Smilart Helios User Guide

Contents

Overview

Smilart Helios version 3.0.0 is a soft-realtime reference implementation of the Smilart Web API version 1.1.

For correct work Smilart Helios requires implementations of Camera Server Erlang API version 1.4 and Phoenix Erlang API version 2.5.

Installation

Smilart Helios product installation is performed by Smilart Application Manager (SAM).

For more information about sam *see* Smilart Application Manager *documentation*

To install the Smilart Helios product one need to use installproduct command which will be available in the system after installation of docker image smilartos-install.
This image also contains information about all available products, their versions and location.

Image smilartos-install

smilartos-install — image, which contains information about all available Smilart products, their versions and locations.

To install it, you need to run several commands:

- Get all versions of smilartos-install and select the one you prefer (usually the latest)

```
$ sam se smilartos-install

      List image versions:

smilartos-install:1772.4.0_126
```

- Install the preferred version

```
$ sam in smilartos-install:1772.4.0_126
```

For more information about sam *see* Smilart Application Manager *documentation*

After installation ends image smilartos-install should appear in the list of images

```
$ sam list | grep smilartos
smilartos-install  |  1772.4.0_126  |  int
```

Then the `installproduct` command becomes available for usage.

Product Helios

- After installing the image `smilartos-install` you need to run the `installproduct` command.
- Product selection window appears where you should select `helios` to install `Smilart Helios` product.
- After that you need to select the required version of the `Smilart Helios` product.
- Screen with information about the selected version appears.
- After that system will collect information about the installation, download necessary files and offer to install collected images into the system.
- After confirmation system will complete the installation of the `Smilart Helios` product.

One can verify that the image was installed successfully by using

```
$ sam list

      List apps:

Name                                | Version       | Repository
----------------------------------------------------------------
helios.x86_64                       | 3.0.0_59      | int
...
```

If any errors occur during the installation one should contact technical support.

Product Phoenix System

To work `Smilart Helios` product requires another product — `Smilart Phoenix System` that provides implementations of `Camera Server Erlang API` version `1.4` and `Phoenix Erlang API` version `2.5`.

One can install that product in the same way as Smilart Helios product.

During installation of `Smilart Phoenix System` check the `Release notes` section for appropriated `Camera Server Erlang API` and `Phoenix Erlang API` versions.

Configuration

To check or modify configuration of `Smilart Helios`, you need to find configuration file, located at `/etc/helios/current/sys.config.orig`.

Text after `%` is a comment.

```
[
```

```
{sml_helios, [
  {phoenix_connection, [ %% Phoenix connection settings
    {node, "sml_phoenix@phoenix"}, %% Erlang node name
    {cookie, "sml_phoenix"} %% Cookie
  ]},
  {camera_server_connection, [ %% Camera_server connection settings
    {node, "sml_camserver@camera_server"}, %% Erlang node name
    {cookie, "sml_camserver"} %% Cookie
  ]},
  {cowboy_settings, [%% Http services settings
    {port, 9999},
    {websocket_idle_timeout_sec, 60}
  ]},
  {person_service_settings, [
    {rpc_timeout_ms, 2000},
    {photos_cache_size_bytes, 52428800}, %% 50*1024*1024 (50Mb)
    {photos_cache_timeout_ms, 60000} %% 1 min
  ]},
  {ipa_service_settings, [
    {rpc_timeout_ms, 2000}
  ]},
  {vca_service_settings, [
    {aggregation_time_ms, 200}
  ]},
  {photobooth_service_settings, [
    {aggregation_time_ms, 200}
  ]},
  {verification_service_settings, [
    {aggregation_time_ms, 200}
  ]},
  {poolboy_settings, [
    {blobstore, [
      {size, 10},
      {max_overflow, 20}
    ]}
  ]},
  {log_tracing, [
      % {trace_name, #{
      %   file=> "PathToLogFile.log",
      % Next opts applied by default
      %   config=> [{formatter, lager_default_formatter},
      %     {formatter_config, [date, " ", time, " ", message, "\r\n"]},
      %     {size, 104857600}, {date, ""}, {count, 4}
      %   ],
      %   level=> debug
      % }},
      {camera_collection, #{file => "api/camera.log"}},
      {person, #{file => "api/persons.log"}},
      {person_collection, #{file => "api/persons.log"}},
      {person_photo_collection, #{file => "api/persons.log"}},
      {person_photo, #{file => "api/persons.log"}},
```

```
        {vca, #{file => "api/vca.log"}},
        {photobooth, #{file => "api/photobooth.log"}},
        {verification, #{file => "api/verification.log"}},
        {ipa, #{file => "api/ipa.log"}}
    ]}
  ]},
  {sml_event_tracer, [ %% System events publishing
    {event_sender, {influx_udp, [
      {host, "${influx_host}"},
      {port, 4444},
      {pool_size, 5},
      {max_overflow, 10}
    ]}}
  ]},
  {lager, [
    {log_root, "log"},
    {async_threshold, 100},
    {async_threshold_window, 25},
    {crash_log_size, 104857600},
    {crash_log_date, ""},
    {crash_log_count, 4},
    {handlers, [
      {lager_file_backend, [{file, "log/info.log"}, {level, info}, {size, 104857600},
{date, ""}, {count, 4}]},
      {lager_file_backend, [{file, "log/warning.log"}, {level, warning}, {size,
104857600}, {date, ""}, {count, 4}]},
      {lager_file_backend, [{file, "log/error.log"}, {level, error}, {size,
104857600}, {date, ""}, {count, 4}]}
    ]},
    {extra_sinks, [
      {sml_lager_lager_event, [
        {async_threshold, 100},
        {async_threshold_window, 25}
      ]}
    ]}
  ]},
  {sasl, [
    {sasl_error_logger, false}
  ]}
].
```

Phoenix connection settings

The section `phoenix_connection` contains `Phoenix` connection settings.

Name	Type	Description
node	string	Phoenix node name in {NAME}@{HOST} format.
cookie	string	Phoenix erlang cookie.

Camera server connection settings

The section `camera_server_connection` contains `Camera Server` connection settings.

Name	Type	Description
node	string	Camera server node name in {NAME}@{HOST} format.
cookie	string	Camera server erlang cookie.

HTTP service settings

The section `cowboy_settings` contains HTTP services settings.

Name	Type	Description
port	integer	HTTP port.
websocket_idle_timeout_sec	integer	The lifetime of a WebSocket connection without PING/PONG messages.

Person Service settings

The section `person_service_settings` contains settings for Person Service.

Name	Type	Description
rpc_timeout_ms	integer	Maximum response time in milliseconds.
photos_cache_size_bytes	integer	Maximum storage capacity in bytes for cached person photos.
photos_cache_timeout_ms	integer	Timeout for cache invalidation in milliseconds.

IPA settings

The section `ipa_service_settings` contains settings for Instant Photo Analytics Service.

Name	Type	Description
rpc_timeout_ms	integer	Maximum response time in milliseconds.

VCA services settings

The section `vca_service_settings` contains settings for Video Content Analytics Service.

Name	Type	Description
aggregation_time_ms	integer	Timeout for an aggregation event to collect all required information.

Photobooth services settings

The section `photobooth_service_settings` contains settings for Photo Booth Service.

Name	Type	Description
aggregation_time_ms	integer	Timeout for an aggregation event to collect all required information.

Verification services settings

The section `verification_service_settings` contains settings for Verification Service.

Name	Type	Description
aggregation_time_ms	integer	Timeout for an aggregation event to collect all required information.

Poolboy Blobstore services settings

The sections `poolboy_settings.blobstore` contains settings for Blobstore Service.

Name	Type	Description
size	integer	Initial number of workers that collect information about images.
max_overflow	integer	Additional number of workers that collect information about images.

Events monitoring settings

The section `sml_event_tracer` contains events monitoring settings. Current events aggregator is InfluxDB (transfer protocol — UDP).

Name	Type	Description
influx_udp.host	string	InfluxDB hostname or IP-address.
influx_udp.port	integer	InfluxDB port.
influx_udp.pool_size	integer	Initial number of parallel connections to the InfluxDB.
influx_udp.max_overflow	integer	Additional number of parallel connections to the InfluxDB.

For more information about `Helios Monitoring` *see* `Helios Monitoring Guide` *documentation*

Implementation details of services

Common Notes

All services distribute all images/photos to clients only in JPEG format.

Verification service

Definition of verification parameter threshold_name

Verification request parameter `threshold_name` is a name of the predefined verification threshold in the `Phoenix Erlang API` implementation.

`Phoenix Erlang API` provides the following list of predefined threshold names from highest to lowest:

1. `UltraHigh`;
2. `High`;
3. `Normal` (recommended);
4. `Low`;
5. `UltraLow`.

Going from highest to lowest the probability to verify person increasing but the probability to falsely accept the wrong person is increasing accordingly.

Besides mentioned names there is capability to use custom threshold names defined in **Phoenix customization file** on the server (see `Phoenix System` documentation).