capacity
capacity = 30

## command to obtain videomanager's grpc ip:port
## type:string env:CFG_MGR_CMD longopt:--mgr-cmd
mgr_cmd =

## videomanager grpc ip:port
## type:string env:CFG_MGR_STATIC longopt:--mgr-static
mgr_static = 127.0.0.1:18811

## ntls server ip:port
```

(continues on next page)

(continued from previous page)

```
## type:string env:CFG_NTLS_ADDR longopt:--ntls-addr
ntls_addr = 127.0.0.1:3133

## debug: save faces to dir
## type:string env:CFG_SAVE_DIR longopt:--save-dir
save_dir =

## minimum face size
## type:number env:CFG_MIN_FACE_SIZE longopt:--min-face-size
min_face_size = 60

## preinit detector for specified resolutions: "640x480;1920x1080"
## type:string env:CFG_RESOLUTIONS longopt:--resolutions
resolutions =

## use `resolutions` as only possible values, others will rescale
## type:bool env:CFG_STRICT_RESOLUTIONS longopt:--strict-resolutions
strict_resolutions = false

## worker labels: labels = k=v;group=enter
## type:string env:CFG_LABELS longopt:--labels
labels =

## use timestamps from SEI packet
## type:bool env:CFG_USE_TIME_FROM_SEI longopt:--use-time-from-sei
use_time_from_sei = false

## reader frame buffer size
## type:number env:CFG_FRAME_BUFFER_SIZE longopt:--frame-buffer-size
frame_buffer_size = 128

## skip count
## type:number env:CFG_SKIP_COUNT longopt:--skip-count
skip_count = 0

#-----------------------------
[streamer]
#-----------------------------
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999

## streamer url - how to access this worker on streamer_port
## type:string env:CFG_STREAMER_URL longopt:--streamer-url
url = 127.0.0.1:18999

## use tracks instead detects for streamer
## type:bool env:CFG_STREAMER_TRACKS longopt:--streamer-tracks
tracks = false

## use tracks with lastFrameId=currentFrameId (.tracks must be true)
## type:bool env:CFG_STREAMER_TRACKS_LAST longopt:--streamer-tracks-last
```

(continues on next page)

(continued from previous page)

```
tracks_last = false

#-----------------------------
[liveness]
#-----------------------------
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk =

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm =

## liveness internal algo param
## type:double env:CFG_LIVENESS_INTERVAL longopt:--liveness-interval
interval = 1.000000

## liveness internal algo param
## type:number env:CFG_LIVENESS_STDEV_CNT longopt:--liveness-stdev-cnt
stdev_cnt = 1

#-----------------------------
[imotion]
#-----------------------------
## use shared decoder for imotion (experimental)
## type:bool env:CFG_IMOTION_SHARED_DECODER longopt:--imotion-shared-decoder
shared_decoder = false

#-----------------------------
[send]
#-----------------------------
## posting faces threads
## type:number env:CFG_SEND_THREADS longopt:--send-threads
threads = 8

## posting faces maximum queue size
## type:number env:CFG_SEND_QUEUE_LIMIT longopt:--send-queue-limit
queue_limit = 256

#-----------------------------
[tracker]
#-----------------------------
## interpolate undetected bboxes in track
## type:bool env:CFG_TRACKER_INTERPOLATE_BBOXES longopt:--tracker-interpolate-bboxes
interpolate_bboxes = true

## max face miss duration, sec
## type:double env:CFG_TRACKER_MISS_INTERVAL longopt:--tracker-miss-interval
miss_interval = 1.000000

## overlap threshold
## type:double env:CFG_TRACKER_OVERLAP_THRESHOLD longopt:--tracker-overlap-threshold
```

(continues on next page)

(continued from previous page)

```
overlap_threshold = 0.250000

#-----------------------------
[models]
#-----------------------------
## path to pedestrian detector fnk
## type:string env:CFG_MODELS_BODY_DETECTOR longopt:--models-body-detector
body_detector =

## path to detector fnk
## type:string env:CFG_MODELS_DETECTOR longopt:--models-detector
detector = /usr/share/findface-data/models/facedet/cheetah_fast.gpu.fnk

## path to quality fnk
## type:string env:CFG_MODELS_QUALITY longopt:--models-quality
quality = /usr/share/findface-data/models/faceattr/quality.v1.gpu.fnk

## path to norm for quality fnk
## type:string env:CFG_MODELS_NORM_QUALITY longopt:--models-norm-quality
norm_quality = /usr/share/findface-data/models/facenorm/bee_fast.gpu.fnk

## path to norm_crop2x fnk, for face send
## type:string env:CFG_MODELS_NORM_CROP2X longopt:--models-norm-crop2x
norm_crop2x = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.gpu.fnk

## path to cache directory
## type:string env:CFG_MODELS_CACHE_DIR longopt:--models-cache-dir
cache_dir = /var/cache/findface/models_cache

#-----------------------------
[video_decoder]
#-----------------------------
## decode video on cpu
## type:bool env:CFG_VIDEO_DECODER_CPU longopt:--video-decoder-cpu
cpu = false
```

When configuring `findface-video-worker` (on CPU/GPU), refer to the following parameters:

CPU	GPU	Description
batch_size		Post faces in batches of the given size.
capacity		Maximum number of video streams to be processed by findface-video-worker.
N/a	cpu	If necessary, decode video on CPU.
N/a	device_number	GPU device number to use.
exit_on_first_finished		(Only if input is specified) Exit on the first finished job.
input		Process streams from file, ignoring stream data from findface-video-manager.
labels		Labels used to allocate a video face detector instance to a certain group of cameras. See *Allocate findface-video-worker to Camera Group*.
liveness -> fnk		Path to the face *liveness* detector.
liveness -> norm		Path to the normalizer used in the *liveness* detector.
mgr-cmd		(Optional, instead of the mgr-static parameter) A command to obtain the IP address of the findface-video-manager host.
mgr-static		IP address of the findface-video-manager host to provide findface-video-worker with settings and the list of to-be-processed streams.
metrics_port		HTTP server port to send metrics. If 0, the metrics are not sent.
min_face_size		Minimum face size to be detected.
ntls-addr		IP address and port of the findface-ntls host.
resize_scale		Rescale video frames with the given coefficient.
resolutions		Preinitialize findface-video-worker to work with specified resolutions. Example: "640x480;1920x1080".
save_dir		(For debug) Save detected faces to the given directory.
streamer -> port, url		IP address and port to access the *video wall*.
use_time_from_sei		(For MPEG-2) Use SEI (supplemental enhancement information) timestamps.

Jobs

The findface-video-manager service provides findface-video-worker with a so-called job, a video processing task that contains configuration settings and stream data.

There are two job types:

- camera:

```
curl http://127.0.0.1:18810/job/ffsec-camera:22 | jq
{
  "id": "ffsec-camera:22",
  "enabled": true,
  "stream_url": "rtsp://ntech:!Ntech11@172.20.77.33:654/
↪000000010000babe0000accc8e9e3a58/live",
  "labels": {},
  "router_url": "http://127.0.0.1/video-detector/frame",
  "single_pass": false,
  "stream_settings": {
    "ffmpeg_params": [],
    "md_threshold": 0.002,
    "md_scale": 0.3,
    "fd_frame_height": -1,
    "uc_max_time_diff": 30,
    "uc_max_dup": 3,
    "uc_max_avg_shift": 10,
```

(continues on next page)

(continued from previous page)


```
    "det_period": 8,
    "realtime": false,
    "npersons": 4,
    "disable_drops": false,
    "tracker_threads": 4,
    "parse_sei": false,
    "image_arg": "photo",
    "additional_headers": [],
    "additional_body": [],
    "api_timeout": 15000,
    "api_ssl_verify": true,
    "post_uniq": true,
    "min_score": -2,
    "min_d_score": -1000,
    "realtime_dly": 500,
    "realtime_post_perm": false,
    "rot": "",
    "roi": "",
    "draw_track": false,
    "send_track": 0,
    "min_face_size": 0,
    "max_face_size": 0,
    "overall": true,
    "only_norm": false,
    "max_candidates": 0,
    "jpeg_quality": 95,
    "ffmpeg_format": ""
  },
  "stream_settings_gpu": {
    "play_speed": -1,
    "filter_min_quality": 0.45,
    "filter_min_face_size": 1,
    "filter_max_face_size": 8192,
    "normalized_only": false,
    "jpeg_quality": 95,
    "overall_only": false,
    "use_stream_timestamp": false,
    "ffmpeg_params": [],
    "router_timeout_ms": 15000,
    "router_verify_ssl": true,
    "router_headers": [
      "Authorization=Token 7db297c4107518b52b4e2195b72c5947"
    ],
    "router_body": [],
    "start_stream_timestamp": 0,
    "imotion_threshold": 0,
    "rot": "",
    "roi": "",
    "realtime_post_interval": 1,
    "realtime_post_every_interval": false,
    "ffmpeg_format": "",
    "disable_drops": false,
```


(continues on next page)

(continued from previous page)

```
    "router_full_frame_png": false,
    "router_disable_normalized": false,
    "crop_fullframe_rot": false,
    "realtime_post_first_immediately": false
  },
  "status": "INPROGRESS",
  "status_msg": "",
  "statistic": {
    "processed_duration": 5729.5728,
    "faces_posted": 43,
    "faces_failed": 0,
    "faces_not_posted": 0,
    "processing_fps": 15.589469,
    "frames_dropped": 0,
    "frames_processed": 87121,
    "frames_imotion_skipped": 0,
    "decoding_soft_errors": 0,
    "frame_width": 1920,
    "frame_height": 1200,
    "job_starts": 1
  },
  "restream_url": "ws://127.0.0.1:18999/stream/ffsec-camera:22",
  "shots_url": "http://127.0.0.1:18999/shot/ffsec-camera:22",
  "worker_id": "tevmenova-ntechlab_gpu_2a1fd5290195670b689d8e01e93b673c",
  "version": "bt1s8kjm51nv2hqc4qg0"
```

- video archive:

```
curl http://127.0.0.1:18810/job/ffsec-video-archive:1 | jq
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2094    0  2094    0     0  2044k      0 --:--:-- --:--:-- --:--:-- 2044k
{
  "id": "ffsec-video-archive:1",
  "enabled": true,
  "stream_url": "file:///var/lib/findface-security/uploads/videos/1.bin",
  "labels": {
    "camera_group_id": "1",
    "camera_id": ""
  },
  "router_url": "http://127.0.0.1/video-detector/frame",
  "single_pass": true,
  "stream_settings": {
    "ffmpeg_params": [],
    "md_threshold": 0.002,
    "md_scale": 0.3,
    "fd_frame_height": -1,
    "uc_max_time_diff": 30,
    "uc_max_dup": 3,
    "uc_max_avg_shift": 10,
    "det_period": 8,
    "realtime": false,
```

(continues on next page)

(continued from previous page)


```
    "npersons": 4,
    "disable_drops": false,
    "tracker_threads": 4,
    "parse_sei": false,
    "image_arg": "photo",
    "additional_headers": [],
    "additional_body": [],
    "api_timeout": 15000,
    "api_ssl_verify": true,
    "post_uniq": true,
    "min_score": -2,
    "min_d_score": -1000,
    "realtime_dly": 500,
    "realtime_post_perm": false,
    "rot": "",
    "roi": "",
    "draw_track": false,
    "send_track": 0,
    "min_face_size": 0,
    "max_face_size": 0,
    "overall": true,
    "only_norm": false,
    "max_candidates": 0,
    "jpeg_quality": 95,
    "ffmpeg_format": ""
  },
  "stream_settings_gpu": {
    "play_speed": -1,
    "filter_min_quality": 0.45,
    "filter_min_face_size": 1,
    "filter_max_face_size": 8192,
    "normalized_only": false,
    "jpeg_quality": 95,
    "overall_only": false,
    "use_stream_timestamp": false,
    "ffmpeg_params": [],
    "router_timeout_ms": 15000,
    "router_verify_ssl": true,
    "router_headers": [
      "Authorization=Token 7db297c4107518b52b4e2195b72c5947"
    ],
    "router_body": [],
    "start_stream_timestamp": 0,
    "imotion_threshold": 0,
    "rot": "",
    "roi": "",
    "realtime_post_interval": 1,
    "realtime_post_every_interval": false,
    "ffmpeg_format": "",
    "disable_drops": true,
    "router_full_frame_png": false,
    "router_disable_normalized": false,
```


(continues on next page)

(continued from previous page)


```
    "crop_fullframe_rot": false,
    "realtime_post_first_immediately": false
  },
  "status": "INPROGRESS",
  "status_msg": "",
  "statistic": {
    "processed_duration": 291,
    "faces_posted": 335,
    "faces_failed": 0,
    "faces_not_posted": 0,
    "processing_fps": 359.69928,
    "frames_dropped": 0,
    "frames_processed": 8731,
    "frames_imotion_skipped": 0,
    "decoding_soft_errors": 0,
    "frame_width": 1280,
    "frame_height": 720,
    "job_starts": 1
  },
  "restream_url": "ws://127.0.0.1:18999/stream/ffsec-video-archive:1",
  "shots_url": "http://127.0.0.1:18999/shot/ffsec-video-archive:1",
  "worker_id": "tevmenova-ntechlab_gpu_2a1fd5290195670b689d8e01e93b673c",
  "version": "bt1tm3bm51nv2hqc4qh0"
```


Each job has the following parameters:

- `id`: job id.
- `enabled`: active status.
- `stream_url`: URL/address of video stream/file to process.
- `labels`: tag(s) that will be used by the `findface-facerouter` component (`findface-security` in the standard FindFace configuration) to find processing directives for faces detected in this stream.
- `single_pass`: if true, disable restarting video processing upon error (by default, false).
- `router_url`: IP address and port of the `findface-facerouter` component (`findface-security` in the standard FindFace configuration) to receive detected faces from the `findface-video-worker` component for processing.
- `stream_settings`: used only for backward compatibility.
- `stream_settings_gpu`: video stream settings that duplicate *those* in the `/etc/findface-video-manager.conf` configuration file (while having priority over them).
- `status`: job status.
- `status_msg`: additional job status info.
- `statistic`: job progress statistics (progress duration, number of posted and not posted faces, processing fps, the number of processed and dropped frames, job start time, etc.).
- `worker_id`: id of the `findface-video-worker` instance executing the job.

findface-ntls

The findface-ntls service is to be installed on a designated host to verify the FindFace license. For verification purposes, findface-ntls uses one of the following sources:

- Ntech Lab global license center if you opt for the online licensing, direct or via a proxy server.
- USB dongle if you opt for the on-premise licensing.

Use the main web interface to manage findface-ntls:

- view the list of purchased features,
- view license limitations,
- upload a license file,
- view the list of currently active components.

The following components are licensable:

- findface-tarantool-server,
- findface-extraction-api,
- findface-video-manager,
- findface-video-worker.

Important: After connection between findface-ntls and a licensable component, or between findface-ntls and the global license server is broken, you will have 6 hours to restore it before the licensable components will be automatically stopped.

The findface-ntls configuration is done through a configuration file /etc/findface-ntls.cfg.

```
## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 127.0.0.1:3133
## Directory where license files are stored
## type:string env:CFG_LICENSE_DIR longopt:--license-dir
license_dir = /opt/ntech/license
## Use specified proxy (MUST support HTTP CONNECT method) to access global license␣
↪server (IP:PORT)
## type:string env:CFG_PROXY longopt:--proxy
proxy =
## Bind address for embedded UI (IP:PORT)
## type:string env:CFG_UI longopt:--ui
ui = 127.0.0.1:3185
```

When configuring findface-ntls, refer to the following parameters:

Parameter	Description
license_dir	Directory to store a license file.
listen	IP address from which licensable services access findface-ntls. To allow access from any IP address, use 0.0.0.0:3133.
proxy	(Optional) IP address and port of your proxy server.
ui	IP address from which accessing the findface-ntls web interface must originate. To allow access from any remote host, set "0.0.0.0".

findface-security

The `findface-security` component serves as a gateway to the FindFace core. It provides interaction between the FindFace Core and the web interface, the system functioning as a whole, HTTP and web socket (along with Django), database update, and *webhooks*.

The `findface-security` component also performs the functions of `findface-facerouter` (part of the FindFace Core), setting processing directives for detected faces. It accepts a face bbox and normalized image along with the original image and other data (for example, the detection date and time) from the `findface-video-worker` service and redirect them to `findface-sf-api` for further processing.

The `findface-security` configuration is done through the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py

#␣
↪==============================================================================
# FindFace Security configuration file
#␣
↪==============================================================================
#
# This config file is written in Python's syntax and interpreted at FindFace␣
↪Security
# service startup. You have to restart the service in order to apply changes.
#
# If you have any questions or suggestions, please contact us at␣
↪support@ntechlab.com


#␣
↪==============================================================================
# GENERAL SETTINGS
#␣
↪==============================================================================

# enables additional logs
DEBUG = False

# media files directory
MEDIA_ROOT = "/var/lib/findface-security/uploads"

# static files directory
STATIC_ROOT = "/var/lib/findface-security/static"

# language code
LANGUAGE_CODE = 'en-us'

# time zone
TIME_ZONE = 'UTC'

# Database is used by FindFace Security to store cameras,
# camera groups, watchlists and so on. Only PostgreSQL is supported.
DATABASES = {
```

(continues on next page)

(continued from previous page)

```
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'DISABLE_SERVER_SIDE_CURSORS': True,
        'NAME': 'ffsecurity',
        'PORT': 5439, 'USER': 'ntech', 'PASSWORD':
↪'ZnUqeWKMzT6T2Bj2G4gbFn7cxRSMcxS7'
    }
}

# Signature key for session encryption
# Use pwgen -sncy 50 1|tr """ "." to generate your own unique key
SECRET_KEY = '8b26839acde2633362bdb176e741a650'

#␣
↪==============================================================================
# FINDFACE SECURITY SETTINGS
#␣
↪==============================================================================

# SERVICE_EXTERNAL_ADDRESS is prioritized for FFSecurity webhooks and Genetec␣
↪plugin.
# EXTERNAL_ADDRESS is used instead if SERVICE_EXTERNAL_ADDRESS is not provided.
# You must provide either SERVICE_EXTERNAL_ADDRESS or EXTERNAL_ADDRESS in order
# to be able to work with FFSecurity webhooks and Genetec plugin.
SERVICE_EXTERNAL_ADDRESS = 'http://172.20.77.120'

# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via␣
↪external links.
EXTERNAL_ADDRESS = ''

# - Base FFSecurity settings -

# enable permissions system
ENABLE_ACL = True

FFSECURITY = {
    # findface-video-worker authorization token
    'VIDEO_DETECTOR_TOKEN': '3243a92b03c3411d4faa3cdd72f967b6',

    # base face matching confidence threshold
    'CONFIDENCE_THRESHOLD': 0.745,

    # episodes specific matching threshold that is used to join faces in an␣
↪episode
    'EPISODES_THRESHOLD': 0.689,

    # minimum face quality sufficient to add it to a dossier
    'MINIMUM_DOSSIER_QUALITY': 0.45,

    # do not save unmatched events (GDPR support)
    'IGNORE_UNMATCHED': False,
```

(continues on next page)

(continued from previous page)

```
    # blur all unmatched faces on the full frame of the matched event (GDPR␣
↪support)
    'BLUR_UNMATCHED_FACES': False,

    # full frame jpeg quality when `BLUR_UNMATCHED_FACES` is enabled
    'BLURRED_FULLFRAME_JPEG_QUALITY': 85,

    # matched events older than EVENTS_MAX_MATCHED_AGE will be automatically
    # deleted (every night at 1:17 am by default)
    'EVENTS_MAX_MATCHED_AGE': 30,

    # same as above but for unmatched events
    'EVENTS_MAX_UNMATCHED_AGE': 30,

    # same as EVENTS_MAX_MATCHED_AGE but for matched full frame images only␣
↪(thumbnails won't be deleted)
    'EVENTS_MAX_FULLFRAME_UNMATCHED_AGE': 30,

    # same as above but for unmatched full frame images only (thumbnails won't␣
↪be deleted)
    'EVENTS_MAX_FULLFRAME_MATCHED_AGE': 30,

    # same as above but for counter records
    'COUNTER_RECORDS_MAX_AGE': 30,

    # same as above but for person events (if no person events left in person,␣
↪it is deleted too)
    'PERSON_EVENTS_MAX_AGE': 90,

    # when closing episode, delete all events except the best episode event
    'EPISODE_KEEP_ONLY_BEST_EVENT': False,

    # NTLS licence server url
    'NTLS_HTTP_URL': 'http://127.0.0.1:3185',

    # findface-video-worker face posting address,
    # it must be set to either FFSecurity EXTERNAL_ADDRESS (by default)
    # or findface-facerouter url (in some specific cases)
    'ROUTER_URL': 'http://127.0.0.1',

    # send serialized dossiers, dossier-lists, camera and camera groups in␣
↪webhooks
    'VERBOSE_WEBHOOKS': False,

    # jpeg quality used when saving thumbnails
    'THUMBNAIL_JPEG_QUALITY': 75,

    # FFServer services urls
    'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
    'SF_API_ADDRESS': 'http://127.0.0.1:18411',
    'FFCOUNTER_ADDRESS': 'http://127.0.0.1:18300',
    'LIVENESS_API_ADDRESS': 'http://127.0.0.1:18301',
```

(continues on next page)

(continued from previous page)

```
    # upload video archives to this path, it differs from media root and
    # you have to change alias in corresponding nginx location also (/videos/)
    'VIDEO_ARCHIVE_UPLOAD_PATH': '/var/lib/findface-security/uploads/videos/',

    # additional events features.
    # make sure that corresponding extractors
    # are licensed and enabled at findface-extraction-api config file.
    # available features are: gender, age, emotions, beard, glasses, medmask.
    'EVENTS_FEATURES': [],

    # feature specific confidence thresholds
    'LIVENESS_THRESHOLD': 0.85,
    'EMOTIONS_THRESHOLD': 0.25,
    'BEARD_THRESHOLD': 0.7,

    # counters full frame saving options:
    # `always` - save always
    # `detect` - save only if faces or silhouettes have been detected
    # `never` - never save full frames
    'COUNTERS_SAVE_FULLFRAME': 'always',
    'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
    'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,

    # max camera frames_dropped percent
    'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
    # max camera faces_failed percent
    'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},

    # -- Persons configuration --

    # rrule (recurrence rule) for scheduling persons clusterization
    # WARNING: all scheduling works with UTC time and NOT aware of any timezone
    'PERSONS_CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;
↪BYHOUR=0;BYMINUTE=0',

    # face to person matching confidence threshold
    'PERSONS_CONFIDENCE_THRESHOLD': 0.745,

    # minimum required face quality for person creation
    'PERSON_EVENT_MIN_QUALITY': 0.45,
    # minimum required number events in episode for person creation
    'PERSON_EVENT_MIN_EPISODE_EVENTS': 1,

    # maximum concurrent video manager jobs for video archives processing
    'MAX_VIDEO_ARCHIVE_JOBS': 3,

    # reports image saving options
    'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
    'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
    'REPORT_FULLFRAME_JPEG_QUALITY': 75,
    'REPORT_FULLFRAME_MAX_HEIGHT': 250,
```

(continues on next page)

(continued from previous page)

```
    # -- Startup tests --

    # required services availability test
    'SERVICES_AVAILABILITY_TEST': True,

    # enable saving audit logs to PostgresSQL
    'ENABLE_AUDIT_LOGS': True,

    # -- FFSecurity Onvif --

    # auth credentials for ffsecurity_onvif
    # ONVIF_CREDENTIALS = [
    #     {
    #         "hostnames": ["192.168.1.64",
↪"2a00:1370:8117:ab87:a614:37ff:fe49:2683"],
    #         "login": "admin",
    #         "password": "admin123"
    #     }
    # ],
    'ONVIF_CREDENTIALS': {},
    # list of all hostnames that will be ignored during Onvif service discovery
    # ONVIF_IGNORE_LIST = ["192.168.1.217"],
    'ONVIF_IGNORE_LIST': [],

    # -- Optional parameters --

    # Edit CUSTOM_FIELDS->dossier_meta section to customize dossier content.
    # Below is an example for integration FindFace Security with Sigur.

    # Edit CUSTOM_FIELDS->dossier_face section to customize dossier face␣
↪content.
    # Below is an example with every field type possible.

    # 'CUSTOM_FIELDS': {
    #     'dossier_meta': {
    #         'items': [
    #             {
    #                 'name': 'personid',
    #                 'default': '',
    #                 'label': 'PersonID',
    #                 'display': ['list', 'form'],
    #                 'description': 'Sigur person ID'
    #             },
    #             {
    #                 'name': 'firstname',
    #                 'default': '',
    #                 'label': 'First Name',
    #                 'display': ['list', 'form'],
    #                 'description': 'Sigur first name'
    #             },
```

(continues on next page)

(continued from previous page)

```
#             {
#                 'name': 'lastname',
#                 'default': '',
#                 'label': 'Last Name',
#                 'display': ['list', 'form'],
#                 'description': 'Sigur last name'
#             },
#             {
#                 'name': 'version',
#                 'default': '',
#                 'label': 'Version',
#                 'display': ['list', 'form'],
#                 'description': 'Sigur photo version'
#             }
#         ],
#         'filters': [
#             {
#                 'name': 'personid',
#                 'label': 'Sigur person ID filter',
#                 'field': 'personid'
#             }
#         ]
#     },
#     'dossier_face': {
#         'items': [
#             {
#                 "field_name": "tag_name_1",
#                 "type": "string",
#                 "default": "change_me"
#             },
#             {
#                 "field_name": "tag_name_2",
#                 "type": "uint",
#                 "default": 123
#             },
#             {
#                 "field_name": "tag_name_3",
#                 "type": "bool",
#                 "default": True
#             },
#         ]
#     }
# },

# maximum event age in seconds than could be added to an episode.
# 'EPISODE_SEARCH_INTERVAL': 60,
# If none of these events matched, new episode is created.

# maximum episode duration (episode is closed after)
# 'EPISODE_MAX_DURATION': 300,

# if no new event added to an episode during this timeout, episode will be␣
```

(continues on next page)

(continued from previous page)

```
↪closed.
    # 'EPISODE_EVENT_TIMEOUT': 30,

    # maximum created thumbnail width
    # 'THUMBNAIL_MAX_WIDTH': 320,

    # url of the backend which is used for social network search.
    # contact support for additional information.
    # 'SOCIAL_BACKEND': None,

    # additional social backend headers.
    # 'SOCIAL_HEADERS': {},

    # unacknowledged events notification interval
    # 'UNACKNOWLEDGED_NOTIFY_INTERVAL': 1,

    # set to True to run all media requests (photos, attachments) through the
    # django application for acl checks.
    # enabling this setting slightly increases security but
    # has severe negative effects on performance.
    # you will also have to mark /uploads/ location as 'internal' in nginx config
    #
    # 'OVERPROTECT_MEDIA': False,
}

# - FindFace Security authorization configuration dictionary -

FFSECURITY_AUTH_CONFIG = {
    # available options: face, password, face_and_password, face_or_password
    'AUTH_TYPE': 'face_or_password',
    'FACE_AUTH_CONFIDENCE': 0.745,
    # 180 days by default
    'MAXIMUM_SESSION_LENGTH': 15552000,
    # session renew works only with face or face_or_password authorization type
    'NEED_SESSION_RENEW': False,
    'RENEW_SESSION_INTERVAL': 0,
    'MAXIMUM_RENEW_ATTEMPTS': 2,
}

# - FindFace Security user interface configuration dictionary -

FFSECURITY_UI_CONFIG = {
    "event": {
        "features": {
            "f_gender_class": ["male", "female"],
            "age": {
                "f_age_gte": "",
                "f_age_lte": ""
            },
            "f_emotions_class": ["angry", "disgust", "fear", "happy", "sad",
↪"surprise", "neutral"],
            "f_glasses_class": ["none", "eye", "sun"],
```

(continues on next page)

(continued from previous page)

```
            "f_beard_class": ["none", "beard"],
            "f_liveness_class": ["real", "fake"],
            "f_medmask_class": ["none", "incorrect", "correct"],
        }
    },

    # Adjustable confidence threshold presets for face matching.
    # Please consult with our support before changing.
    "confidence_display": [
        {"confidence": 0.00, "color": "#000000", "label": {"ru": " ", "en":
↪"Very Low"}},
        {"confidence": 0.65, "color": "#FF0300", "label": {"ru": "", "en": "Low
↪"}},
        {"confidence": 0.70, "color": "#FFB700", "label": {"ru": "", "en":
↪"Slightly Low"}},
        {"confidence": 0.72, "color": "#B8FA00", "label": {"ru": "", "en":
↪"Moderate"}},
        {"confidence": 0.75, "color": "#7EFF2B", "label": {"ru": "", "en":
↪"Slightly High"}},
        {"confidence": 0.80, "color": "#4DFF60", "label": {"ru": "", "en":
↪"High"}},
        {"confidence": 0.85, "color": "#1DFF96", "label": {"ru": " ", "en":
↪"Very High"}},
    ]
}

# -- ASGI-server configuration --
# consult support before changing these settings.

# per worker thread pool size.
ASGI_THREADS = 32

UVICORN_SETTINGS = {
    # worker processes count, 'auto' sets it to logical cpu count
    'workers': 'auto',
    'host': 'localhost',
    'port': 8002,
    # websocket worker processes count,
    # 'auto' sets it to logical cpu count, but not more than 8.
    'ws-workers': 'auto',
    'ws-host': 'localhost',
    'ws-port': 8003,
}

# disable unused services to increase
# overall system performance in some cases.
SERVICES = {
    "ffsecurity": {
        "episodes": True,
        "webhooks": True,
        # use queue manager to prevent drops of video archive events
        "video_archive_events_manager": True,
```

(continues on next page)

(continued from previous page)

```
        "persons": False,
    }
}

# -- Other settings --

# The number of threads in the night clusterization.
# Not recommended values are greater than the number of cores in the processor.
# Consult with support before changing this value.
NUMPY_OMP_NUM_THREADS = 'auto'


#␣
↪==============================================================================
# FINDFACE SECURITY PLUGINS
#␣
↪==============================================================================
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.

# =============== Axxon ================
# INSTALLED_APPS.append('ffsecurity_axxon')

# AXXON = [
#     {
#         'name': 'server_name',
#         'api': 'http://example.com/',
#         'rtsp': 'rtsp://example.com:554/',
#         'user': 'user',
#         'password': 'password',
#     }
# ]

# FFSECURITY_UI_CONFIG['dossier'] = {
#    'video': True,
# }


# =============== Genetec ================
# INSTALLED_APPS.append('ffsecurity_genetec')


# ================ Sova ==================
# INSTALLED_APPS.append('ffsecurity_sova')


# ================ Sigur =================
# keep in mind, that SIGUR plugin also uses CUSTOM_FIELDS and THUMBNAIL_MAX_
↪WIDTH settings
# INSTALLED_APPS.append('ffsecurity_sigur')
# SIGUR = {
#     'LOGIN': 'admin',
```

(continues on next page)

(continued from previous page)

```
#     'PASSWORD': 'admin',
#     'MF_SELECTOR': 'biggest',  # what to do with several faces in sigur person␣
↪photo; allowed ['biggest', 'reject']
#     'ONLY_RT_EVENTS': True,  # only events with bs_type == realtime,
#     'EVENT_DELAY': 0.004 # minimum time between two events of same person in␣
↪seconds. If interval between two events with same person is less, than this␣
↪value, second event will be dropped
# }


# ======= CryptoPRO authentication =======
# INSTALLED_APPS.append('ffsecurity_cproauth')
# REST_FRAMEWORK['DEFAULT_AUTHENTICATION_CLASSES'] = [
#     'ffsecurity.auth.TokenAuthentication',
#     'ffsecurity_cproauth.auth.CryptoProOrTokenAuthentication'
# ]


# ========== DossierLists sync ===========
# INSTALLED_APPS.append('ffsecurity_sync')

# token must be identical on master and slave
# use pwgen -s 64 1
# SYNC_TOKEN = 'change_me'
# rrule that defines sync schedule
# SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=4;BYMINUTE=0'
# if True synchronization will occur on FindFace Security startup and restart
# SYNC_AT_STARTUP = False
# if True synchronization will occur immediately after creating␣
↪synchronization for dossier list
# SYNC_AT_CREATION = False


# ============== Puppeteer ===============
# INSTALLED_APPS.append('ffsecurity_puppeteer')

# PUPPETEER_CONFIG = {
#     'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600,        # maximum lifetime of␣
↪search results not saved involuntarily
#     'REMOTE_MONITORING_SYNC_INTERVAL': 600,          # monitoring data␣
↪synchronization interval, seconds
#     'ENABLE_DAILY_SEARCH': True,                     # daily search activation␣
↪(default False)
#     'DAILY_SEARCH_PUSH_HOUR': 2,                     # daily search dossiers␣
↪synchronization hour
#     'DAILY_SEARCH_PULL_HOUR': 6,                     # hour in which results␣
↪of daily search will be obtained
#     'puppets': [
#         {
#             'id': 'first_puppet',                   # puppet ID
#             'url': 'http://1.1.1.1:8010/',          # puppet URL
#             'token': 'first_puppet_token',          # use pwgen -s 64 1␣
```

(continues on next page)

(continued from previous page)

```
↪(should match the token in puppet)
#             'facen_model': 'jackfruit_480'           # face model in puppet
#         },
#         {
#             'id': 'second_puppet',
#             'url': 'http://1.1.1.1:8010/',
#             'token': 'second_puppet_token',
#
#              # if remote installation has a different face model than the␣
↪one used in FFSecurity -
#              # you need to specify its name and ExtractionAPI URL where the␣
↪corresponding face model is specified
#             'facen_model': 'grapefruit_480',
#             'extractor': 'http://127.0.0.1:18667',
#         },
#     ]
# }
#
# ================= Vns ==================
# A plugin for using FindFace Security as a puppeteer server
# INSTALLED_APPS.append('ffsecurity_vns')

# VNS_CONFIG = {
#     'USERS': {
#         'user1': 'token1',
#         'user2': 'token2'
#     },
#     'MONITORING_THRESHOLD': 0.75,
#     'DAILY': {
#         'ENABLED': False,
#         'THRESHOLD': 0.75,
#         'START_TIME': "00:00:00"
#     }
# }
```

When configuring `findface-security`, refer to the following parameters:

Parameter	Description
`BEARD_THRESHOLD`	The presence of a beard on a face is determined with a certain level of confidence. De
`CONFIDENCE_THRESHOLD`	Face similarity threshold for verification in events.
`COUNTERS_FULLFRAME_JPEG_QUALITY`	JPEG quality of full frames in counters.
`COUNTER_RECORDS_MAX_AGE`	The age of counter records at which they are automatically purged from the database.
`COUNTERS_SAVE_FULLFRAME`	Saving options of full frames in counters: `always`, `detect` - only save if faces or silho
`COUNTERS_THUMBNAIL_JPEG_QUALITY`	JPEG quality of thumbnails in counters.
`CUSTOM_FIELDS`	Uncomment and modify this section to customize dossier content. See *Dossier Custo*
`DATABASES` (section)	Database settings. Fill in as such: `'PORT': 5439, 'USER': 'ntech', 'PASSWORD`
`EMOTIONS_THRESHOLD`	Emotions are determined with a certain level of confidence. Depending on the confide
`EPISODE_EVENT_TIMEOUT`	The maximum time in seconds since the last event has been added to an episode. Afte
`EPISODE_KEEP_ONLY_BEST_EVENT`	When closing an episode, delete all events in it except the best event.
`EPISODE_MAX_DURATION`	The maximum episode duration in seconds. After this time, an episode automatically
`EPISODE_SEARCH_INTERVAL`	The period of time preceding an event, within which the system searches the biometric

Parameter	Description
EPISODES_THRESHOLD	Face similarity threshold for verification in episodes.
EVENTS_FEATURES	If you enabled recognition models in the /etc/findface-extraction-api.ini co
EVENTS_MAX_FULLFRAME_MATCHED_AGE	Same as EVENTS_MAX_MATCHED_AGE but only for full frames.
EVENTS_MAX_FULLFRAME_UNMATCHED_AGE	Same as EVENTS_MAX_UNMATCHED_AGE but only for full frames.
EVENTS_MAX_MATCHED_AGE	The age of matched events at which they are automatically purged from the database.
EVENTS_MAX_UNMATCHED_AGE	The age of unmatched events at which they are automatically purged from the database
EXTERNAL_ADDRESS	External IP address or URL that will be used to access the FindFace web interface.
EXTRACTION_API	IP address of the findface-extraction-api host.
IGNORE_UNMATCHED	Disable logging events for faces which have no match in the dossiers (negative verifica
LIVENESS_THRESHOLD	The liveness detector will estimate a face liveness with a certain level of confidence. D
MAX_CAMERA_DROPPED_FRAMES	Color representation of camera statuses (yellow and red), based on the percentage of d
MAX_CAMERA_FAILED_FACES	Color representation of camera statuses (yellow and red), based on the percentage of fa
MAX_VIDEO_ARCHIVE_JOBS	Maximum concurrent findface-video-manager jobs for video archive processing.
MINIMUM_DOSSIER_QUALITY	Minimum quality of a face in a dossier photo. Photos containing faces of worse quality
NTLS_HTTP_URL	IP address of the findface-ntls host.
PERSONS_CLUSTERIZATION_SCHEDULE	Recurrence rule (RRULE) for scheduling person clusterization.
PERSONS_CONFIDENCE_THRESHOLD	Confidence threshold to match a face to a person.
PERSON_EVENT_MIN_EPISODE_EVENTS	Minimum number of events in episodes used in person clusterization.
PERSON_EVENT_MIN_QUALITY	Minimum quality of faces used in person clusterization. Do not modify the default val
PERSON_EVENTS_MAX_AGE	The age of person events at which they are automatically purged from the database.
ROUTER_URL	IP address of the findface-security host that will receive detected faces from the f
SERVICE_EXTERNAL_ADDRESS	(Optional) IP address prioritized for webhooks and Genetec integration.
SF_API_ADDRESS	IP address of the findface-sf-api host.
THUMBNAIL_JPEG_QUALITY	Thumbnail JPEG quality.
THUMBNAIL_MAX_WIDTH	Maximum thumbnail width.
VERBOSE_WEBHOOKS	Send serialized dossiers, watch lists, cameras, and camera groups in *webhooks*.
VIDEO_DETECTOR_TOKEN	To authorize the video face detection module, come up with a token and specify it here
VIDEO_MANAGER_ADDRESS	IP address of the findface-video-manager host.

findface-facerouter

Important: The findface-facerouter is not included in the FindFace Security standard configuration. Use it for integration if necessary. See *Custom Plugins*.

The findface-facerouter service sets processing directives for faces detected in video. The directives are set through custom plugins.

The findface-facerouter service accepts a face bbox and normalized image along with the original image and other data (for example, the detection date and time) from the findface-video-worker service. In general, findface-facerouter allows you to apply arbitrary face processing directives, including directly sending faces to a partner application. In the basic configuration, findface-facerouter is pre-configured to redirect faces to findface-sf-api for further processing, but you will still have to set processing directives by creating a plugin.

Functionality:

- sets processing directives for faces detected in video,
- redirects faces detected in video to findface-sf-api or other service (including a third-party application) for further processing.

The findface-facerouter configuration is done through a configuration file /etc/findface-facerouter.py.

```
# main.py options:
# debug                          = False
## debug - debug mode
# detector                       = ''
## detector - Detector to use if client fails to provide normalized face
## (nnd).Use "nnd" if you need to detect faces in such requests. Empty value
## rejects requests without normalized.
# host                           = ''
## host - host to listen
# port                           = 18820
## port - port to listen
# prometheus_timing_buckets      = None
## prometheus_timing_buckets - prometheus histogram buckets (python list of
## numbers, e.g. [1,2,3])
# sfapi_url                      = 'http://localhost:18411'
## sfapi_url - SF-API URL
# version                        = False
## version - print version
# plugin_dir.py options:
# plugin_dir                     = ''
## plugin_dir - Plugin directory for plugin_source='dir'
# abstract_define.py options:
# plugin_source                  = 'dir'
## plugin_source - Plugin source (dir)
# log.py options:
# log_file_max_size              = 100000000
## log_file_max_size - max size of log files before rollover
# log_file_num_backups           = 10
## log_file_num_backups - number of log files to keep
# log_file_prefix                = None
## log_file_prefix - Path prefix for log files. Note that if you are running
## multiple tornado processes, log_file_prefix must be different for each of
## them (e.g. include the port number)
# log_rotate_interval            = 1
## log_rotate_interval - The interval value of timed rotating
# log_rotate_mode                = 'size'
## log_rotate_mode - The mode of rotating files(time or size)
# log_rotate_when                = 'midnight'
## log_rotate_when - specify the type of TimedRotatingFileHandler interval other
## options:('S', 'M', 'H', 'D', 'W0'-'W6')
# log_to_stderr                  = None
## log_to_stderr - Send log output to stderr (colorized if possible). By default
## use stderr if --log_file_prefix is not set and no other logging is
## configured.
# logging                        = 'info'
## logging - Set the Python log level. If 'none', tornado won't touch the
## logging configuration.
```

When configuring `findface-facerouter`, refer to the following parameters:

Parameter	Description
`sfapi_url`	IP address and port of the `findface-sf-api` host.
`plugin_dir`	Directory with plugins to define face processing directives.

1.7.2 Installation File

FindFace installation configuration is automatically saved to a file `/tmp/<findface-installer-*>.json`. You can edit this file and use it to install FindFace on other hosts without having to answer the installation questions again.

Tip: See *Deploy from Console Installer* to learn more about the FindFace installer.

Important: Be sure to remove fields `*.config`, `exp_ip`, and `int_ip` before installing FindFace on a host with a different IP address.

Here is an example of the installation file:

```
{
  "product": "security",
  "python3-pil.variant": "avx2",
  "ignore_nolicense": true,
  "type": "stand-alone",
  "ext_ip.bind": "0.0.0.0",
  "ext_ip.advertised": "172.20.77.19",
  "inter_ip.advertised": "127.0.0.1",
  "inter_ip.bind": "127.0.0.1",
  "components": [
    "findface-data",
    "memcached",
    "etcd",
    "redis",
    "postgresql",
    "pgbouncer",
    "jq",
    "python3-pil",
    "findface-ntls",
    "findface-extraction-api",
    "findface-sf-api",
    "findface-counter",
    "findface-liveness-api",
    "findface-upload",
    "findface-video-manager",
    "findface-video-worker",
    "findface-security",
    "findface-tarantool-server"
  ],
  "tnt_instances": 4,
  "findface-video-worker.variant": "gpu",
  "findface-extraction-api.variant": "gpu",
  "findface-data.models": [
    "./findface-data-age.v1-cpu_3.0.0_all.deb",
    "./findface-data-age.v1-gpu_3.0.0_all.deb",
    "./findface-data-beard.v0-cpu_3.0.0_all.deb",
    "./findface-data-beard.v0-gpu_3.0.0_all.deb",
    "./findface-data-detector-cheetah-cpu_3.0.0_all.deb",
    "./findface-data-detector-cheetah-fast-cpu_3.0.0_all.deb",
```

(continues on next page)

(continued from previous page)

```
    "./findface-data-detector-cheetah-fast-gpu_3.0.0_all.deb",
    "./findface-data-detector-cheetah-gpu_3.0.0_all.deb",
    "./findface-data-detector-mtcnn-cpu_3.0.0_all.deb",
    "./findface-data-emotions.v1-cpu_3.0.0_all.deb",
    "./findface-data-emotions.v1-gpu_3.0.0_all.deb",
    "./findface-data-gender.v2-cpu_3.0.0_all.deb",
    "./findface-data-gender.v2-gpu_3.0.0_all.deb",
    "./findface-data-glasses3.v0-cpu_3.0.0_all.deb",
    "./findface-data-glasses3.v0-gpu_3.0.0_all.deb",
    "./findface-data-ifruit-320-cpu_3.0.0_all.deb",
    "./findface-data-ifruit-320-gpu_3.0.0_all.deb",
    "./findface-data-jackfruit-160-cpu_3.0.0_all.deb",
    "./findface-data-jackfruit-160-gpu_3.0.0_all.deb",
    "./findface-data-jackfruit-320-cpu_3.0.0_all.deb",
    "./findface-data-jackfruit-320-gpu_3.0.0_all.deb",
    "./findface-data-jackfruit-480-cpu_3.0.0_all.deb",
    "./findface-data-jackfruit-480-gpu_3.0.0_all.deb",
    "./findface-data-liveness.alleyn.v2-cpu_3.0.0_all.deb",
    "./findface-data-liveness.alleyn.v2-gpu_3.0.0_all.deb",
    "./findface-data-medmask3.v2-cpu_3.0.0_all.deb",
    "./findface-data-medmask3.v2-gpu_3.0.0_all.deb",
    "./findface-data-normalization-ant.v2-cpu_3.0.0_all.deb",
    "./findface-data-normalization-ant.v2-gpu_3.0.0_all.deb",
    "./findface-data-normalization-bee-fast-cpu_3.0.0_all.deb",
    "./findface-data-normalization-bee-fast-gpu_3.0.0_all.deb",
    "./findface-data-normalization-bee.v2-cpu_3.0.0_all.deb",
    "./findface-data-normalization-bee.v2-gpu_3.0.0_all.deb",
    "./findface-data-normalization-crop2x.v2-maxsize400-cpu_3.0.0_all.deb",
    "./findface-data-normalization-crop2x.v2-maxsize400-gpu_3.0.0_all.deb",
    "./findface-data-pedet-edie-rc2-cpu_3.0.0_all.deb",
    "./findface-data-pedet-edie-rc2-gpu_3.0.0_all.deb",
    "./findface-data-quality.v1-cpu_3.0.0_all.deb",
    "./findface-data-quality.v1-gpu_3.0.0_all.deb"
  ],
  "facen_model": "face/jackfruit_480.gpu.fnk",
  "findface-extraction-api.config": {
    "listen": "127.0.0.1:18666",
    "license_ntls_server": "127.0.0.1:3133",
    "detectors": {
      "instances": 1,
      "quality_estimator": true
    },
    "normalizers": {
      "instances": 1
    },
    "extractors": {
      "instances": 1,
      "models": {
        "gender": "",
        "age": "",
        "emotions": "",
        "face": "face/jackfruit_480.gpu.fnk",
```

(continues on next page)

(continued from previous page)

```
        "quality": "faceattr/quality.v1.gpu.fnk"
      }
    }
  },
  "findface-video-manager.config": {
    "listen": "127.0.0.1:18810",
    "master": {
      "self_url": "127.0.0.1:18811",
      "self_url_http": "127.0.0.1:18810"
    },
    "rpc": {
      "listen": "127.0.0.1:18811"
    },
    "ntls": {
      "enabled": false,
      "url": "http://127.0.0.1:3185/"
    }
  },
  "findface-facerouter.config": {
    "plugin_source": "dir",
    "host": "127.0.0.1",
    "port": "18820",
    "sfapi_url": "http://127.0.0.1:18411",
    "plugin_dir": "/etc/findface-facerouter-plugins"
  },
  "memcached.config": {
    "max_memory": 1024,
    "item_size": 16,
    "listen_host": "127.0.0.1"
  },
  "findface-video-worker.config": {
    "FKVD_MGR_ADDR": "127.0.0.1:18811",
    "FKVD_NTLS_ADDR": "127.0.0.1:3133",
    "FKVD_WRK_CAP": "30",
    "streamer": [
      "port = 18999",
      "url = 127.0.0.1:18999"
    ]
  },
  "findface-ntls.config": {
    "NTLS_LISTEN": "127.0.0.1:3133",
    "NTLS_LISTEN_UI": "127.0.0.1:3185",
    "NTLS_LICENSE_DIR": "/opt/ntech/license"
  },
  "findface-tarantool-server.config": {
    "shard-001": {
      "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-001",
      "TNT_LISTEN": "127.0.0.1:32001",
      "TNT_FF_LISTEN_IP": "127.0.0.1",
      "TNT_FF_LISTEN_PORT": "8101",
      "TNT_FF_NTLS": "127.0.0.1:3133",
      "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
```

(continues on next page)

(continued from previous page)


```
      "TNT_META_SCHEME": "meta_scheme",
      "TNT_META_INDEXES": "meta_indexes"
    },
    "shard-002": {
      "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-002",
      "TNT_LISTEN": "127.0.0.1:32002",
      "TNT_FF_LISTEN_IP": "127.0.0.1",
      "TNT_FF_LISTEN_PORT": "8102",
      "TNT_FF_NTLS": "127.0.0.1:3133",
      "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
      "TNT_META_SCHEME": "meta_scheme",
      "TNT_META_INDEXES": "meta_indexes"
    },
    "shard-003": {
      "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-003",
      "TNT_LISTEN": "127.0.0.1:32003",
      "TNT_FF_LISTEN_IP": "127.0.0.1",
      "TNT_FF_LISTEN_PORT": "8103",
      "TNT_FF_NTLS": "127.0.0.1:3133",
      "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
      "TNT_META_SCHEME": "meta_scheme",
      "TNT_META_INDEXES": "meta_indexes"
    },
    "shard-004": {
      "TNT_DATA_DIR": "/opt/ntech/var/lib/tarantool/shard-004",
      "TNT_LISTEN": "127.0.0.1:32004",
      "TNT_FF_LISTEN_IP": "127.0.0.1",
      "TNT_FF_LISTEN_PORT": "8104",
      "TNT_FF_NTLS": "127.0.0.1:3133",
      "TNT_EXTRA_LUA": "\ndofile(\"/etc/findface-security/tnt_schema.lua\")\n",
      "TNT_META_SCHEME": "meta_scheme",
      "TNT_META_INDEXES": "meta_indexes"
    }
  },
  "findface-sf-api.config": {
    "listen": "127.0.0.1:18411",
    "extraction-api": {
      "extraction-api": "http://127.0.0.1:18666"
    },
    "storage-api": {
      "shards": [
        {
          "master": "http://127.0.0.1:8101/v2/",
          "slave": ""
        },
        {
          "master": "http://127.0.0.1:8102/v2/",
          "slave": ""
        },
        {
          "master": "http://127.0.0.1:8103/v2/",
          "slave": ""
```


(continues on next page)

(continued from previous page)

```
        },
        {
          "master": "http://127.0.0.1:8104/v2/",
          "slave": ""
        }
      ]
    },
    "limits": {
      "allow-return-facen": true
    }
  },
  "findface-counter.config": {
    "listen": "127.0.0.1:18300",
    "database": {
      "connection_string": "dbname=ffcounter host=/var/run/postgresql sslmode=disable"
    }
  },
  "pgbouncer.config": {
    "ntech_password": "hva1xogfMUO3Mt6WdS4zyOXcaHs6zHOG"
  },
  "postgresql.config": {
    "ntech_password": "hva1xogfMUO3Mt6WdS4zyOXcaHs6zHOG"
  },
  "findface-security.config": {
    "SERVICE_EXTERNAL_ADDRESS": "http://172.20.77.19",
    "FFSECURITY": {
      "ROUTER_URL": "http://127.0.0.1"
    },
    "ntech_password": "hva1xogfMUO3Mt6WdS4zyOXcaHs6zHOG"
  }
}
```

1.7.3 Neural Network Models

Here you can see a summary for neural network models created by our Lab and used in FindFace:

Important: The default face biometrics model upon a clean install is `jackfruit_480`.

Model	Type
face/jackfruit_160	Face biometrics
face/jackfruit_320	
face/jackfruit_480	
face/ifruit_320	
faceattr/age.v1	Age recognition
faceattr/beard.v0	Beard recognition
faceattr/emotions.v1	Emotions recognition
faceattr/gender.v2	Gender recognition
faceattr/glasses3.v0	Glasses recognition
faceattr/medmask3.v2	Face mask detection
faceattr/liveness.alleyn.v2	Liveness standalone service
faceattr/quality.v1	Face quality estimation

Note: The CPU and GPU benchmark setup is the following:

- CPU - Intel® Core™ i7-6700 CPU @ 3.4GHz 4 cores
- GPU - GeForce GTX 1080 Ti

Benchmark results for CPU-accelerated models:

Note: Features in the benchmark result below: faceattr/age.v1.cpu, faceattr/beard.v0.cpu, faceattr/emotions.v1.cpu, faceattr/gender.v2.cpu, faceattr/glasses3.v0.cpu.

Number of instances	1	2	3	4	5
Model	RAM, Mb	RAM, Mb	RAM, Mb	RAM, Mb	RAM, Mb
face/ifruit_320.cpu	4300	5020	5698	7606	8279
face/ifruit_320.cpu + features	6254	7019	7725	8597	9220
face/ifruit_320.cpu + features + edie_rc2.cpu.fnk	6513	7454	9165	9685	10777
features	3034	3799	4460	5091	5774
features + medmask3.v2.cpu	3191	3942	4699	5313	6003
face/ifruit_320.cpu + medmask3.v2.cpu	4447	5108	5809	7642	8262
face/ifruit_320.cpu + edie_rc2.cpu.fnk	4583	5557	7746	8721	9592

Benchmark results for GPU-accelerated models:

Note: Features in the benchmark result below: faceattr/age.v1.gpu.fnk, faceattr/beard.v0.gpu.fnk, faceattr/emotions.v1.gpu.fnk, faceattr/gender.v2.gpu.fnk, faceattr/glasses3.v0.gpu.fnk.

Model	GPU RAM, Mb
face/ifruit_320.gpu.fnk	3169
face/ifruit_320.gpu.fnk + edie_rc2.gpu.fnk	4243
face/ifruit_320.gpu.fnk + features	3901
face/ifruit_320.gpu.fnk + features + edie_rc2.gpu.fnk	5102
face/ifruit_320.gpu.fnk + faceattr/medmask3.v2.gpu.fnk	3313
features	2345
features + faceattr/medmask3.v2.gpu.fnk	2617

1.7.4 FindFace Data Storages

In this section:

- *List of Storages*
- *Biometric Database Galleries*

List of Storages

FindFace uses the following data storages:

- Tarantool-based biometric database that stores biometric samples (feature vectors) and face identification events.
- Main system database based on PostgreSQL, that stores internal system data, dossiers, user accounts, and camera settings.
- Directory `/var/lib/findface-security/uploads` that stores uploaded dossier photos, video files, and such event artifacts as full frames, face thumbnails, and normalized face images.
- Directory `/var/lib/ffupload/` that stores only such event artifacts as face thumbnails.

Biometric Database Galleries

There are 3 galleries in the Tarantool-based biometric database:

- `ffsec_dossier_face`: biometric samples extracted from dossier photos.
- `ffsec_events`: biometric samples extracted from faces detected in the video.
- `ffsec_persons`: centroids of persons (virtual biometric samples averaged across all person's faces) and metadata.

1.7.5 Backup Options

To backup the biometric database, you need the `findface-storage-api-dump` utility. It can be launched with the following options:

Note: You can find the detailed information on the `findface-storage-api-dump` usage in *Back Up and Recover Data Storages*.

```
findface-storage-api-dump --help

Usage of findface-storage-api-dump:
  -cache string
        Cache type: inmemory, redis or memcache (default "memcache")
  -cache-inmemory-size int
        Maximum number of items in ARC cache (default 16384)
  -cache-memcache-nodes value
        Comma-separated list of memcache shards (default 127.0.0.1:11211)
  -cache-memcache-timeout duration
        Specifies read/write timeout (default 100ms)
  -cache-redis-addr string
        Host:Port address (default "localhost:6379")
  -cache-redis-db int
        Database to be selected after connecting to the server.
  -cache-redis-network string
        Network type, either tcp or unix (default "tcp")
  -cache-redis-password string
        Optional password. Must match the password specified in the requirepass server␣
↪configuration option.
  -cache-redis-timeout duration
        Specifies dial/read/write timeout (default 5s)
  -config string
        Path to config file
  -config-template
        Output config template and exit
  -continue-on-errors
        Continue on errors instead of exiting
  -cpu-profile string
        Enable CPU profile and set output file
  -extraction-api-extraction-api string
        Extraction API address (default "http://127.0.0.1:18666")
  -extraction-api-timeouts-connect duration
        extraction-api-timeouts-connect (default 5s)
  -extraction-api-timeouts-idle-connection duration
        extraction-api-timeouts-idle-connection (default 10s)
  -extraction-api-timeouts-overall duration
        extraction-api-timeouts-overall (default 35s)
  -extraction-api-timeouts-response-header duration
        extraction-api-timeouts-response-header (default 30s)
  -limits-allow-return-facen
        Allow returning raw feature vectors to detect responses if ?return_facen=true
  -limits-body-image-length int
        Maximum length of image supplied in request body (default 33554432)
  -limits-deny-networks string
        Comma-separated list of subnets that are not allowed to fetch from (default "127.
↪0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8")
  -limits-url-length int
        Maximum supported url length in bytes (default 4096)
  -listen string
        IP:port to listen on (default ":18411")
  -normalized-storage-enabled
        Enables normalize saving (default true)
```

(continues on next page)

(continued from previous page)

```
  -normalized-storage-s3-access-key string
        Access key for the object storage
  -normalized-storage-s3-bucket-name string
        S3 storage bucket name
  -normalized-storage-s3-endpoint string
        S3 compatible object storage endpoint
  -normalized-storage-s3-operation-timeout int
        Storage operations (Get,Put,Delete) timeout in seconds (default 30)
  -normalized-storage-s3-public-url string
        Storage public url
  -normalized-storage-s3-region string
        Storage region
  -normalized-storage-s3-secret-access-key string
        Secret key for the object storage
  -normalized-storage-s3-secure
        If 'true' API requests will be secure (HTTPS), and insecure (HTTP) otherwise␣
↪(default true)
-normalized-storage-webdav-timeouts-connect duration
        normalized-storage-webdav-timeouts-connect (default 5s)
  -normalized-storage-webdav-timeouts-idle-connection duration
        normalized-storage-webdav-timeouts-idle-connection (default 10s)
  -normalized-storage-webdav-timeouts-overall duration
        normalized-storage-webdav-timeouts-overall (default 35s)
  -normalized-storage-webdav-timeouts-response-header duration
        normalized-storage-webdav-timeouts-response-header (default 30s)
  -normalized-storage-webdav-upload-url string
        webdav storage for normalized, disable normalized if empty string (default
↪"http://127.0.0.1:3333/uploads/")
  -normalized_storage string
        Normalized storage type: webdav, s3 (default "webdav")
  -output-dir string
        Output directory (default ".")
  -storage-api-max-idle-conns-per-host int
        storage-api-max-idle-conns-per-host (default 20)
  -storage-api-timeouts-connect duration
        storage-api-timeouts-connect (default 5s)
  -storage-api-timeouts-idle-connection duration
        storage-api-timeouts-idle-connection (default 10s)
  -storage-api-timeouts-overall duration
        storage-api-timeouts-overall (default 35s)
  -storage-api-timeouts-response-header duration
        storage-api-timeouts-response-header (default 30s)
```

1.7.6 Restore Options

To restore the biometric database from a backup, you need the `findface-storage-api-restore` utility. It can be launched with the following options:

Note: You can find the detailed information on the `findface-storage-api-restore` usage in *Back Up and Recover Data Storages*.

```
findface-storage-api-restore --help

Usage of findface-storage-api-restore:
  -cache string
        Cache type: inmemory, redis or memcache (default "memcache")
  -cache-inmemory-size int
        Maximum number of items in ARC cache (default 16384)
  -cache-memcache-nodes value
        Comma-separated list of memcache shards (default 127.0.0.1:11211)
  -cache-memcache-timeout duration
        Specifies read/write timeout (default 100ms)
  -cache-redis-addr string
        Host:Port address (default "localhost:6379")
  -cache-redis-db int
        Database to be selected after connecting to the server.
  -cache-redis-network string
        Network type, either tcp or unix (default "tcp")
  -cache-redis-password string
        Optional password. Must match the password specified in the requirepass server␣
↪configuration option.
  -cache-redis-timeout duration
        Specifies dial/read/write timeout (default 5s)
  -config string
        Path to config file
  -config-template
        Output config template and exit
  -cpu-profile string
        Enable CPU profile and set output file
  -dont-create-gallery
        Don't create gallery, fail if doesn't exist
  -extraction-api-extraction-api string
        Extraction API address (default "http://127.0.0.1:18666")
  -extraction-api-timeouts-connect duration
        extraction-api-timeouts-connect (default 5s)
  -extraction-api-timeouts-idle-connection duration
        extraction-api-timeouts-idle-connection (default 10s)
  -extraction-api-timeouts-overall duration
        extraction-api-timeouts-overall (default 35s)
  -extraction-api-timeouts-response-header duration
        extraction-api-timeouts-response-header (default 30s)
  -limits-allow-return-facen
        Allow returning raw feature vectors to detect responses if ?return_facen=true
  -limits-body-image-length int
        Maximum length of image supplied in request body (default 33554432)
```

(continues on next page)

(continued from previous page)

```
  -limits-deny-networks string
        Comma-separated list of subnets that are not allowed to fetch from (default "127.
↪0.0.0/8,192.168.0.0/16,10.0.0.0/8,::1/128,fe00::/8")
  -limits-url-length int
        Maximum supported url length in bytes (default 4096)
  -listen string
        IP:port to listen on (default ":18411")
  -normalized-storage-enabled
        Enables normalize saving (default true)
  -normalized-storage-s3-access-key string
        Access key for the object storage
  -normalized-storage-s3-bucket-name string
        S3 storage bucket name
  -normalized-storage-s3-endpoint string
        S3 compatible object storage endpoint
  -normalized-storage-s3-operation-timeout int
        Storage operations (Get,Put,Delete) timeout in seconds (default 30)
  -normalized-storage-s3-public-url string
        Storage public url
  -normalized-storage-s3-region string
        Storage region
  -normalized-storage-s3-secret-access-key string
        Secret key for the object storage
  -normalized-storage-s3-secure
        If 'true' API requests will be secure (HTTPS), and insecure (HTTP) otherwise␣
↪(default true)
  -normalized-storage-webdav-timeouts-connect duration
        normalized-storage-webdav-timeouts-connect (default 5s)
  -normalized-storage-webdav-timeouts-idle-connection duration
        normalized-storage-webdav-timeouts-idle-connection (default 10s)
  -normalized-storage-webdav-timeouts-overall duration
        normalized-storage-webdav-timeouts-overall (default 35s)
  -normalized-storage-webdav-timeouts-response-header duration
        normalized-storage-webdav-timeouts-response-header (default 30s)
  -normalized-storage-webdav-upload-url string
        webdav storage for normalized, disable normalized if empty string (default
↪"http://127.0.0.1:3333/uploads/")
  -normalized_storage string
        Normalized storage type: webdav, s3 (default "webdav")
  -rename string
        Ignore dump header and use this string as gallery name
  -storage-api-max-idle-conns-per-host int
        storage-api-max-idle-conns-per-host (default 20)
  -storage-api-timeouts-connect duration
        storage-api-timeouts-connect (default 5s)
  -storage-api-timeouts-idle-connection duration
        storage-api-timeouts-idle-connection (default 10s)
  -storage-api-timeouts-overall duration
        storage-api-timeouts-overall (default 35s)
  -storage-api-timeouts-response-header duration
        storage-api-timeouts-response-header (default 30s)
```

CHAPTER

TWO

USER'S GUIDE

This chapter describes how to work with the FindFace web interface, including its advanced possibilities, and will be of interest to administrators, analysts, operators, and other users.

2.1 First Steps after Deployment

Once you have successfully deployed FindFace, it is time to open the *web interface* and get started. In this chapter, you can find a recommended sequence of steps that will help you harness your system's complete functionality.

In this chapter:

- *Gear Up for Work*
- *Create Users and Ensure System Security*
- *Organize Cameras*
- *Organize Watch Lists and Dossiers*
- *Start Monitoring Faces*
- *Organize Video Surveillance*
- *Start Counting Faces and Silhouettes*
- *Start Analyzing People*
- *FindFace in Action*
- *Basic Maintenance*
- *Go Further*

2.1.1 Gear Up for Work

Perform the primary configuration of your system:

1. *Set up* the left side navigation bar.
2. *Adapt* general preferences.
3. *Choose* the language.

2.1.2 Create Users and Ensure System Security

1. Check out the list of *predefined user roles* and *create new roles* if necessary.
2. *Add users* to the system and grant them privileges.
3. *Configure* authentication and user session monitoring. Authentication is possible by password, face, face or password, face and password.

You may also need:

1. *Enable* SSL data encryption.
2. *Enable* dossier security. If the dossier security is disabled, the dossier photos and attachments will be available by direct link regardless of the user rights.
3. *Disable* FindFace ACL if you do not need it, as the constant permission checks consume a significant amount of system resources.

2.1.3 Organize Cameras

1. *Create a new camera group* or use the default one. A camera group is an entity that allows you to group cameras subject to their physical location. For example, cameras at the same entrance to a building can be combined into one camera group.
2. *Add cameras* to the camera group and *check their statuses*.

You may also need:

1. Configure your system to process video from the group of cameras at their physical location. It may come in handy in a distributed architecture. *Learn more*.
2. Consider enabling event deduplication if observation scenes of cameras within the group overlap. This feature allows you to exclude coinciding facial recognition events among cameras belonging to the same group. *Learn more*.

2.1.4 Organize Watch Lists and Dossiers

1. *Create a new watch list* or use the default one. A watch list is an entity that allows you to classify people by arbitrary criteria: blacklist, wanted, VIP, staff, etc.
2. Upload dossiers and add them in the watch list either *manually*, *in bulk via the web interface*, or use the *console bulk upload* function.

You may also need:

1. *Distribute dossier database* among several hosts. The dossier database will be available for editing on the `master` server and reading and monitoring on the `slaves`.
2. *Customize dossier content*. Create additional fields, tabs, and search filters.

2.1.5 Start Monitoring Faces

By default, FindFace is monitoring only *unmatched faces*. To enable a custom watch list monitoring, simply make this list *active*. You can also turn on sound notifications and request manual acknowledgment for the events associated with the list.

You may also need:

1. Make events more informative by enabling recognition of gender, age, emotions, beard, face mask, and glasses. *Learn more*.
2. Protect your system from spoofing by enabling the Face Liveness Detection functionality. *Learn more*.
3. Support laws related to the processing of personal data of individuals (GDPR and similar). *Learn more*.

2.1.6 Organize Video Surveillance

Create a camera layout for essential video surveillance.

2.1.7 Start Counting Faces and Silhouettes

Set up *counters* to count faces and silhouettes on connected cameras. This functionality can apply to a wide range of situations, such as people counting in queues and waiting areas, monitoring public gatherings, crowding prevention, and more.

2.1.8 Start Analyzing People

FindFace provide a set of people-related analytical tools:

1. *Enable* person recognition to build a person gallery. The system databases will hold a new entity `person event` linked to all *episodes* that feature a person's face. You can work with the person gallery similarly as with events and episodes.
2. *Analyze* social interactions. Examine a circle of people with whom a person has previously been in contact.
3. View 'know your customer' analytics (KYC). It is analytics on the number of visitors, their gender, average age, most frequently visited zones, and the character of visits (first-timers or returners). *Learn more*.

2.1.9 FindFace in Action

1. *Automatically identify faces in live video* and check them against watch lists. Work with the event history by using various filters.
2. Harness the *episodes*. An episode is a set of identification events that feature faces of the same person, detected within a certain period. As events on the *Events* tab show up in an arbitrary order, a large number of miscellaneous events can make the work challenging and unproductive. With the Episodes, the system uses AI to organize incoming events based on the faces similarity and detection time. This allows for the effortless processing of diverse events, even in large numbers.
3. Search for faces in the following databases:
   - Database of detected faces. *Learn more*.
   - Dossier database. *Learn more*.
4. *Search archived videos* for faces under monitoring.
5. Manually *compare two faces* and verify that they belong to the same person.
6. *Build* detailed reports on face recognition events, episodes, search events, persons, counters, cameras, dossiers, and KYC analytics.
7. Use the *mobile app*.

2.1.10 Basic Maintenance

1. *Configure* automatic cleanup of events, episodes, and full frames.
2. Manually *purge* events, episodes, and full frames.
3. Regularly *backup* the database.
4. *Harness* the FindFace comprehensive and searchable audit logs to enhance your system protection.

2.1.11 Go Further

1. Set up *webhooks* to automatically send notifications about specific events, episodes, and counter records to a given URL. In this case, when such an event occurs, FindFace will send an HTTP request to the URL configured for the webhook. You can use webhooks for various purposes, for example, to notify a user about a particular event, invoke required behavior on a target website, and solve security tasks such as automated access control. *Learn more*.
2. Harness the FindFace functions through *HTTP API*.
3. Check out the list of our *partner integrations*.
4. Harness *plugins* to set your directives that determine how FindFace processes detected faces.

See also:

- *Primary Configuration*
- *User Management and System Security*
- *Camera Management*
- *Configure Face Monitoring. Dossier Database*
- *Face Identification in Offline Videos*
- *Face and Silhouette Counters*

- *Person Recognition and People-Related Analytics*
- *Advanced Functionality*
- *Maintenance and Troubleshooting*

2.2 Work with FindFace

Use the web interface to interact with FindFace. To open the web interface, enter its basic address in the address bar of your browser, and log in.

Note: The basic address is set during *deployment*.

Important: To log in for the first time, use the admin account created during *deployment*. To create more users, refer to *User Management*.

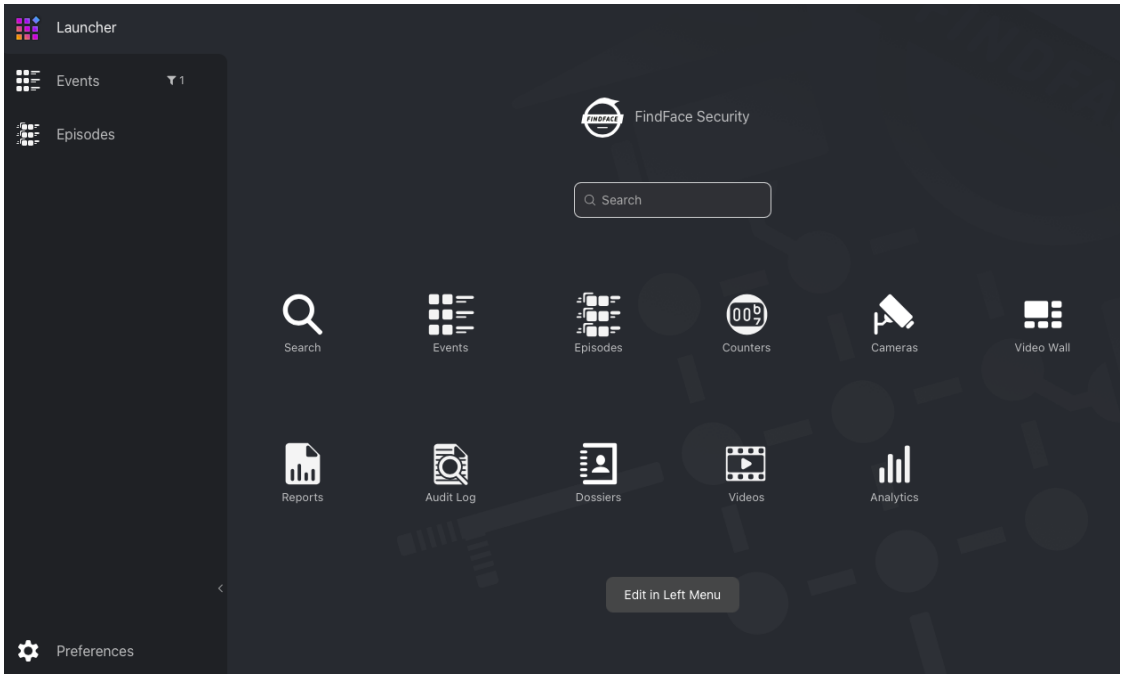
Tip: Work with FindFace on-the-go, using the *mobile app*.

2.2.1 Primary Configuration

This section is about the FindFace primary configuration. Learn how to configure the left side navigation bar, face recognition thresholds, schedule the automatic database cleanup, switch the language, and more.

Navigation

By default, there are only two items in the left side navigation bar: *Events* and *Episodes*. Use *Launcher* to get access to the other FindFace tabs.



Through *Launcher*, you can also make your favorite tabs permanently available from the navigation bar. Do the following:

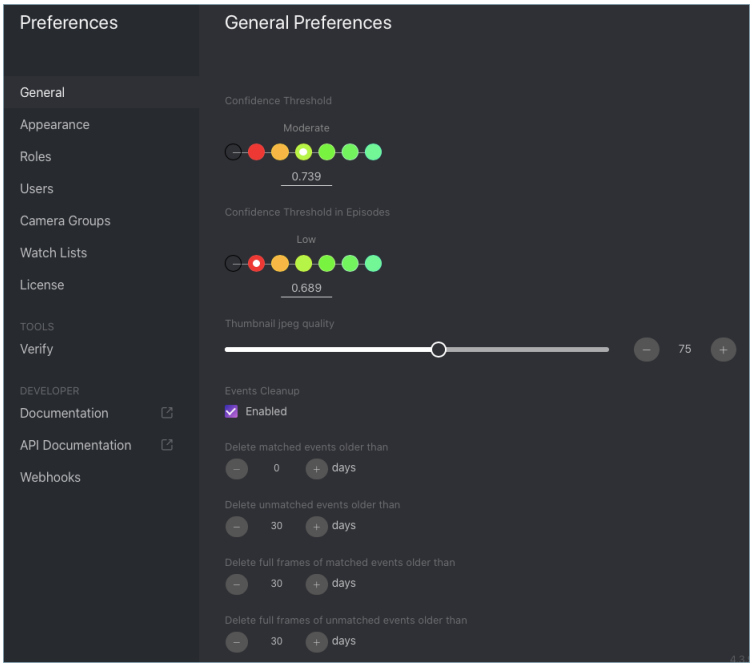
1. Click *Edit in Left Menu*.
2. Check and uncheck navigation items, subject to your needs.
3. Click *Finish Editing*.

General Preferences

The FindFace general preferences determine your system functioning and resource consumption. Here they are:

- generic confidence threshold for face verification
- confidence threshold for episodes
- thumbnail JPEG quality
- schedule for automatic events/episodes cleanup

To configure the general preferences, navigate to the *Preferences* tab and click *General*. After you are finished with adjustments, click *Update*. Find the detailed explanation of each setting below.



In this section:

- *Generic Confidence Threshold*
- *Confidence Threshold for Episodes*
- *Thumbnail JPEG Quality*
- *Automatic Event And Episode Cleanup*

Generic Confidence Threshold

FindFace verifies that a detected face and some face from the dossiers belong to the same person (i. e. the faces match), based on the pre-defined similarity threshold. The default threshold is set to 0.745. If necessary, you can change the generic threshold.

Note: The higher is the threshold, the less are chances that a wrong person will be positively verified, however, some valid photos may also fail verification.

Tip: You can configure the confidence threshold individually for each *camera*, *camera group*, and *watch list*.

Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts by support@ntechlab.com.

Confidence Threshold for Episodes

To construct an *episode*, the system searches the biometric database for recent events with similar faces with a predefined similarity threshold. The default threshold is set to 0.689, which is considered as optimum. If necessary, you can change this value. Be sure to consult with our technical experts prior (support@ntechlab.com).

Thumbnail JPEG Quality

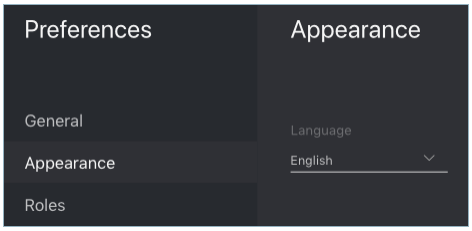
Subject to JPEG quality, thumbnails may take up a significant amount of disc volume. Use the *General* tab to configure the parameter.

Automatic Event And Episode Cleanup

Use the same tab to schedule purging old events and related episodes from the database on a regular basis. You can purge matched and unmatched events/episodes on different schedules, as well as purge only full frames for matched and unmatched events/episodes.

Switch Language

To switch the interface language, navigate to the *Preferences -> Appearance*.



2.2.2 User Management and System Security

Important: Although FindFace provides tools to ensure its protection from unauthorized access, they are not replacing a properly configured firewall. Be sure to use a firewall to heighten the FindFace network protection.

User Management

In this chapter:

- *Predefined Roles*
- *Create Custom Role*
- *Primary and Additional User Privileges*
- *Create User*
- *Deactivate or Delete User*

- *Enable Administrator Privileges for System Plugins*

Predefined Roles

FindFace provides the following predefined roles:

- Administrator has rights to *manage cameras*, events, FindFace users, the *dossier database*, and full access to all other functions.

  Important: Whatever the role, the first administrator (Super Administrator) cannot be deprived of its rights.

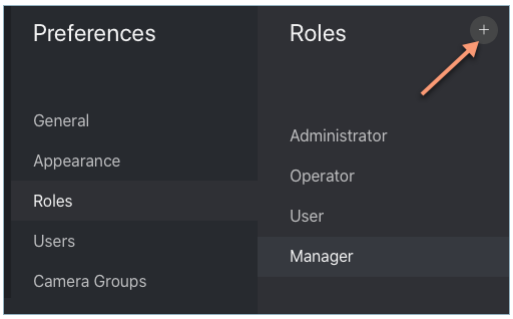
- Operator can create dossiers manually, receive and acknowledge events, and search for faces on the event list. The other data is available read-only. The *batch dossier creation* is unavailable.
- User has a right to receive and acknowledge events, and to search for faces on the event list. The other data is available read-only.

You can change the predefined roles privileges, as well as create various custom roles.

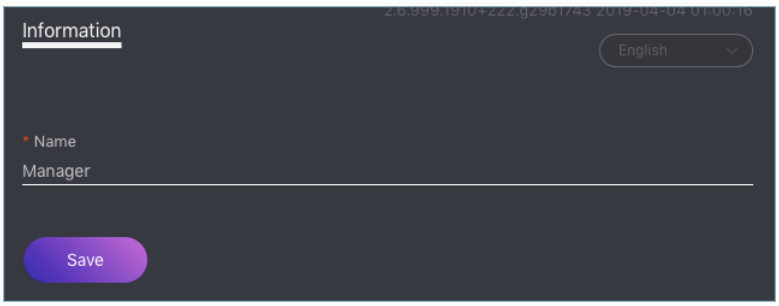
Create Custom Role

To create a custom role, do the following:

1. Navigate to the *Preferences* tab. Click *Roles*.
2. Click +.



3. On the *Information* tab, specify the role name.



4. Click *Save*. You will see additional tabs appear next to the *Information* tab. You can use these tabs to assign the role privileges for specific watch lists (the *Watch Lists* tab) and camera groups (*Camera Groups*), as well as for entire system functions and entities (*Permissions*).

Note: For example, if you set `None` for a certain camera group on the *Camera Groups* tab, users with this role won't be able to work with **this** very group of cameras. Setting `None` for `cameragroup` on the *Permissions* tab will prevent users from viewing and working with **all** camera groups.

Note: The right for an event consists of the rights for a corresponding camera and watch list. To see unmatched events, you only need the rights for a camera.

The full list of the FindFace entities is as follows:

- `dossierlist`: *watch list*
- `dossier`: *dossier*
- `dossierface`: *photo in a dossier*
- `cameragroup`: *camera group*
- `camera`: *camera*
- `listevent`: *event list*
- `eventepisode`: *episodes*
- `person`: *person gallery*
- `uploadlist`: list of photos in *batch upload*
- `upload`: item (photo) in batch photo upload
- `user`: *user*
- `webhook`: *webhook*
- `videoarchive`: *face identification in offline video*
- `counter`: *counters picking statistics on faces and silhouettes*

You can also enable and disable rights for the following functionality:

- `configure_ntls`: configuration of the `findface-ntls` *license server*
- `batchupload_dossier`: *batch photo upload*
- `view_runtimesetting`: viewing the FindFace *general preferences*
- `change_runtimesetting`: changing the FindFace general preferences
- `view_auditlog`: viewing and working with the *audit logs*.

Primary and Additional User Privileges

You assign privileges to a user by using roles:

- *Primary role*: main user role, mandatory for assignment. You can assign only one primary role to a user.
- *Role*: additional user role, optional for assignment. You can assign several roles to one user. The rights associated with the additional roles will be added to the primary privileges.

All users belonging to a particular primary role automatically get access to camera groups (and cameras within the group) and watch lists (and dossiers assigned to the watchlist) created by a user with the same primary role, subject to the privileges defined by their additional role(s).

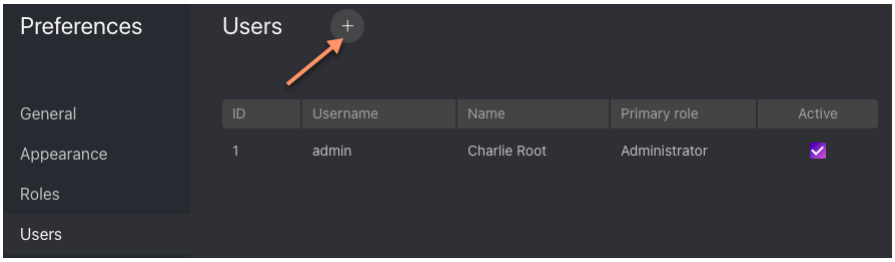
See also:

Create User

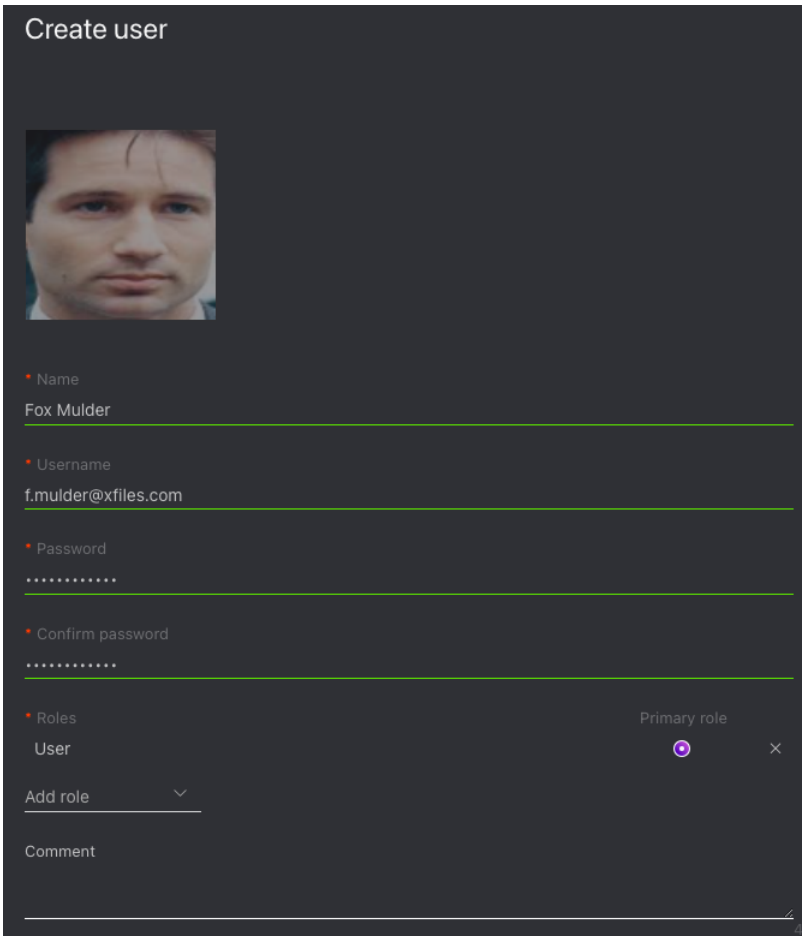
Create User

To create a user, do the following:

1. Navigate to the *Preferences* tab. Click *Users*.
2. Click +.



3. Specify such user data as name, login and password. If necessary, add a comment. Attach the user's photo.



Important: A face in the photo must be of high quality, i.e. close to a frontal position. Distance between pupils: 60 px. Supported formats: WEBP, JPG, BMP, PNG. Photos that do not meet the requirements will be rejected with a detailed error description.

Tip: The photo can be used for *biometric authentication*.

4. From the *Roles* drop-down menu, select one or several user roles. Set one of them as the *Primary role*.
5. Check *Active*.
6. Click *Create*.

Deactivate or Delete User

In order to deactivate a user, uncheck *Active* on the user list (*Preferences -> Users*).

To delete a user from FindFace, click on the user login on the list. Click *Delete*.

Enable Administrator Privileges for System Plugins

The FindFace package incorporates an extensive set of system plugins that provide the following functionality:

- *partner integrations*,
- management of *distributed dossier database*,
- log-in through a crypto certificate (contact your manager for details).

Note: You have to manually enable the system plugins via the `/etc/findface-security/config.py` configuration file.

By default, the Administrator role is granted no privileges for any of the plugins. To assign relevant privileges to Administrator, do the following:

1. Enable a system plugin in the `/etc/findface-security/config.py` configuration file, following the step-by-step instructions provided by our team.
2. Re-migrate the main database architecture from FindFace to **`PostgreSQL`**.

```
sudo findface-security migrate
```

3. Re-create user groups in the main database.

```
sudo findface-security create_groups
```

4. Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

Authentication and Session Monitoring

In this section:

- *Authentication Types*
- *Configure Authentication and Session Renewal*
- *Log out All Users*

Authentication Types

FindFace provides the following authentication types:

- `password`: standard login/password authentication.
- `face`: authentication is possible only by the user's face.
- `face_or_password`: authentication is possible using either a face or login/password.
- `face_and_password`: two-factor authentication. After a face is successfully recognized, the user must enter their credentials.

Important: The *standalone liveness service* (`findface-liveness-api`) must be installed for all the authentication types based on face recognition.

Important: Before using face recognition for authentication, you need to *attach photos* to users' profiles and equip their workplaces with webcams.

Note: You can enable a work session monitoring for the authentication types `face` and `face_or_password`. In this case, the system will be periodically renewing the session after verifying that the face of a person at the workplace matches the user's face that has logged in (see *Configure Authentication and Session Renewal* for details).

Tip: FindFace also provides a certificate-based authentication that is configured independently. Contact our support team for details (support@ntechlab.com).

Configure Authentication and Session Renewal

To configure authentication and session monitoring, do the following:

1. Open the `/etc/findface-security/config.py` configuration file. Find the `FFSECURITY_AUTH_CONFIG` section.

```
sudo vi /etc/findface-security/config.py


FFSECURITY_AUTH_CONFIG = {
```

(continues on next page)

(continued from previous page)

```
    # available options: face, password, face_and_password, face_or_password
    'AUTH_TYPE': 'face_or_password',
    'FACE_AUTH_CONFIDENCE': 0.745,
    # 180 days by default
    'MAXIMUM_SESSION_LENGTH': 15552000,
    # session renew works only with face or face_or_password authorization type
    'NEED_SESSION_RENEW': False,
    'RENEW_SESSION_INTERVAL': 0,
    'MAXIMUM_RENEW_ATTEMPTS': 2,
}
```

2. Set the following authentication parameters:
   - `AUTH_TYPE`: authentication type. Available options: `face`, `password`, `face_and_password`, `face_or_password`.
   - `FACE_AUTH_CONFIDENCE`: after a face in the webcam video is detected as alive, the system checks this face against the database of user photos with this confidence threshold.
   - `MAXIMUM_SESSION_LENGTH`: the maximum session length, in seconds. After a session expires, the user will be automatically logged out unless the session is renewed.
3. Set the following session monitoring parameters:
   - `NEED_SESSION_RENEW`: if `True`, a session can be renewed and prolonged by the time equal to `MAXIMUM_SESSION_LENGTH`, after verifying that the face of a person at the workplace matches the user's face that has logged in.
   - `RENEW_SESSION_INTERVAL`: period in seconds before the expected time of the session expiry, during which the system will attempt to renew the session by enabling the webcam and verifying the user's face.
   - `MAXIMUM_RENEW_ATTEMPTS`: the number of user verification attempts. The attempts occur in a row during the renewal interval.

Note: A verification attempt takes about 3 seconds to complete.

Tip: We recommend you to set up the monitoring parameters so that `MAXIMUM_RENEW_ATTEMPTS` multiplied by the attempt duration is less than `RENEW_SESSION_INTERVAL`. Otherwise, the system will extend the renewal interval x2, x3, and so on, subject to the number of attempts.

4. Restart `findface-security`.

```
sudo systemctl restart findface-security.service
```

Log out All Users

To automatically log out all users, execute the following command on the FindFace principal server console:

```
sudo findface-security logout_all_users
```

Tip: This command comes in handy when switching to a different authentication type.

Enable Data Encryption

To ensure data security, we recommend you enabling SSL encryption. Do the following:

1. Under the nginx configuration directory, create a directory that will be used to hold all of the SSL data:

   ```
   sudo mkdir /etc/nginx/ssl
   ```

2. Create the SSL key and certificate files:

   ```
   sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/my-
   ↪example-domain.com.key -out /etc/nginx/ssl/my-example-domain.com.crt
   ```

   You will be asked a few questions about your server in order to embed the information correctly in the certificate. Fill out the prompts appropriately. The most important line is the one that requests the `Common Name`. You need to enter the domain name or public IP address that you want to be associated with your server. Both of the files you created (`my-example-domain.com.key` and `my-example-domain.com.crt`) will be placed in the `/etc/nginx/ssl` directory.

3. Configure nginx to use SSL. Open the nginx configuration file `/etc/nginx/sites-available/ffsecurity-nginx.conf`. Apply the following modifications to the file:

   1. Add the new `server {...}` section that contains the URL replacement rule:

      ```
      server {
              listen 80;
              server_name my-example-domain.com www.my-example-domain.com;
              rewrite ^(.*) https://my-example-domain.com$1 permanent;
              access_log off;
      }
      ```

   2. Comment out the following lines in the existing `server {...}` section:

      ```
      # listen 80 default_server;
      # listen [::]:80 default_server;
      ```

   3. Add the following lines, including the paths to the certificate and the key, to the existing `server {...}` section:

      ```
      listen 443 ssl;

      ssl_certificate     /etc/nginx/ssl/my-example-domain.com.crt;
      ssl_certificate_key /etc/nginx/ssl/my-example-domain.com.key;
      ```

   4. In the generic nginx configuration file `/etc/nginx/nginx.conf`, find the `SSL Settings` section and append the following lines:

```
ssl_session_cache   shared:SSL:10m;
ssl_session_timeout 1h;
```

The example of the configuration file `/etc/nginx/sites-available/ffsecurity-nginx.conf` with correctly configured SSL settings is shown below:

```
upstream ffsecurity {
 server 127.0.0.1:8002;
}

upstream ffsecurity-ws {
  server 127.0.0.1:8003;
}

map $http_upgrade $ffsec_upstream {
  default "http://ffsecurity-ws";
  "" "http://ffsecurity";
}

server {
  listen 80;
  server_name my-example-domain.com www.my-example-domain.com;
  rewrite ^(.*) https://my-example-domain.com$1 permanent;
  access_log off;
}

server {
  # listen 80 default_server;
  # listen [::]:80 default_server;
  listen 443 ssl;
  ssl_certificate /etc/nginx/ssl/my-example-domain.com.crt;
  ssl_certificate_key /etc/nginx/ssl/my-example-domain.com.key;

  root /var/lib/findface-security;

  autoindex off;

  server_name _;

  location = / {

    alias /usr/share/findface-security-ui/;
    try_files /index.html =404;
     }
  location /static/ {

  }
  location /uploads/ {
    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'GET';
    add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-
↪Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range,
↪Authorization';
```

(continues on next page)

(continued from previous page)

```
    add_header 'Access-Control-Expose-Headers' 'Content-Length,
↪Content-Range';
    add_header 'Access-Control-Max-Age' 2592000;
  }
  location /ui-static/ {
    alias /usr/share/findface-security-ui/ui-static/;
  }
  location /doc/ {
    alias /opt/findface-security/doc/;
  }
  location ~ /videos/(?<video_id>[0-9]+)/upload/(.*)$ {
    if ($request_method = 'OPTIONS') {
      add_header 'Content-Type' 'text/plain; charset=utf-8';
      add_header 'Content-Length' 0;
      return 204;
    }
    set $auth_request_uri "http://ffsecurity/videos/$video_id/auth-
↪upload/";
    auth_request /video-upload-auth/;

    alias "/var/lib/findface-security/uploads/videos/$video_id.bin";
    client_max_body_size 15g;

    dav_access user:rw group:rw all:rw;
    dav_methods PUT;

    create_full_put_path on;
    autoindex off;
    autoindex_exact_size off;
    autoindex_localtime on;
    charset utf-8;

    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'PUT, OPTIONS';
    add_header 'Access-Control-Allow-Headers' 'authorization';
  }
  location = /video-upload-auth/ {
    internal;
    client_max_body_size 15g;
    proxy_set_header Content-Length "";
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass_request_body off;
    proxy_pass $auth_request_uri;
  }

  location / {
    client_max_body_size 300m;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
```

(continues on next page)

(continued from previous page)

```
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_pass $ffsec_upstream;
    proxy_read_timeout 5m;

    location ~ ^/(cameras|videos)/([0-9]+)/stream/?$ {
      proxy_set_header Host $http_host;
      proxy_set_header X-Forwarded-For $remote_addr;
      proxy_set_header X-Forwarded-Proto $scheme;
      proxy_pass http://ffsecurity;
    }

    location ~ ^/streams/(.*)$ {
      internal;
      proxy_pass $1;
    }
  }
}
```

4. Restart nginx.

```
sudo systemctl restart nginx.service
```

5. Edit the `/etc/findface-security/config.py` configuration file. In the `EXTERNAL_ADDRESS` and `ROUTER_URL` parameters, substitute the `http://` prefix with `https://`.

```
sudo vi /etc/findface-security/config.py

...
EXTERNAL_ADDRESS="https://my-example-domain.com"
...
ROUTER_URL="https://IP_address"
```

6. If there are running `findface-video-worker` services in the system, you need to either recreate cameras in the web interface, or change the `router_url` parameter in relevant video processing jobs, substituting the `http://` prefix with `https://`. This can be done with the following command:

```
curl -s localhost:18810/jobs | jq -r '.[]["id"]' | xargs -I {} curl -X PATCH -d '{
↪"router_url": "https://my-example-domain.com/video-detector/frame"}' http://
↪localhost:18810/job/{}
```

Enable Dossier Security

If the dossier security is disabled, the dossier photos and attachments will be available by direct link regardless of the user rights. To increase dossier security, configure FindFace to run all media requests through the DJANGO application for ACL checks.

Important: Enable the dossier media security only if you need it, as this setting has a severe negative impact on the system performance.

To enable dossier security, do the following:

1. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

2. Uncomment `OVERPROTECT_MEDIA` and set it `True`.

```
...

'OVERPROTECT_MEDIA': False,
```

3. Open the nginx configuration file `/etc/nginx/sites-available/ffsecurity-nginx.conf`. Uncomment `internal` in the `location /uploads` section.

```
location /uploads/ {
   internal; # Uncomment if you intend to enable OVERPROTECT_MEDIA
   ...
}
```

4. Restart `findface-security` and nginx.

```
sudo systemctl restart findface-security.service
sudo systemctl restart nginx.service
```

5. After the new security policy is applied, logged-in users must re-authenticate. To make the users do so, execute the logout-all command:

```
sudo findface-security logout_all_users
```

Disable ACL

You can turn off FindFace ACL if you do not need it, as the constant permission checks consume a significant amount of system resources.

Do the following:

1. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

2. Set `ENABLE_ACL = False`.

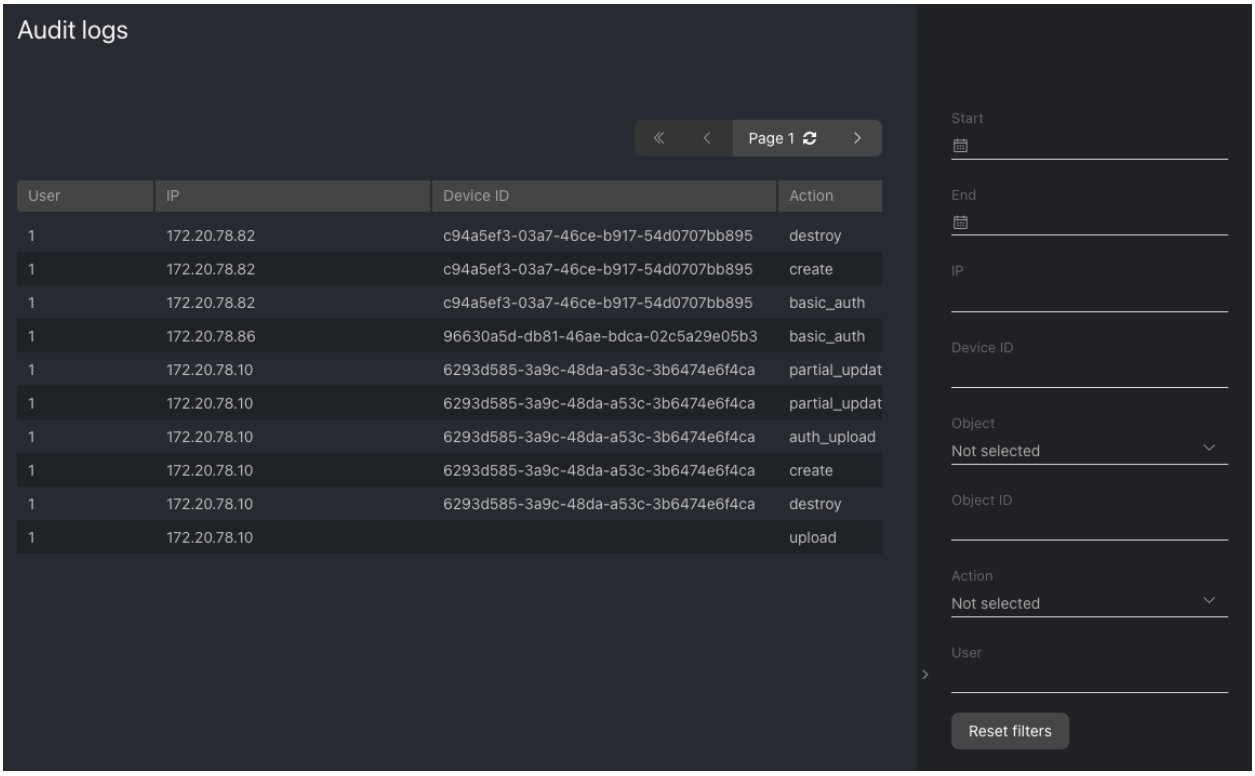
```
...

ENABLE_ACL = False
```

3. Restart `findface-security`.

```
sudo systemctl restart findface-security.service
```

Audit Logs

The FindFace comprehensive and searchable audit logs are an excellent complementary tool for user management that provides you with a thorough audit of the user actions and strengthens your system protection. You can access this functionality on the *Audit Logs* tab.



Each record provides the following data:

- id of the user who performed the action
- IP address where the request came from
- device id: the unique identifier of the client device
- action type such as authorization, search, object modification, restart, and so on
- object type to which the action applies, for example, a dossier or a camera
- object identifier
- details, subject to the action type
- timestamp

Use the filter panel to the right to set up the search conditions.

2.2.3 Camera Management

To configure video-based biometric identification, add cameras to FindFace, grouping them subject to their location.

Note: Privileges to create camera groups and cameras are managed in user's permissions (see *User Management*).

In this chapter:

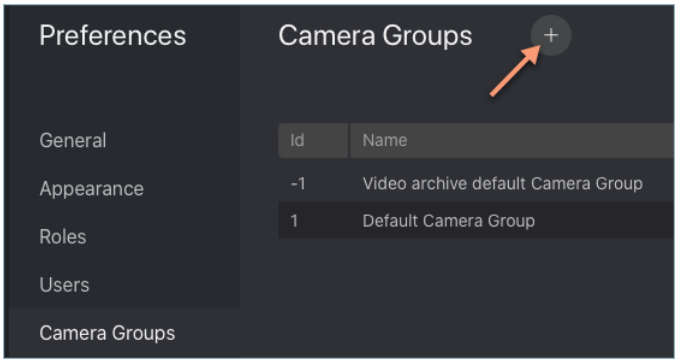
- *Create Camera Group*
- *Add Camera*
- *Monitor Camera Operation*

Create Camera Group

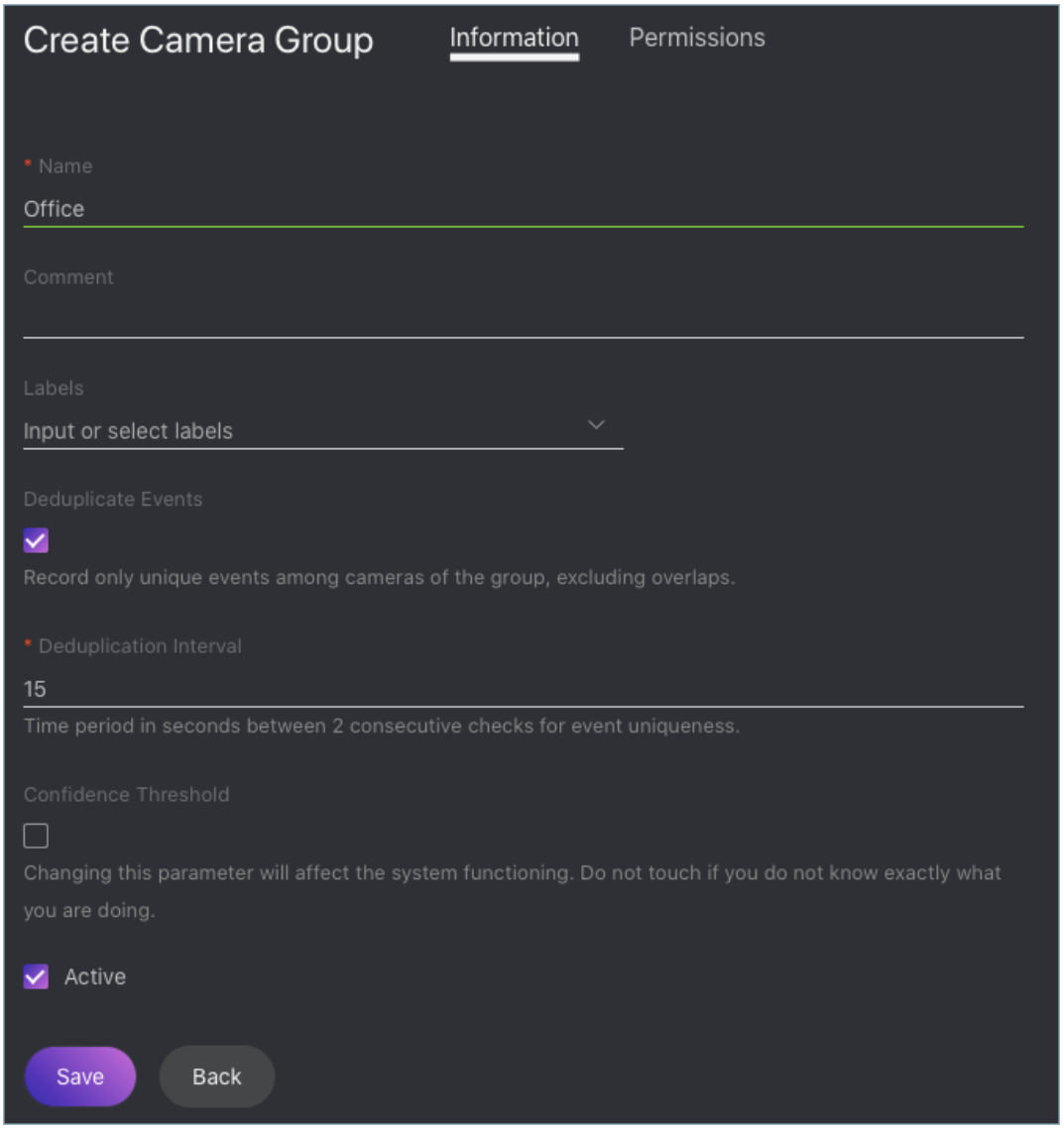
Tip: A default preconfigured camera group is available in the system.

To create a group of cameras, do the following:

1. Navigate to the *Preferences* tab. Click *Camera Groups*.
2. Click +.



3. On the *Information* tab, specify the group name. Add a comment if needed.



4. If you want to allocate a certain `findface-video-worker` instance to process video streams from the group, create or select one or several allocation labels.

Note: To complete the allocation, list the labels in the `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`) configuration file. See *Allocate findface-video-worker to Camera Group* for details.

---

5. If you want to deduplicate events from cameras that belong to the same group, i. e. exclude coinciding events, check *Deduplicate Events* and specify the deduplication interval (interval between 2 consecutive checks for event uniqueness).

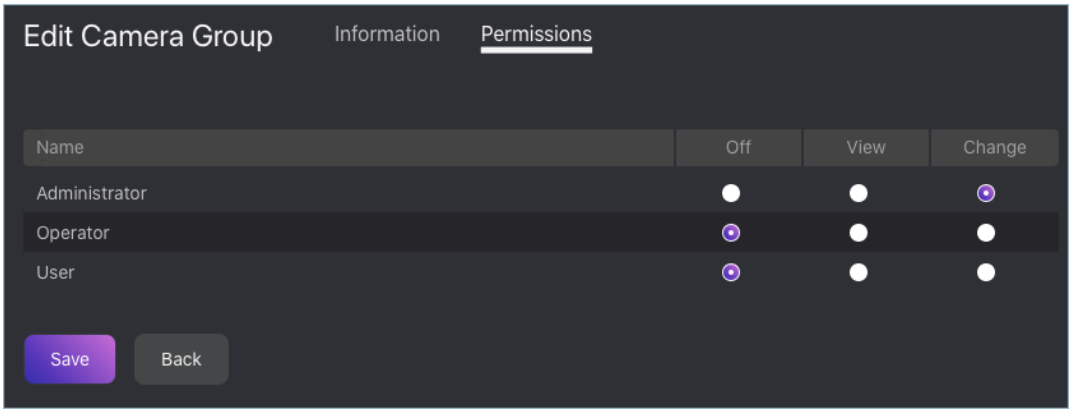
> **Warning:** Use deduplication with extreme caution. If cameras within a group observe different scenes, some faces may be skipped. See *Deduplicate Events* for details.

6. By default, all camera groups in the system are applied the *generic confidence threshold*. To set an individual

threshold for the camera group, check *Confidence Threshold* and specify the threshold value.

Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts by support@ntechlab.com.

7. Check *Active*.
8. Click *Save*.
9. On the *Permissions* tab, assign privileges on the camera group, specifying which user roles are allowed to change/view the camera group settings.

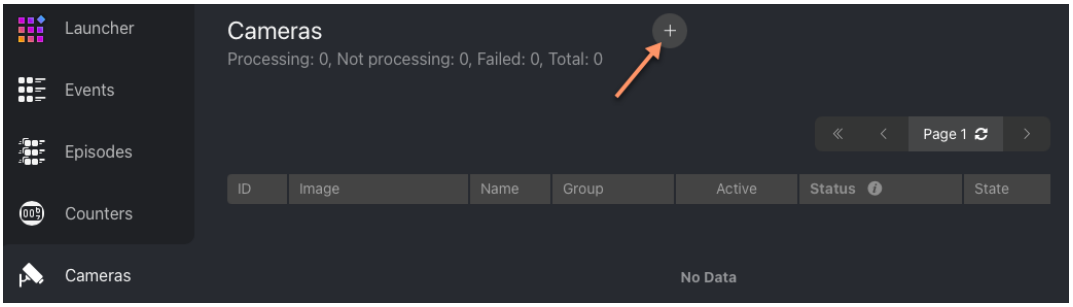


10. Click *Save*.

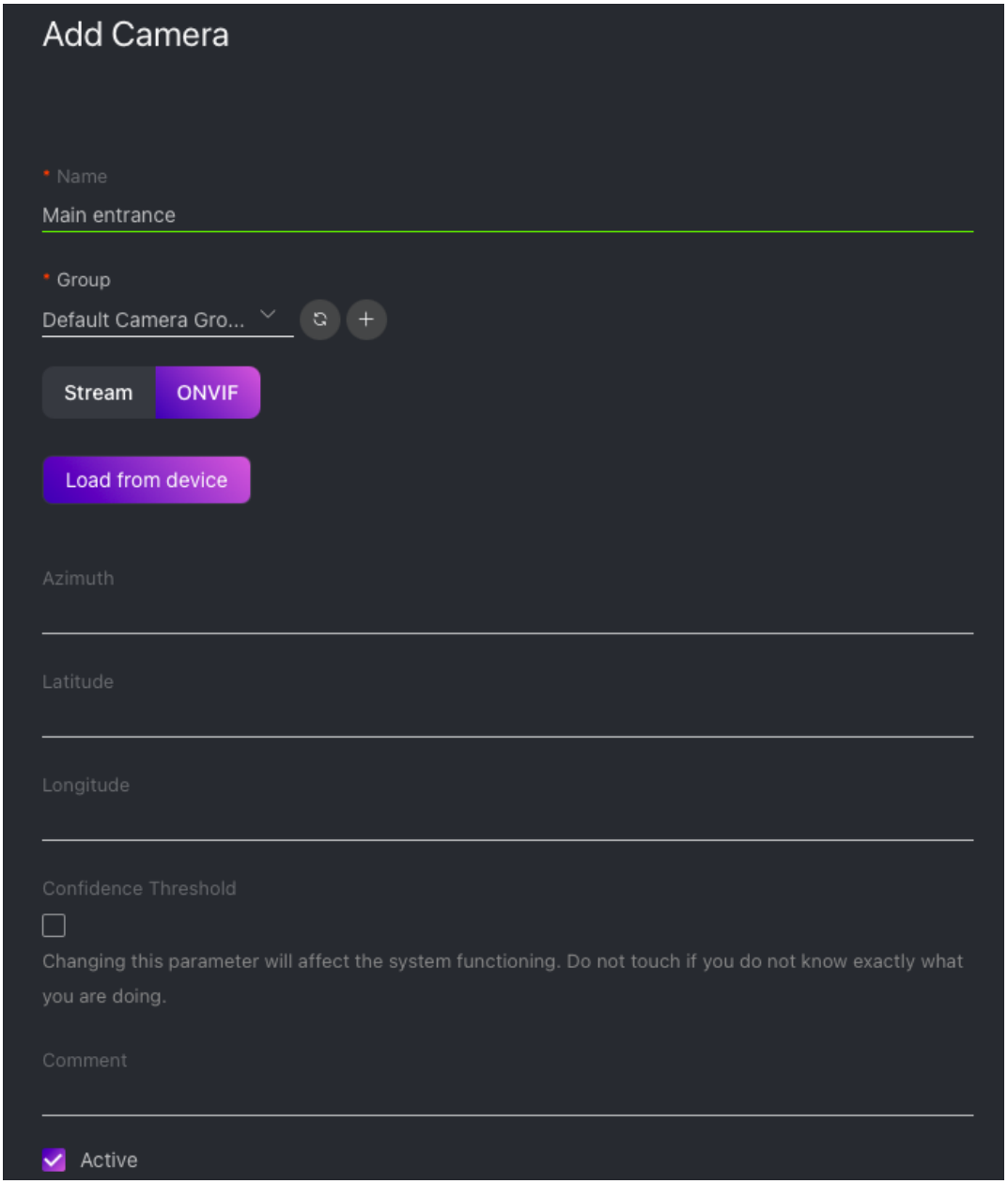
Add Camera

To add a camera, do the following:

1. Navigate to the *Cameras* tab.
2. Click +.



3. Specify the name of a camera and add it to a group. If necessary, add a comment.



4. Specify the camera URL (*Stream*). Using an ONVIF camera, select it from the list of detected devices to automatically load available settings and streams.
5. (Optional) Specify the camera geographical location.
6. By default, all cameras in the system are applied the *generic confidence threshold*. To set an individual threshold for the camera, check *Confidence Threshold* and specify the threshold value.

   Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts by support@ntechlab.com.

7. Check *Active*.
8. To configure video processing, click *Parameters* and make adjustments:

- *Minimum face snapshot quality* (`filter_min_quality`): Minimum quality of a face snapshot to post. Do not change the default value (0.45) without consulting with our technical experts (support@ntechlab.com).
- *Minimum face size* (`filter_min_face_size`): Minimum face size in pixels to post.
- *Maximum face size* (`filter_max_face_size`): Maximum face size in pixels in post.
- *Compression quality* (`jpeg_quality`): Full frame compression quality.
- *FFMPEG options* (`ffmpeg_params`): FFMPEG options for a video stream in the key-value format ["rtsp_transpotr=tcp", "ss=00:20:00"].
- *Offline mode* (`overall_only`): Offline mode. Enable posting one snapshot of the best quality for each face.
- *Time interval* (`realtime_post_interval`): Time interval in seconds (integer or decimal) within which the face tracker picks up the best snapshot in realtime mode.
- *Post first face immediately* (`realtime_post_first_immediately`): If `true`, post the first face from a track immediately after it passes through the quality, size, and ROI filters, without waiting for the first `realtime_post_interval` to complete. The way the subsequent snapshots are posted depends on the `realtime_post_every_interval` value. If false, post the first face after the first `realtime_post_interval` completes.
- *Post best snapshot* (`realtime_post_every_interval`): If true, post the best snapshot obtained within each Time interval (realtime_post_interval) in realtime mode. If false, post the best snapshot only if its quality has improved comparing to the previously posted snapshot.
- *Posting timeout* (`router_timeout_ms`): Timeout in milliseconds for posting faces.
- *Retrieve timestamps from stream* (`use_stream_timestamp`): If true, retrieve and post timestamps from a video stream. If false, post the actual date and time.
- *Add to timestamps* (`start_stream_timestamp`): Add the specified number of seconds to timestamps from a stream.
- *Play speed limit* (`play_speed`): If less than zero, the speed is not limited. In other cases, the stream is read with the given `play_speed`. Not applicable for live streams.
- *Region of Tracking* (`ROT`): Enable detecting and tracking faces only inside a clipping rectangle. Use this option to reduce the video face detector load.
- *Region of Interest* (`ROI`): Enable posting faces detected only inside a region of interest.

Tip: To specify ROT/ROI, use the visual wizard. First, create a camera without ROT/ROI. Then open it for editing and click *Parameters*. You will see the visual wizard appear.

If necessary, specify optional parameters for video processing. Click *Advanced Parameters*.

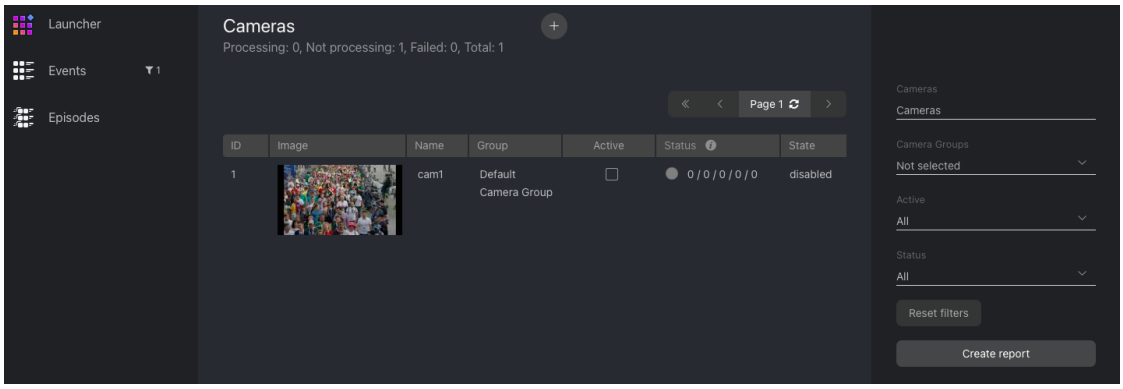
- *Force input format* (`ffmpeg_format`): Pass FFMPEG format (mxg, flv, etc.) if it cannot be detected automatically.
- *Verify SSL* (`router_verify_ssl`): If true, enable verification of the server SSL certificate when the face tracker posts faces to the server over https. If false, a self-signed certificate can be accepted.
- *Minimum motion intensity* (`imotion_threshold`): Minimum motion intensity to be detected by the motion detector.

9. Click *Save*.

Note: Each created camera is associated with a so-called job, a video processing task that contains configuration settings and stream data and is assigned to `findface-video-worker`. This task can be restarted (see *Monitor Camera Operation*).

Monitor Camera Operation

To monitor the operation of cameras, navigate to the *Cameras* tab.



Camera statuses:

- Green: the video stream is being processed without errors.
- Yellow: the video stream is being processed for less than 30 seconds, or one or more errors occurred when posting a face.
- Red: the video stream cannot be processed.
- Grey: camera disabled.

Tip: You can configure the yellow and red statuses based on the portion of dropped frames and failed face postings. To do so, modify the following parameters in the `/etc/findface-security/config.py` configuration file:

```
sudo vi /etc/findface-security/config.py

FFSECURITY = {
    ...
    # max camera frames_dropped percent
    'MAX_CAMERA_DROPPED_FRAMES': {'yellow': 0.1, 'red': 0.3},
    # max camera faces_failed percent
    'MAX_CAMERA_FAILED_FACES': {'yellow': 0.1, 'red': 0.3},
    ...
}
```

For each camera, you will be provided with complete statistics such as current session duration, the number of successfully posted faces, the number of faces processed with errors after the last job restart, the number of frame drops, and other data.

Note: Each created camera is associated with a so called job, a video processing task that contains configuration

settings and stream data and is assigned to `findface-video-worker`. This task can be restarted.

To restart a job, click  in the *Action* column. In this case, the number of errors will be reset to `0`.

With a large number of cameras in the system, use the following filters:

- *Camera groups*,
- *Active*,
- *Status*.

See also:

- *Allocate findface-video-worker to Camera Group*
- *Deduplicate Events*

2.2.4 Configure Face Monitoring. Dossier Database

This chapter is all about configuring face monitoring and creating the dossier database. Each dossier must contain one or several photos of a person and belong to a specific classification list (watch list), for example, wanted, VIP, etc. You can create as many watch lists as necessary.

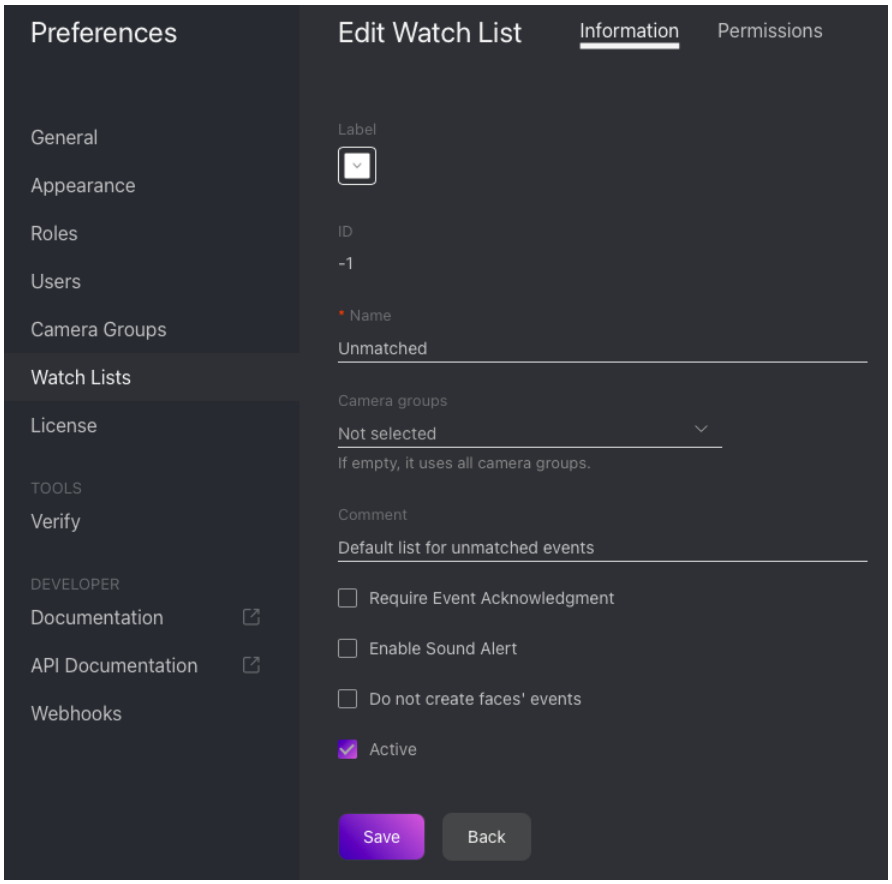
Tip: To create dossiers in bulk, use the *batch photo upload* functionality.

In this section:

- *Monitoring Unmatched Faces*
- *Create Watch List*
- *Create Dossier Manually*
- *Batch Photo Upload*
- *Filter Dossiers by Watch List*
- *Purge Dossier Database*
- *Disable Event Creation for Specific Faces*

Monitoring Unmatched Faces

FindFace features one pre-configured watch list that is used for monitoring only unmatched faces. This watch list cannot be removed from the system. To edit its settings or deactivate it, navigate to the *Preferences* tab. Click *Watch Lists* and then click *Unmatched* in the table.



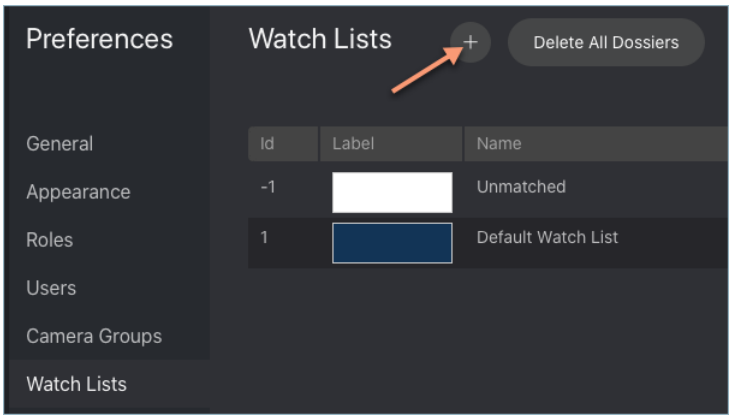
Note: To view only unmatched faces in the event list, select *Unmatched* in the *Watch lists* filter on the *Events* tab (refer to *Work with Events* for details).

Create Watch List

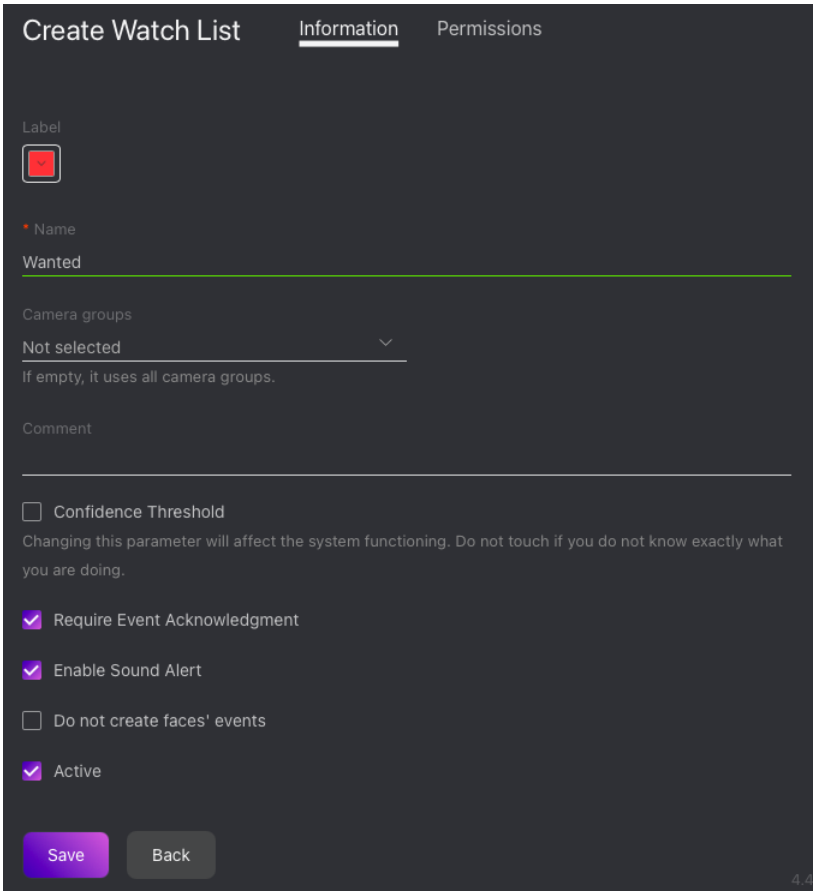
To create a custom watch list, do the following:

Tip: Besides the *Unmatched* watch list, there is also a default preconfigured watch list to monitor matched faces. This watch list cannot be removed from the system.

1. Navigate to the *Preferences* tab. Click *Watch Lists*.
2. Click +.



3. From the *Label* palette, select a color which will be shown in notifications for this list. Keep in mind that the right color makes for a quicker response of the person on duty.

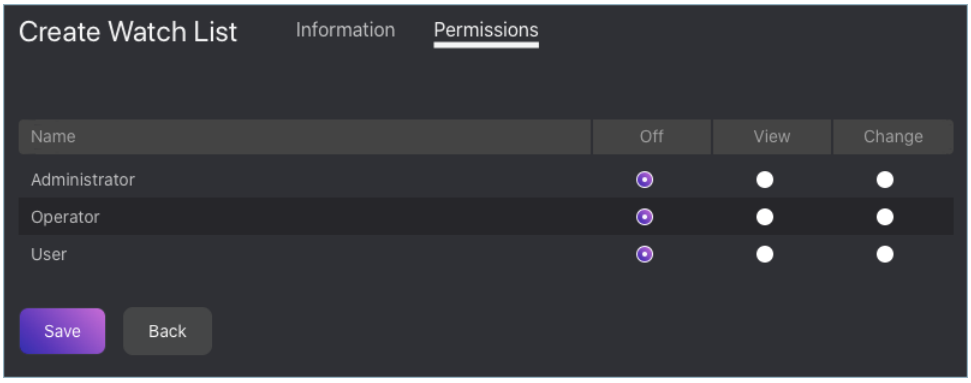


4. Specify the watch list name. Add a comment if needed.
5. Select a camera group(s) that will be used to monitor the watch list. If no groups specified, the watch list will be monitored by all active cameras in the system.
6. Check *Require acknowledgment* if it is mandatory that events associated with the list be manually acknowledged.
7. Check *Enable sound alert* to turn on sound notifications for the list if needed.

8. By default, all watch lists in the system are applied the *generic confidence threshold*. To set an individual threshold for the watch list, check *Confidence Threshold* and specify the threshold value.

   Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts by support@ntechlab.com.

9. Check *Active*.
10. Click *Save*.
11. On the *Permissions* tab, assign privileges on the watch list, specifying which user roles are allowed to change/view the watch list settings.

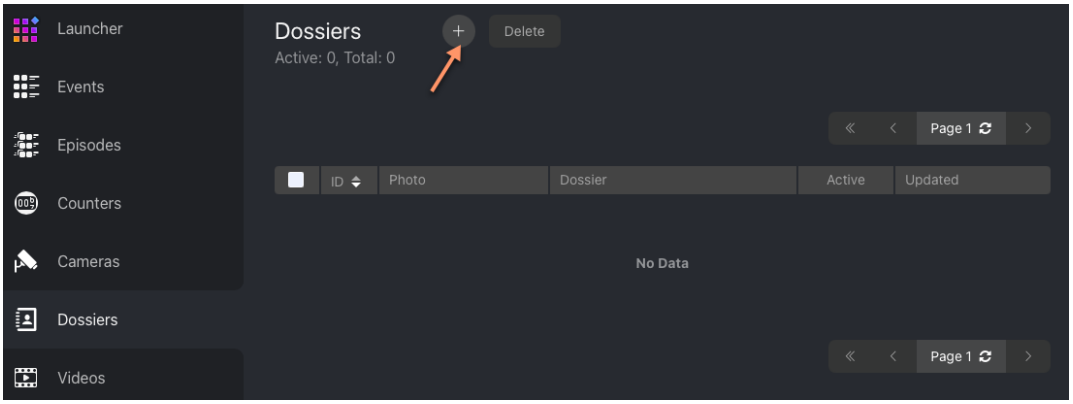


12. Click *Save*.

Create Dossier Manually

To create a dossier manually, do the following:

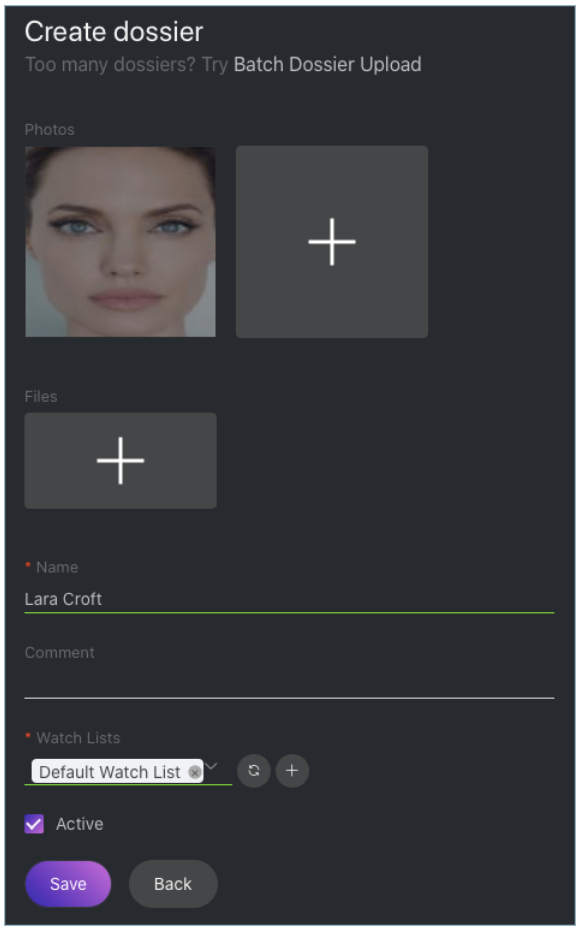
1. Navigate to the *Dossiers* tab.
2. Click +.



3. Attach a photo, related documents, and specify the person's name. If necessary, add a comment.

   Important: A face in the photo must be of high quality, i.e. close to a frontal position. Distance between pupils:

60 px. Supported formats: WEBP, JPG, BMP, PNG. Photos that do not meet the requirements will be rejected with a detailed error description.



4. From the *Watch lists* drop-down menu, select a classification list (or several lists, one by one) for the dossier.
5. Check *Active*. If a dossier is inactive, it is excluded from the real time face identification.
6. Click *Save*. If a similar dossier already exists in the database, you will be able to merge it with the new one, create the new dossier anyway, or cancel creation.

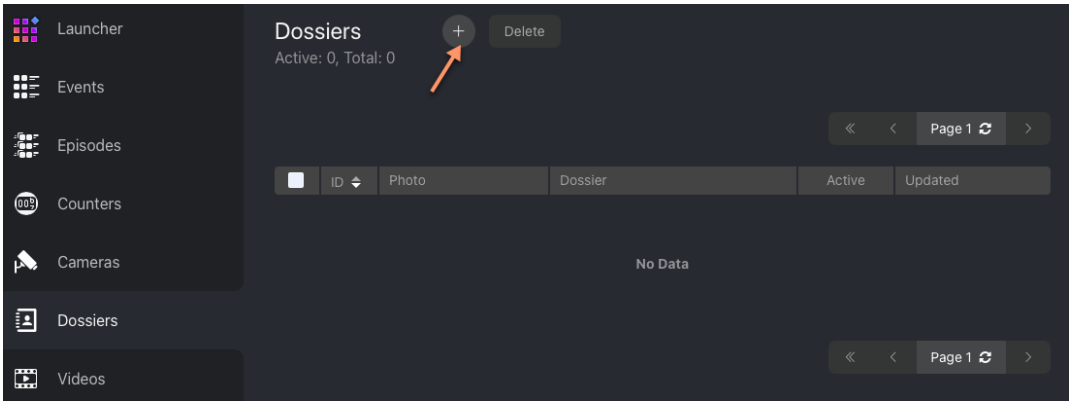
Batch Photo Upload

To create dossiers in bulk, use the batch photo upload. Do the following:

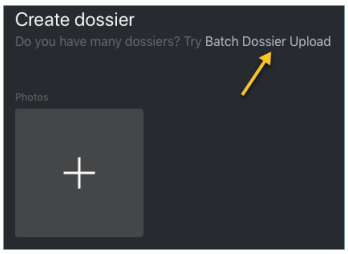
Tip: If you need to upload a large number of photos (more than 10,000), use *Console Bulk Photo Upload*.

Important: Faces in photos must be of high quality, i.e. close to a frontal position. Distance between pupils: 60 px. Supported formats: WEBP, JPG, BMP, PNG. Photos that do not meet the requirements will be rejected with a detailed error description.

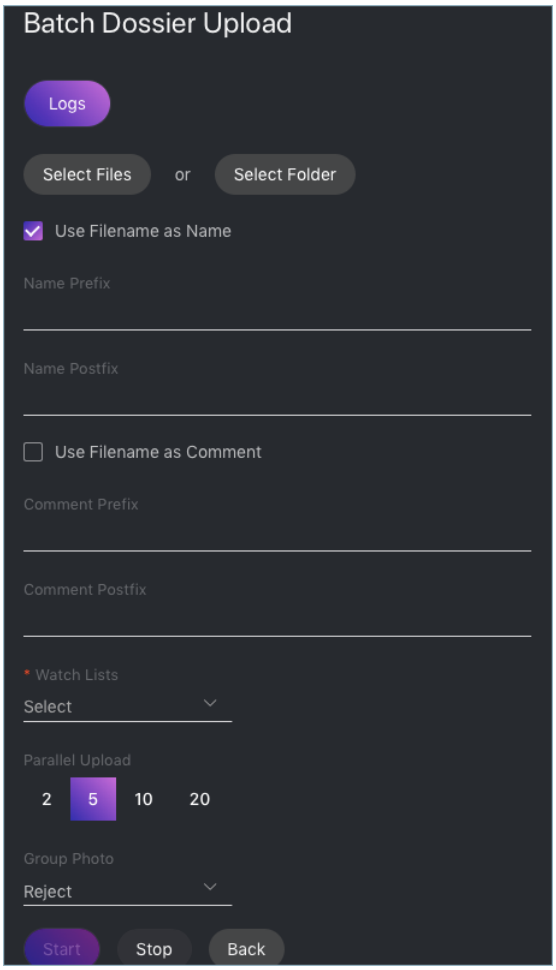
1. Navigate to the *Dossiers* tab.
2. Click +.



3. Click *Batch Dossier Upload*.



4. Select multiple image files, or a folder.

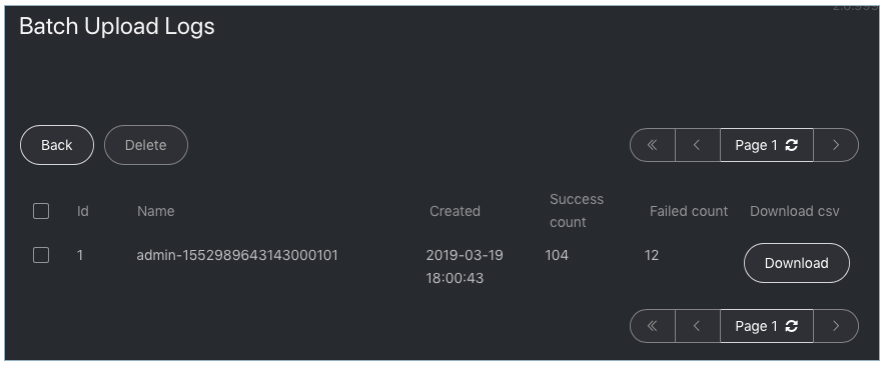


5. You can use image file names as a basis for names and/or comments in dossiers to be created. Select the necessary option(s). Then configure the automatic name/comment generation rule by appending a custom prefix and/or postfix to the file name.

   Tip: To avoid merging the 3 words into one, use underscore or another symbol in the prefix and postfix.

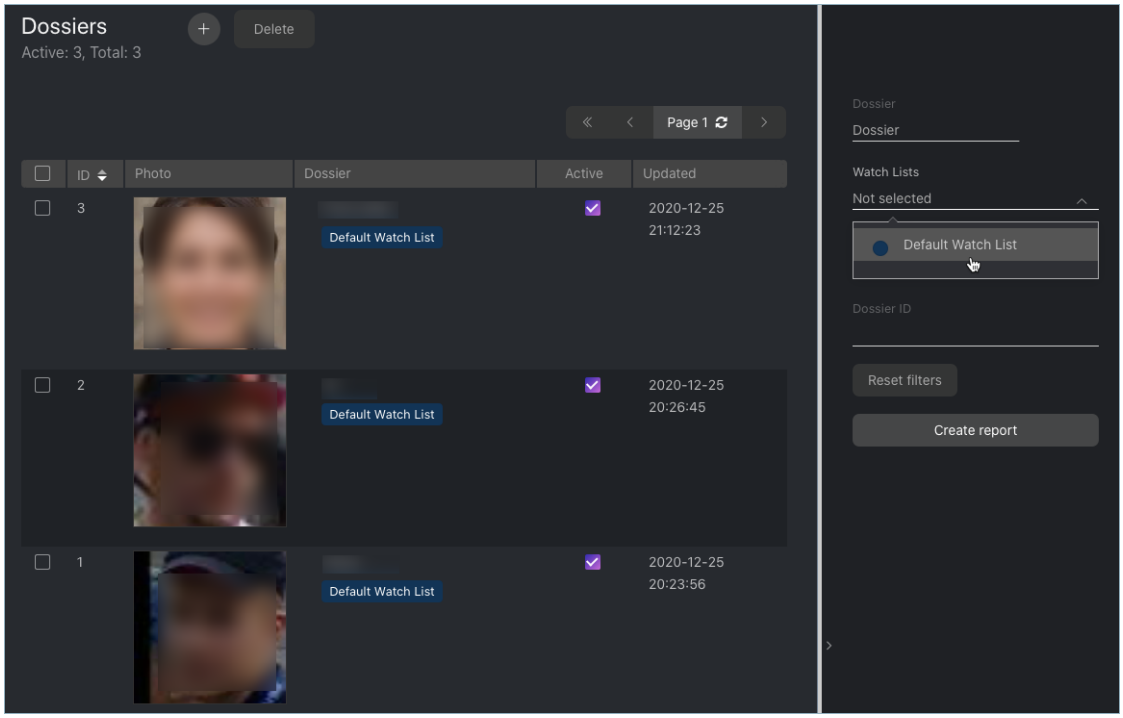
6. From the *Watch lists* drop-down menu, select a classification list for the dossiers.
7. Use the *Parallel Upload* option to specify the number of photo upload streams. The more streams you use, the faster it takes to complete the upload, however it requires more resources as well.
8. From the *Group Photo* drop-down menu, select the system behavior upon detecting several faces in a photo: reject the photo, upload the biggest face, or upload all faces.
9. Click *Start* to launch the photo upload.

   Important: To view the batch photo upload log, click *Logs*. You can then download the log in the `.csv` format if needed.



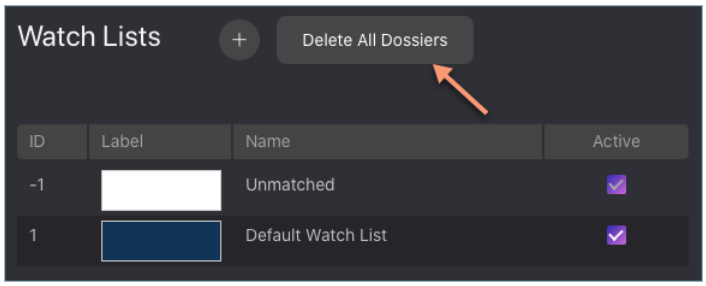
Filter Dossiers by Watch List

You can find all dossiers created in FindFace on the *Dossiers* tab. Use the *Watch lists* filter to filter dossiers by list.



Purge Dossier Database

You can purge the entire dossier database in one click. To do so, navigate to the *Preferences* tab. Click *Watch Lists*. Click *Delete All Dossiers*.

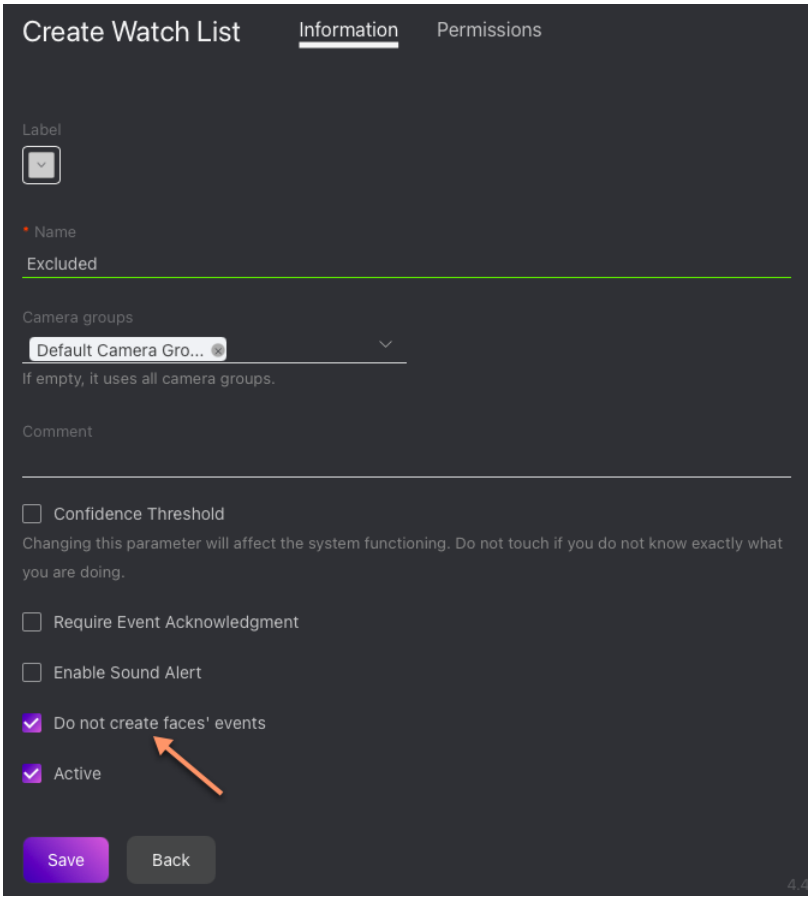


Disable Event Creation for Specific Faces

Sometimes, it is necessary to omit certain faces during monitoring. One of the most common cases is the faces in advertisement media located in the camera field. Being detected continuously by your system, they can easily overflow the event feed and the database.

To prevent this from happening, do the following:

1. *Create a watch list* that will store the faces excluded from detection. In its settings, check *Do not create face events*.



2. For each excluded face, *create a dossier* and add it to the watch list.

2.2.5 Face and Silhouette Counters

Important: To be able to count human silhouettes, you first have to enable *silhouette detection*.

FindFace allows you to count faces and silhouettes on connected cameras. This functionality can apply to a wide range of situations, such as people counting in queues and waiting areas, monitoring public gatherings, crowding prevention, and more.

The counting method is based on time slices, which means that the system counts faces and silhouettes in static screenshots taken with a given `count interval`. The counter shows how the number of faces and silhouettes changes over time.

In this section:

- *Configure Counters*
- *Create Counter*
- *Counter Chart*
- *Work with Counter Records*
- *Set Webhook for Counter*

Configure Counters

To configure counters, open the `/etc/findface-security/config.py` configuration file and modify the following parameters:

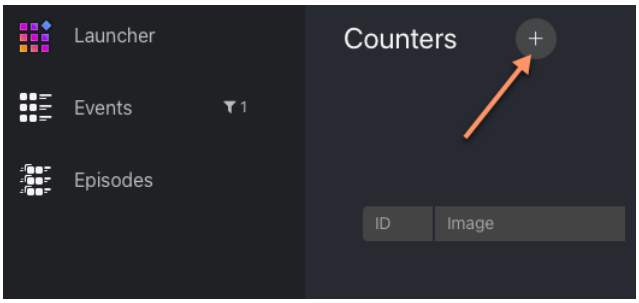
- `COUNTERS_SAVE_FULLFRAME` determines saving options of full frames in counters: `always`, `detect` - only save if faces or silhouettes have been detected, `never`.
- `COUNTERS_FULLFRAME_JPEG_QUALITY`: JPEG quality of full frames,
- `COUNTERS_THUMBNAIL_JPEG_QUALITY`: JPEG quality of thumbnails.

```
sudo vi /etc/findface-security/config.py

# counters full frame saving options:
# `always` - save always
# `detect` - save only if faces or silhouettes have been detected
# `never` - never save full frames
'COUNTERS_SAVE_FULLFRAME': 'always',
'COUNTERS_FULLFRAME_JPEG_QUALITY': 75,
'COUNTERS_THUMBNAIL_JPEG_QUALITY': 75,
...
```

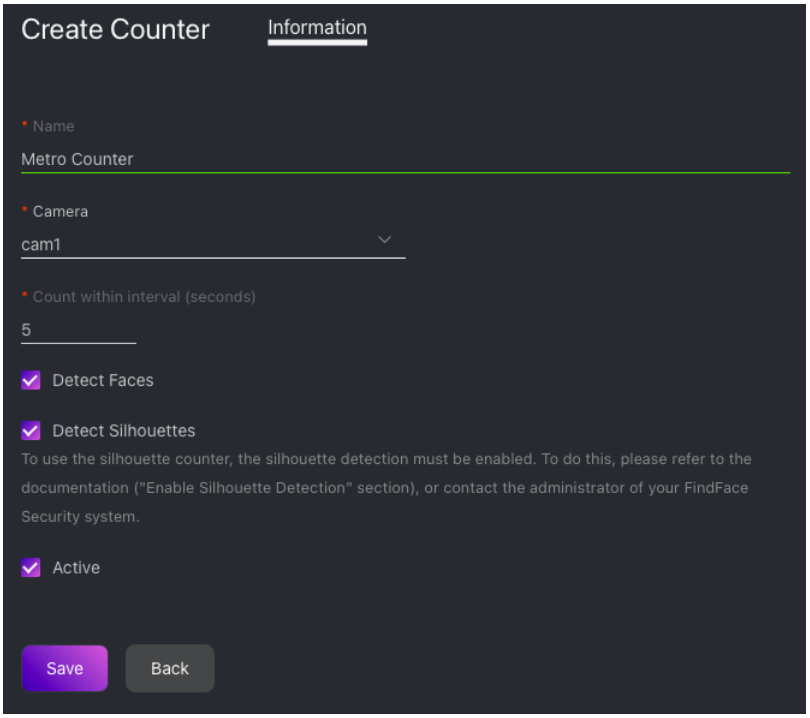
Create Counter

To set up a counter, do the following:

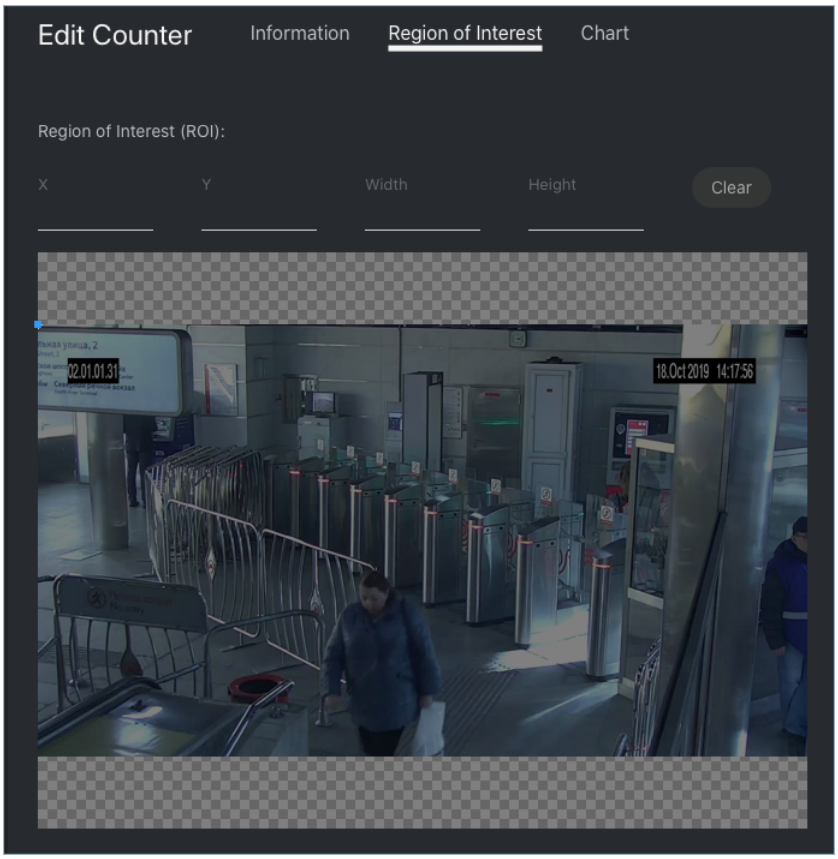
1. Navigate to the *Counters* tab.
2. Click +.



3. Specify the counter name.

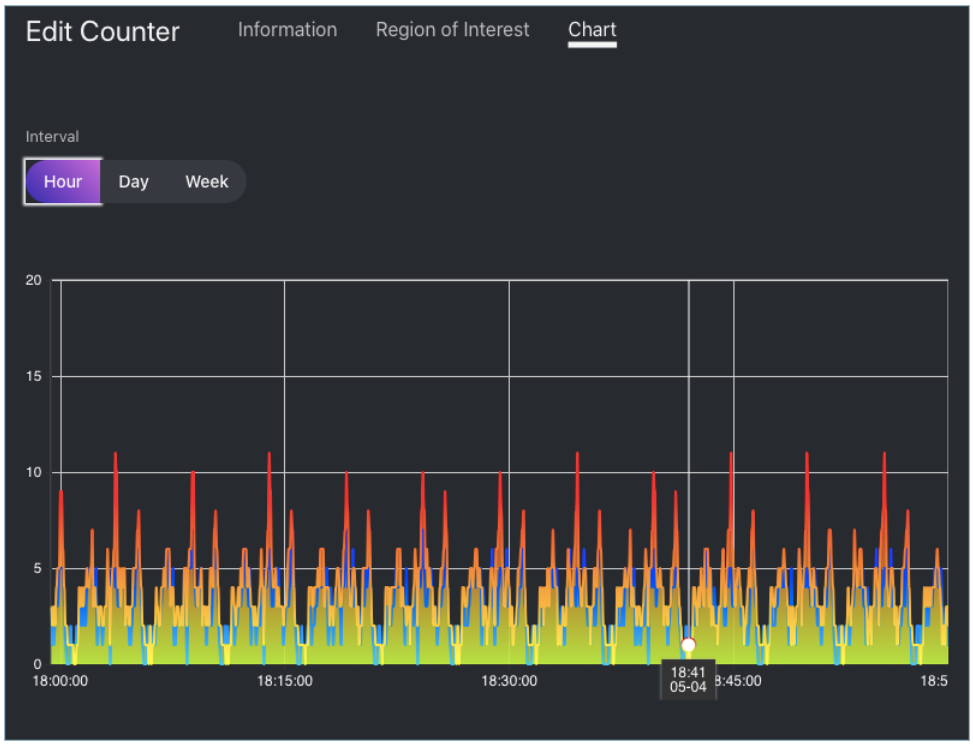


4. Select the camera for counting.
5. Specify the interval between two consecutive screenshots used for counting.
6. Check *Detect Faces* to count faces.
7. Check *Detect Silhouettes* to count silhouettes. Silhouettes detection has to be *enabled*.
8. Make sure that the counter is *Active*.
9. Click *Save*. You will see two new tabs appear.
10. (Optional) Navigate to the *Region of Interest* tab to specify the face/silhouette tracking region within the camera field of view.



Counter Chart

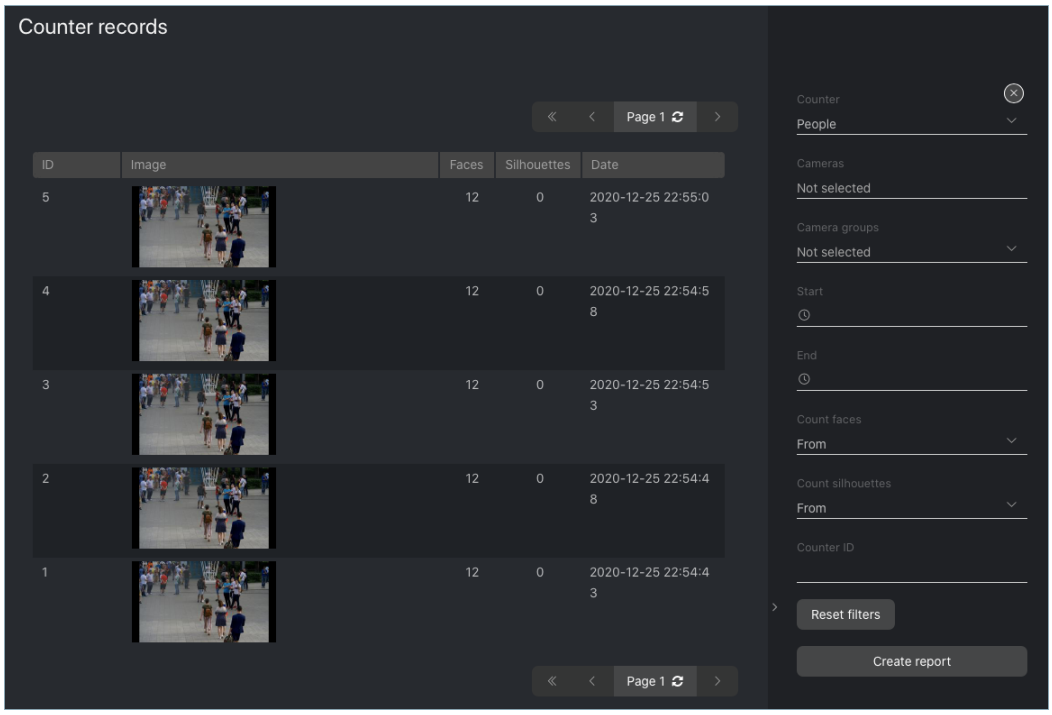
To see the counter chart for the last hour, 24 hours, or week, navigate to the *Chart* tab in the counter settings.



Work with Counter Records

Static screenshots taken by a counter, with the number of faces and silhouettes in them, are saved as counter records.

To see the counter records, navigate to the *Counters* tab and click on the required counter.



To work with counter records, use the following filters:

- Counter
- Cameras
- Camera groups
- Time period
- Number of faces in record
- Number of silhouettes in record
- Record id

Set Webhook for Counter

To take it up a notch, *configure a webhook* for counter records with a specific number of faces and silhouettes.

See also:

- *Silhouette Detection*
- *Webhooks*

2.2.6 Events and Episodes of Facial Recognition

To monitor the real-time face identification in live videos, use the *Events* and *Episodes* tabs. Besides monitoring, both tabs allow you to access the history of identification events.

Tip: Search for faces through the event database and dossier database on the *Search* tab.

Tip: To perform the face identification in archived videos, see *Face Identification in Offline Videos*.

Work with Events

This section is about the *Events* tab.

Tip: Take your security up a notch with *episodes*.

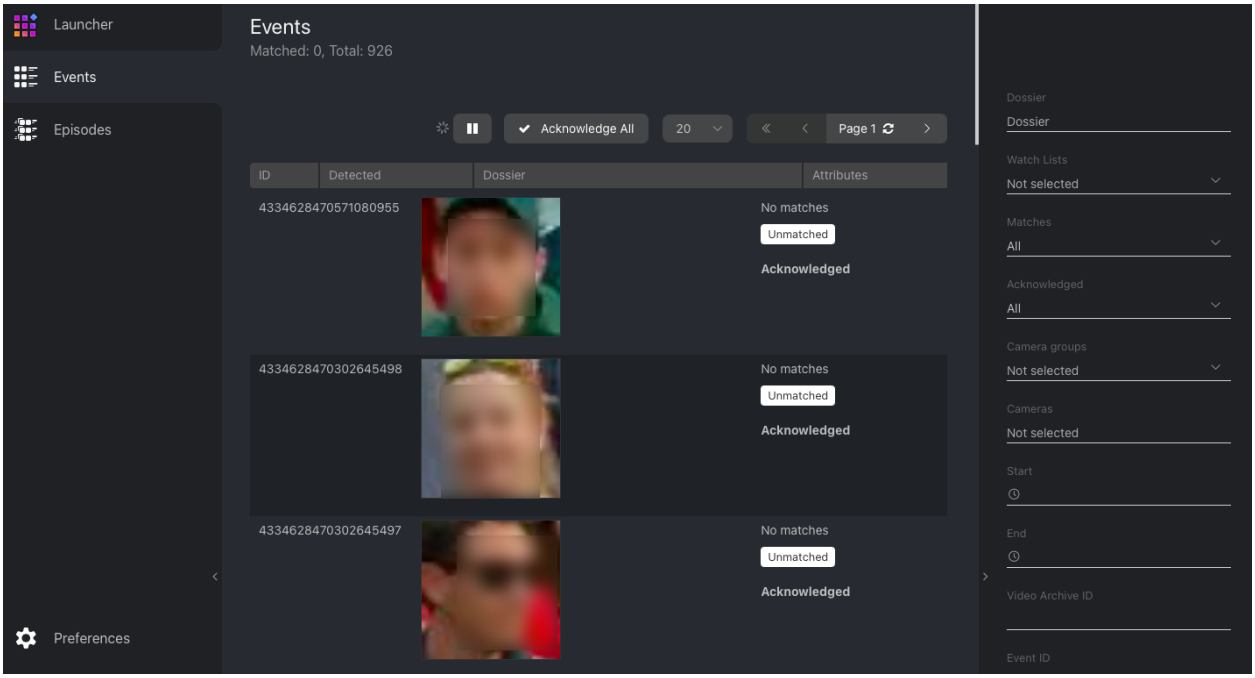
Important: You can *enable sound notifications* for events related to specific watch lists. In some browsers, the tab with events has to remain in focus to get a sound played. To put a tab in focus, open it, and click anywhere on the page.

In this chapter:

- *View Identification Events*
- *Face Liveness and Face Features Recognition*
- *Event Ticket. Acknowledging Event*
- *Event Ticket. Face Search*

View Identification Events

Once a face detected, you will see a notification on the event list.



A notification can feature different pieces of information, depending on whether a detected face has a match in the database:

- Match not found: a normalized face image, detection date and time, the name of a camera group.
- Match found: a normalized face image, the photo from a dossier, the name of a person, similarity between faces, the comment from a dossier, the name of a dossier list, detection date and time, the name of a camera group.

Note: You can configure the system in such a way that you will get notifications only for the faces with a match.

Important: In order to pause the notifications thread, click above the list of events.

When working with events, the following default filters may come in handy:

- *Dossier*: display events only for a selected dossier.

- *Watch lists*: display events only for a selected dossier category (watch list).

  Note: To view only unmatched faces on the event list, select *Unmatched* in this filter.

- *Matches*: display events only with/without matches, or all events.
- *Acknowledged*: display only acknowledged/unacknowledged events, or all events.
- *Cameras*: display only events from a selected camera.
- *Camera groups*: display only events from a selected group of cameras.
- *Start*, *End*: display only events that occurred within a certain time period.
- *Video Archive ID*: display events from the video archive with a given ID.
- *Event ID*: display an event with a given ID.
- *Episode ID*: display events from the episode with a given ID.

Face Liveness and Face Features Recognition

Depending on the system settings, you can see an estimation of face liveness and/or a result of such face features recognition as gender, age, emotions, glasses, and/or beard.

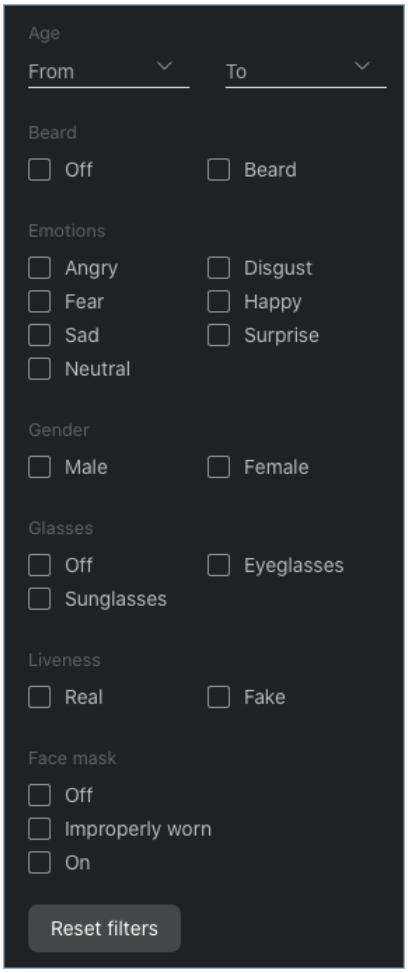
The face liveness detector automatically spots fake faces and prevents photo attacks by distinguishing a live face from a face image.

Note: The liveness score can be `null`. It is so when the liveness detector is disabled or unable to estimate the face liveness in the provided image.

The face feature recognition result is in the following format:

Face feature	Result format	Example
Age	Feature: `age`: number of years	age: 33
Gender	Result: `male/female` (feature: `gender`): algorithm confidence in result	female (gender): 0.95
Emotions	Result: `angry/disgust/fear/happy/sad/surprise/neutral` (feature: `emotions`): algorithm confidence in result	happy (emotions): 0.99
Glasses	Result: `eye/sun/none` (feature: `glasses`): algorithm confidence in result	none (glasses): 0.87
Beard	Result: `beard/none` (feature: `beard`): algorithm confidence in result	none (beard): 0.91

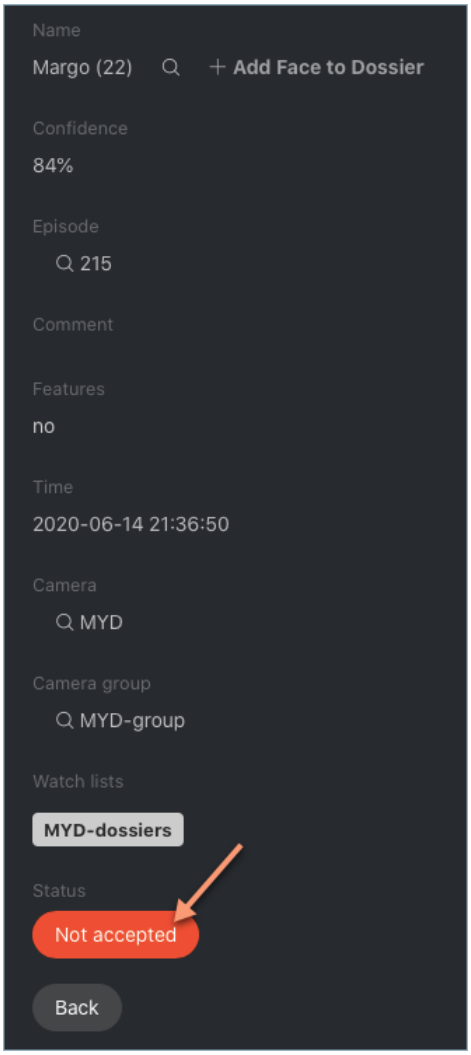
Filter events by face features and liveness when needed.



Event Ticket. Acknowledging Event

In order to navigate to an event ticket from the list of events, click on the face recognition result in a notification (*No matches* or the name of a matching person).

An event ticket contains the same data as a relevant *notification*. It also allows for acknowledging the event. To do so, click *Not accepted* to change the event acknowledgment status. Click *Save*.



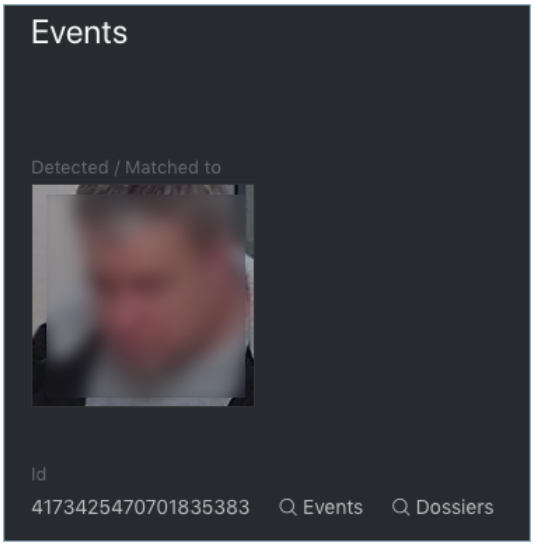
Tip: If a detected face has a match in the dossiers, you can navigate into a relevant one by clicking on the person's name in the event ticket.

Tip: In order to acknowledge all the events, click  above the list of events.

Note: Event acknowledgment can be automated for selected watch lists.

Event Ticket. Face Search

FindFace allows you to search detected faces through the list of events and dossier database. To navigate from an event ticket to the search tab, click *Events* or *Dossiers* respectively.



See also:

- *Search Faces in Databases*.

Organize Events with Episodes

This section is about the *Episodes* tab.

See also:

- *Work with Events*

An episode is a set of identification events that feature faces of the same person, detected within a specific period of time. As events on the *Events* tab show up in an arbitrary order, a large number of miscellaneous events can make the work difficult and unproductive. With the episodes, the system uses AI to organize incoming events based on the faces similarity and detection time. This allows for easy processing of diverse events, even in large numbers.

Tip: Search for faces through the event database and dossier database on the *Search* tab.

Tip: To perform the face identification in archived videos, see *Face Identification in Offline Videos*.

In this chapter:

- *About Episodes*
- *Episode Settings*
- *Grant Rights for Episodes*
- *View Episodes*
- *Event and Episode Acknowledging*
- *Filter Events by Episode ID*

About Episodes

An episode is a set of identification events that feature faces of the same person, detected within a certain period of time.

There are two types of episodes:

- LIVE: an episode is currently active, with more events to be possibly added.
- Closed: an episode is closed, no events can be added.

Episode Settings

To configure the episodes, use the `/etc/findface-security/config.py` configuration file. You need the following parameters into the `FFSECURITY` section:

- `EPISODE_SEARCH_INTERVAL`: The period of time preceding an event, within which the system searches the biometric database for events with similar faces. If no such an event is found, the system creates a new episode. Otherwise, it picks up the most relevant event from a LIVE episode after sorting out the 100 most recent similar faces.

  Note: The threshold similarity in episodes differs from that for face verification. See *General Preferences*.

- `EPISODE_MAX_DURATION`: The maximum episode duration in seconds. After this time, an episode automatically closes.
- `EPISODE_EVENT_TIMEOUT`: The maximum time in seconds since the last event has been added to an episode. After this time, an episode automatically closes.
- `EPISODE_KEEP_ONLY_BEST_EVENT`: When closing an episode, delete all events in it, except the one with the best face. Use this option to save disk space.

```
sudo vi /etc/findface-security/config.py

...

FFSECURITY = {
      ...
      'EPISODE_KEEP_ONLY_BEST_EVENT': True,
```

(continues on next page)

(continued from previous page)

```
    'EPISODE_SEARCH_INTERVAL': 60,
    'EPISODE_MAX_DURATION': 300,
    'EPISODE_EVENT_TIMEOUT': 30,
    ...
}

...
```

See also:

To see episodes work, navigate to the *Episodes* tab. See *Organize Events with Episodes* for details.

Grant Rights for Episodes

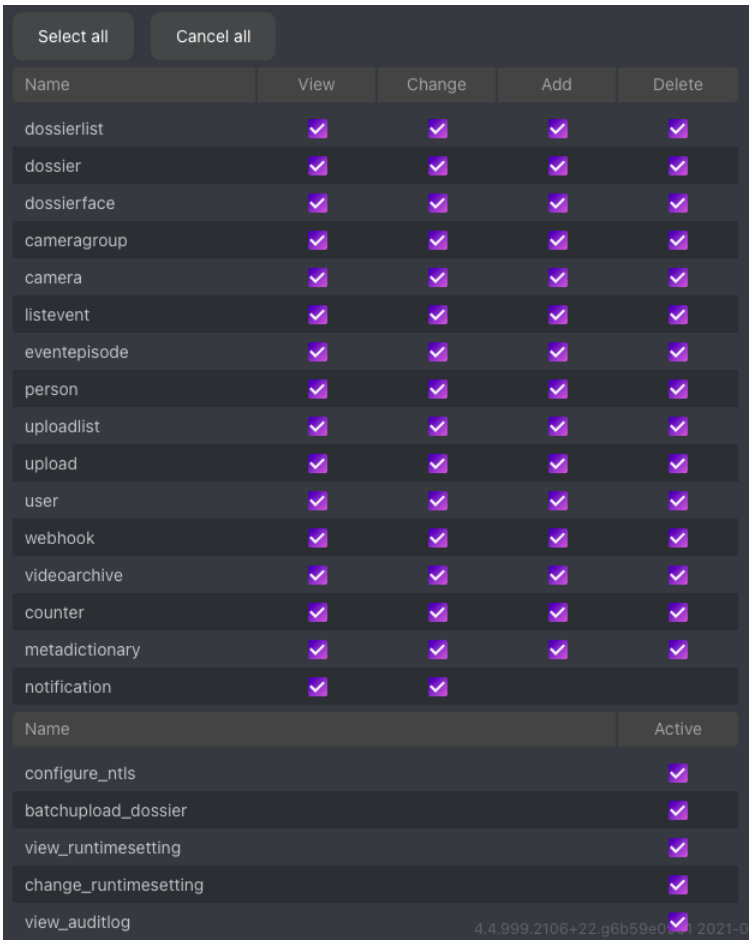
A user receives a notification of a new episode if they have rights for the first event. Viewing new events in the episode also requires proper rights.

The right for an event consists of the rights for a corresponding camera and watch list.

Note: To see unmatched events, you only need the rights for a camera.

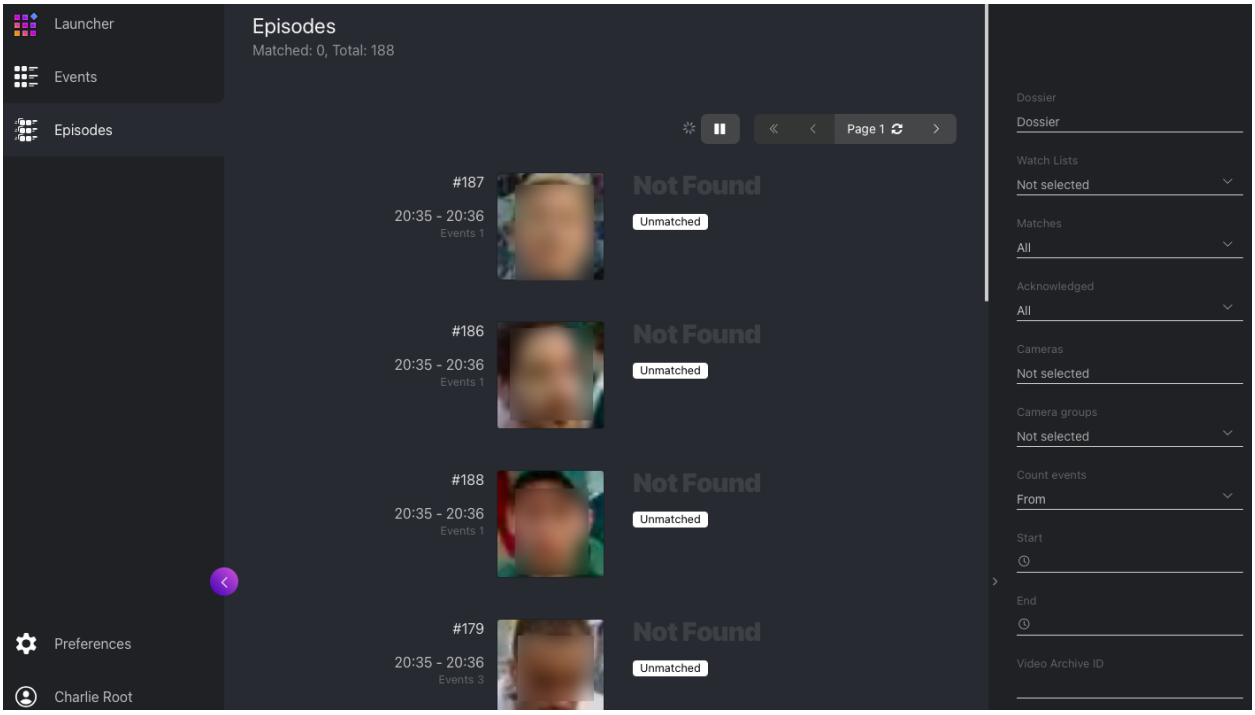
To manage rights of a role for the entire `Episode` entity, open permissions for this role and adjust the `eventepisode` permission.

Tip: See *User Management*.



View Episodes

You can find the list of episodes with filters and statistics on the *Episodes* tab. Once a face is detected, it is either added to an existing LIVE episode, or used as a starting point of a new episode. Each episode is assigned an identifier which can be later used to filter events and episodes.



When working with episodes, the following default filters may come in handy:

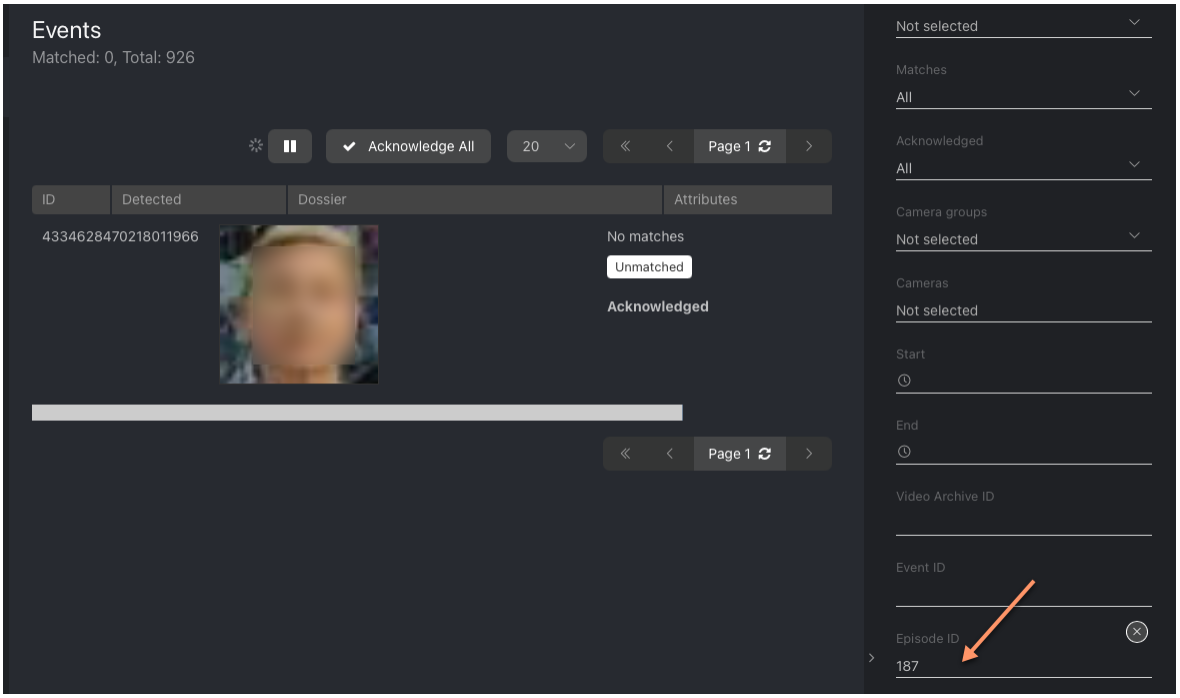
- *Dossier*: display episodes only for a selected dossier.
- *Watch lists*: display episodes only for a selected dossier category (watch list).

Note: To view only unmatched faces on the episode list, select *Unmatched* in this filter.

- *Matches*: display episodes only with/without matches, or all episodes.
- *Acknowledged*: display only acknowledged/unacknowledged episodes, or all episodes.
- *Cameras*: display only episodes from a selected camera.
- *Camera groups*: display only episodes from a selected group of cameras.
- *Start*, *End*: display only episodes that occurred within a certain time period.
- *Count from*: display only episodes with a given number of events.
- *Video Archive ID*: display episodes related to the video archive with a given ID.
- *Episode ID*: display an episode with a given ID.

You can also filter episodes by face liveness and face features (if applicable).

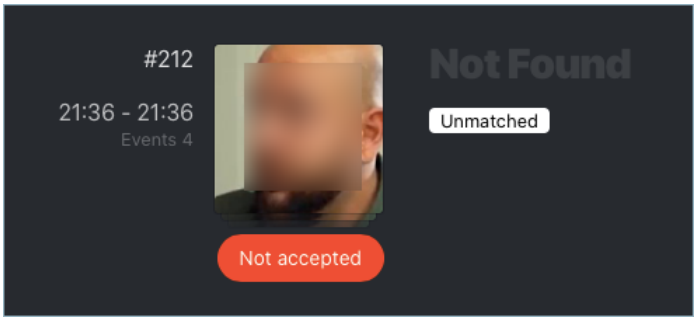
To view the events added to an episode, click it on the list. You will be redirected to the *Events* tab with the corresponding episode ID set in the *Episode* filter:



Work with the *Events* tab as described in *Work with Events*.

Event and Episode Acknowledging

To acknowledge an entire episode, click *Not accepted* for this episode on the list. As a result, all events in the episode will be automatically acknowledged, including those that are yet-to-appear (in the case of a LIVE episode).



An episode is also automatically acknowledged after acknowledging all its events one by one.

Filter Events by Episode ID

To display events by episode ID, either use the *id* filter on the *Episodes* tab or the *Episode ID* filter on the *Events* tab.

2.2.7 Face Identification in Offline Videos

Besides real-time face identification, FindFace allows for offline video processing. This functionality has a wide range of possible applications, among which the most common case is face detection and recognition in archived videos.

In this chapter:

- *Configure Offline Video Processing*
- *Process Video File*

Configure Offline Video Processing

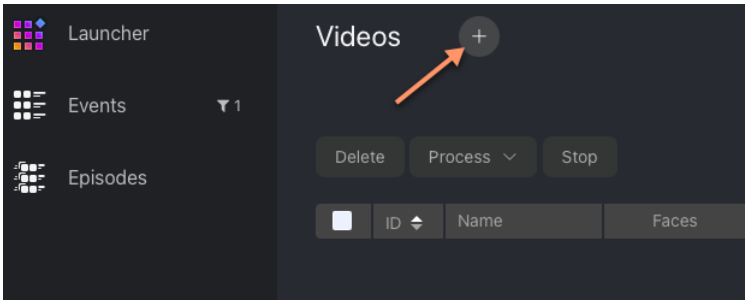
By default, video files are processed in a queued mode to prevent event drops due to resource overconsumption. You can modify the default number of simultaneously processed video files. To do so, open the `/etc/findface-security/config.py` configuration file and change the `MAX_VIDEO_ARCHIVE_JOBS` parameter. Please contact our experts prior (support@ntechlab.com) to make sure your resources are enough.

```
sudo vi /etc/findface-security/config.py

...

FFSECURITY = {
...
    # maximum concurrent video manager jobs for video archives processing
    'MAX_VIDEO_ARCHIVE_JOBS': 3,
    ...
}
...
```

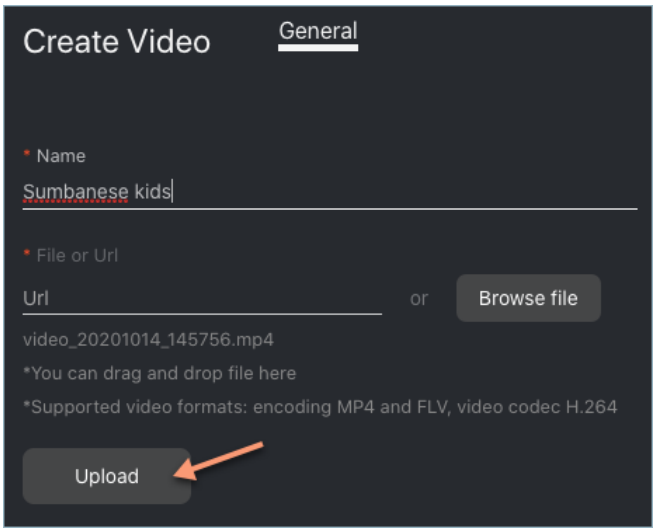
Process Video File

To identify faces in an offline video, do the following:

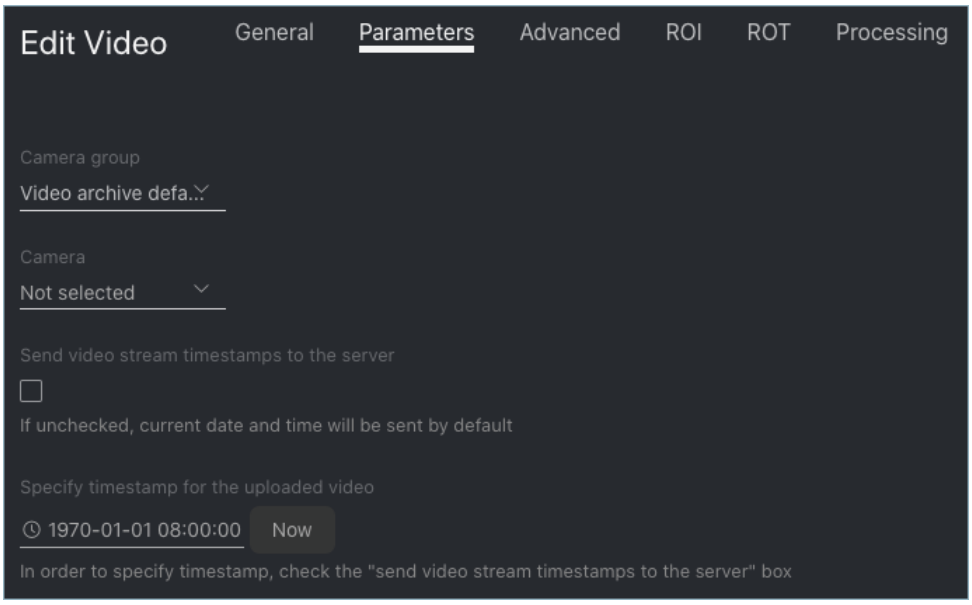
1. To process offline videos, you need a camera group. You can create a new camera group with basic settings or use the Video Archive default camera group. After the camera group is chosen, assign it to all *watch lists* that you want to monitor when processing the video.
2. Create a video in FindFace by uploading it from a file or online storage/cloud. To do so, navigate to the *Videos* tab.
3. Click +.



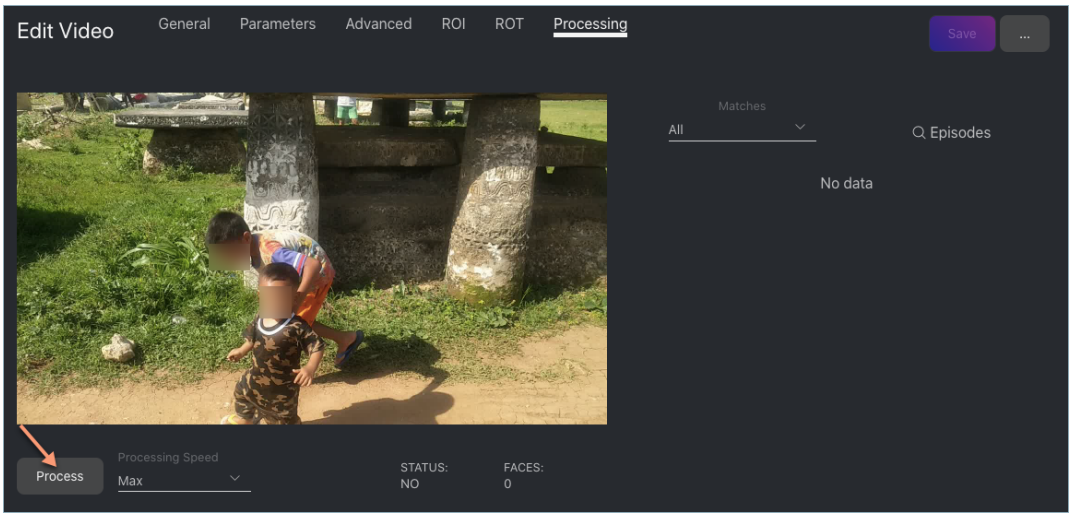
4. Specify the video name.



5. Specify the video URL in an online storage, or select a video file.
6. Click *Upload*.
7. After the video is uploaded, navigate to the *Parameters* tab. Select the camera group you have chosen.



8. (Optional) Select a camera to which you want to attribute the face recognition events found in the video.
9. (Optional) Configure the timestamps for face recognition events.
10. (Optional) On the *Advanced*, *ROI*, *ROT* tabs, specify parameters of video processing in the same manner as you do when configuring a *camera*.
11. Navigate to the *Processing* tab. Click *Process* to start face identification.



You can view face identification events right here, as well as on the *Events* and *Episodes* tabs by filtering the list of events by the camera group/camera associated with the video.

2.2.8 Search Faces in Databases

FindFace allows you to search for faces in the following databases:

- Database of detected faces (the *Events* tab).
- Dossier database (the *Dossiers*). Contains face reference images.

To find a face in a database, navigate to the *Search* tab.

In this chapter:

- *Search for Faces in Event List*
- *Search for Faces in Dossier List*

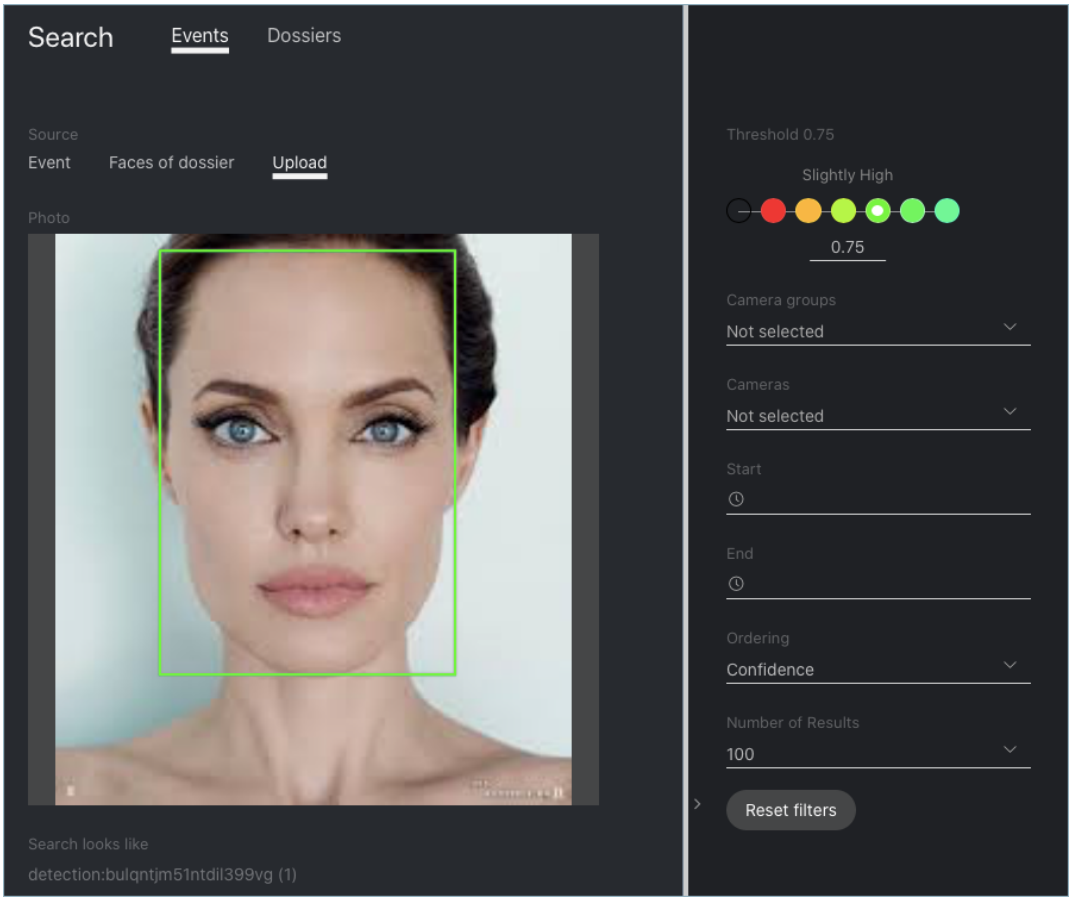
Search for Faces in Event List

FindFace allows you to search the database of detected faces.

Note: You can access this database by navigating to the event list (the *Events* tab).

To find a face, do the following:

1. Navigate to the *Search* tab.



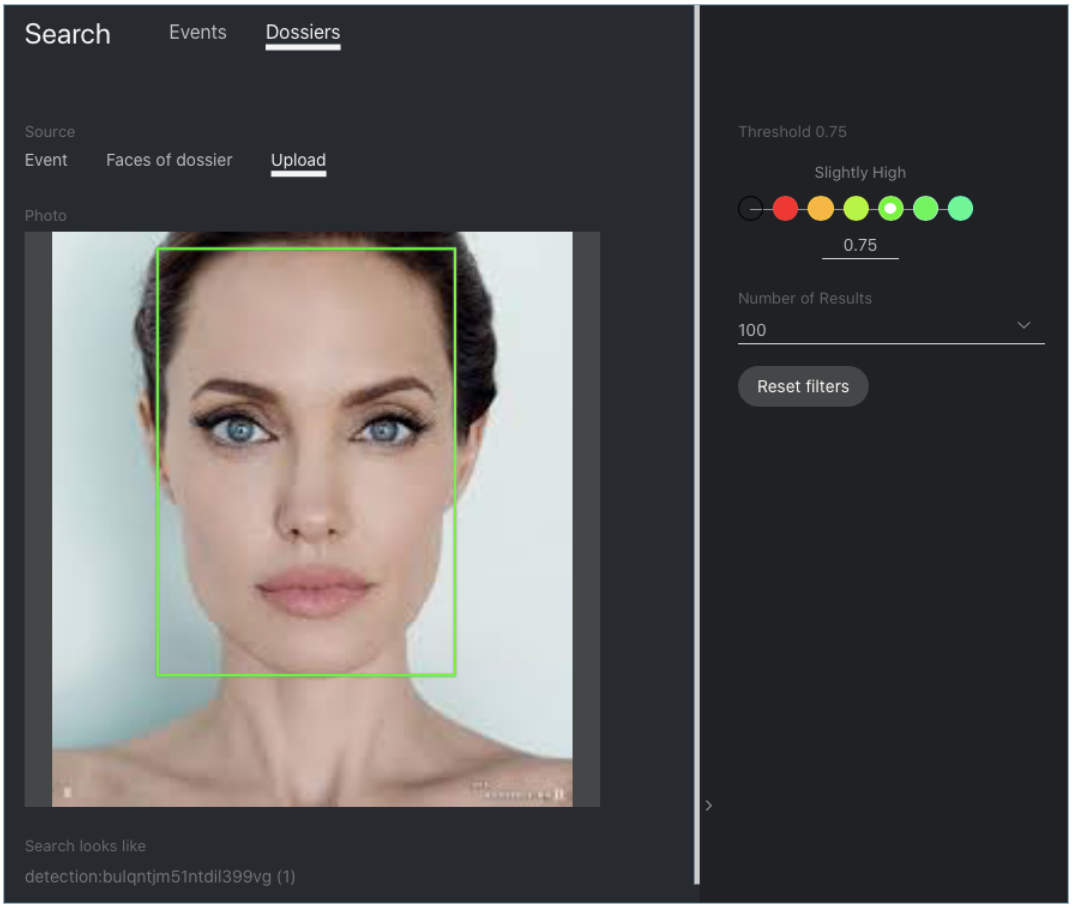
2. Specify a database to search: *Events*.
3. Specify a face to search for in one of the following ways:
   - By event ID with the face.
   - By dossier ID with the face. Should the dossier contain multiple photos, select some of them to use in the search.
   - By uploading a photo. It will be displayed in the *Photo* area. If there are multiple faces in the image, select the one of your interest.
4. By default, the system searches for faces using the identification threshold 0.75. If necessary, set your own value using the *Threshold* filter.
5. (Optional) Specify a group of cameras, camera and a time period within which the event occurred.
6. Select the method for ordering the search results: by confidence (similarity between faces) or date.
7. Specify the maximum number of dossiers in the search results.
8. Click *Search*. You will see the search results appear below. For each face found, the matching confidence level is provided.

Search for Faces in Dossier List

FindFace allows you to search the database of dossiers containing face reference images.

To find a face, do the following:

1. Navigate to the *Search* tab.

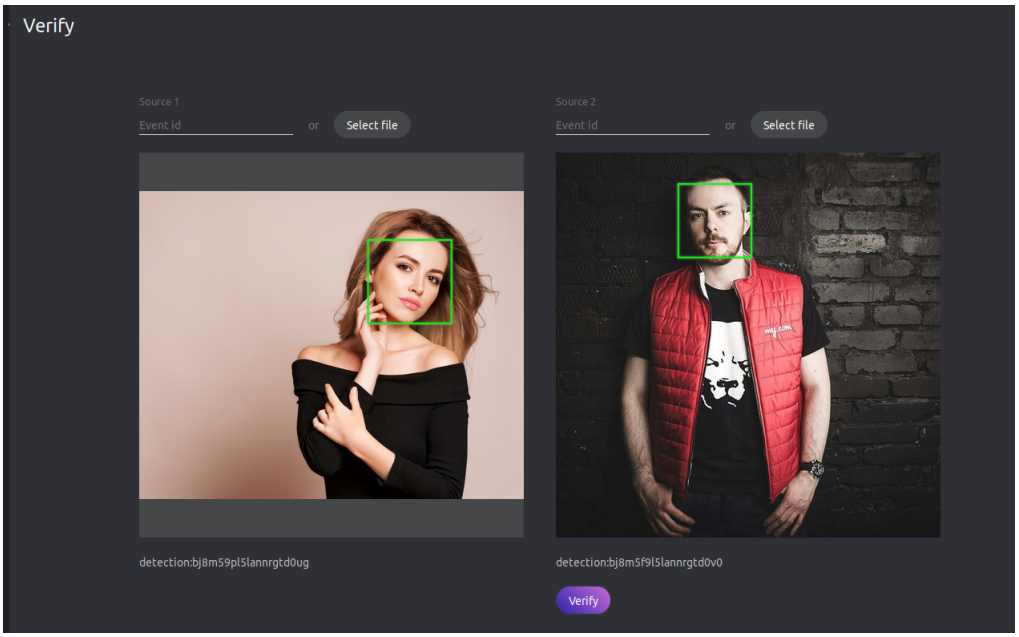


2. Specify a database to search: *Dossiers*.
3. Specify a face to search for in one of the following ways:
   - By event ID with the face.
   - By dossier ID with the face. Should the dossier contain multiple photos, select some of them to use in the search.
   - By uploading a photo. It will be displayed in the *Photo* area. If there are multiple faces in the image, select the one of your interest.
4. By default, the system searches for faces using the identification threshold 0.75. If necessary, set your own value using the *Threshold* filter.
5. Select the method for ordering the search results: by confidence (similarity between faces) or date.
6. Specify the maximum number of dossiers in the search results.
7. Click *Search*. You will see the search results appear below. For each face found, the matching confidence level is provided.

2.2.9 Compare Two Faces

FindFace allows you to compare 2 faces. Do the following:

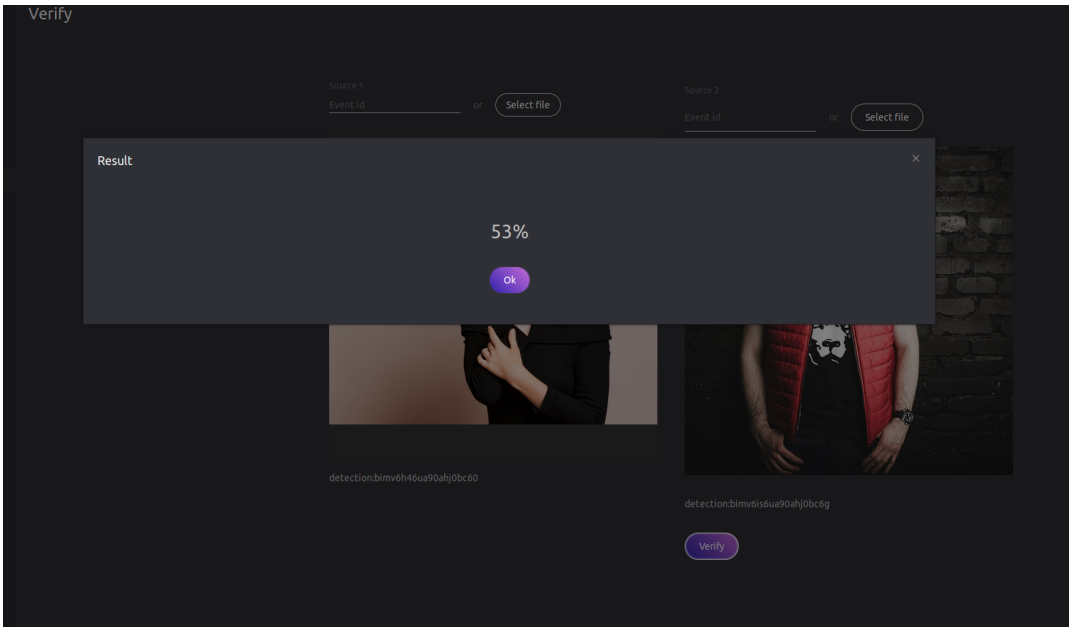
1. Navigate to the *Preferences* tab. Click *Verify*.



2. Specify the IDs of events that feature the faces you want to compare, and/or upload photos with the faces.

Tip: You can find event IDs on the *Events* tab.

3. Click *Verify*. You will see the probability of the faces belonging to the same person appear.



2.2.10 Person Recognition and People-Related Analytics

FindFace is an ideal tool to gather people-related analytics. Enable person recognition first and then make the most of it with our analytical features.

Person Recognition

FindFace allows for automatic person recognition. The system on-the-fly recognizes faces belonging to the same person and clusters them, building a person gallery. You can work with the person gallery on the *Persons* tab.

Note: In the person clusterization is enabled, the system databases will hold the `person event` entity linked to all *episodes* that feature a person's face.

Important: By default, person clusterization is disabled. *Enable and configure it* via the `/etc/findface-security/config.py` configuration file.

In this section:

- *Clusterization Methods*
- *Enable and Configure Person Clusterization*
- *Work with Person Gallery*

Clusterization Methods

FindFace uses the following methods to cluster faces belonging to the same person:

- Dynamic clusterization. The clusterization takes place on-the-fly after an episode is closed. The result of dynamic clusterization is shown in real-time on the *Persons* tab.

  Note: The technical details are as follows. Not every episode is qualified: the number of events in it must be equal or greater than `PERSON_EVENT_MIN_EPISODE_EVENTS` (set up via the `/etc/findface-security/config.py` configuration file). If an episode meets this requirement, the system selects the best quality event and performs the following operations:

  - Creates a new entity `PersonEvent` in the main system database **`PostgreSQL`**. The entity contains the event metadata, a link to the parent episode, face biometric sample, and thumbnail.
  - Searches for a similar face centroid in the `person_events` gallery of the **`Tarantool`** biometric database. A face centroid is a virtual biometric sample averaged across all person's faces that have been detected so far. If a similar centroid is found, the system updates it using the new event. Otherwise, it creates a new centroid.

- Scheduled clusterization. We recommend scheduling it on late night hours as it takes up a lot of CPU resources and time.

Note: The schedule is defined in the RRULE format as `PERSONS_CLUSTERIZATION_SCHEDULE` in the `/etc/findface-security/config.py` configuration file. The rest of the technical implementation resembles the dynamic method. However, the face centroid quality is better in the scheduled method as centroids are averaged across a larger array of accumulated biometric samples.

Important: The scheduled clusterization completely overwrites the person gallery, including ids.

Enable and Configure Person Clusterization

By default, person clusterization is disabled. To enable it, open the `/etc/findface-security/config.py` configuration file and modify the `SERVICES` section as such:

```
sudo vi /etc/findface-security/config.py

...
SERVICES = {
    "ffsecurity": {
        ...
        "persons": True,
    }
...
```

You will see the *Persons* tab appear in the FindFace web interface.

In the same configuration file, you can modify the following parameters:

- `PERSON_EVENT_MIN_QUALITY`: minimum quality of faces used in person clusterization.
- `PERSON_EVENT_MIN_EPISODE_EVENTS`: minimum number of events in episodes used in person clusterization.
- `PERSONS_CONFIDENCE_THRESHOLD`: confidence threshold to match a face to a person.

  Warning: Consult with our experts by support@ntechlab.com before changing this parameter.

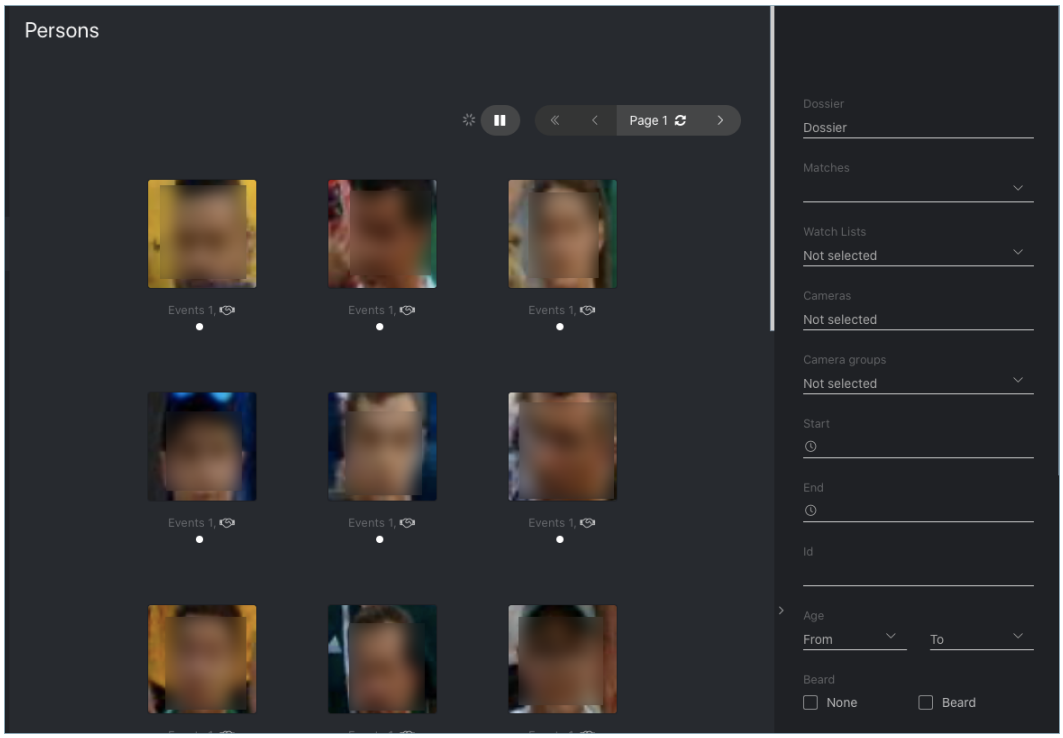
- `PERSONS_CLUSTERIZATION_SCHEDULE`: recurrence rule (RRULE) for scheduling person clusterization.

  Tip: See the RRULE calculator here.

```
# -- Persons configuration --
# rrule (recurrence rule) for scheduling persons clusterization
# WARNING: all scheduling works with UTC time and NOT aware of any timezone
'PERSONS_CLUSTERIZATION_SCHEDULE': 'RRULE:FREQ=DAILY;INTERVAL=1;WKST=MO;BYHOUR=0;
↪BYMINUTE=0',
# face to person matching confidence threshold
'PERSONS_CONFIDENCE_THRESHOLD': 0.739,
# minimum required face quality for person creation
'PERSON_EVENT_MIN_QUALITY': 0.45,
# minimum required number events in episode for person creation
'PERSON_EVENT_MIN_EPISODE_EVENTS': 1,
```

Work with Person Gallery

To see the person gallery, navigate to the *Persons* tab.



To work with the person gallery, use the following filters:

- Dossier
- Matches
- Cameras
- Camera groups
- Watch lists
- Time period
- Person id
- Face features (if enabled)
- Liveness (if enabled)

See also:

- *Configuration file of findface-security*
- *Webhooks*
- *Social Interaction Analysis*
- *Video Analytics*

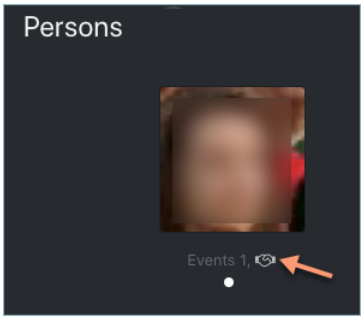
Social Interaction Analysis

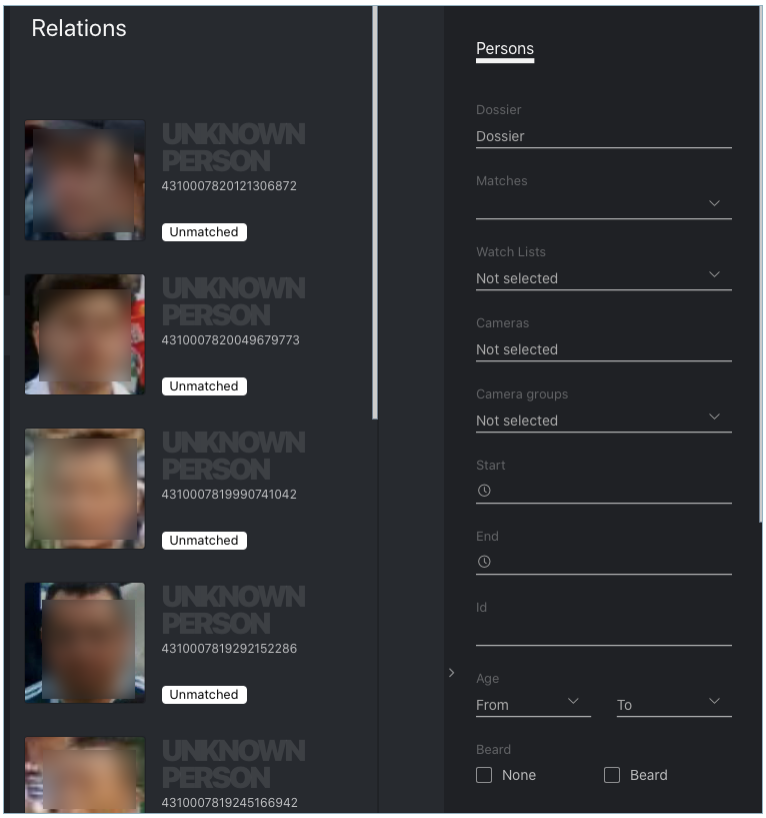
It is possible to see a circle of people with whom a person has previously been in contact. For each person from the first circle, the system determines another circle of connected people, and so on. Overall, social interaction analysis is three-circle deep.

Important: The social interaction analysis is provided only when the *person recognition* is enabled.

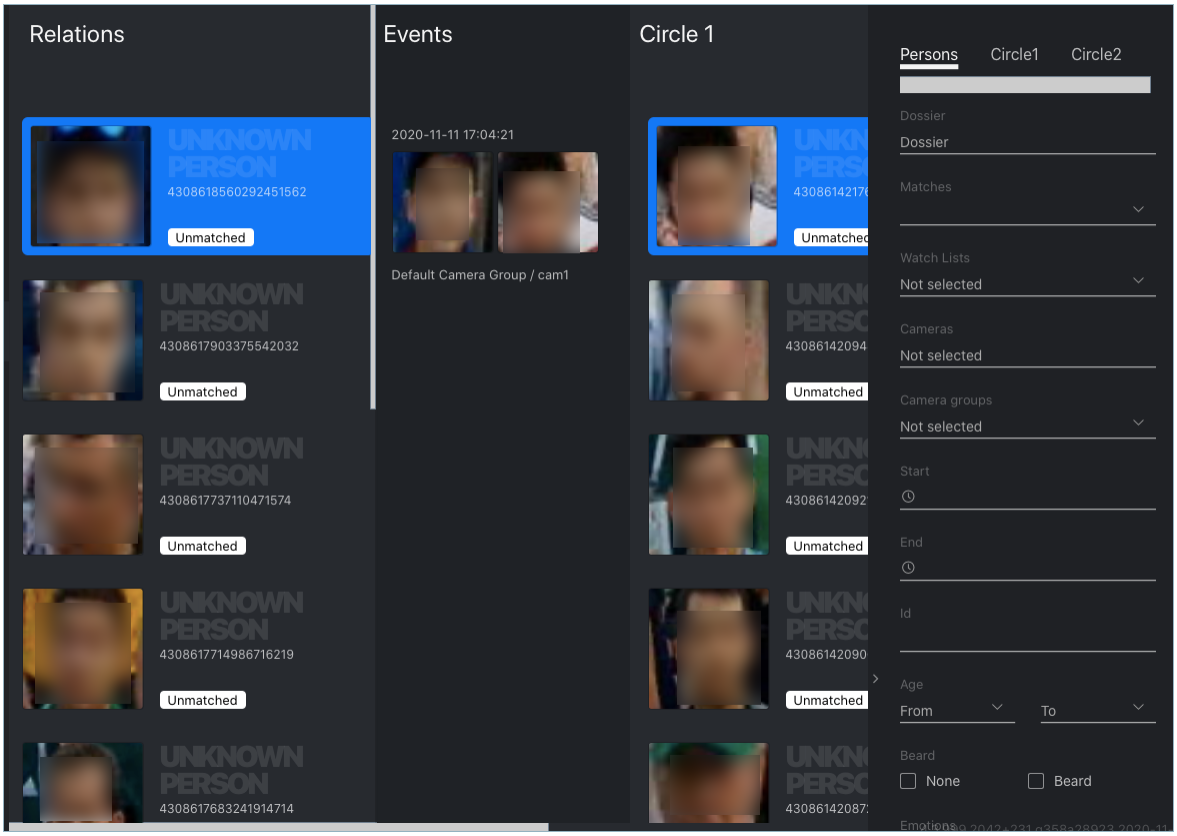
The social interaction analysis is available on the *Relations* tab.

Tip: You can also display the circle of connected people right from the *Persons* tab by clicking on the handshake icon.





On the *Relations* tab, click on a person to display their first circle of relations. Keep on to unveil the entire tree of social interactions.

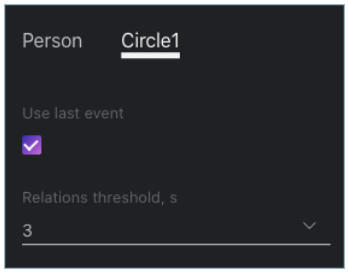


You can apply available filters to every circle.

Tip: For example, you can find older adults or people without a face mask who are directly or indirectly related to a potentially contagious person.

When searching through a circle of relations, apply the following settings:

- *Use the last event*: use the last event of an episode to analyze contacts between individuals. In this case, having found truly associated people is most probable as they simultaneously leave a camera's field of view. If the option is disabled, the system will use the best event of an episode for relations search.
- *Relations threshold*: maximum time in seconds between the appearance of individuals to consider them related.

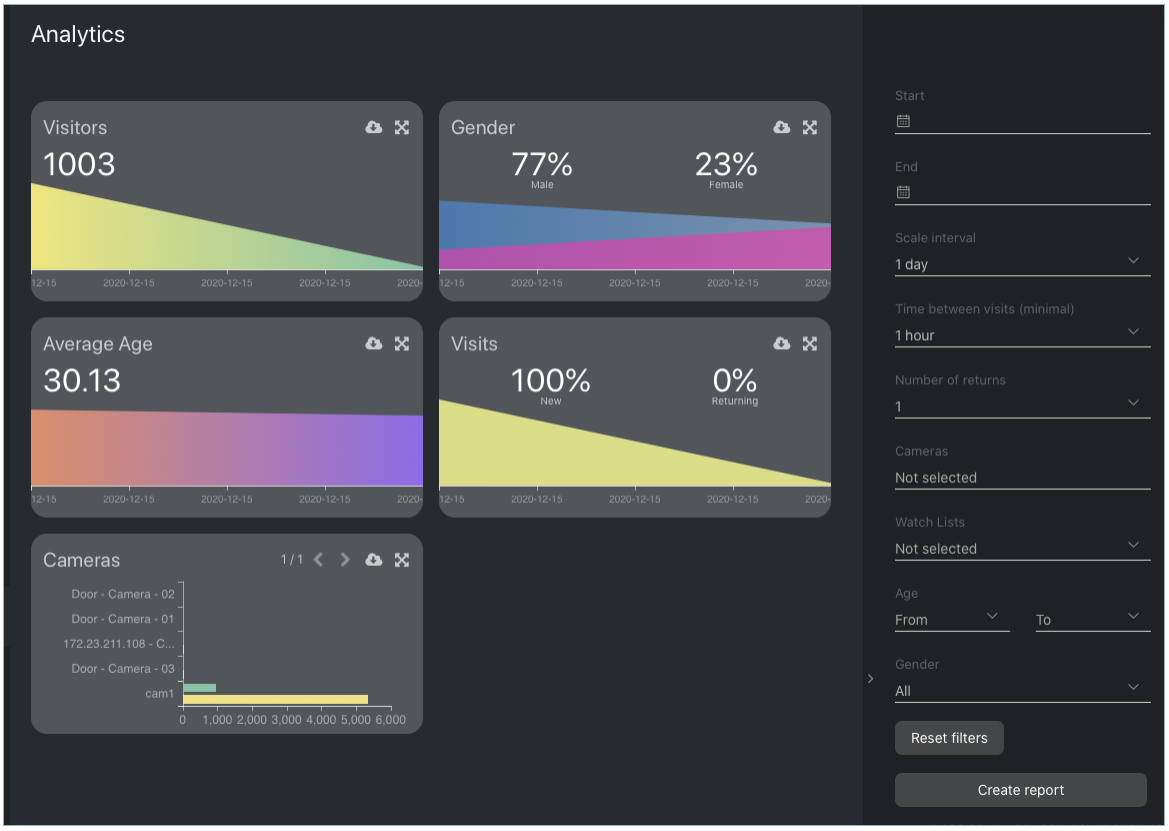


Video Analytics

Video analytics provided by FindFace includes statistics on the number of visitors, their gender, average age, most frequently visited zones (judged by most active cameras), and the character of visits (first-timers or returners). It is a great starter tool to incorporate the know your customer guidelines into your business.

The analytical data charts are available on the *Analytics* tab.

Important: The analytics is built only when the *person recognition* is enabled.



To work with the analytical data, use the following filters:

- Time period
- Scale interval
- Time between visits
- Number of returns
- Cameras
- Watch lists
- Age
- Gender

See also:

- *Person Recognition*

2.2.11 Reports

FindFace allows you to build reports on the following system entities:

- face recognition events
- episodes
- search events
- persons
- cameras
- dossiers
- analytical data

In this chapter:

- *Configure Saving Images in Reports*
- *Build Report*

Configure Saving Images in Reports

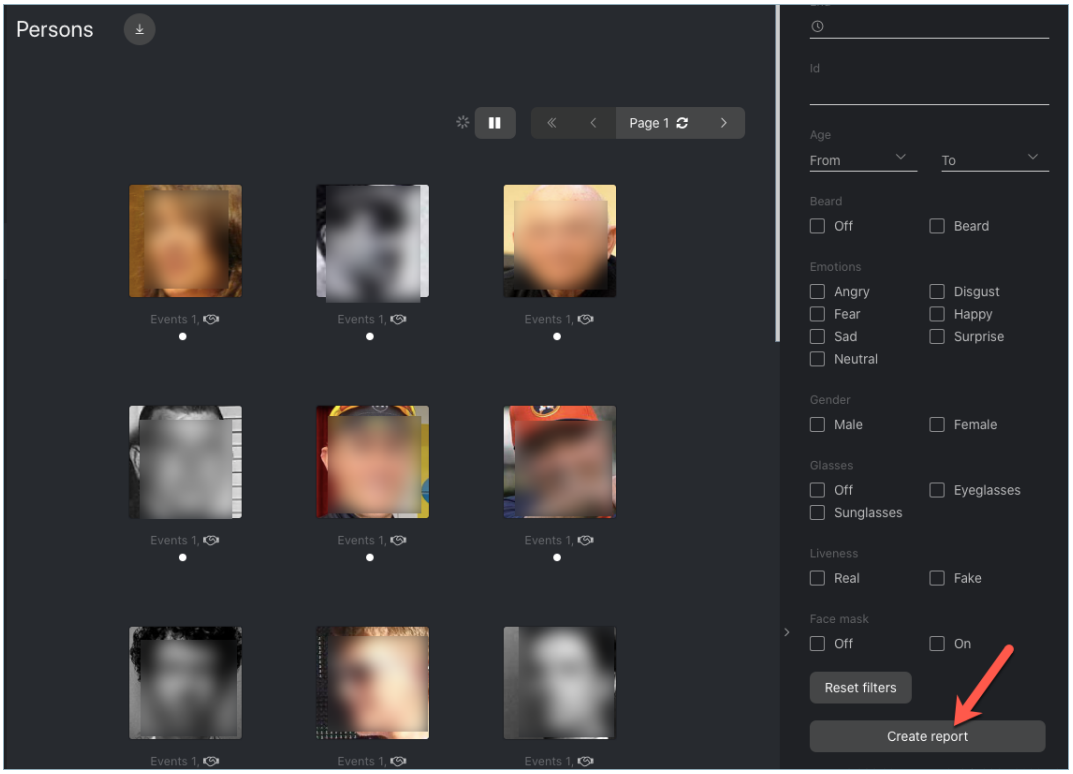
When building a report, you will be able to choose to save the report images as links, thumbnails, or full frames. It is possible to configure the image parameters. To do so, open the `/etc/findface-security/config.py` configuration file and alter the default JPEG quality and the maximum height of thumbnails and full frames, subject to your free disc space.

```
sudo vi /etc/findface-security/config.py

# reports image saving options
'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
'REPORT_FULLFRAME_JPEG_QUALITY': 75,
'REPORT_FULLFRAME_MAX_HEIGHT': 250,
```

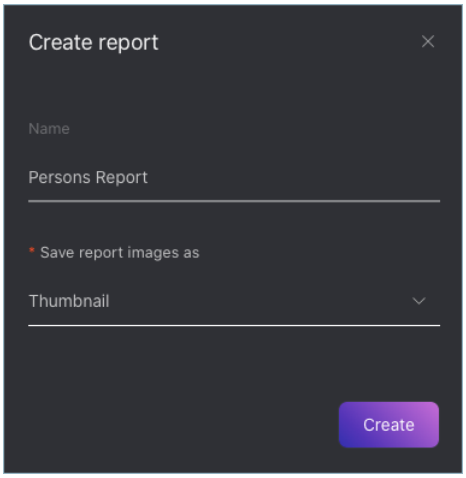
Build Report

To build a report, do the following:

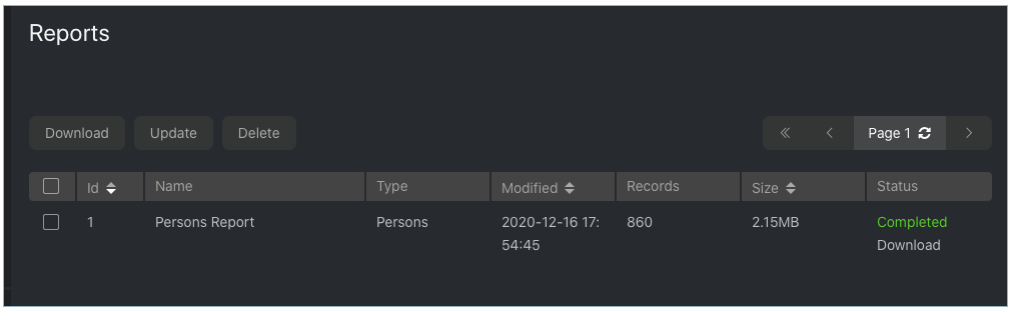
1. Navigate to the tab associated with the required entity: *Events*, *Episodes*, *Search*, *Persons*, *Cameras*, *Dossiers*, or *Analytics*.
2. Set the filters for the report.
3. Click *Create Report*.



4. Specify the report name. Choose whether to save the report images as links, thumbnails, or full frames. Click *Create*.



5. The report will be available for download on the *Reports* tab.



2.2.12 Video Wall

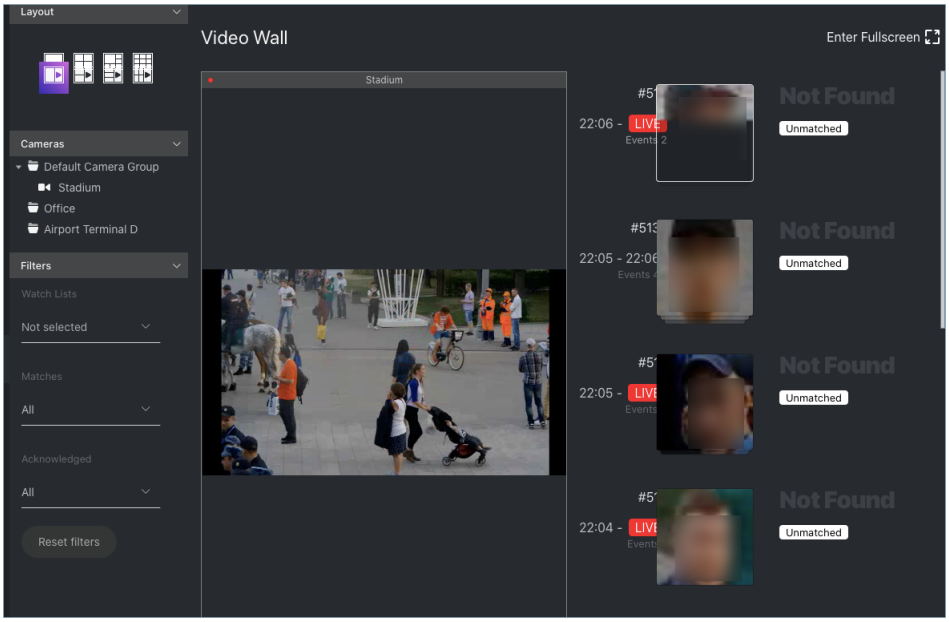
FindFace allows for basic video surveillance. Use the Video Wall to display the video image from cameras and video files.

The Video Wall offers two modes, 4 predefined layouts in each:

- video streaming,
- video streaming with face detection and episode feed.

To display video on the Video Wall, do the following:

1. Navigate to the *Video Wall* tab.
2. Select a Video Wall mode and camera layout.



3. Drag-n-drop cameras of your choice to the Video Wall.

You can work with the episode feed on the Video Wall in the *same manner* as with the *Episodes* tab, including the following basic filters:

- *Watch Lists*
- *Matches*.

- *Acknowledged*.

2.2.13 FindFace Mobile App

To interact with FindFace on the go, use the FindFace mobile app. The app is available on request for Android.

In this section:

- *Recommended System Requirements*
- *Authentication*
- *Settings*
- *Functions*
- *App Logs*

Recommended System Requirements

- OS: Android 6.

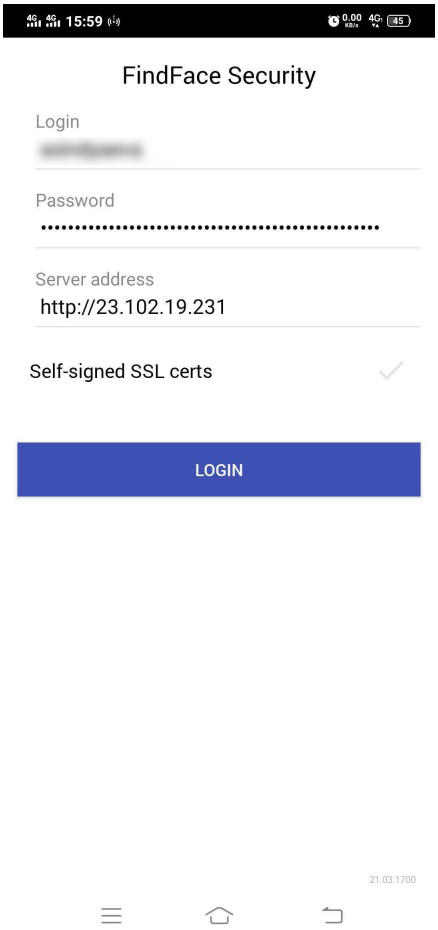
  Note: Functioning on versions 4.4-5. is possible but not guaranteed.

- CPU: 2 cores.
- RAM: 2 GB.
- Battery capacity: at least 4000 mAh for an 8-hour shift.

Authentication

To log in to the mobile app, enter your FindFace login and password and the FindFace server URL.

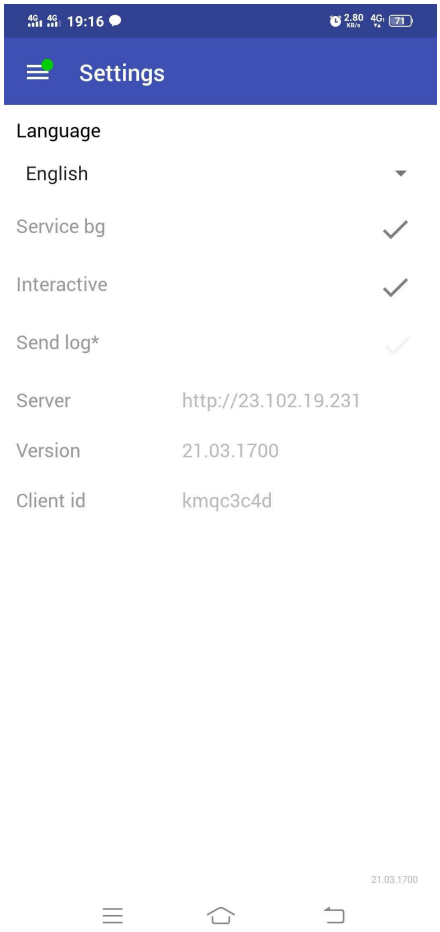
Important: By default, the app doesn't trust self-signed SSL certificates. If the FindFace server uses such a certificate, check *Self-signed SSL certs* and restart the app before logging in.



Settings

To set up your mobile app, do the following:

1. Click *Settings* in the main menu.
2. Enter the PIN code to open the settings (`1234` by default).
3. If necessary, switch the app language.

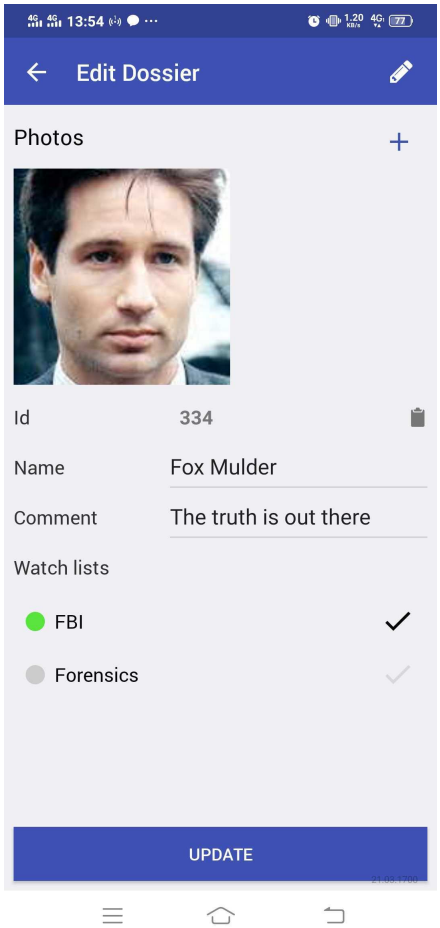


4. For the app to keep working after being switched to the background, check *Service bg* (enabled by default).
5. To add the interactive glasses mode to the app, check *Interactive*. In this mode, the application will be sending the information about new face matches to the smart glasses connected to your smartphone. If you need this functionality, be sure to contact your manager prior, as there are some restrictions.
6. Check *Send log* to send the app logs to the FindFace server.

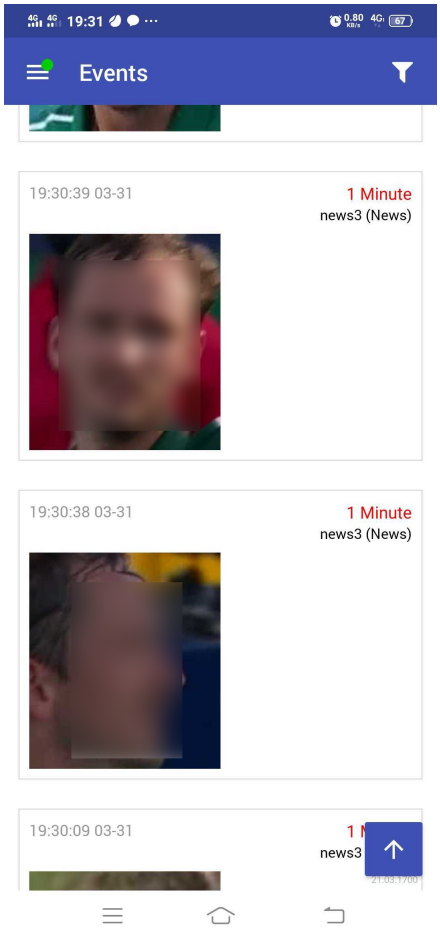
Functions

The mobile app has a highly intuitive and handy design and provides the following on-the-go functionality:

- View and create a dossier on a person.

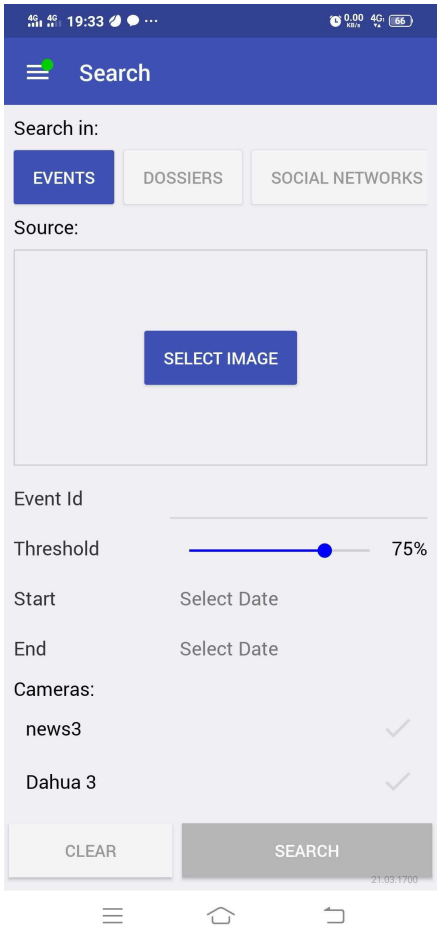


- Get notifications on face identification events.



Important: To receive the push notifications on events in the mobile version, open a relevant watch list settings in the full version, and check *Require Event Acknoledgment* and *Enable Sound Alert*.

- Search for faces in the event list and dossier database.



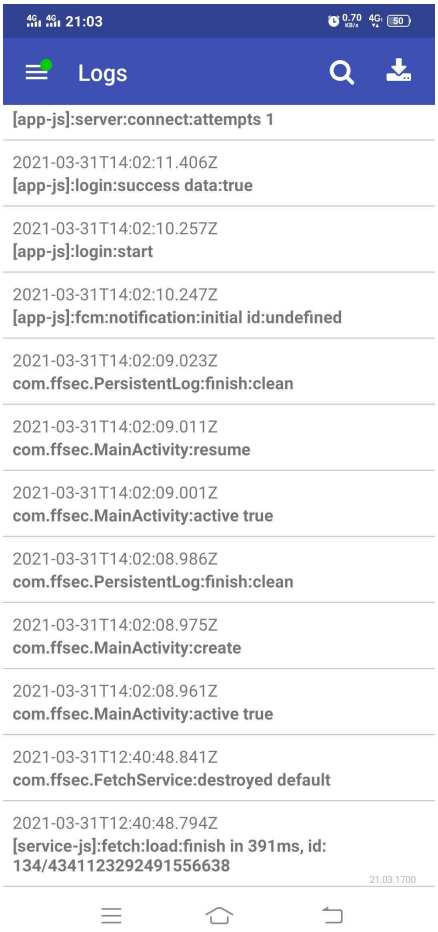
Working with the mobile app is similar to the full version.

See also:

- *Create Dossier Manually*
- *Work with Events*
- *Search Faces in Databases*

App Logs

In the case of the app failure, navigate to the *Logs* tab. Export the app logs and send them to our support team (support@ntechlab.com).



2.3 Advanced Functionality

2.3.1 Allocate `findface-video-worker` to Camera Group

In a distributed architecture, it is often necessary that video streams from a group of cameras be processed *in situ*, without being redistributed across remote `findface-video-worker` instances by the principal server.

Note: Among typical use cases are hotel chains, chain stores, several security checkpoints in the same building, etc.

In this case, allocate the local `findface-video-worker` to the camera group.

Do the following:

1. Navigate to the *Preferences* tab. Click *Camera Groups*.
2. Open the camera group settings.
3. In the *Labels*, create or select one or several allocation labels. Save changes.
4. Open the `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) configuration file and specify the allocation labels in the following format: `label_name=true` (label `terminal_1` in the example below).

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini

labels = terminal_1=true
```

5. Restart `findface-video-worker`.

```
sudo systemctl restart findface-video-worker-cpu.service
sudo systemctl restart findface-video-worker-gpu.service
```

Note: If a camera is assigned an allocation label, its video stream can be processed by a `findface-video-worker` instance with the same label, as well as by all unlabeled `findface-video-worker` instances.

Warning: If a labeled camera is processed by an unlabeled `findface-video-worker` instance and a free similar-labeled instance appears, the camera won't automatically switch to the latter. To switch the camera, restart the similar-labeled `findface-video-worker` instance.

2.3.2 Distributed Dossier Database

In a distributed architecture, it is often necessary to have the dossier database distributed among several hosts.

In the current implementation, the dossier database is available for editing only on the principal server known as `master`. It is in sync with several additional FindFace instances that serve as `slaves`. On the slaves, the dossier database is available only for reading and monitoring.

Important: You will be able to delete dossiers on the slaves if the master is unavailable.

Important: If a watch list on the future slave already contains dossiers, they will be replaced with those from the master during the first synchronization. All information in the previous dossiers will be lost.

Warning: Neural networks on the master and slaves must be identical.

In this section:

- *Configure Master/Slave Synchronization*
- *Replicate Watch List from Master to Slaves*
- *Set Synchronization Time*
- *Cancel Watch List Replication and Synchronization*
- *Duplicate Functionality to Web Interface*

Configure Master/Slave Synchronization

To configure master/slave synchronization, do the following:

1. On the master, open the `/etc/findface-security/config.py` configuration file. Come up with a synchronization token and specify it in the `SYNC_TOKEN` parameter (be sure to uncomment it prior).

```
sudo vi /etc/findface-security/config.py

...

# ========== DossierLists sync ===========
...
# token must be identical on master and slave
# use pwgen -s 64 1
SYNC_TOKEN = 'ABC_123456789'
...
```

2. On the slave(s), uncomment the `SYNC_TOKEN` parameter in the `/etc/findface-security/config.py` configuration file and paste the created synchronization token into it. The tokens on the master and slaves must be identical.

The master/slave sync is now set and will be enabled once you configure a watch list replication from the master to slave(s).

Replicate Watch List from Master to Slaves

To replicate a watch list from the master to slave instances, send a POST request to the slave with the following parameters in the body:

- `remote_dossier_list`: id of the original watch list on the master
- `remote_url`: master URL
- `slave_dossier_list`: id of the watch list on the slave, which is to be a replica of the original watch list

```
POST /sync/dossier-lists/
{remote_dossier_list: 1,
remote_url: "http://172.17.46.14",
slave_dossier_list: 3}
```

Set Synchronization Time

To schedule dossier synchronization, do the following:

1. Open the `/etc/findface-security/config.py` configuration file on the master.

```
sudo vi /etc/findface-security/config.py
```

2. Uncomment and set the following parameters:
   - `SYNC_SCHEDULE`: recurrence rule (RRULE) that defines the sync schedule.

   Tip: See the RRULE calculator here.

- SYNC_AT_STARTUP: if True, synchronization occurs on the FindFace startup and restart.
- SYNC_AT_CREATION: if True, synchronization immediately occurs after you set up synchronization for a watch list.

```
...

# ========== DossierLists sync ===========
...

# rrule that defines sync schedule
SYNC_SCHEDULE = 'RRULE:FREQ=DAILY;WKST=MO;BYHOUR=11;BYMINUTE=0'
# if True synchronization will occur on FindFace startup and restart
SYNC_AT_STARTUP = True
# if True synchronization will occur immediately after creating synchronization for␣
↪dossier list
SYNC_AT_CREATION = True
```

3. Uncomment the mentioned above parameters on each slave. The parameter values can be arbitrary.

Cancel Watch List Replication and Synchronization

To cancel a watch list replication and synchronization, send the following API request to the slave with the {id} of the watch list on the slave:

```
DELETE /sync/dossier-lists/{id}/
```

Duplicate Functionality to Web Interface

By default, you can enable and disable watch list replication only via API. To make the functionality available in the web interface as well, do the following:

1. Open the /etc/findface-security/config.py configuration file on the master.

```
sudo vi /etc/findface-security/config.py
```

2. Enable the ffsecurity_sync plugin by uncommenting the line INSTALLED_APPS.append('ffsecurity_sync') into the plugins section:

```
...

# ========== DossierLists sync ===========
INSTALLED_APPS.append('ffsecurity_sync')
...
```

3. Do the same on each slave.
4. On each host, migrate the main database architecture from FindFace to **PostgreSQL**. Re-create user groups in the main database. Restart the findface-security service.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo systemctl restart findface-security.service
```

2.3.3 Dossier Custom Tabs, Fields, and Filters

It is often necessary that a dossier feature additional tabs and fields in the web interface.

See also:

To create dossier custom fields in your Tarantool-based biometric database, see *Dossier Face Custom Metadata in Tarantool*.

To add custom tabs and fields to a dossier, do the following:

1. Prepare the list of custom tabs and fields you want to add to a dossier.
2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

3. Into the `FFSECURITY` section, uncomment the `CUSTOM_FIELDS` section and modify the exemplary content, considering the following:
   - `'items'`: the list of fields in a dossier. Describe each field with the following parameters:
     - `'name'`: field's internal name, string.
     - `'label'`: field's label in the web interface, string.
     - `'display'`: display format (`form` or `list`), string or array.
     - `'tab'`: tab that features the field. If not specified, the field appears on the main dossier page (that with a photograph).
     - `'editable'`: field's editability, boolean.
     - `'type'`: field data type, string. Possible values:
       * `list`: requires `items`, additional parameter for lists (see below), expects objects {id, name} in dictionaries;
       * `valuelist`: expects elements of primitive types.
       * `objectlist`: allows for creating arrays of objects of required types.
       * `datetime`: primitive data type displayed as a datetime list.
       * `date`: primitive data type displayed as a date picker.
       * `boolean`: primitive data type displayed as a checkbox.
       * `string`: primitive data type `string`.
     - additional parameters for lists (type=list, type=valuelist):
       * `multiple`: possibility of selecting several items in the list, boolean.
       * `items`: dictionary used as a data source for the list.
       * `allow_create`: possibility of adding new items to the list.
       * `custom_id`: custom field for id (type=list).
     - additional parameters for object lists (type=objectlist).
       * `object`: objects used as a data source for the object list.
       * `simple`: indicator that the field expects data of a primitive type instead of objects, for example, expects strings with phone numbers.
   - `'filters'`: the list of search filters associated with the custom fields. Parameters:

  - 'name': filter's internal name,
  - 'label': filter's label in the web interface,
  - 'field': associated field in the format [field name].

- 'tabs': the list of tabs in a dossier. The first listed tab corresponds to the main dossier page.

```
FFSECURITY = {

...

    # Edit CUSTOM_FIELDS section to customize dossier content.
    # Below is an example for integration FindFace Security with Sigur.
    'CUSTOM_FIELDS': {
        'dossier_meta': {
            'items': [
                {
                    'name': 'personid',
                    'default': '',
                    'label': 'PersonID',
                    'display': ['list', 'form'],
                    'description': 'Sigur person ID'
                },
                {
                    'name': 'firstname',
                    'default': '',
                    'label': 'First Name',
                    'display': ['list', 'form'],
                    'description': 'Sigur first name'
                },
                {
                    'name': 'lastname',
                    'default': '',
                    'label': 'Last Name',
                    'display': ['list', 'form'],
                    'description': 'Sigur last name'
                },
                {
                    'name': 'version',
                    'default': '',
                    'label': 'Version',
                    'display': ['list', 'form'],
                    'description': 'Sigur photo version'
                }
            ],
            'filters': [
                {
                    'name': 'personid',
                    'label': 'Sigur person ID filter',
                    'field': 'personid'
                }
            ]
        }
    },
```

4. You will see the custom content appear in the web interface.

2.3.4 Dossier Face Custom Metadata in Tarantool

It is often necessary to assign additional metadata to the dossier faces in your Tarantool-based biometric database.

See also:

To create dossier custom tabs, fields, and filters in the web interface, see *Dossier Custom Tabs, Fields, and Filters*.

To set the face custom metadata, do the following:

1. Prepare the list of custom meta fields you want to assign to a dossier face in Tarantool.
2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

3. Into the `FFSECURITY` section, uncomment the `CUSTOM_FIELDS` -> `dossier_face` section and modify the exemplary content, considering the following:
   - `field_name`: field name;
   - `type`: data type;
   - `default`: field default value. If a default value exceeds '1e14 – 1', use a string data type to specify it, for example, `"123123.."` instead of `123123...`

```
FFSECURITY = {

...

'CUSTOM_FIELDS': {
  ...
        'dossier_face': {
            'items': [
                {
                    "field_name": "tag_name_1",
                    "type": "string",
                    "default": "change_me"
                },
                {
                    "field_name": "tag_name_2",
                    "type": "uint",
                    "default": 123
                },
                {
                    "field_name": "tag_name_3",
                    "type": "bool",
                    "default": True
                },
            ]
        }
},
```

4. *Add the new meta fields* to the Tarantool database structure.
5. You can work with the new meta fields through *HTTP API* using the `dossier-faces` methods.

2.3.5 Console Bulk Photo Upload

To bulk-upload photos to the dossier database, you can use the **findface-security-uploader** utility from the FindFace package (in addition to the web interface upload functionality). Use this utility when you need to upload a large number of photos (more than 10,000).

Tip: To view the **findface-security-uploader** help, execute:

```
findface-security-uploader --help
```

Do the following:

1. Write the list of photos and metastrings to a CSV or TSV file.

   Important: The file used as a metadata source must have the following format: `path to photo | metastring`.

   To prepare a TSV file, use either a `script` or the `find` command.

   Note: Both the script and the command in the examples below create the `images.tsv` file. Each image in the list will be associated with a metastring coinciding with the image file name in the format `path to photo | metastring`.

   To build a TSV file listing photos from a specified directory (`/home/user/25_celeb/` in the example below), run the following command:

   ```
   python3 tsv_builder.py /home/user/25_celeb/
   ```

   The `find` usage example:

   ```
   find photos/ -type f -iname '*g' | while read x; do y="${x%.*}"; printf "%s\t%s\n" "
   ↪$x" "${y##*/}"; done
   ```

2. Create a job file out of a CSV or TSV file by using `add`. As a result, a file `enroll-job.db` will be created and saved in a current directory.

   ```
   findface-security-uploader add images.tsv
   ```

   The `add` options:

   - `--format`: input file format, `tsv` by default,
   - `--delimiter`: field delimiter, by default `"\t"` for TSV, and `","` for CSV.

   Note: A job file represents a sqlite database which can be opened on the **sqlite3** console.

3. Process the job file by using `run`.

   ```
   findface-security-uploader run --dossier-lists 2 --api http://127.0.0.1:80 --user␣
   ↪admin --password password
   ```

   The `run` options:

- `--parallel`: the number of photo upload threads, 10 by default. The more threads you use, the faster the bulk upload is completed, however it requires more resources too.
- `--all-faces`: upload all faces from a photo if it features several faces.
- `--api`: `findface-security` API URL, `http://127.0.0.1:80/` by default.
- `--user`: login.
- `--password`: password.
- `--dossier-lists`: comma-separated list of the watch lists id's.
- `--failed`: should an error occur during the job file processing, correct the mistake and try again with this option.

2.3.6 Deduplicate Events

In this section:

- *Enable Deduplication*
- *How It Works*

Consider enabling Deduplication to exclude coinciding facial recognition events within one camera group.

Enable Deduplication

To enable event deduplication, do the following:

1. Enable the offline mode of video face detection for each camera in the group. See *Add Camera* for details.
2. Navigate to the *Preferences* tab. Click *Camera Groups*.
3. Open the camera group settings.
4. Check *Deduplicate Events* and specify the deduplication interval in seconds.

How It Works

The deduplication algorithm works as follows:

1. In the offline mode, the server receives one best face snapshot per tracking session on a camera (a tracking session continues until a face disappears from the camera field of view).
2. If within the same camera group, there are several tracking sessions on a camera(s) within the specified deduplication interval, FindFace will handle the received snapshots in the following way:
   - If there is a match with a dossier within the preceding deduplication interval, FindFace drops a newly acquired snapshot. Otherwise, the snapshot is saved to the database.
   - For unmatched faces, when performing deduplication, FindFace considers both similarity between faces in the snapshots and face quality. FindFace drops all snapshots of similar faces within the deduplication interval unless a new face is of higher quality. This guarantees that the system deduplicates events without skipping high-quality faces, which are essential for further video analytics.

2.3.7 Real-time Face Liveness Detection

Note: The *liveness detector* is much slower on CPU than on GPU.

To spot fake faces and prevent photo attacks, use the integrated 2D anti-spoofing system that distinguishes a live face from a face image. Due to the analysis of not one, but a number of frames, the algorithm captures any changes in a facial expression and skin texture. This ensures that it is a live person in front of a camera and eliminates the possibility of fraud using images on paper or mobile device screens.

The liveness detector estimates a face liveness with a certain level of confidence and returns the confidence score along with a binary result `real/fake`, depending on the pre-defined liveness threshold.

In this section:

- *Enable Face Liveness Detector*
- *Configure Liveness Threshold*
- *Face Liveness in Web Interface*

Enable Face Liveness Detector

To enable the face liveness detector, do the following:

1. Open the `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`) configuration file. In the `liveness` section, specify the path to the neural network model (`fnk`) and normalizer (`norm`) which are used in the face liveness detector.

```
sudo vi /etc/findface-video-worker-gpu.ini

#-----------------------------
[liveness]
#-----------------------------
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.alleyn.v2.gpu.fnk

## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.gpu.fnk
```

```
sudo vi /etc/findface-video-worker-cpu.ini

#-----------------------------
[liveness]
#-----------------------------
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.alleyn.v2.cpu.fnk
```

(continues on next page)

(continued from previous page)

```
## path to normalization for liveness
## type:string env:CFG_LIVENESS_NORM longopt:--liveness-norm
norm = /usr/share/findface-data/models/facenorm/crop2x.v2_maxsize400.cpu.fnk
```

2. Restart `findface-video-worker`.

```
sudo systemctl restart findface-video-worker-gpu
sudo systemctl restart findface-video-worker-cpu
```

Configure Liveness Threshold

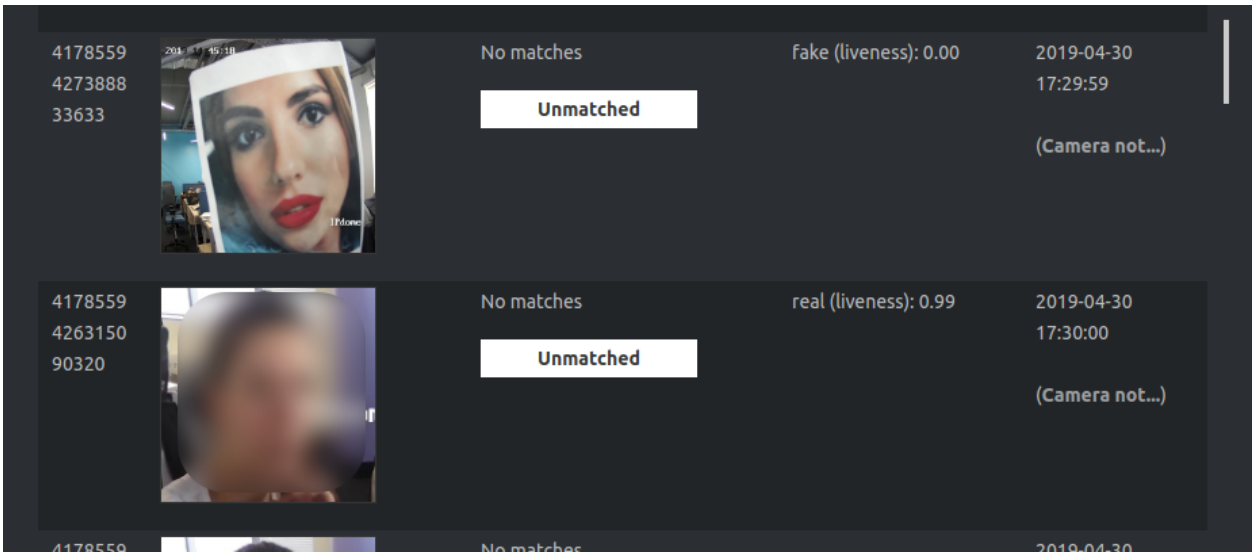
If necessary, you can adjust the liveness `threshold` in the `/etc/findface-security/config.py` configuration file. The liveness detector will estimate a face liveness with a certain level of confidence. Depending on the threshold value, it will return a binary result `real` or `fake`.

Note: The default value is optimal. Before changing the threshold, we recommend you to seek advice from our experts by support@ntechlab.com.

```
sudo vi /etc/findface-security/config.py

'LIVENESS_THRESHOLD': 0.85,
```

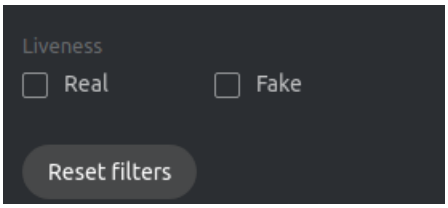
Face Liveness in Web Interface

Once the face liveness detector configured, you will see liveness estimation for each event.



Note: The liveness score is `null` when the liveness detector is unable to estimate the face liveness in the provided image.

Use the *Liveness* filter to display only real or only fake faces in the event list.



See also:

Liveness Detection as Standalone Service

2.3.8 Liveness Detection as Standalone Service

See also:

Real-time Face Liveness Detection

Besides the *integrated* anti-spoofing system that distinguishes a live face from a face image, FindFace provides an API-based face liveness detection service `findface-liveness-api`.

The `findface-liveness-api` service takes a specific number of frames from a video fragment and returns the best quality face, and decimal liveness result averaged across the taken frames. If configured, the service can also return full-frame and normalized face images and save the detection result in the `findface-sf-api` cache, returning `detection_id`.

FindFace uses `findface-liveness-api` for face-based *authentication*. If needed, you can install and use this service standalone.

In this section:

- *Install and Configure `findface-liveness-api`*
- *Liveness API Usage*

Install and Configure `findface-liveness-api`

The `findface-liveness-api` service is automatically installed with FindFace.

To install the service standalone, install the FindFace *APT repository* and execute the following commands:

```
sudo apt update
sudo apt install findface-liveness-api
```

You can configure the `findface-liveness-api` parameters in the `/etc/findface-liveness-api.ini` configuration file:

```
sudo vi /etc/findface-liveness-api.ini

listen: :18301
liveness-threshold: 0.95
fullframe-jpeg-quality: 75
max-decoded-frames: 30
min-selected-frames: 10
```

(continues on next page)

(continued from previous page)

```
mf-selector: reject
extraction-api:
  request-batch-size: 16
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  extraction-api: http://127.0.0.1:18666
sf-api:
  timeouts:
    connect: 5s
    response_header: 30s
     overall: 35s
    idle_connection: 10s
  sf-api: http://127.0.0.1:18411
limits:
  video-size: 10485760
  video-length-sec: 60
  video-fps: 30
  video-width-px: 1920
  video-height-px: 1080
```

Parameter	Description
`fullframe-jpeg-quality`	JPEG quality of full frames in the `photo` field.
`max-decoded-frames`	Finish decoding after reaching the specified number of frames.
`min-selected-frames`	The minimum number of final frames successfully passed through decoding and liveness extraction. Must be equal or less than `max-decoded-frames`.
`mf-selector`	Service behavior upon having multiple faces in the video frame: `reject` - reject this frame, `biggest` - use the biggest face for liveness detection.
`extraction-api -> request-batch-size`	Batch size for liveness extraction.
`limits -> video-size`	Maximum video size, bytes.
`limits -> video-length-sec`	Maximum video length, seconds.
`limits -> video-fps`	Maximum video FPS.
`limits -> video-width-px`	Maximum video width, pixels.
`limits -> video-height-px`	Maximum video height, pixels.

To start the `findface-liveness-api` service and enable its autostart, execute:

```
sudo systemctl enable findface-liveness-api.service && sudo systemctl start findface-
↪liveness-api.service
```

Liveness API Usage

To interact with the `findface-liveness-api` service, use HTTP API requests. In the example below, the POST request is sent with the following optional parameters:

- `return_detection` (`default=False`): save the best face in the `findface-sf-api` cache and return its `detection_id`.
- `return_normalized` (`default=False`): return the face normalized image in the `normalized` field.
- `return_photo` (`default=False`): return the full frame in the `photo` field.

Example

Request

```
curl -i -X POST \
  'http://127.0.0.1:18301/v1/video-liveness?return_detection=true&return_normalized=true&
↪return_photo=true' \
  -H 'Content-Type: video/mp4' \
  --data-binary '@/home/my_video.mp4'
```

Response

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Content-Type: application/json
X-Request-Id: LA:WSP2NcHc
Date: Mon, 07 Sep 2020 15:30:05 GMT
Transfer-Encoding: chunked
{
        "alive": true,
        "average_liveness": 0.9706386,
        "best_face": {
                "liveness": 0.97768883,
                "quality": 0.89638597,
                "bbox": {
                        "left": 0,
                        "top": 578,
                        "right": 307,
                        "bottom": 1154
                },
                "detection_id": "btb53vbp688s1njt3bv0",
                "photo": "/9j/2wCEAAgGBgcGBQgHBwcJ...",
                "normalized": "iVBORw0KGgoAAAANSUhEU...",
                "frame_no": 1,
                "frame_ts": 0.033275817
        }
}
```

2.3.9 Face Features Recognition

Subject to your needs, you can enable automatic recognition of such face features as gender, age, emotions, glasses, beard, and face mask. This functionality can be activated on both GPU- and CPU-accelerated video face detectors.

In this section:

- *Enable Face Features Recognition*
- *Display Features Recognition Results in Events*
- *Face Features in Events*

Enable Face Features Recognition

Important: This step will enable face features recognition via HTTP API.

To enable automatic recognition of face features, open the `/etc/findface-extraction-api.ini` configuration file and enable relevant recognition models: gender, age, emotions, glasses, beard, and face mask. Be sure to choose the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models.

```
sudo vi /etc/findface-extraction-api.ini

models:
  age: faceattr/age.v1.cpu.fnk
  emotions: faceattr/emotions.v1.cpu.fnk
  face: face/jackfruit_480.cpu.fnk
  gender: faceattr/gender.v2.cpu.fnk
  beard: faceattr/beard.v0.cpu.fnk
  glasses3: faceattr/glasses3.v0.cpu.fnk
  medmask3: faceattr/medmask3.v2.cpu.fnk
```

The following models are available:

Note: You can find face features recognition models at `/usr/share/findface-data/models/faceattr/`.

```
ls /usr/share/findface-data/models/faceattr/
age.v1.cpu.fnk  age.v1.gpu.fnk  beard.v0.cpu.fnk  beard.v0.gpu.fnk  emotions.v1.cpu.fnk ␣
↪emotions.v1.gpu.fnk  gender.v2.cpu.fnk  gender.v2.gpu.fnk  glasses3.v0.cpu.fnk ␣
↪glasses3.v0.gpu.fnk medmask3.v2.cpu.fnk medmask3.v2.gpu.fnk liveness.alleyn.v2.gpu.fnk␣
↪quality.v1.cpu.fnk quality.v1.gpu.fnk
```

Face feature	Acceleration	Configuration file parameter
face (biometry)	CPU	`face: face/ifruit_320.cpu.fnk face: face/jackfruit_160.cpu.fnk face: face/jackfruit_320.cpu.fnk face: face/jackfruit_480.cpu.fnk`
	GPU	`face: face/ifruit_320.gpu.fnk face: face/jackfruit_160.gpu.fnk face: face/jackfruit_320.gpu.fnk face: face/jackfruit_480.gpu.fnk`
age	CPU	`age: faceattr/age.v1.cpu.fnk`
	GPU	`age: faceattr/age.v1.gpu.fnk`
gender	CPU	`gender: faceattr/gender.v2.cpu.fnk`
	GPU	`gender: faceattr/gender.v2.gpu.fnk`
emotions	CPU	`emotions: faceattr/emotions.v1.cpu.fnk`
	GPU	`emotions: faceattr/emotions.v1.gpu.fnk`
glasses	CPU	`glasses3: faceattr/glasses3.v0.cpu.fnk`
	GPU	`glasses3: faceattr/glasses3.v0.gpu.fnk`
beard	CPU	`beard: faceattr/beard.v0.cpu.fnk`
	GPU	`beard: faceattr/beard.v0.gpu.fnk`
face mask	CPU	`medmask3: faceattr/medmask3.v2.cpu.fnk`
	GPU	`medmask3: faceattr/medmask3.v2.gpu.fnk`

Tip: To disable a recognition model, simply pass an empty value to a relevant parameter. Do not remove the parameter itself as in this case the system will be searching for the default model.

```
models:
  gender: ""
  age: ""
  emotions: ""
```

Restart `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api
```

Once the models are enabled, be sure to *configure* the web interface to display the recognition results.

Display Features Recognition Results in Events

To display the face features recognition results in the event list, add the following line into the `FFSECURITY` section: `'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask']`, subject to the list of enabled models.

Warning: This line must be placed between `SF_API_ADDRESS` and `LIVENESS_THRESHOLD` as shown in the example.

```
sudo vi /etc/findface-security/config.py

...
FFSECURITY = {
```

(continues on next page)

(continued from previous page)

```
...
'SF_API_ADDRESS': 'http://127.0.0.1:18411',
'EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses', 'medmask'],
'LIVENESS_THRESHOLD': 0.85,
'BEARD_THRESHOLD': 0.7,
 }
```

Restart `findface-security`.

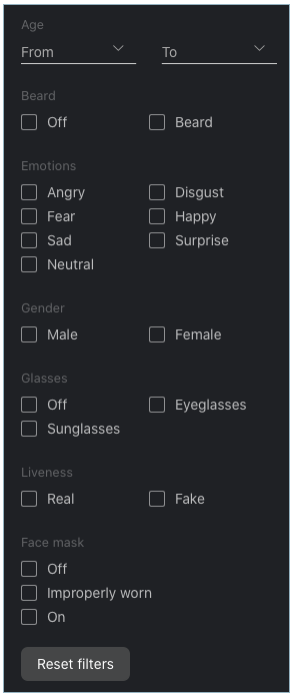
```
sudo systemctl restart findface-security
```

Face Features in Events

Once the face features recognition configured, you will see the recognition result for each found face in the following format:

Face feature	Result format	Example
Age	Feature: `age`: number of years	age: 33
Gender	Result: `male/female` (feature: `gender`): algorithm confidence in result	female (gender): 0.95
Emotions	Result: `angry/disgust/fear/happy/sad/surprise` (feature: `emotions`): algorithm confidence in result	happy (emotions): 0.99
Glasses	Result: `eye/sun/none` (feature: `glasses`): algorithm confidence in result	none (glasses): 0.87
Beard	Result: `beard/none` (feature: `beard`): algorithm confidence in result	none (beard): 0.91
Face mask	Result: `none/correct/incorrect` (feature: `face mask`): algorithm confidence in result	none (face mask): 0.93

Filter events by face features when needed.



2.3.10 Silhouette Detection

To collect real-time *statistics* on human silhouettes, you first have to enable silhouette detection.

In this section:

- *Enable Silhouette Detection*

Enable Silhouette Detection

To enable detection of human silhouettes, do the following:

1. Open the `/etc/findface-extraction-api.ini` configuration file and add a new detector `edie` (`edie_rc2.cpu.fnk`/`edie_rc2.gpu.fnk`) with relevant settings, as shown in the example below.

```
sudo vi /etc/findface-extraction-api.ini

detectors:
  max_batch_size: 1
  instances: 1
  models:
    cheetah:
      model: facedet/cheetah.cpu.fnk
      options:
        min_object_size: 32
        resolutions:
        - 256x256
```

(continues on next page)

(continued from previous page)

```
        - 384x384
        - 512x512
        - 768x768
        - 1024x1024
        - 1536x1536
        - 2048x2048
    edie:
      model: pedet/edie_rc2.cpu.fnk
      options:
        min_object_size: 16
        resolutions:
        - 1280x720
        - 1920x1080
```

Note: The values of the `edie` options in the example are considered optimal. You can adjust them, subject to your cameras' observation scenes. Contact our experts by support@ntechlab.com if you have any questions.

2. Restart `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api
```

2.3.11 Multiple Video Cards Usage

Should you have several video cards installed on a physical server, you can create additional `findface-extraction-api-gpu` or `findface-video-worker-gpu` instances and distribute them across the video cards, one instance per card.

In this section:

- *Distribute `findface-extraction-api-gpu` Instances Across Several Video Cards*
- *Allocate `findface-video-worker-gpu` to Additional Video Card*

Distribute `findface-extraction-api-gpu` Instances Across Several Video Cards

To distribute `findface-extraction-api-gpu` instances across several video cards, do the following:

1. Stop the initial `findface-extraction-api-gpu` service.

```
sudo service findface-extraction-api stop
```

2. Create several copies of the `/etc/findface-extraction-api.ini` configuration file, subject to how many video cards you are going to use for biometric samples extraction. Append the appropriate GPU device numbers to the new configuration files names as shown in the example below (GPU devices #0 and #6).

```
/etc/findface-extraction-api@0.ini
/etc/findface-extraction-api@6.ini
```

3. Open the new configuration files. Specify the GPU device numbers and adjust the listening ports.

```
sudo vi /etc/findface-extraction-api@0.ini

listen: 127.0.0.1:18666
...

gpu_device: 0
...
```

```
sudo vi /etc/findface-extraction-api@6.ini

listen: 127.0.0.1:18667
...

gpu_device: 6
...
```

4. Start the new services.

```
sudo service findface-extraction-api@0 start
sudo service findface-extraction-api@6 start
```

Allocate `findface-video-worker-gpu` to Additional Video Card

To create an additional `findface-video-worker-gpu` instance and allocate it to a different video card, do the following:

1. Display the status of the `findface-video-worker-gpu` primary service by executing:

```
sudo systemctl status findface-video-worker-gpu.service
```

2. Find the full path to the service in the following line:

```
Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service); enabled;
↪ vendor preset: enabled
```

It is `findface-video-worker-gpu.service` in our example (name may vary). Create a copy of the service under a new name.

```
sudo cp /usr/lib/systemd/system/findface-video-worker-gpu.service /usr/lib/systemd/
↪system/findface-video-worker-gpu2.service`
```

3. In the same manner, create a copy of the primary service configuration file under a new name.

```
sudo cp /etc/findface-video-worker-gpu.ini /etc/findface-video-worker-gpu2.ini
```

4. Open the just created configuration file and actualize the GPU device number to use. Modify the streamer port number by the following formula: `18999 (port number for GPU #0) - GPU device number`, i.e. for the GPU #1, `port = 18998`, for the GPU #2, `port = 18997`, and so on.

```
sudo vi /etc/findface-video-worker-gpu2.ini

## cuda device number
```

(continues on next page)

(continued from previous page)

```
device_number = 1

...

#-----------------------------
[streamer]
#-----------------------------
## streamer/shots webserver port, 0=disabled
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port
port = 18999
...
```

5. Open the new service and specify the just created configuration file.

```
sudo vi /usr/lib/systemd/system/findface-video-worker-gpu2.service

ExecStart=/usr/bin/findface-video-worker-gpu --config /etc/findface-video-worker-
↪gpu2.ini
```

6. Reload the `systemd` daemon to apply the changes.

```
sudo systemctl daemon-reload
```

7. Enable the new service autostart.

```
sudo systemctl enable findface-video-worker-gpu2.service

Created symlink from /etc/systemd/system/multi-user.target.wants/findface-video-
↪worker-gpu2.service to /usr/lib/systemd/system/findface-video-worker-gpu2.service
```

8. Launch the new service.

```
sudo systemctl start findface-video-worker-gpu2.service
```

9. Check the both `findface-video-worker-gpu` services status.

```
sudo systemctl status findface-video-worker-* | grep -i 'Active:' -B 3

 findface-video-worker-gpu2.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu2.service;␣
↪enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-07-18 10:32:02 MSK; 1min 11s ago

...

 findface-video-worker-gpu.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service;␣
↪enabled; vendor preset: enabled)
  Active: active (running) since Mon 2019-07-15 15:18:33 MSK; 2 days ago
```

2.3.12 Direct API Requests to Tarantool

You can use HTTP API to extract data directly from the Tarantool Database.

In this section:

- *General Information*
- *Add Face*
- *Remove Face*
- *Face Search*
- *Edit Face Metadata and Feature Vector*
- *List Galleries*
- *Get Gallery Info*
- *Create Gallery*
- *Remove Gallery*

General Information

API requests to Tarantool are to be sent to `http://<tarantool_host_ip:port>`.

Tip: The port for API requests can be found in the `FindFace.start` section of the Tarantool configuration file `/etc/tarantool/instances.available/*.lua`:

```
cat /etc/tarantool/instances.available/*.lua

##8101:
FindFace.start("127.0.0.1", 8101)
```

Note: In the case of the standalone deployment, you can access Tarantool by default only locally (127.0.0.1). If you want to access Tarantool remotely, *alter* the Tarantool configuration file (`/etc/tarantool/instances.available/*.lua`).

API requests to Tarantool may contain the following parameters in path segments:

- `:ver`: API version (v2 at the moment).
- `:name`: gallery name.

  Tip: To list gallery names on a shard, type in the following command in the address bar of your browser:

  ```
  http://<tarantool_host_ip:shard_port>/stat/list/1/99
  ```

  The same command on the console is as such:

```
curl <tarantool_host_ip:shard_port>/stat/list/1/99 \| jq
```

You can also list gallery names by using a direct request to Tarantool:

```
echo 'box.space.galleries:select()' | tarantoolctl connect <tarantool_host_ip:shard_
↪port>
```

Note that if there is a large number of shards in the system, chances are that a randomly taken shard does not contain all the existing galleries. In this case, just list galleries on several shards.

Add Face

```
POST /:ver/faces/add/:name
```

Parameters in body:

JSON-encoded array of faces with the following fields:

- `"id"`: face id in the gallery, uint64_t,
- `"facen"`: raw feature vector, base64,
- `"meta"`: face metadata, dictionary.

Returns:

- HTTP 200 and empty body on success.
- HTTP 404 if a gallery with the given name doesn't exist.
- HTTP with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -s 'http://localhost:8101/v2/faces/add/testgal' --data '
[
  {
    "id": 9223372036854776000,
    "facen": "qgI3vZRv/z...NpO9MdHavW1WuT0=",
    "meta": {
"cam_id": "223900",
"person_name": "Mary Ostin",

    }
  }
]
```

Response

```
HTTP/1.1 200 Ok
Content-length: 1234
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

Remove Face

```
POST /v2/faces/delete/:name
```

Parameters in body:

JSON-encoded array of face ids to be removed

Returns:

- HTTP 200 and empty body on success.
- HTTP 404 if a face with the given id is not found in the gallery.
- HTTP with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D -  -s 'http://localhost:8101/v2/faces/delete/testgal' --data '[1, 4, 922, 3]'
```

Response

```
HTTP/1.1 200 Ok
Content-length: 111
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

Face Search

```
POST /v2/faces/search/:name
```

Parameters in body:

JSON-encoded search request with the following fields:

- `limit`: maximum number of faces in the response.
- `sort`: sorting order. Pass one of the following values: `id`: increasing order by id, `-id`: decreasing order by id, `-score`: decreasing order by face similarity (only if you search for faces with similar feature vectors).
- `filter` (filters):
  - `facen`: (optional) search for faces with similar feature vectors. Pass a dictionary with the following fields: `data`: raw feature vector, base64; `score`: range of similarity between faces [threshold similarity; 1], where 1 is 100% match.
  - `id` and `meta/<meta_key>`: search by face id and metastring content. To set this filter, use the following operators:
    * `range`: range of values, only for numbers.
    * `set`: id or metastring must contain at least one value from a given set, for numbers and strings.
    * `subset`: id or metastring must include all values from a given subset, for numbers and strings.
    * `like`: by analogy with `like` in SQL requests: only 'aa%', '%aa', and '%aa%' are supported. Only for strings and set[string]. In the case of set[string], the filter will return result if at least one value meets the filter condition.
    * `ilike`: by analogy with `like` but case-insensitive, only for strings and set[string].

Returns:

- JSON-encoded array with faces on success. The value in the `X-search-stat` header indicates whether the fast index was used for the search: `with_index` or `without_index`.

  Note: Fast index is not used in API v2.

- HTTP with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -s 'http://localhost:8101/v2/testgal/search' --data '
{
        "limit": 2,
    "sort": {
                "score": -1
        },
        "filter": {
                "facen": {
                        "data": "qgI3vZRv/z0BQTk9rcirOyZrNpO9MdHavW1WuT0=",
                        "score": [0.75, 1]
                },
                "id": {
```

(continues on next page)

(continued from previous page)

```
                        "range": [9223372036854000000, 9223372036854999000]
                },
                "meta": {
                        "person_id": {
                                "range": [444, 999]
                        },
                        "cam_id": {
                                "set": ["12767", "8632", "23989"]
                        }
                }
        }
}'
```

Response

```
HTTP/1.1 200 Ok
Content-length: 1234
X-search-stat: without_index
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{
  "results": [
    {
      "facen": " qgI3vZRv/z0BQTk9rcirOyZrNpO9MdHavW1WuT0=",
      "meta": {
        "timestamp": 0,
        "photo_hash": "",
        "person_id": 777,
        "cam_id": "8632"
      },
      "score": 0.9964,
      "id": 9223372036854776000
    }
  ]
}
```

Edit Face Metadata and Feature Vector

```
POST /v2/faces/update/:name
```

Parameters in body:

JSON-encoded array with faces with the following fields:

- `"id"`: face id, uint64_t.
- `"facen"`: (optional) new feature vector, base64. If omitted or passed as `null`, the relevant field in the database won't be updated.
- `"meta"`: dictionary with metadata to be updated. If some metastring is omitted or passed as `null`, the relevant field in the database won't be updated.

Returns:

- HTTP 200 and dictionary with all face parameters, including not updated, on success.
- HTTP 404 and error description if a face with the given id doesn't exist.
- HTTP with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -s 'http://localhost:8101/v2/faces/update/sandbox' --data '[{"id":1,"facen
↪":null,"meta":{"m:timestamp":1848}}]'
```

Response

```
HTTP/1.1 200 Ok
Content-length: 151
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{"meta":{"m:timestamp":1848,"normalized_id":"1_b9pkrf00mjt6h1vmq1kg.png","m:cam_id":
↪"a9f7a973-f07e-469d-a3bd-41ddd510b26f","feat":"{\"score\":0.123}"}, "id":1, ... }
```

List Galleries

```
POST /v2/galleries/list
```

Returns:

JSON-encoded array with galleries with the following fields: `name`: gallery name, `faces`: number of faces in a gallery.

Example

Request

```
curl -D - -s -X POST http://localhost:8101/v2/galleries/list
```

Response

```
HTTP/1.1 200 Ok
Content-length: 42
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{
  "results": [
    {
      "name": "testgal",
      "faces": 2
    }
  ]
}
```

Get Gallery Info

```
POST /v2/galleries/get/:name
```

Returns:

- HTTP 200 and dictionary with gallery parameters on success.
- HTTP 404 and error description if a gallery with the given name doesn't exist.
- HTTP with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -s -X POST http://localhost:8101/v2/galleries/get/testgal
```

```
HTTP/1.1 200 Ok
Content-length: 11
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

(continues on next page)

(continued from previous page)

```
{"faces":2}
```

Create Gallery

```
POST /v2/galleries/add/:name
```

Returns:

- HTTP 200 and empty body on success.
- with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -X POST -s 'http://localhost:8101/v2/galleries/add/123'
```

Response

```
HTTP/1.1 409 Conflict
Content-length: 57
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive

{"error":{"message":"gallery already exists","code":409}}
```

Remove Gallery

```
POST /v2/galleries/delete/:name
```

Returns:

- HTTP 200 and empty on success.
- with a status other than 200 and error description in the body on failure.

Example

Request

```
curl -D - -X POST -s 'http://localhost:8101/v2/galleries/delete/123'
```

Response

```
HTTP/1.1 204 No content
Content-length: 0
Server: Tarantool http (tarantool v1.7.3-673-g23cc4dc)
Connection: keep-alive
```

2.3.13 Enable Personal Data Protection

FindFace supports laws related to the processing of personal data of individuals (GDPR and similar).

To apply personal data protection to your system, do the following:

1. Open the `/etc/findface-security/config.py` configuration file.

   ```
   sudo vi /etc/findface-security/config.py
   ```

2. Disable saving unmatched events by setting `'IGNORE_UNMATCHED': True`.

   ```
   ...

   FFSECURITY = {
       ...

       # do not save unmatched events (GDPR support)
       'IGNORE_UNMATCHED': False,

       ...

   }
   ```

3. For events with matches, enable blurring all unmatched faces in full frames. To do so, set `'BLUR_UNMATCHED_FACES': True`. Optionally, you can modify the default JPEG quality of those frames.

   ```
   ...

   FFSECURITY = {
       ...

       # blur all unmatched faces on the full frame of the matched event (GDPR support)
       'BLUR_UNMATCHED_FACES': False,

       # full frame jpeg quality when `BLUR_UNMATCHED_FACES` is enabled
       'BLURRED_FULLFRAME_JPEG_QUALITY': 85,
       ...
   ```

(continues on next page)

(continued from previous page)

```
}
```

4. Restart `findface-security`.

```
sudo systemctl restart findface-security.service
```

CHAPTER

THREE

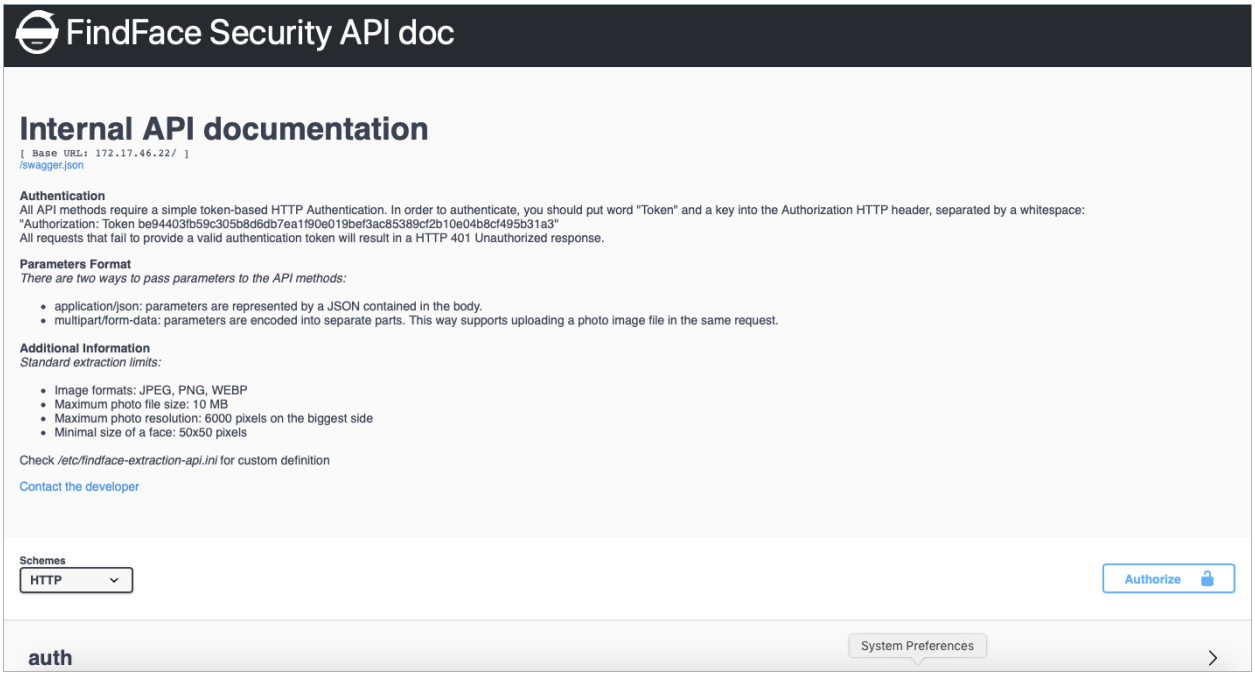
INTEGRATIONS

This chapter is all about integration with FindFace. Integrate your system through HTTP API, webhooks, and plugins, or check out our turnkey partner integrations.

3.1 HTTP API

Detailed interactive documentation on the FindFace HTTP API is available after installation at `http://<ffsecurity_ip:port>/api-docs`. Learn and try it out.

Tip: You can also find it by navigating to *Preferences -> API Documentation* in the web interface.



3.2 Webhooks

You can set up FindFace to automatically send notifications about specific events, episodes, and counter records to a given URL. To do so, create and configure a webhook. In this case, when such an event, episode, or counter record occurs, FindFace will send an HTTP request to the URL configured for the webhook.

You can use webhooks for various purposes, for instance, to notify a user about a specific event, invoke required behavior on a target website, and solve security tasks such as automated access control.

In this section:

- *Configure Webhook*
- *Webhook in Action*
- *Verbose Webhooks*

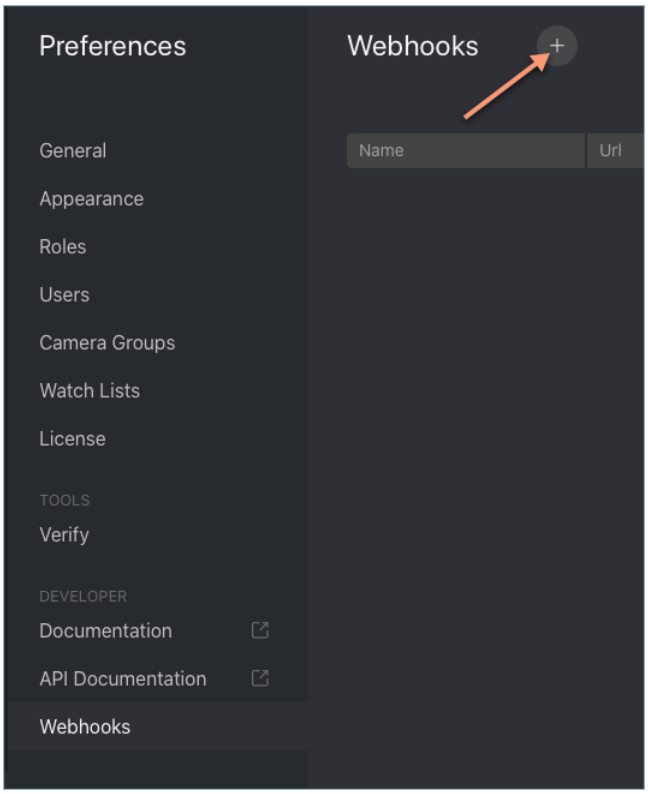
3.2.1 Configure Webhook

Important: You need Administrator privileges to create a webhook.

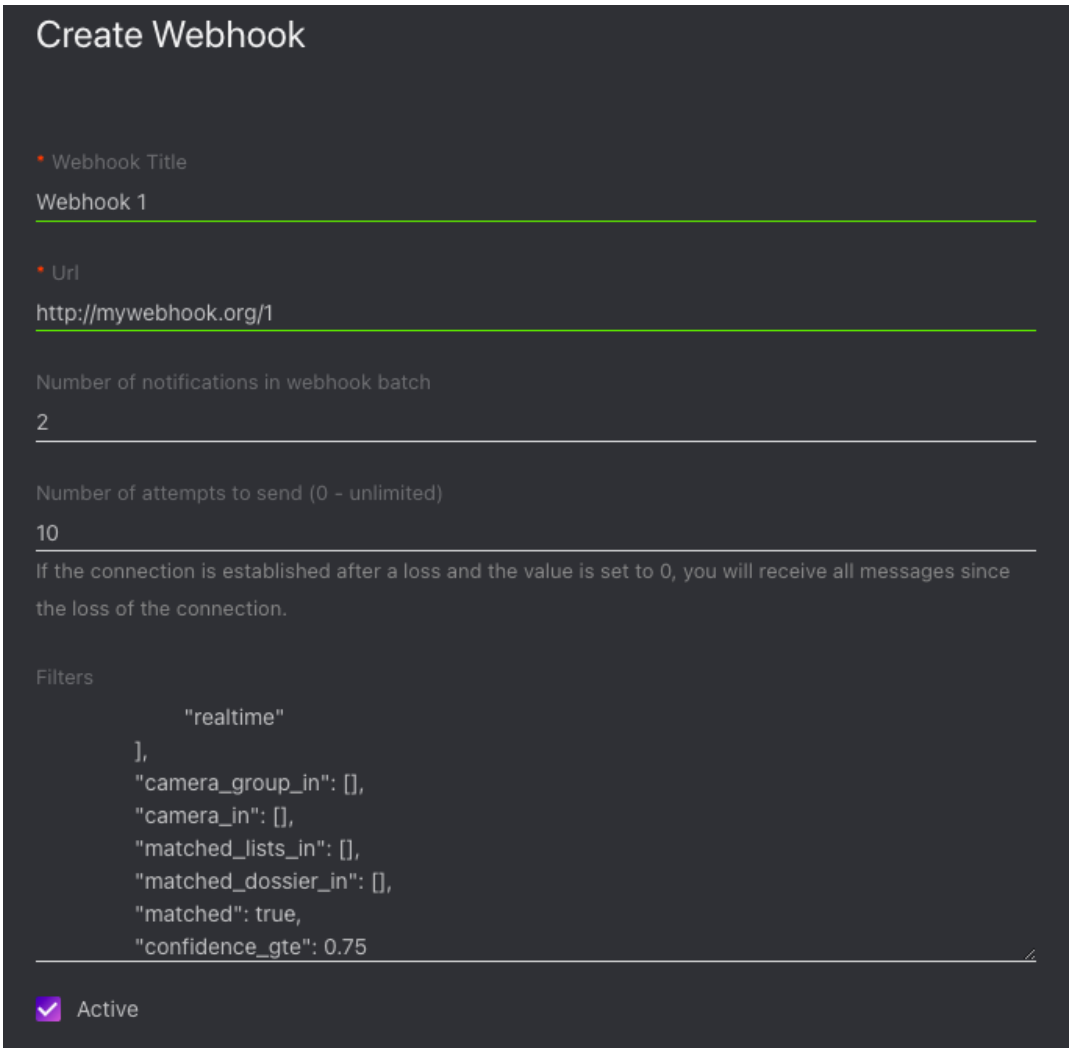
Note: To use the webhooks, be sure that at least one of the following parameters is specified in `/etc/findface-security/config.py`: `SERVICE_EXTERNAL_ADDRESS` or `EXTERNAL_ADDRESS`.

To create and configure a webhook, do the following:

1. Navigate to the *Preferences* tab. Click *Webhooks*.
2. Click +.



3. Specify the webhook title.



4. Specify URL to automatically send notifications to.
5. You can send notifications in batches. Specify the maximum number of notifications in a webhook batch. The actual number may be less.
6. Specify the maximum number of attempts to send a notification. The interval between attempts increases exponentially with a maximum of 100 seconds.

Important: To receive all messages since the connection loss, should it happen, set `0`. Set `1` to omit old messages.

---

7. FindFace will be automatically sending notifications on events, episodes, and counters which match given filters. The following filters are available:

Events:

- `allowed_bs_types`: *face tracking mode*, possible values: `overall`, `realtime`.
- `camera_group_in`: camera group id, number.
- `camera_in`: camera id, number.
- `matched_lists_in`: watch list id, number.
- `matched_dossier_in`: matched dossier id, number.
- `matched`: event `matched` status (`true` or `false`), boolean.
- `confidence_gte`: minimum confidence, number.

Episodes:

- `allowed_types`: episode status, possible values: an episode opening (`episode_open`), adding a new event into an episode (`episode_event`), an episode closing (`episode_close`).
- `camera_group_in`: camera group id, number.
- `camera_in`: camera id, number.
- `matched_lists_in`: watch list id, number.
- `matched`: event `matched` status (`true` or `false`), boolean.
- `events_count_gte`: minimum number of events in an episode, number.
- `events_count_lte`: maximum number of events in an episode, number.

Counters:

- `counter_in`: counter id, number
- `camera_group_in`: camera group id, number.
- `camera_in`: camera id, number
- `faces_gte`: minimum number of faces in a counter record, number.
- `faces_lte`: maximum number of faces in a counter record, number.
- `silhouettes_gte`: minimum number of silhouettes in a counter record, number.
- `silhouettes_lte`: maximum number of silhouettes in a counter record, number.

```
{
        "events": {
                "allowed_bs_types": [
                        "overall",
                        "realtime"
                ],
                "camera_group_in": [],
                "camera_in": [],
                "matched_lists_in": [],
                "matched_dossier_in": [],
                "matched": true,
                "confidence_gte": 0.75
```

(continues on next page)

(continued from previous page)

```
        },
        "episodes": {
                "allowed_types": [
                        "episode_open",
                        "episode_event",
                        "episode_close"
                ],
                "camera_group_in": [],
                "camera_in": [],
                "matched_lists_in": [],
                "matched": true,
                "events_count_gte": 0,
                "events_count_lte": 999
        },
        "counters": {
                "counter_in": [],
                "camera_group_in": [],
                "camera_in": [],
                "faces_gte": 0,
                "faces_lte": 0,
                "silhouettes_gte": 0,
                "silhouettes_lte": 0
        }
}
```

Important: Use only filters which match your search needs. To turn off a filter, remove it from a webhook. Do not leave a filter empty (`[]`) as in this case the result of filtration will be empty as well.

Note: To get all notifications, pass only curly braces without any enclosed filters:

```
{}
```

Tip: Example #1. Get notifications about all events:

```
{ "events": {} }
```

Example #2. Get notifications of the opening of matched episodes:

```
{ "episodes": { "allowed_types": ["episode_open"], "matched": true }}
```

Note: You can specify several values for filters with square braces. In this case, the webhook will be triggered once one of the values from this filter has been matched. In the example below, you will get an event from the camera group `1` or `3` if a matched dossier is `12` or `25`.

```
{
        "events": {
                "camera_group_in": [1, 3],
                "matched_dossier_in": [12, 25],
        },
}
```

8. Check *Active*.
9. Click *Save*.

3.2.2 Webhook in Action

Try out a webhook by capturing event notifications with a simple web server in Python:

```
from pprint import pprint
from aiohttp import web


async def handle(request):
    pprint(await request.json())
    return web.Response(status=200)


app = web.Application()
# for aiohttp v 3.x
# app.add_routes([web.post('/', handle)])

# for aiohttp v 2.x
app.router.add_post('/', handle)

web.run_app(app, port=8888)
```

Important: A webhook catcher that you use must return an `HTTP 200` response after receiving the webhook request from FindFace, like in the example above.

If no filters are configured for a webhook, this web server will be getting notifications about all events, episodes, and counter records that occur in the system. The notifications have the following format:

Event

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': '2020-05-18T15:08:38+00:00',
  'acknowledged_reaction': '',
  'bs_type': 'overall',
  'camera': None,
  'camera_group': 1,
  'confidence': 0.0,
  'created_date': '2020-05-18T15:08:38+00:00',
```

(continues on next page)

(continued from previous page)


```
  'episode': None,
  'event_type': 'event_created',
  'face': 'http://172.17.46.134/uploads/2020/05/18/event/150842_face_AgohWm.jpg',
  'features': {'age': None,
               'beard': None,
               'emotions': None,
               'gender': None,
               'glasses': None,
               'liveness': None,
               'medmask': None},
  'frame': 'http://172.17.46.134/uploads/2020/05/18/event/150842_full_frame_Y3vtGe.jpg',
  'frame_coords_bottom': 320,
  'frame_coords_left': 117,
  'frame_coords_right': 170,
  'frame_coords_top': 242,
  'id': '4267625862518432158',
  'looks_like_confidence': None,
  'matched': False,
  'matched_dossier': None,
  'matched_face': '',
  'matched_lists': [-1],
  'quality': -0.000766,
  'scores': {'liveness_score': None,
             'quality': -0.000766480341553,
             'track': {'first_timestamp': '2020-05-18T15:08:38',
                       'id': '43277e17b1c2-44',
                       'last_timestamp': '2020-05-18T15:08:39'},
             'track_duration_seconds': 2.502499999999997},
  'video_source': 1,
  'webhook_type': 'events'}]
```


Episode opening


```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': '4267637154774219594',
  'camera_groups': [1],
  'cameras': [],
  'closed_date': None,
  'created_date': '2020-05-18T16:18:49.111880Z',
  'event_type': 'episode_open',
  'events_count': 1,
  'features': None,
  'id': 2118,
  'last_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-05-18T16:18:46+00:00',
                 'acknowledged_reaction': '',
```


(continues on next page)

(continued from previous page)


```
                 'camera': None,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-05-18T16:18:46+00:00',
                 'episode': 2118,
                 'face': 'http://172.17.46.134/uploads/2020/05/18/event/161849_face_
↪j2TQwk.jpg',
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.46.134/uploads/2020/05/18/event/161849_full_
↪frame_vTfuH9.jpg',
                 'frame_coords_bottom': 327,
                 'frame_coords_left': 778,
                 'frame_coords_right': 901,
                 'frame_coords_top': 161,
                 'id': '4267637154774219594',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': -0.000311,
                 'scores': {'liveness_score': None,
                            'quality': -0.000311948591843,
                            'track': {'first_timestamp': '2020-05-18T16:18:46',
                                      'id': '1ee9a3612af3-9',
                                      'last_timestamp': '2020-05-18T16:18:47'},
                            'track_duration_seconds': 2.039999999999999},
                 'video_source': 2},
  'matched_event': None,
  'matched_lists': [-1],
  'open': True,
  'webhook_type': 'episodes'}]
```


Episode closing


```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-05-18T15:09:57+00:00',
                 'acknowledged_reaction': '',
                 'camera': None,
```


(continues on next page)

(continued from previous page)


```
                'camera_group': 1,
                'confidence': 0.0,
                'created_date': '2020-05-18T15:09:57+00:00',
                'episode': 518,
                'face': 'http://172.17.46.134/uploads/2020/05/18/event/151012_face_
↪5LlHQL.jpg',
                'features': {'age': None,
                             'beard': None,
                             'emotions': None,
                             'gender': None,
                             'glasses': None,
                             'liveness': None,
                             'medmask': None},
                'frame': 'http://172.17.46.134/uploads/2020/05/18/event/151012_full_
↪frame_CdNn2N.jpg',
                'frame_coords_bottom': 299,
                'frame_coords_left': 917,
                'frame_coords_right': 1005,
                'frame_coords_top': 179,
                'id': '4267626103667833809',
                'looks_like_confidence': None,
                'matched': False,
                'matched_dossier': None,
                'matched_face': '',
                'matched_lists': [-1],
                'quality': -0.653877,
                'scores': {'liveness_score': None,
                           'quality': -0.653877139091491,
                           'track': {'first_timestamp': '2020-05-18T15:09:57',
                                     'id': '43277e17b1c2-231',
                                     'last_timestamp': '2020-05-18T15:09:57'},
                           'track_duration_seconds': 0.250255555555554},
                'video_source': 1},
  'camera_groups': [1],
  'cameras': [],
  'closed_date': '2020-05-18T15:10:42.870851Z',
  'created_date': '2020-05-18T15:10:12.201230Z',
  'event_type': 'episode_close',
  'events_count': 1,
  'features': None,
  'id': 518,
  'last_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-05-18T15:09:57+00:00',
                 'acknowledged_reaction': '',
                 'camera': None,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-05-18T15:09:57+00:00',
                 'episode': 518,
                 'face': 'http://172.17.46.134/uploads/2020/05/18/event/151012_face_
↪5LlHQL.jpg',
```


(continues on next page)

(continued from previous page)

```
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.46.134/uploads/2020/05/18/event/151012_full_
↪frame_CdNn2N.jpg',
                 'frame_coords_bottom': 299,
                 'frame_coords_left': 917,
                 'frame_coords_right': 1005,
                 'frame_coords_top': 179,
                 'id': '4267626103667833809',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': -0.653877,
                 'scores': {'liveness_score': None,
                            'quality': -0.653877139091491,
                            'track': {'first_timestamp': '2020-05-18T15:09:57',
                                      'id': '43277e17b1c2-231',
                                      'last_timestamp': '2020-05-18T15:09:57'},
                            'track_duration_seconds': 0.250255555555554},
                 'video_source': 1},
  'matched_event': None,
  'matched_lists': [-1],
  'open': False,
  'webhook_type': 'episodes'}]
```

Counter record

```
[{'camera': 3,
  'camera_group': 1,
  'counter': 2,
  'counter_name': 'smosh',
  'created_date': '2020-05-18T16:15:06.679592Z',
  'event_type': 'counter_record',
  'faces_bbox': [[[700, 210], [894, 210], [894, 464], [700, 464]],
                 [[160, 190], [304, 190], [304, 394], [160, 394]]],
  'faces_count': 2,
  'fullframe': 'http://172.17.46.134/uploads/2020/05/18/counters/161506_fullframe_7Z8n7X.
↪jpg',
  'id': 16,
  'silhouettes_bbox': [[[15, 135], [584, 135], [584, 709], [15, 709]],
                       [[585, 80], [1194, 80], [1194, 684], [585, 684]],
                       [[0, 380], [69, 380], [69, 714], [0, 714]]],
  'silhouettes_count': 3,
```

(continues on next page)

(continued from previous page)

```
  'thumbnail': 'http://172.17.46.134/uploads/2020/05/18/counters/161506_thumb_XLMFwE.jpg
↪',
  'webhook_type': 'counters'}]
```

To view a webhook pulling status, execute:

```
sudo journalctl -u findface-security.service | grep 'WebhooksManager'
```

Success:

```
May 18 21:21:38 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] Updating "events"␣
↪workers for webhooks: {2}
May 18 21:21:52 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:9KHqkQg7-
↪VW:aa3af58f] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.002617
May 18 21:21:52 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685965791894192']
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:v59UsC1V-
↪VW:75c4a9ec] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.002386
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685968207813297']
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:vKNlXiIn-
↪VW:c0219d31] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.004499
May 18 21:21:53 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685968837561053']
May 18 21:21:55 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:zZO8v4LJ-
↪VW:feff75dd] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.001905
May 18 21:21:55 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685973269790230']
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:EbpDel24-
↪VW:083688e2] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.002017
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685977917324748']
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:L5XoQTdq-
↪VW:6f1e397f] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.009237
May 18 21:21:57 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685979796372941']
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:ZZ33mwuv-
↪VW:a4cad3a2] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.008542
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685980899116054']
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:BfAQRgp0-
↪VW:4c19b207] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.003183
```

(continues on next page)

(continued from previous page)

```
May 18 21:21:58 qa-2 ffsecurity[17851]: INFO     [WebhooksManager:2] <queue:  0> Webhook␣
↪worker(id-2, type-"events") sent batch(len-1, type-"events"): ['4267685982215838395']
```

Failure:

```
May 18 21:29:09 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:jrGdiC7e-
↪VW:2def51cf] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.003909
May 18 21:29:09 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue:  1> Webhook␣
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt␣
↪2 out of 10. Next attempt in 0.729 seconds.
May 18 21:29:10 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:jgqLszI7-
↪VW:6a7fea19] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.002402
May 18 21:29:10 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue:  2> Webhook␣
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt␣
↪3 out of 10. Next attempt in 1.968 seconds.
May 18 21:29:10 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:LLGB1RRR-
↪VW:053d7c7d] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.003794
May 18 21:29:11 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:4Vl23NQD-
↪VW:a4640479] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.037162
May 18 21:29:11 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:QKY577ed-
↪VW:41cd531a] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.005274
May 18 21:29:12 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:SVSrlj1n-
↪VW:973ae0dd] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.004273
May 18 21:29:12 qa-2 ffsecurity[17851]: WARNING  [WebhooksManager:2] <queue:  6> Webhook␣
↪worker(id-2, type-"events") Error sending webhook: 405, message='Not Allowed'. Attempt␣
↪4 out of 10. Next attempt in 5.314 seconds.
May 18 21:29:12 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:phoO3HFd-
↪VW:9c6812d1] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.019604
May 18 21:29:13 qa-2 ffsecurity[17851]: INFO     [WebhooksManager] [SC:WDMmZ5MO-
↪VW:842b3397] Webhook updater processing message(type-"events:event_created"). Consumer␣
↪reception delta: 0.231164
```

3.2.3 Verbose Webhooks

By default, webhook notifications contain only ids of such entities as dossiers, watch lists, cameras, and camera groups. It is possible to get whole entities in notifications by switching webhooks to the verbose mode.

To do so, open the `/etc/findface-security/config.py` configuration file and set `'VERBOSE_WEBHOOKS': True`:

```
sudo vi /etc/findface-security/config.py


...
FFSECURITY = {
```

(continues on next page)

(continued from previous page)

```
    ...
    # send serialized dossiers, dossier-lists, camera and camera groups in webhooks
    'VERBOSE_WEBHOOKS': True,
    ...
}
...
```

In the verbose mode, the format of webhook notifications is the following:

Event (Verbose)


```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': '2020-07-30T14:41:52+00:00',
  'acknowledged_reaction': '',
  'bs_type': 'overall',
  'camera': {'active': True,
             'azimuth': None,
             'comment': '',
             'created_date': '2020-07-12T05:57:50.459974Z',
             'group': 1,
             'health_status': {'code': 'yellow',
                               'code_desc': 'Some faces from this camera '
                                            "couldn't be processed",
                               'enabled': True,
                               'msg': '',
                               'statistic': {'decoding_soft_errors': 0,
                                             'faces_failed': 17,
                                             'faces_not_posted': 0,
                                             'faces_posted': 185,
                                             'frames_dropped': 1066,
                                             'frame_height': 1080,
                                             'frames_imotion_skipped': 0,
                                             'frames_processed': 36421,
                                             'frame_width': 1920
                                             'job_starts': 1,
                                             'processed_duration': 1499.44,
                                             'processing_fps': 38.595585},
                               'status': 'INPROGRESS'},
             'id': 1,
             'latitude': None,
             'longitude': None,
             'modified_date': '2020-07-20T06:38:58.607160Z',
             'name': '1',
             'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
             'stream_settings': {'api_timeout': 15000,
                                 'draw_track': False,
                                 'fd_frame_height': -1,
                                 'ffmpeg_format': '',
                                 'ffmpeg_params': [],
                                 'jpeg_quality': 95,
```


(continues on next page)

(continued from previous page)


```
                                'max_face_size': 0,
                                'md_scale': 0.3,
                                'md_threshold': 0.002,
                                'min_d_score': -1000,
                                'min_face_size': 0,
                                'min_score': -2,
                                'npersons': 4,
                                'overall': True,
                                'realtime': False,
                                'realtime_dly': 500,
                                'realtime_post_perm': False,
                                'roi': '',
                                'rot': '',
                                'tracker_threads': 4},
            'stream_settings_gpu': {'ffmpeg_format': '',
                                    'ffmpeg_params': [],
                                    'filter_max_face_size': 8192,
                                    'filter_min_face_size': 1,
                                    'filter_min_quality': 0.45,
                                    'imotion_threshold': 0,
                                    'jpeg_quality': 95,
                                    'overall_only': True,
                                    'play_speed': -1,
                                    'realtime_post_every_interval': False,
                                    'realtime_post_first_immediately': False,
                                    'realtime_post_interval': 1,
                                    'roi': '',
                                    'rot': '',
                                    'router_timeout_ms': 15000,
                                    'router_verify_ssl': True,
                                    'start_stream_timestamp': 0,
                                    'use_stream_timestamp': False},
            'threshold': None,
            'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪chunklist_b1200000.m3u8?codec=mpeg4'},
 'camera_group': {'active': True,
                  'comment': '',
                  'created_date': '2020-07-12T05:48:09.537724Z',
                  'deduplicate': True,
                  'deduplicateDelay': 10,
                  'id': 1,
                  'labels': {},
                  'modified_date': '2020-07-17T01:41:22.944825Z',
                  'name': 'Default Camera Group',
                  'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                  'threshold': None},
 'confidence': 0.0,
 'created_date': '2020-07-30T14:41:52+00:00',
 'episode': None,
 'event_type': 'event_created',
 'face': 'http://172.17.47.245/uploads/2020/07/30/event/144203_face_5ks7RN.jpg',
 'features': {'age': None,
```


(continues on next page)

(continued from previous page)


```
                'beard': None,
                'emotions': None,
                'gender': None,
                'glasses': None,
                'liveness': None,
                'medmask': None},
  'frame': 'http://172.17.47.245/uploads/2020/07/30/event/144203_full_frame_OC4sG3.jpg',
  'frame_coords_bottom': 427,
  'frame_coords_left': 367,
  'frame_coords_right': 600,
  'frame_coords_top': 119,
  'id': '4284552331019521692',
  'looks_like_confidence': None,
  'matched': False,
  'matched_dossier': None,
  'matched_face': '',
  'matched_lists': [{'acknowledge': False,
                     'active': True,
                     'camera_groups': [],
                     'color': 'ffffff',
                     'comment': 'Default list for unmatched evenets',
                     'created_date': '2020-07-12T05:48:09.324264Z',
                     'id': -1,
                     'modified_date': '2020-07-12T05:48:09.324369Z',
                     'name': 'Unmatched',
                     'notify': False,
                     'permissions': {},
                     'remote_url': None,
                     'threshold': None}],
  'quality': -0.000112,
  'scores': {'liveness_score': None,
             'quality': -0.00011235895362900001,
             'track': {'first_timestamp': '2020-07-30T14:41:51',
                       'id': '313e117d86b3-203',
                       'last_timestamp': '2020-07-30T14:42:01'},
             'track_duration_seconds': 4.799999999999272},
  'temperature': None,
  'webhook_type': 'events'}]
```


Episode Opening (Verbose)


```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': '4284565234541639834',
  'camera_groups': [{'active': True,
                     'comment': '',
                     'created_date': '2020-07-12T05:48:09.537724Z',
                     'deduplicate': True,
```


(continues on next page)

(continued from previous page)


```
                    'deduplicateDelay': 10,
                    'id': 1,
                    'labels': {},
                    'modified_date': '2020-07-17T01:41:22.944825Z',
                    'name': 'Default Camera Group',
                    'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                    'threshold': None}],
 'cameras': [{'active': True,
              'azimuth': None,
              'comment': '',
              'created_date': '2020-07-12T05:57:50.459974Z',
              'group': 1,
              'health_status': {'code': 'yellow',
                                'code_desc': 'Some faces from this camera '
                                             "couldn't be processed",
                                'enabled': True,
                                'msg': '',
                                'statistic': {'decoding_soft_errors': 0,
                                              'faces_failed': 20,
                                              'faces_not_posted': 0,
                                              'faces_posted': 1027,
                                              'frames_dropped': 4082,
                                              'frame_height': 1080,
                                              'frames_imotion_skipped': 0,
                                              'frames_processed': 153841,
                                              'frame_width': 1920,
                                              'job_starts': 1,
                                              'processed_duration': 6320.04,
                                              'processing_fps': 49.990253},
                                'status': 'INPROGRESS'},
              'id': 1,
              'latitude': None,
              'longitude': None,
              'modified_date': '2020-07-20T06:38:58.607160Z',
              'name': '1',
              'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
              'stream_settings': {'api_timeout': 15000,
                                  'draw_track': False,
                                  'fd_frame_height': -1,
                                  'ffmpeg_format': '',
                                  'ffmpeg_params': [],
                                  'jpeg_quality': 95,
                                  'max_face_size': 0,
                                  'md_scale': 0.3,
                                  'md_threshold': 0.002,
                                  'min_d_score': -1000,
                                  'min_face_size': 0,
                                  'min_score': -2,
                                  'npersons': 4,
                                  'overall': True,
                                  'realtime': False,
                                  'realtime_dly': 500,
```


(continues on next page)

(continued from previous page)


```
                                    'realtime_post_perm': False,
                                    'roi': '',
                                    'rot': '',
                                    'tracker_threads': 4},
                'stream_settings_gpu': {'ffmpeg_format': '',
                                        'ffmpeg_params': [],
                                        'filter_max_face_size': 8192,
                                        'filter_min_face_size': 1,
                                        'filter_min_quality': 0.45,
                                        'imotion_threshold': 0,
                                        'jpeg_quality': 95,
                                        'overall_only': True,
                                        'play_speed': -1,
                                        'realtime_post_every_interval': False,
                                        'realtime_post_first_immediately': False,
                                        'realtime_post_interval': 1,
                                        'roi': '',
                                        'rot': '',
                                        'router_timeout_ms': 15000,
                                        'router_verify_ssl': True,
                                        'start_stream_timestamp': 0,
                                        'use_stream_timestamp': False},
                'threshold': None,
                'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪chunklist_b1200000.m3u8?codec=mpeg4'}],
  'closed_date': None,
  'created_date': '2020-07-30T16:01:52Z',
  'event_type': 'episode_open',
  'events_count': 1,
  'features': None,
  'id': 104229,
  'last_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-07-30T16:01:52+00:00',
                 'acknowledged_reaction': '',
                 'camera': 1,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-07-30T16:01:52+00:00',
                 'episode': 104229,
                 'face': 'http://172.17.47.245/uploads/2020/07/30/event/160210_face_
↪LnGXeX.jpg',
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160210_full_
↪frame_2N1x5Y.jpg',
                 'frame_coords_bottom': 254,
```


(continues on next page)

(continued from previous page)

```
                 'frame_coords_left': 223,
                 'frame_coords_right': 319,
                 'frame_coords_top': 137,
                 'id': '4284565234541639834',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': 0.000206,
                 'scores': {'liveness_score': None,
                            'quality': 0.00020667129138,
                            'track': {'first_timestamp': '2020-07-30T16:01:48',
                                      'id': '313e117d86b3-1047',
                                      'last_timestamp': '2020-07-30T16:02:09'},
                            'track_duration_seconds': 19.56000000000131},
                 'temperature': None,
                 'video_source': None},
                 ,
  'matched_event': None,
  'matched_lists': [{'acknowledge': False,
                     'active': True,
                     'camera_groups': [],
                     'color': 'ffffff',
                     'comment': 'Default list for unmatched evenets',
                     'created_date': '2020-07-12T05:48:09.324264Z',
                     'id': -1,
                     'modified_date': '2020-07-12T05:48:09.324369Z',
                     'name': 'Unmatched',
                     'notify': False,
                     'permissions': {},
                     'remote_url': None,
                     'threshold': None}],
  'open': True,
  'temperature': None,
  'webhook_type': 'episodes'}]
```

Episode Closing (Verbose)

```
[{'acknowledged': True,
  'acknowledged_by': None,
  'acknowledged_date': None,
  'acknowledged_reaction': '',
  'best_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-07-30T16:05:09+00:00',
                 'acknowledged_reaction': '',
                 'camera': 1,
                 'camera_group': 1,
                 'confidence': 0.0,
```

(continues on next page)

(continued from previous page)


```
                'created_date': '2020-07-30T16:05:06+00:00',
                'episode': 104236,
                'face': 'http://172.17.47.245/uploads/2020/07/30/event/160509_face_
↪PXGmaZ.jpg',
                'features': {'age': None,
                             'beard': None,
                             'emotions': None,
                             'gender': None,
                             'glasses': None,
                             'liveness': None,
                             'medmask': None},
                'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160509_full_
↪frame_YeIRKk.jpg',
                'frame_coords_bottom': 322,
                'frame_coords_left': 465,
                'frame_coords_right': 574,
                'frame_coords_top': 190,
                'id': '4284565716150084776',
                'looks_like_confidence': None,
                'matched': False,
                'matched_dossier': None,
                'matched_face': '',
                'matched_lists': [-1],
                'quality': -0.000468,
                'scores': {'liveness_score': None,
                           'quality': -0.000468814512714,
                           'track': {'first_timestamp': '2020-07-30T16:04:56',
                                     'id': '313e117d86b3-1071',
                                     'last_timestamp': '2020-07-30T16:05:08'},
                           'track_duration_seconds': 8.159999999999854},
                'temperature': None},
  'camera_groups': [{'active': True,
                     'comment': '',
                     'created_date': '2020-07-12T05:48:09.537724Z',
                     'deduplicate': True,
                     'deduplicateDelay': 10,
                     'id': 1,
                     'labels': {},
                     'modified_date': '2020-07-17T01:41:22.944825Z',
                     'name': 'Default Camera Group',
                     'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                     'threshold': None}],
  'cameras': [{'active': True,
               'azimuth': None,
               'comment': '',
               'created_date': '2020-07-12T05:57:50.459974Z',
               'group': 1,
               'health_status': {'code': 'yellow',
                                 'code_desc': 'Some faces from this camera'
                                              'cannot be processed',
                                 'enabled': True,
                                 'msg': '',
```


(continues on next page)

(continued from previous page)


```
                                'statistic': {'decoding_soft_errors': 0,
                                              'faces_failed': 20,
                                              'faces_not_posted': 0,
                                              'faces_posted': 1051,
                                              'frames_dropped': 4253,
                                              'frame_height': 1080,
                                              'frames_imotion_skipped': 0,
                                              'frames_processed': 158401,
                                              'frame_width': 1920,
                                              'job_starts': 1,
                                              'processed_duration': 6509.28,
                                              'processing_fps': 48.428185},
                                'status': 'INPROGRESS'},
                'id': 1,
                'latitude': None,
                'longitude': None,
                'modified_date': '2020-07-20T06:38:58.607160Z',
                'name': '1',
                'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
                'stream_settings': {'api_timeout': 15000,
                                    'draw_track': False,
                                    'fd_frame_height': -1,
                                    'ffmpeg_format': '',
                                    'ffmpeg_params': [],
                                    'jpeg_quality': 95,
                                    'max_face_size': 0,
                                    'md_scale': 0.3,
                                    'md_threshold': 0.002,
                                    'min_d_score': -1000,
                                    'min_face_size': 0,
                                    'min_score': -2,
                                    'npersons': 4,
                                    'overall': True,
                                    'realtime': False,
                                    'realtime_dly': 500,
                                    'realtime_post_perm': False,
                                    'roi': '',
                                    'rot': '',
                                    'tracker_threads': 4},
                'stream_settings_gpu': {'ffmpeg_format': '',
                                        'ffmpeg_params': [],
                                        'filter_max_face_size': 8192,
                                        'filter_min_face_size': 1,
                                        'filter_min_quality': 0.45,
                                        'imotion_threshold': 0,
                                        'jpeg_quality': 95,
                                        'overall_only': True,
                                        'play_speed': -1,
                                        'realtime_post_every_interval': False,
                                        'realtime_post_first_immediately': False,
                                        'realtime_post_interval': 1,
                                        'roi': '',
```


(continues on next page)

(continued from previous page)


```
                                        'rot': '',
                                        'router_timeout_ms': 15000,
                                        'router_verify_ssl': True,
                                        'start_stream_timestamp': 0,
                                        'use_stream_timestamp': False},
                'threshold': None,
                'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪chunklist_b1200000.m3u8?codec=mpeg4'}],
  'closed_date': '2020-07-30T16:05:24.077331Z',
  'created_date': '2020-07-30T16:04:02Z',
  'event_type': 'episode_close',
  'events_count': 4,
  'features': None,
  'id': 104236,
  'last_event': {'acknowledged': True,
                 'acknowledged_by': None,
                 'acknowledged_date': '2020-07-30T16:05:09+00:00',
                 'acknowledged_reaction': '',
                 'camera': 1,
                 'camera_group': 1,
                 'confidence': 0.0,
                 'created_date': '2020-07-30T16:05:06+00:00',
                 'episode': 104236,
                 'face': 'http://172.17.47.245/uploads/2020/07/30/event/160509_face_
↪PXGmaZ.jpg',
                 'features': {'age': None,
                              'beard': None,
                              'emotions': None,
                              'gender': None,
                              'glasses': None,
                              'liveness': None,
                              'medmask': None},
                 'frame': 'http://172.17.47.245/uploads/2020/07/30/event/160509_full_
↪frame_YeIRKk.jpg',
                 'frame_coords_bottom': 322,
                 'frame_coords_left': 465,
                 'frame_coords_right': 574,
                 'frame_coords_top': 190,
                 'id': '4284565716150084776',
                 'looks_like_confidence': None,
                 'matched': False,
                 'matched_dossier': None,
                 'matched_face': '',
                 'matched_lists': [-1],
                 'quality': -0.000468,
                 'scores': {'liveness_score': None,
                            'quality': -0.000468814512714,
                            'track': {'first_timestamp': '2020-07-30T16:04:56',
                                      'id': '313e117d86b3-1071',
                                      'last_timestamp': '2020-07-30T16:05:08'},
                            'track_duration_seconds': 8.159999999999854},
                 'temperature': None},
```


(continues on next page)

(continued from previous page)

```
  'matched_event': None,
  'matched_lists': [{'acknowledge': False,
                     'active': True,
                     'camera_groups': [],
                     'color': 'ffffff',
                     'comment': 'Default list for unmatched evenets',
                     'created_date': '2020-07-12T05:48:09.324264Z',
                     'id': -1,
                     'modified_date': '2020-07-12T05:48:09.324369Z',
                     'name': 'Unmatched',
                     'notify': False,
                     'permissions': {},
                     'threshold': None}],
  'open': False,
  'temperature': None,
  'webhook_type': 'episodes'}]
```

Counter Record (Verbose)

```
[{'camera': {'active': True,
             'azimuth': None,
             'comment': '',
             'created_date': '2020-07-12T05:57:50.459974Z',
             'group': 1,
             'health_status': {'code': 'green',
                               'code_desc': '  ',
                               'enabled': True,
                               'msg': '',
                               'statistic': {'decoding_soft_errors': 0,
                                             'faces_failed': 0,
                                             'faces_not_posted': 0,
                                             'faces_posted': 55,
                                             'frames_dropped': 309,
                                             'frame_height': 1080,
                                             'frames_imotion_skipped': 0,
                                             'frames_processed': 3181,
                                             'frame_width': 1920,
                                             'job_starts': 3,
                                             'processed_duration': 139.68,
                                             'processing_fps': 184.37485},
                               'status': 'INPROGRESS'},
             'id': 1,
             'latitude': None,
             'longitude': None,
             'modified_date': '2020-07-30T19:56:40.773455Z',
             'name': '1',
             'screenshot': 'http://172.17.47.245/cameras/1/screenshot/',
             'stream_settings': {'api_timeout': 15000,
                                 'draw_track': False,
                                 'fd_frame_height': -1,
```

(continues on next page)

(continued from previous page)

```
                                'ffmpeg_format': '',
                                'ffmpeg_params': [],
                                'jpeg_quality': 95,
                                'max_face_size': 0,
                                'md_scale': 0.3,
                                'md_threshold': 0.002,
                                'min_d_score': -1000,
                                'min_face_size': 0,
                                'min_score': -2,
                                'npersons': 4,
                                'overall': True,
                                'realtime': False,
                                'realtime_dly': 500,
                                'realtime_post_perm': False,
                                'roi': '',
                                'rot': '',
                                'tracker_threads': 4},
            'stream_settings_gpu': {'ffmpeg_format': '',
                                    'ffmpeg_params': [],
                                    'filter_max_face_size': 8192,
                                    'filter_min_face_size': 1,
                                    'filter_min_quality': 0.45,
                                    'imotion_threshold': 0,
                                    'jpeg_quality': 95,
                                    'overall_only': False,
                                    'play_speed': -1,
                                    'realtime_post_every_interval': False,
                                    'realtime_post_first_immediately': False,
                                    'realtime_post_interval': 1,
                                    'roi': '',
                                    'rot': '',
                                    'router_timeout_ms': 15000,
                                    'router_verify_ssl': True,
                                    'start_stream_timestamp': 0,
                                    'use_stream_timestamp': False},
            'threshold': None,
            'url': 'http://a3569458063-s26881.cdn.ngenix.net/live/smil:r24.smil/
↪chunklist_b1200000.m3u8?codec=mpeg4'},
 'camera_group': {'active': True,
                  'comment': '',
                  'created_date': '2020-07-12T05:48:09.537724Z',
                  'deduplicate': True,
                  'deduplicateDelay': 10,
                  'id': 1,
                  'labels': {},
                  'modified_date': '2020-07-17T01:41:22.944825Z',
                  'name': 'Default Camera Group',
                  'permissions': {'1': 'view', '2': 'view', '3': 'view'},
                  'threshold': None},
 'counter': {'active': True,
             'camera': 1,
             'count_interval': 5,
```

(continues on next page)

(continued from previous page)

```
                'created_date': '2020-07-30T17:56:01.469273Z',
                'detect_faces': True,
                'detect_silhouettes': False,
                'id': 1,
                'modified_date': '2020-07-30T19:58:31.544631Z',
                'name': '123',
                'roi': [[0, 0], [1024, 0], [1024, 576], [0, 576]]},
  'counter_name': '123',
  'created_date': '2020-07-30T20:17:01.443532Z',
  'event_type': 'counter_record',
  'faces_bbox': [[[327, 289], [474, 289], [474, 485], [327, 485]]],
  'faces_count': 1,
  'fullframe': 'http://172.17.47.245/uploads/2020/07/30/counters/201701_fullframe_fDH9f4.
↪jpg',
  'id': 1253,
  'silhouettes_bbox': None,
  'silhouettes_count': 0,
  'thumbnail': 'http://172.17.47.245/uploads/2020/07/30/counters/201701_thumb_I5fzIP.jpg
↪',
  'webhook_type': 'counters'}]
```

3.3 Partner Integrations

3.3.1 Genetec Security Center

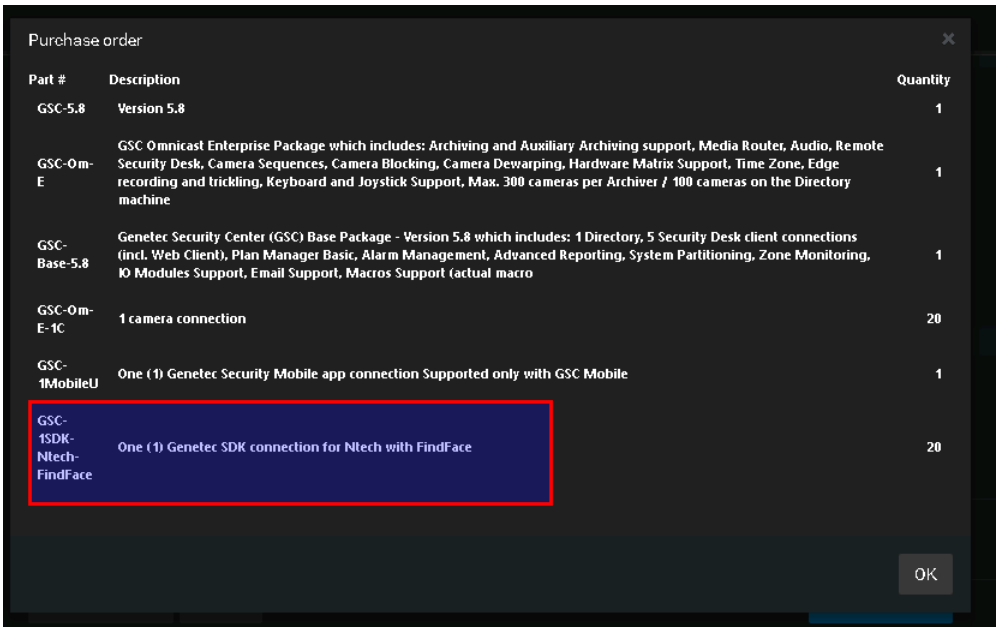
FindFace integration with Genetec Security Center allows you to expand the capabilities of your Genetec-based security system with face recognition functionality.

Configure Integration

Integration with Genetec Security Center is implemented via the `findface-genetec` plugin. By default, the plugin is disabled.

Before getting started with the integration on the FindFace side, deploy the Genetec Web SDK and Media Gateway packages, and create an `Alarm` entity that will be triggered in Genetec Security Center when a face recognition event occurs in FindFace.

Important: For the Genetec-FindFace integration to work, you also need to purchase a proper license from Genetec (license part number `GSC-1SDK-Ntech-FindFace`) and activate it in Genetec Security Center.



Purchase order

Part #	Description	Quantity
GSC-5.8	Version 5.8	1
GSC-Om-E	GSC Omnicast Enterprise Package which includes: Archiving and Auxiliary Archiving support, Media Router, Audio, Remote Security Desk, Camera Sequences, Camera Blocking, Camera Dewarping, Hardware Matrix Support, Time Zone, Edge recording and trickling, Keyboard and Joystick Support, Max. 300 cameras per Archiver / 100 cameras on the Directory machine	1
GSC-Base-5.8	Genetec Security Center (GSC) Base Package - Version 5.8 which includes: 1 Directory, 5 Security Desk client connections (incl. Web Client), Plan Manager Basic, Alarm Management, Advanced Reporting, System Partitioning, Zone Monitoring, IO Modules Support, Email Support, Macros Support (actual macro	1
GSC-Om-E-1C	1 camera connection	20
GSC-1MobileU	One (1) Genetec Security Mobile app connection Supported only with GSC Mobile	1
GSC-1SDK-Ntech-FindFace	One (1) Genetec SDK connection for Ntech with FindFace	20

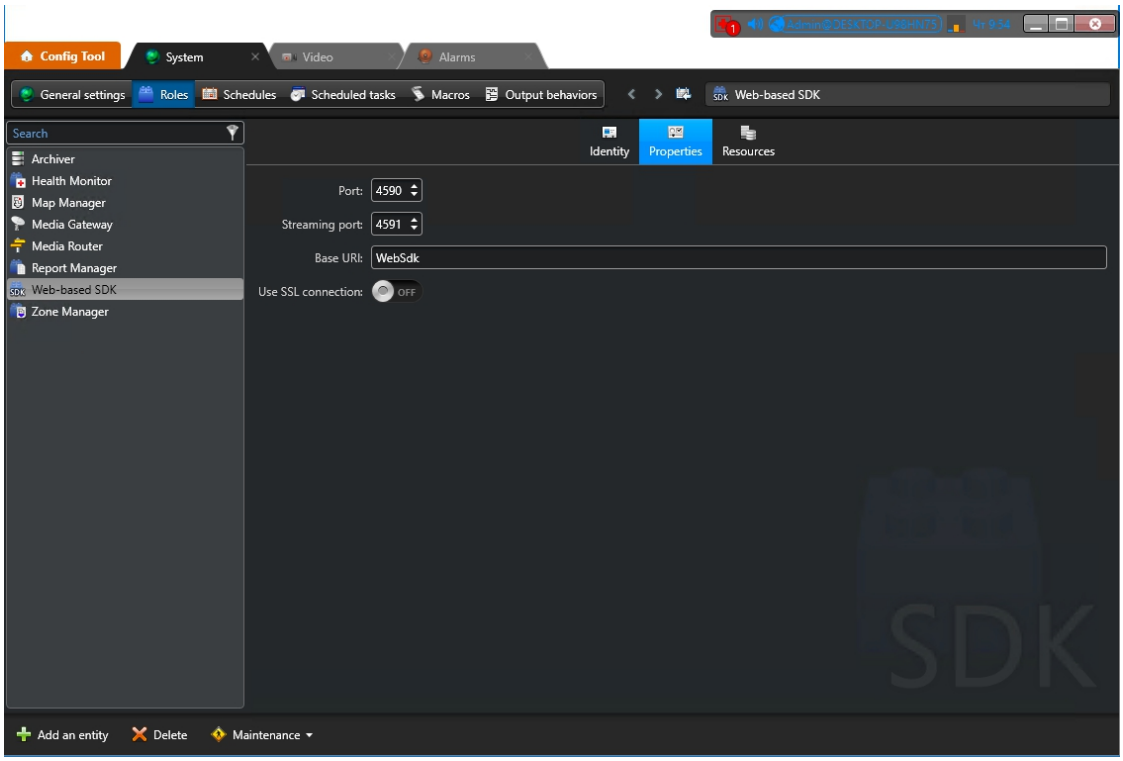
OK

In this chapter:

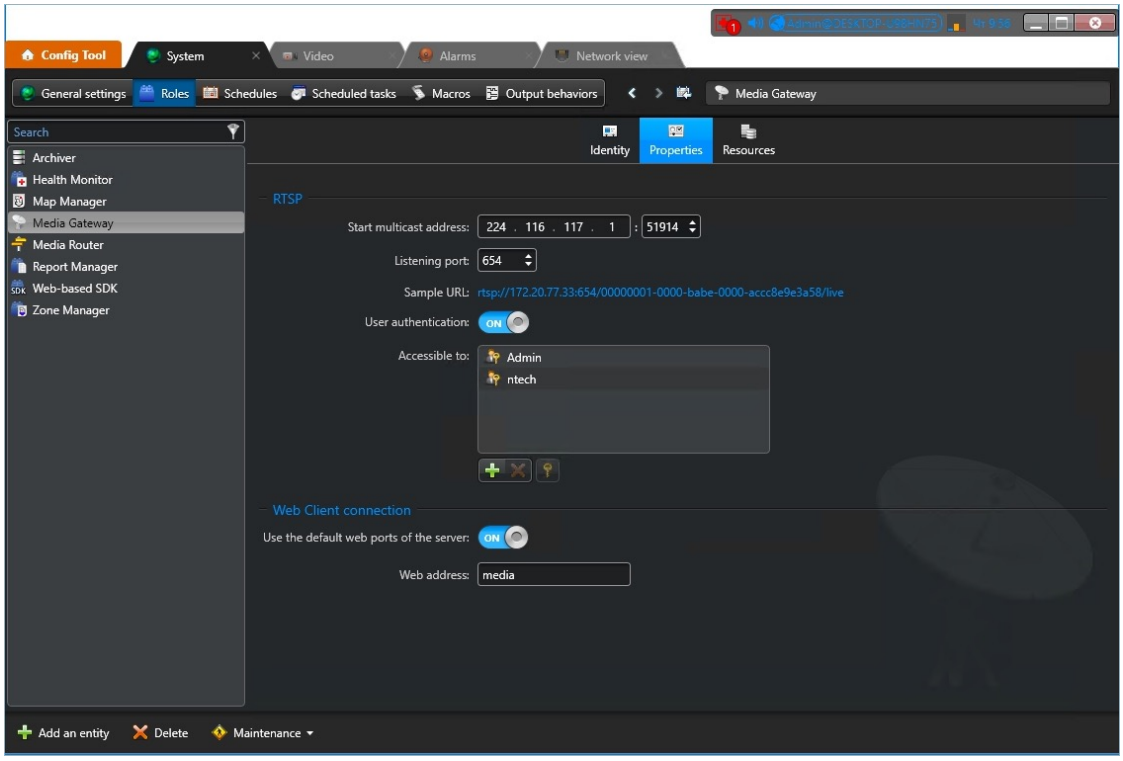
- *Configure Genetec Web SDK and Media Gateway*
- *Create Alarm in Genetec Security Center*
- *Enable Genetec Integration in FindFace*
- *Configure Endpoints in FindFace*
- *Import Cameras from Genetec Security Center*
- *Create Watch Lists and Dossiers in FindFace*

Configure Genetec Web SDK and Media Gateway

To enable and configure Web SDK, use Genetec Config Tool. For details, refer to *Security Center Administrator Guide -> Chapter 52: Role Types -> Web-based SDK configuration tabs*.



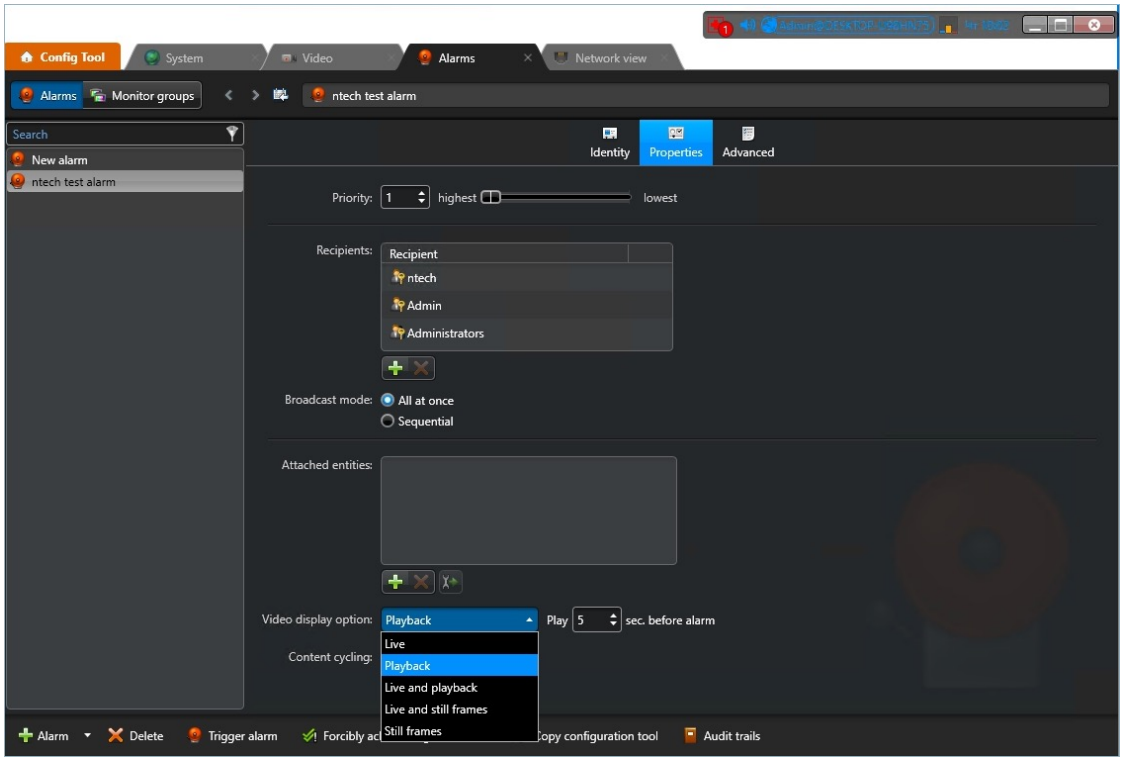
When enabling and configuring Media Getaway in Genetec Config Tool, refer to *Security Center Administrator Guide -> Chapter 24: Video Deployment*.



Important: Make sure that the firewall is configured so that the ports for the WebSDK and Media Gateway are open.

Create Alarm in Genetec Security Center

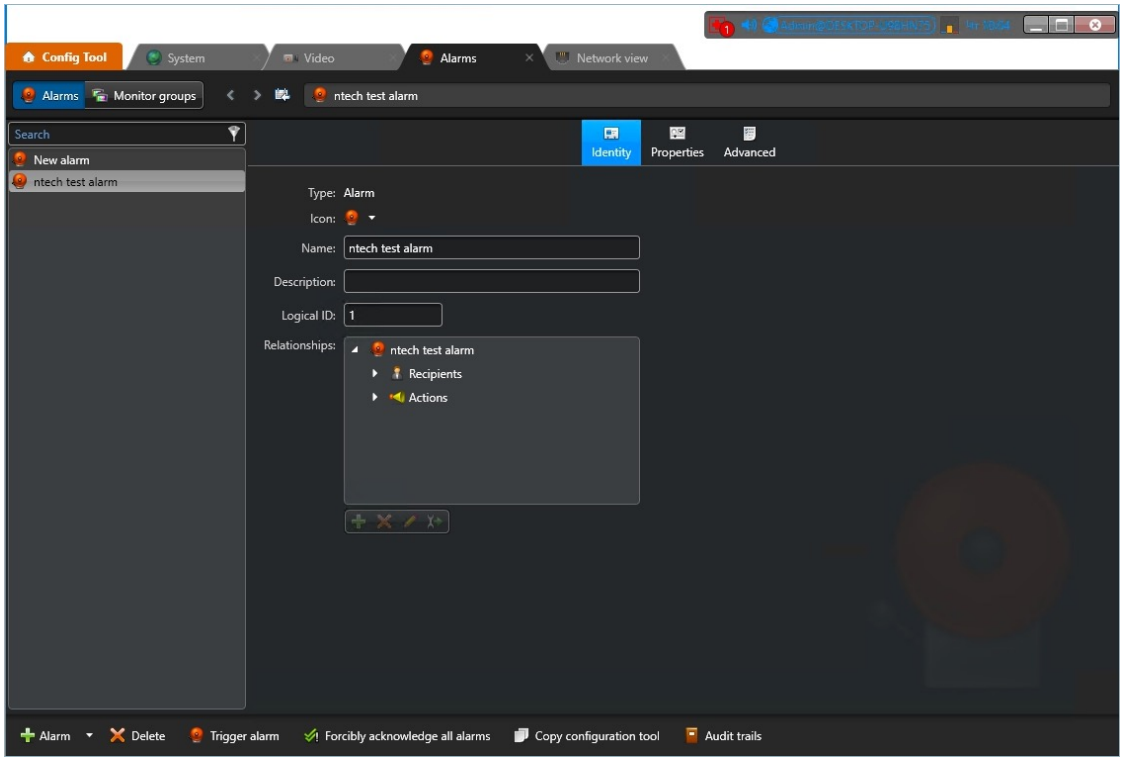
Create and configure a new `Alarm` entity in Genetec Config Tool. Refer to *Security Center Administrator Guide -> Chapter 48: Alarms -> Creating Alarms* for details.



Tip: On the *Properties* tab, select the *Video display option* that suits your needs the best. Available options are *Live*, *Playback*, etc.

Tip: To enable alarm procedures and auto rotation of video right within the alarm pop-up window, enable *Content cycling*.

When configuring the integration in FindFace, you will have to enter the alarm logical id that is specified on the *Identity* tab.



Enable Genetec Integration in FindFace

To enable the Genetec integration in FindFace, do the following:

1. Enable the `findface-genetec` plugin. To do so, open the `/etc/findface-security/config.py` configuration file and uncomment the `INSTALLED_APPS.append('ffsecurity_genetec')` line. Make sure that at least one of the following parameters is specified: `SERVICE_EXTERNAL_ADDRESS` or `EXTERNAL_ADDRESS`.

```
sudo vi /etc/findface-security/config.py

...
# SERVICE_EXTERNAL_ADDRESS is prioritized for FFSecurity webhooks and Genetec␣
↪plugin.
# EXTERNAL_ADDRESS is used instead if SERVICE_EXTERNAL_ADDRESS is not provided.
# You must provide either SERVICE_EXTERNAL_ADDRESS or EXTERNAL_ADDRESS in order
# to be able to work with FFSecurity webhooks and Genetec plugin.
SERVICE_EXTERNAL_ADDRESS = 'http://127.0.0.1'
# EXTERNAL_ADDRESS is used to access objects created inside FFSecurity via external␣
↪links.
EXTERNAL_ADDRESS = ''

...
# FindFace PLUGINS
# ==============================================================================
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.
...
# =============== Genetec ================
```

(continues on next page)

(continued from previous page)

```
INSTALLED_APPS.append('ffsecurity_genetec')
```

2. Migrate the main database architecture from FindFace to **PostgreSQL** and re-create user groups with *predefined* rights.

```
sudo findface-security migrate
sudo findface-security create_groups
```

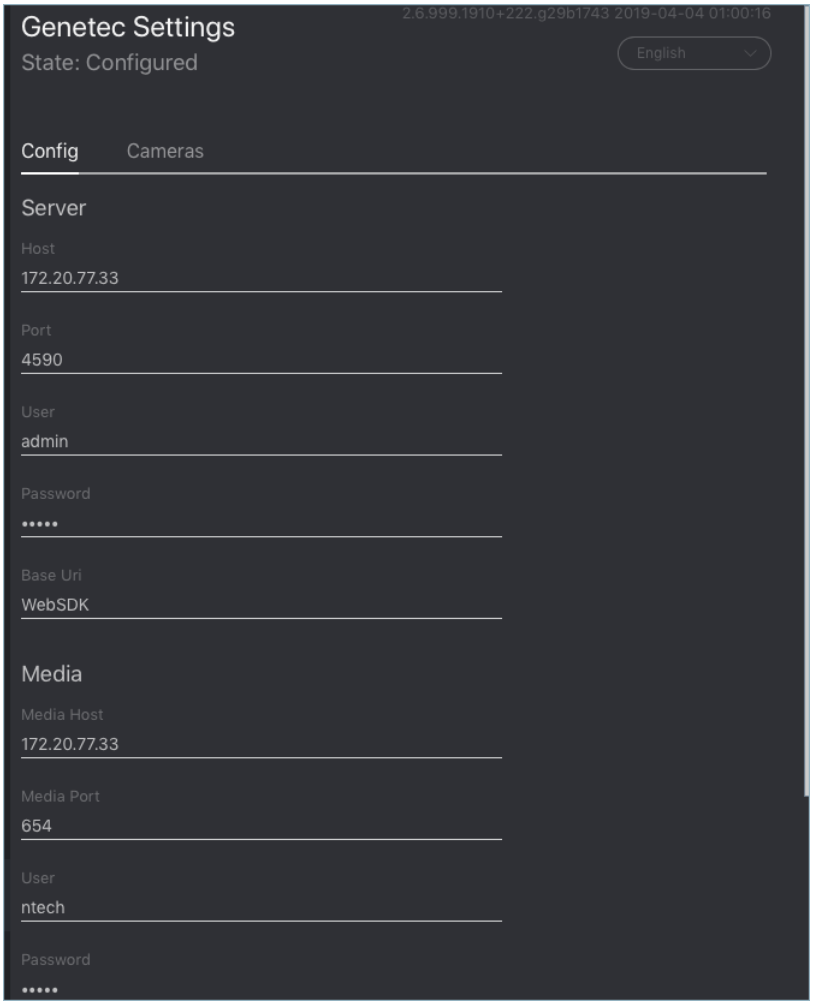
3. Restart `findface-security`.

```
sudo systemctl restart findface-security.service
```

Configure Endpoints in FindFace

To establish connection between FindFace and Genetec Security Center, do the following:

1. Navigate to the *Preferences* tab. Click *Genetec*.



2. In the *Server* and *Media* sections, specify *settings* of the Web SDK and Media Gateway endpoints.

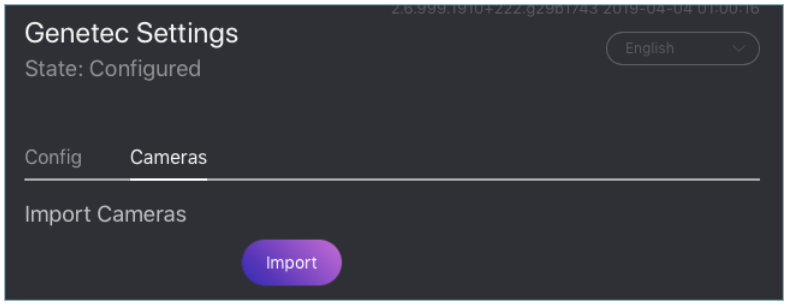
Important: The ports for the WebSDK and Media Gateway need to be open.

3. In the *Ids* section, specify the *logical id* of the `Alarm` entity that will be triggered in Genetec Security Center when a face recognition event occurs in FindFace.

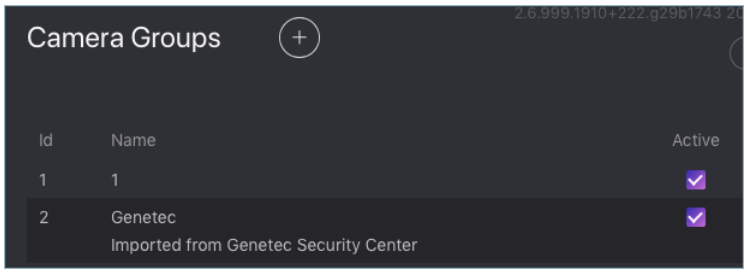
4. Click *Save*. If the connection to Genetec Security Center is successfully established, you will see the *State* change to *Configured*.

Import Cameras from Genetec Security Center

Once the connection to Genetec Security Center is established, import cameras. To do so, click *Cameras* on the *Genetec* tab. Click *Import*.



This action will create a *group of cameras* `Genetec` listing all the cameras from Genetec Security Center.



To view this list of cameras, navigate to the *Cameras* tab on the FindFace navigation bar. If you want to exclude a camera from face recognition, simply deactivate it in the list.

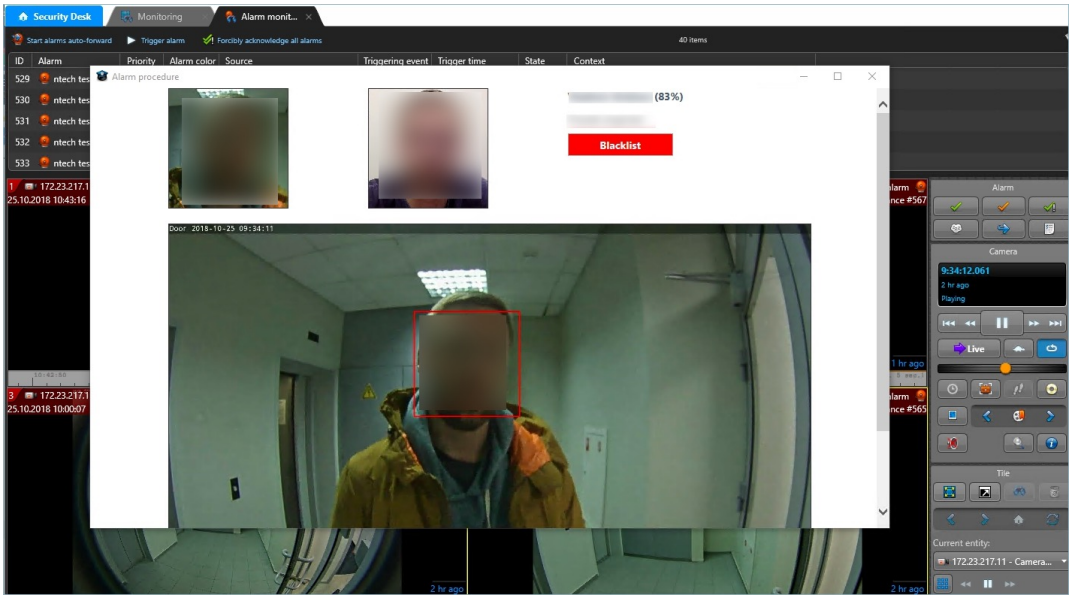
Create Watch Lists and Dossiers in FindFace

After you have configured the endpoints and camera settings, finish the integration by creating a *dossier database*. Notifications about face recognition events will be automatically sent to Genetec Security Center. See *Notifications in Genetec Security Center*.

Notifications in Genetec Security Center

Each face recognition event from a Genetec camera, that has a match with a dossier, triggers a relevant alarm in Genetec Security Center. Every alarm triggered by FindFace is associated with a relevant camera (source of the face recognition event) so you can instantly watch live or playback video within the `Alarm Monitoring` task in Genetec Security Desk. FindFace also utilizes `Alarm Procedures` to provide a user with additional content related to the alarm, such as:

- face detected in video
- matching face from the dossier database
- person's name and comment from the dossier
- matching confidence
- watch list's name
- full frame



After you receive a face recognition alarm, process it as you usually do with other alarms in Genetec Security Center.

3.3.2 Axxon Next

FindFace integration with Axxon Next allows you to detect and identify faces in video streams from an Axxon-based security system.

Important: One FindFace instance supports interaction with only one Axxon Next server.

Integration with Axxon Next is implemented via the `ffsecurity_axxon` plugin.

To configure the FindFace integration with Axxon Next in Ubuntu, do the following:

1. Activate the plugin by uncommenting the `INSTALLED_APPS.append('ffsecurity_axxon')` line in the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py

...

# ==============================================================================
# FindFace PLUGINS
# ==============================================================================
# Uncomment lines below to enable plugins. Please consult documentation for
# a plugin specific settings.
# =============== Axxon ================
INSTALLED_APPS.append('ffsecurity_axxon')
```

2. Uncomment the `FFSECURITY->AXXON` section in the configuration file. Fill it out as shown in the example below. In the `api` parameter, specify the IP address of the Axxon Next server that will provide FindFace with Axxon API and HLS-archive streams. In the `rtsp` parameter, specify the common segment of Axxon video stream addresses. `name`, `user`, `password`: the Axxon Next server name and credentials to access it.

```
FFSECURITY['AXXON'] = [
    {
        'name': 'server_name',
        'api': 'http://example.com/',
        'rtsp': 'rtsp://example.com:554/',
        'user': 'user',
        'password': 'password',
    }
]
```

3. (Optional). If facial recognition events are required to contain video from Axxon Next, uncomment the `FFSECURITY_UI_CONFIG['dossier']` section.

```
FFSECURITY_UI_CONFIG['dossier'] = {
   'video': True,
}
```

4. Create representations of Axxon Next cameras in FindFace (see *Camera Management*). A camera representation URL must be specified in the format `axxon:<friendlyNameLong>`, where `friendlyNameLong` is a camera name on the Axxon Next server. Find out this name in the Axxon user interface, or via Axxon API by executing:

```
curl http://user:password@127.0.0.1/video-origins/
```

(continues on next page)

(continued from previous page)

```
{
   "OLOLOE-DEV/DeviceIpint.vhod_1/SourceEndpoint.video:0:0" : {
      "friendlyNameLong" : "vhod_1.Vhod_1",
      "friendlyNameShort" : "Vhod_1",
      "origin" : "OLOLOE-DEV/DeviceIpint.vhod_1/SourceEndpoint.video:0:0",
      "state" : "signal_restored"
   }
}
```

For the camera from the example above, URL must be specified as `axxon:vhod_1.Vhod_1`.

The configuration is now finished. If the integration is properly configured, FindFace will be detecting and identifying faces in Axxon Next video streams, and facial recognition events will be featuring video clips from Axxon Next (upon relevant settings).

3.4 Custom Plugins

In the course of configuring the system, you can set directives on how the system processes a face after detecting it in the video. To do so, write one or several plugins in Python. The plugins will allow you to configure video face detection outcomes individually for each use case.

To enable your plugin, use the `/etc/findface-facerouter.py` configuration file.

Tip: As an example, check out the default plugin `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`.

3.4.1 Deploy `findface-facerouter` in FindFace

To deploy the `findface-facerouter` component, do the following:

1. Install `findface-facerouter` either from the *console installer* or from the apt repository as such:

```
sudo apt update
sudo apt install -y findface-facerouter
```

2. Open the `/etc/findface-facerouter.py` configuration file.

```
sudo vi /etc/findface-facerouter.py
```

3. If the `findface-facerouter` and `findface-sf-api` components are installed on different hosts, uncomment the `sfapi_url` parameter and specify the `findface-sf-api` host IP address.

```
sfapi_url                       = 'http://localhost:18411'
```

4. Open the `/etc/findface-security/config.py` configuration file. In the `ROUTER_URL` parameter, actualize the `findface-facerouter` IP address and port (18820 by default). Specify either external or internal IP address, subject to the network through which `findface-video-worker` interacts with `findface-facerouter`.

```
sudo vi /etc/findface-security/config.py

...
```

(continues on next page)

(continued from previous page)

```
FFSECURITY = {
   'ROUTER_URL': 'http://127.0.0.1:18820/v0/frame?',
```

5. Enable the `findface-facerouter` service autostart and launch the service.

```
sudo systemctl enable findface-facerouter.service && sudo systemctl start findface-
↪facerouter.service
```

6. Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

3.4.2 Configure `findface-facerouter` to Use Plugins

Tip: To get started, try the default plugin `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`. Or go ahead and *create your own*.

Important: Be sure to *change* the Tarantool database structure prior, according to the processing directive in the plugin.

Important: The `findface-facerouter` component must be *installed and configured*.

To configure `findface-facerouter` to use plugins, do the following:

1. Put a plugin into a directory of your choice. All plugins in use have to be in the same directory.
2. Open the `/etc/findface-facerouter.py` configuration file. Uncomment the `plugin_dir` parameter and specify the plugin directory.

   Warning: The `findface-facerouter.py` content must be correct Python code.

```
sudo vi /etc/findface-facerouter.py

plugin_dir                    = '/etc/findface/plugins/'
```

3. Restart `findface-facerouter`.

```
sudo systemctl restart findface-facerouter.service
```

3.4.3 Basics

In this section:

- *Plugin Architecture*
- *The `preprocess` method*
- *The `process` method*
- *The `shutdown` method*

Plugin Architecture

After the `findface-video-worker` component detects a face, the face is posted to the `findface-facerouter` component via an HTTP API request. To process this request, each `findface-facerouter` plugin must export the `activate(app, ctx, plugin_name, plugin_source)` function.

The `activate` function has the following parameters:

- `app`: a `tornado.web.Application` entity of the `findface-facerouter` component.
- `ctx`: data context to be passed to a plugin upon activation.
- `plugin_name`: the name of the plugin to be activated.
- `plugin_source`: source object to load the plugin from.

Upon activation, a plugin is passed the following data context:

1. `request.ctx.sfapi`: a set up `ntech.sfapi_client.Client` instance that can be invoked directly to process the result of video face detection (for example, to create a new gallery, add a face to a gallery, etc.).
2. `plugins`: `OrderedDict` with all the plugins as (`key`: plugin name, `value`: the result returned by the `activate` function).
3. `idgen`: id generator that can be invoked as `ctx.idgen()`.

The `activate(app, ctx, plugin_name, plugin_source)` function must return an object with the following methods:

1. `preprocess`,
2. `process`,
3. `shutdown` (optional).

The `preprocess` method

In this method, a `findface-facerouter` plugin decides if it is interested in the face received from the `findface-video-worker` component. If so, it returns a tuple or a list that contains one or several strings `'facen'`, `'gender'`, `'age'`, `'emotions'`. This means that it is necessary to extract a biometric sample, recognize gender, age, emotions respectively. If the returned tuple/list is non-empty, the `findface-facerouter` redirects the face to the `findface-sf-api` in a `/detect POST` request with relevant query string parameters (`facen=on`, `gender=on`, `age=on`, `emotions=on`).

The basic `preprocess` method to inherit from has the following syntax (see the `Plugin` class):

preprocess(*self*, *request: FrHTTPRequest*, *labels: Mapping[str, str]*) → Tuple[str]

Parameters

- **FrHTTPRequest** (*tornado.httpserver.HTTPRequest*) – a HTTP API request that includes an extra argument `params`
- **labels** (*dictionary*) – a custom set of a frame labels, which are initially specified in a job parameters for `findface-video-worker` and then assigned to the frame

The `params` argument of FrHTTPRequest includes the following fields:

Parameters

- **photo** (*bytes*) – JPEG video frame featuring a detected face
- **face0** (*bytes*) – normalized face image
- **bbox** (list of integers [[x1,y1,x2,y2]], where `x1`: x coordinate of the top-left corner, `y1`: y coordinate of the top-left corner, `x2`: x coordinate of the bottom-right corner, `y2`: y coordinate of the bottom-right corner) – coordinates of the face region in the video frame
- **cam_id** (*string*) – camera id
- **timestamp** (*datetime.datetime*) – video frame timestamp
- **detectorParams** (*dictionary*) – debug information from the video face detector
- **bs_type** (*string*) – best face search mode. Available options: overall (the `findface-video-worker` posts only one snapshot per track, but of the highest quality.), realtime (the `findface-video-worker` posts the best snapshot within each of consecutive time intervals).
- **labels** (*dictionary*) – (duplicates params.labels) a custom set of a frame labels, which are specified in a job parameters for `findface-video-worker` and then assigned to the frame

The decision about face processing is made based on the data in the `request.params`, including the custom set of labels, as well as for any other reasons.

The process method

This method is called if the `preprocess` method returns a non-empty tuple or list (i.e. with 'facen', 'gender', 'age', an/or 'emotions' strings). After the `findface-sf-api` returns a response with the result of face detection (see the `/detect POST` request) with all the requested face features, the `findface-facerouter` component calls the `process` method of the plugin in order to the perform face processing itself.

To process a face, a plugin uses `request.ctx.sfapi`.

The basic `process` method to inherit from has the following syntax (see the `Plugin` class):

process(*self*, *request: FrHTTPRequest*, *photo: bytes*, *bbox: List[int]*, *event_id: int*, *detection:* DetectFace)

The shutdown method

This method is only called before the findface-facerouter shutdown.

The basic shutdown method to inherit from has the following syntax (see the Plugin class):

shutdown(*self*)

3.4.4 Classes and Methods

In this section:

- *Basic Classes*
- *Object Classes*
- *Face Detection and Gallery Management*
- *Filters for Database Search*
- *Display Error Messages*

Basic Classes

class facerouter.plugin.**Plugin**

Provides the basic methods for writing a plugin (see *Basics*). A custom class that wraps a plugin must inherit from the Plugin class.

preprocess(*self*, *request: FrHTTPRequest*, *labels: Mapping[str, str]*) → Tuple[str]

Returns a tuple that contains one or several strings 'facen', 'gender', 'age', 'emotions'. This means that findface-facerouter must request findface-extraction-api to extract a biometric sample, recognize gender, age, emotions respectively.

Parameters

- **FrHTTPRequest** (*tornado.httpserver.HTTPRequest*) – a HTTP API request that includes an extra argument params
- **labels** (*dictionary*) – a custom set of a frame labels from request.params

Returns
one or several strings 'facen', 'gender', 'age', 'emotions'

Return type
tuple

The params argument of FrHTTPRequest includes the following fields:

Parameters

- **photo** (*bytes*) – JPEG video frame featuring a detected face
- **face0** (*bytes*) – normalized face image
- **bbox** (list of integers [[x1,y1,x2,y2]], where x1: x coordinate of the top-left corner, y1: y coordinate of the top-left corner, x2: x coordinate of the bottom-right corner, y2: y coordinate of the bottom-right corner) – coordinates of the face region in the video frame
- **cam_id** (*string*) – camera id

- **timestamp** (*datetime.datetime*) – video frame timestamp
- **detectorParams** (*dictionary*) – debug information from the video face detector
- **bs_type** (*string*) – best face search mode. Available options: overall (the `findface-video-worker` posts only one snapshot per track, but of the highest quality.), realtime (the `findface-video-worker` posts the best snapshot within each of consecutive time intervals).
- **labels** (*dictionary*) – (duplicates params.labels) a custom set of a frame labels, which are specified in a job parameters for `findface-video-worker` and then assigned to the frame

process(*self*, *request: FrHTTPRequest*, *photo: bytes*, *bbox: List[int]*, *event_id: int*, *detection:* DetectFace)

Accepts the detected face features.

Parameters

- **request** (*tornado.httpserver.HTTPRequest*) – a HTTP API request from `findface-video-worker`
- **photo** (*bytes*) – JPEG video frame featuring a detected face, from `request.params`
- **bbox** (list of integers [[x1,y1,x2,y2]], where `x1`: x coordinate of the top-left corner, `y1`: y coordinate of the top-left corner, `x2`: x coordinate of the bottom-right corner, `y2`: y coordinate of the bottom-right corner) – coordinates of the face region in the video frame, from `request.params`
- **event_id** (*uint64*) – id of the face automatically set by `findface -facerouter` upon receiving it from `findface-video-worker`. Can be used as a face custom identifier in the biometric database.
- **detection** (`objects.DetectFace`) – detection result received from `findface-sf-api`, that contains requested face features such as faces, gender, age and emotions.

Returns

n/a

Return type

n/a

shutdown(*self*)

This method is invoked before the `findface-facerouter` shutdown.

Parameters

n/a

Returns

n/a

Object Classes

class objects.**BBox**

Represents coordinates of the rectangle around a face.

class objects.**DetectFace**

Represents a detection result with the following fields:

Parameters

- **id** (*string*) – id of the detection result in memcached
- **bbox** (*objects.Bbox*) – coordinates of the rectangle around a face
- **features** (*dictionary*) – (optional) information about gender, age and emotions

class objects.**DetectResponse**

Represents a list of `objects.DetectionFace` objects with an additional field `orientation` featuring information about the face EXIF orientation in the image.

Parameters

orientation (*EXIF orientation*) – orientation of a detected face

class objects.**FaceId**(*namedtuple('FaceId', ('gallery', 'face'))*)

Represents a custom face identifier object in the gallery.

Parameters

- **gallery** (*string*) – gallery name
- **face** (*integer*) – custom face identifier in the gallery

class objects.**Face**

Represents a result of database search by biometric sample

Parameters

- **id** (`objects.FaceId`) – FaceId object.
- **features** (*dictionary*) – information about gender, age and emotions
- **meta** (*dictionary*) – face meta data
- **confidence** (*float*) – similarity between the biometric sample and a face in the search result

class objects.**ListResponse**

Represents a list of `objects.Face` objects (i.e. a list of biometric sample search results) with an additional field `next_page` featuring the cursor for the next page with search results.

Parameters

next_page (*string*) – cursor for the next page with search results

Face Detection and Gallery Management

class ntech.sfapi_client.client.**Client**

Represents basic methods to detect faces in images and work with galleries.

detect(*self*, *, *url=None*, *image=None*, *facen=False*, *gender=False*, *age=False*, *emotions=False*, *return_facen=False*, *autorotate=False*, *detector: str = None*, *timeout=None*) → *DetectResponse*

Detects a face and returns the result of detection.

Parameters

- **url** (*URL*) – image URL if you pass an image that is publicly accessible on the internet
- **image** (*bytes*) – PNG/JPG/WEBP image file is you pass an image as a file
- **facen** (*boolean*) – extract a biometric sample from the detected face. To save the detection result in memcached pass `facen=True`
- **gender** (*boolean*) – extract and return information about gender
- **age** (*boolean*) – extract and return information about age
- **emotions** (*boolean*) – extract and return information about emotions
- **return_facen** (*boolean*) – return facen in the method result
- **autorotate** (*boolean*) – automatically rotate the image in 4 different orientations to detect faces in each of them. Overlapping detections with IOU > 0.5 will be merged
- **detector** (*boolean*) – nnd or normalized. The `normalized` detector is used to process normalized images, for example, those which are received from `fkvideo_worker`.
- **timeout** (*number*) – FindFace core response timeout, in seconds (if `none`, the default value is used)

Returns
Detection result

Return type
`DetectorResponse` object.

gallery(*self*, *name*)

Returns a gallery object `sfapi_client.Gallery` to refer to it later (for example, to list gallery faces).

Parameters
name (*string*) – gallery name

Returns
a gallery object

Return type
sfapi_client.Gallery

list_galleries(self, timeout=None):

Returns the list of galleries.

Parameters
timeout (*number*) – FindFace core response timeout, in seconds (if `none`, the default value is used)

Returns
list of galleries with the fields `name` (a gallery name, string) and `number` (the number of faces in the gallery, number)

Return type
list of GalleryListItem

class ntech.sfapi_client.gallery.**Gallery**

Provides methods to work with galleries and faces.

list(*self*, **, filters: Iterable[*filters.Filter*] = None, limit: int = 1000, sort: str = '', page=None, ignore_errors=False, timeout=None*) → *ListResponse*

Returns a list-like object with faces from the gallery, that match the given filters. The returned list-like object has an additional property `next_page` which can be used as a value for the `page` parameter in next requests.

Parameters

- **filters** (`sfapi_client.filters.Filter`) – list of filters
- **limit** (*integer*) – maximum number of returned faces
- **sort** (*string*) – sorting order. Pass one of the following values: `id`: increasing order by id, `-id`: decreasing order by id (sorting by id is used if you have NOT specified a feature vector to search for), `-confidence`: decreasing order by face similarity (only if you have specified a feature vector to search for). By default, the method uses the `id` order (no feature vector specified), or `-confidence` (with feature vector).
- **page** – cursor of the next page with search results. The `page` value is returned in the response in the `next_page` parameter along with the previous page results.
- **ignore_errors** (*boolean*) – By default, if one or several `findface-tarantool-server` shards are out of service during face identification, `findface-sf-api` returns an error. Enable this Boolean parameter to use available `findface-tarantool-server` shards to obtain face identification results.
- **timeout** (*number*) – FindFace core response timeout, in seconds (if `none`, the default value is used)

Returns
list with faces from the gallery, that match the given filters.

Return type
ListResponse object

add(*self*, *new_id: Union[int, Callable], source: Union[*DetectFace*,* Face*, str], *, meta: Dict[str, Union[int, str, List[str]]] = None, regenerate_attempts=None, timeout=None*) → *Face*

Creates a face in the gallery.

Parameters

- **new_id** (*integer or callable*) – custom face identifier (Face ID) in the database gallery. May be a (async) callable which returns the id. To generate id, you can use the `ctx.idgen()` function delivered with the context.
- **source** (*sfapi_client.DetectFace, sfapi_client.Face, sfapi_client.FaceId, or string*) – face source: create a face using another face in the database or a detection result as a source.
- **meta** (*dictionary*) – face metadata. Keys must be strings and values must be either ints, strings or lists of strings. Metadata keys and types must be previously specified in the storage configuration files (`/etc/tarantool/instances.available/*.lua`).
- **regenerate_attempts** – number of attempts to regenerate a unique Face ID with the `ctx.idgen()` function if new_id is callable

- **timeout** (*number*) – FindFace core response timeout, in seconds (if none, the default value is used)

Returns
representation of the newly created face

Return type
Face object

delete(*self*, *face: Union[Face, int]*, *timeout=None*) → None

Removes a face from the gallery.

Parameters

- **face** (*sfapi_client.Face, sfapi_client.FaceId or id in integer*) – face to be removed
- **timeout** (*number*) – FindFace core response timeout, in seconds (if none, the default value is used)

Returns
None

get(*self*, *face: Union[Face, int]*, *timeout=None*) → *Face*

Retrieves a face from the gallery.

Parameters

- **face** (*sfapi_client.Face, sfapi_client.FaceId or id in integer*) – face to be retrieved
- **timeout** (*number*) – FindFace core response timeout, in seconds (if none, the default value is used)

Returns
representation of the face

Return type
Face object

create(*self*, *timeout=None*) → None

Creates a gallery in `findface-sf-api` as a `sfapi_client.Gallery` object. Being a proxy object, `sfapi_client.Gallery` doesn't require a gallery to be existing on the server.

Parameters
timeout (*number*) – FindFace core response timeout, in seconds (if none, the default value is used)

Returns
None

drop(*self*, *timeout=None*) → None:

Removes a gallery from `findface-sf-api`.

Parameters
timeout (*number*) – FindFace core response timeout, in seconds (if none, the default value is used)

Returns
None

update(*self*, *face: Union[Face, str]*, *, *meta: Dict[str, Union[int, str, List[str]]] = None*, *timeout=None*) → *Face*

Update face meta data in the gallery.

Parameters

- **face** (*sfapi_client.Face, sfapi_client.FaceId or id in integer*) – face to be updated
- **meta** (*dictionary*) – face meta data to be updated. Keys must be strings and values must be either ints, strings or lists of strings. If a meta string is not passed or passed as null, it won't be updated in the database.
- **timeout** (*number*) – FindFace core response timeout, in seconds (if `none`, the default value is used)

Returns
representation of the updated face

Return type
Face object

Filters for Database Search

class ntech.sfapi_client.filters.**Filter**

Generic class. Represents a list of filters (with assigned values) that have to be applied to the gallery content.

serialize(*self*)

Method that passes the list of filters with assigned values to the `findface-sf-api` component.

Returns
filter names and filter values

Return type
tuple ('filtername', ["value1", "value2"]).

class ntech.sfapi_client.filters.**Id**

Represents methods for filtering gallery content by id. Don't instantiate, use relevant classmethods to call a filter.

classmethod lte(*cls*, *value: int*) → *Filter*

LTE filter. Select all faces with `id` less or equal to `value`.

Parameters
value (*integer*) – id value

Returns
filter name (LTE) and its value.

Return type
object of `Filter` class.

Example: `Id.lte(1234)` selects faces with id less or equal to `1234`.

classmethod gte(*cls*, *value: int*) → *Filter*

GTE filter. Select all faces with `id` greater or equal to `value`.

Parameters
value (*integer*) – id value

Returns
filter name (GTE) and its value.

Return type
object of `Filter` class.

Example: `Id.lte(1234)` selects faces with id greater or equal to `1234`.

classmethod oneof(*cls*, **value: Union[int]*) → *Filter*

IN filter. Select a face(s) with id from a given set.

Parameters
value (*list of integers*) – list of id values

Returns
filter name (IN) and its value.

Return type
object of `Filter` class.

Example: `Id.oneof(1234, 5678)` selects a face(s) with id `1234` and/or `5678`.

class ntech.sfapi_client.filters.**Meta**

Represents methods for filtering gallery content by metadata. Don't instantiate, use relevant classmethods to call a filter.

classmethod lte(*self*, *value: Union[str, int]*) → *Filter*

LTE filter. Select all faces with a metastring less or equal to `value`

Parameters
value (*string or integer*) – metastring value

Returns
filter name (LTE) and its value.

Return type
object of `Filter` class.

Example: `Meta('foo').lte(1234)` selects faces with a metastring `foo` less or equal to `1234`.

classmethod gte(*self*, *value: Union[str, int]*) → *Filter*

GTE filter. Select all faces with a metastring greater or equal to `value`

Parameters
value (*string or integer*) – metastring value

Returns
filter name (GTE) and its value.

Return type
object of `Filter` class.

Example: `Meta('foo').gte(1234)` selects faces with a metastring `foo` greater or equal to `1234`.

classmethod oneof(*self*, **value: Union[str, int]*) → *Filter*

IN filter. Select a face(s) with a metastring from a given set.

Parameters
value (*list of strings or integers*) – list of metastring values

Returns
filter name (IN) and its value.

Return type
object of `Filter` class.

Example: `Meta.oneof(1234, 5678)` selects a face(s) with a metastring `1234` and/or `5678`.

classmethod subset(*self*, **value: str*) → *Filter*

SUBSET filter. Select all faces with a metastring featuring all values from a given set.

Parameters
value (*list of strings or integers*) – list of metastring values

Returns
filter name (SUBSET) and its value.

Return type
object of `Filter` class.

Example: `Meta('foo').subset("male", "angry")` selects face with a metastring `foo` featuring all values from the set ["male", "angry"].

class ntech.sfapi_client.filters.**Detection**(*Filter*)

Represents a method that identifies a detected face (searches the database for similar faces).

__init__(*self*, *id: Union[str,* objects.DetectFace*]*, *threshold: float*)

Parameters

- **id** (`objects.DetectFace` or temporary face id in memcached returned by sfapi_client.Client.detect(), string) – face (detection result) to be identified
- **threshold** (*float*) – identification threshold similarity between faces from 0 to 1.

Example: `Detection(det1, 0.77)` selects faces similar to the detection result `det1` with similarity greater or equal to `0.77`.

class ntech.sfapi_client.filters.**Face**(*Filter*)

Represents a method that searches the database for faces similar to a given face from a gallery.

__init__(*self*, *id: Union[str,* objects.Face*]*, *threshold: float*)

Parameters

- **id** (`objects.Face`, `objects.FaceId` or custom face id in the gallery, string) – face from a gallery to be identified
- **threshold** (*float*) – identification threshold similarity between faces from 0 to 1.

Example: Detection(FaceId("gal1", 1234), 0.77) selects faces similar to the face 1234 from the `gal1` gallery with similarity greater or equal than 0.77.

Several Filters Usage Example

```
filters=[filters.Id.gte(123456), filters.Meta('age').gte(45), filters.Meta('camera').
↪oneof('abc', 'def')]
```

Display Error Messages

class sfapi_client.**SFApiRemoteError**

This error message appears if the error occurred for a reason other than a network failure.

The error body always includes at least two fields:

- `code` is a short string in `CAPS_AND_UNDERSCORES`, usable for automatic decoding.
- `reason` is a human-readable description of the error and should not be interpreted automatically.

Common Error Codes

Error code	Description
`UNKNOWN_ERROR`	Error with unknown origin.
`BAD_PARAM`	The request can be read, however, some method parameters are invalid. This response type contains additional attributes `param` and``value`` to indicate which parameters are invalid.
`CONFLICT`	Conflict.
`EXTRACTION_ERROR`	Error upon a face feature vector extraction.
`LICENSE_ERROR`	The system configuration does not match license.
`MALFORMED_REQUEST`	The request is malformed and cannot be read.
`OVER_CAPACITY`	The `findface-extraction-api` queue length has been exceeded.
`SOURCE_NOT_FOUND`	The face in the `from` parameter does not exist.
`SOURCE_GALLERY_NOT_FOUND`	The gallery in the `from` parameter does not exist.
`STORAGE_ERROR`	The biometric database not available.
`CACHE_ERROR`	Memcached not available.
`NOT_FOUND`	Matching faces not found.
`NOT_IMPLEMENTED`	This functionality not implemented.
`GALLERY_NOT_FOUND`	Matching galleries not found.

class sfapi_client.**SFApiMalformedResponseError**

This error message appears if the error occurred due to a network failure, or if Client was unable to read an API response from `findface-sf-api`.

3.4.5 Example

The following example illustrates the basics of writing a plugin, as well as the use of classes and methods. This plugin requests face features from `findface-sf-api` and then sends a request to `<FFSEC_URL>/video-detector/process` to create an event with the data obtained from `findface-sf-api`.

You can find this plugin at `/opt/findface-security/fr_plugin/ffsec_fr_plugin.py`. Embed it as described *here* and try it out.

Important: Make sure that the `FFSEC_URL` variable contains the actual IP address and port of the `findface-security` host.

```
import datetime
import logging
import aiohttp
from dateutil.tz import tzutc
```

(continues on next page)

(continued from previous page)

```
from facerouter.plugin import Plugin
from ntech import sfapi_client
from ntech.asyncio_utils import wrap_futures
from ntech.asyncio_utils.noop_cookie import NoopCookieJar
from ntech.tornado_utils import asyncio_to_tornado
# change this if your ffsecurity is located on another host or listens on a non-default␣
↪port
FFSEC_URL = 'http://127.0.0.1:8002'
logger = logging.getLogger(__name__)
class FFSecurityPlugin(Plugin):
    def __init__(self, ctx, ffsec_url):
        super().__init__(ctx)
        self.ffsec_url = ffsec_url.rstrip('/')
        self.session = aiohttp.ClientSession(cookie_jar=NoopCookieJar())
        self.future_wrapper = asyncio_to_tornado
    def deactivate(self, *args):
        self.session.close()
    def request_headers(self, request):
        return {
            "Authorization": request.headers['Authorization'],
            'X-Request-ID': request.request_id,
        }
    @wrap_futures
    async def preprocess(self, request, labels):
        # somewhat hacky way to pass data between preprocess and process:
        request.ffsec_reception_timestamp = datetime.datetime.now(tzutc())
        headers = self.request_headers(request)
        async with self.session.post(self.ffsec_url + '/video-detector/preprocess',␣
↪headers=headers) as resp:
            resp.raise_for_status()
            resp_json = await resp.json()
            logger.debug("request_id=%r preprocess: ffsecurity response: %r", request.
↪request_id, resp_json)
            plugin_wants = resp_json['plugin_wants']
            request.ffsec_plugin_wants = plugin_wants
            logger.info("request_id=%r preprocess: ffsecurity requested features: %r",␣
↪request.request_id, plugin_wants)
        return plugin_wants
    @wrap_futures
    async def process(self, request, photo, bbox, event_id, detection: sfapi_client.
↪DetectFace):
        headers = self.request_headers(request)
        with aiohttp.MultipartWriter('form-data') as mpwriter:
            part = aiohttp.payload.BytesPayload(request.params.photo)
            part.set_content_disposition('form-data', name='photo', filename='photo.jpg')
            mpwriter.append(part)
            part = aiohttp.payload.BytesPayload(b'')
            part.set_content_disposition('form-data', name='normalized', filename='norm.
↪png')
            mpwriter.append(part)
            part = aiohttp.payload.JsonPayload(request.params.detectorParams)
            part.set_content_disposition('form-data', name='detectorParams')
```

(continues on next page)

(continued from previous page)

```
            mpwriter.append(part)
            part = aiohttp.payload.JsonPayload([list(bbox)])
            part.set_content_disposition('form-data', name='bbox')
            mpwriter.append(part)
            part = aiohttp.payload.StringPayload(request.params.cam_id)
            part.set_content_disposition('form-data', name='cam_id')
            mpwriter.append(part)
            part = aiohttp.payload.StringPayload(request.params.timestamp.isoformat())
            part.set_content_disposition('form-data', name='timestamp')
            mpwriter.append(part)
            part = aiohttp.payload.StringPayload(request.ffsec_reception_timestamp.
↪isoformat())
            part.set_content_disposition('form-data', name='reception_timestamp')
            mpwriter.append(part)
            part = aiohttp.payload.JsonPayload(request.ffsec_plugin_wants)
            part.set_content_disposition('form-data', name='plugin_wants')
            mpwriter.append(part)
            if request.params.bs_type is not None:
                part = aiohttp.payload.StringPayload(request.params.bs_type)
                part.set_content_disposition('form-data', name='bs_type')
                mpwriter.append(part)
            part = aiohttp.payload.JsonPayload({
                'id': getattr(detection, 'id', None),
                'features': detection.features,
                'bbox': detection.bbox._asdict(),
                'facen': getattr(detection, 'facen', None),
                'attributes': detection.attributes,
            })
            part.set_content_disposition('form-data', name='detection')
            mpwriter.append(part)
            async with self.session.post(
                    self.ffsec_url + '/video-detector/process',
                    data=mpwriter,
                    headers=headers
            ) as resp:
                await resp.read()
                resp.raise_for_status()
        logger.info("request_id=%r process: ffsecurity accepted event", request.request_
↪id)
async def activate(app, ctx, plugin_name, plugin_source):
    plugin = FFSecurityPlugin(ctx=ctx, ffsec_url=FFSEC_URL)
    return plugin
```

PYTHON MODULE INDEX

f

facerouter.plugin, 266

n

ntech.sfapi_client.client, 269
ntech.sfapi_client.filters, 272
ntech.sfapi_client.gallery, 270

o

objects, 268

INDEX

Symbols

__init__() (*ntech.sfapi_client.filters.Detection method*), 274
__init__() (*ntech.sfapi_client.filters.Face method*), 274

A

add() (*ntech.sfapi_client.gallery.Gallery method*), 270

B

BBox (*class in objects*), 268

C

Client (*class in ntech.sfapi_client.client*), 269
create() (*ntech.sfapi_client.gallery.Gallery method*), 271

D

delete() (*ntech.sfapi_client.gallery.Gallery method*), 271
detect() (*ntech.sfapi_client.client.Client method*), 269
Detection (*class in ntech.sfapi_client.filters*), 274
drop() (*ntech.sfapi_client.gallery.Gallery method*), 271

F

Face (*class in ntech.sfapi_client.filters*), 274
facerouter.plugin
    module, 266
Filter (*class in ntech.sfapi_client.filters*), 272

G

Gallery (*class in ntech.sfapi_client.gallery*), 270
gallery() (*ntech.sfapi_client.client.Client method*), 269
get() (*ntech.sfapi_client.gallery.Gallery method*), 271
gte() (*ntech.sfapi_client.filters.Id class method*), 272
gte() (*ntech.sfapi_client.filters.Meta class method*), 273

I

Id (*class in ntech.sfapi_client.filters*), 272

L

list() (*ntech.sfapi_client.gallery.Gallery method*), 270
lte() (*ntech.sfapi_client.filters.Id class method*), 272
lte() (*ntech.sfapi_client.filters.Meta class method*), 273

M

Meta (*class in ntech.sfapi_client.filters*), 273
module
    facerouter.plugin, 266
    ntech.sfapi_client.client, 269
    ntech.sfapi_client.filters, 272
    ntech.sfapi_client.gallery, 270
    objects, 268

N

ntech.sfapi_client.client
    module, 269
ntech.sfapi_client.filters
    module, 272
ntech.sfapi_client.gallery
    module, 270

O

objects
    module, 268
objects.DetectFace (*class in objects*), 268
objects.DetectResponse (*class in objects*), 268
objects.Face (*class in objects*), 268
objects.FaceId (*class in objects*), 268
objects.ListResponse (*class in objects*), 268
oneof() (*ntech.sfapi_client.filters.Id class method*), 273
oneof() (*ntech.sfapi_client.filters.Meta class method*), 273

P

Plugin (*class in facerouter.plugin*), 266
preprocess(), 264
preprocess() (*facerouter.plugin.Plugin method*), 266
process(), 265
process() (*facerouter.plugin.Plugin method*), 267

S

serialize() (*ntech.sfapi_client.filters.Filter method*), 272

sfapi_client.SFApiMalformedResponseError (*class in ntech.sfapi_client.filters*), 275
sfapi_client.SFApiRemoteError (*class in ntech.sfapi_client.filters*), 275
shutdown(), 266
shutdown() (*facerouter.plugin.Plugin method*), 267
subset() (*ntech.sfapi_client.filters.Meta class method*), 274

U

update() (*ntech.sfapi_client.gallery.Gallery method*), 271