Smilart Phoenix Journal User Guide

Contents

Overview

`Smilart Phoenix Journal` subscribes for events produced by Phoenix System and stores them its' internal database for a user to view them later.
Currently Journal records events from Person Service, Video Content Analytics Service and Verification Service only. `Smilart Phoenix Journal` requires `Phoenix Erlang API` version `2.4` to work correctly.

Installation

`Smilart Phoenix Journal` product installation is performed by `Smilart Application Manager (SAM)`.

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

To install the `Smilart Phoenix Journal` product `installproduct` command is used which will be available in 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*

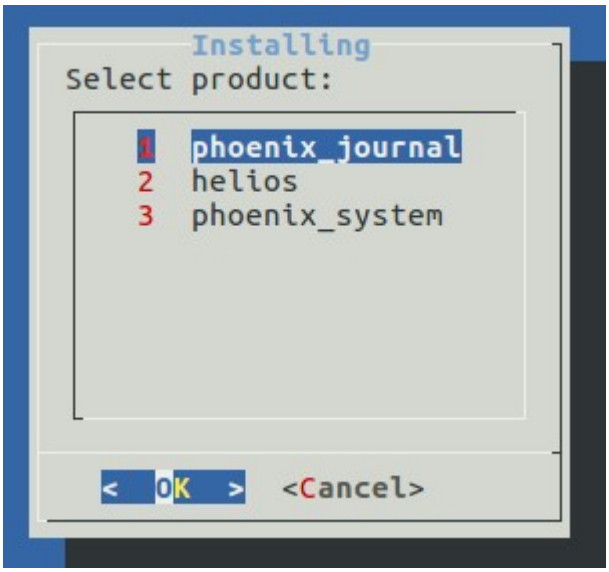
After installation ends image `smilartos-install` should appear in the list of images

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

Then the `installproduct` command becomes available.

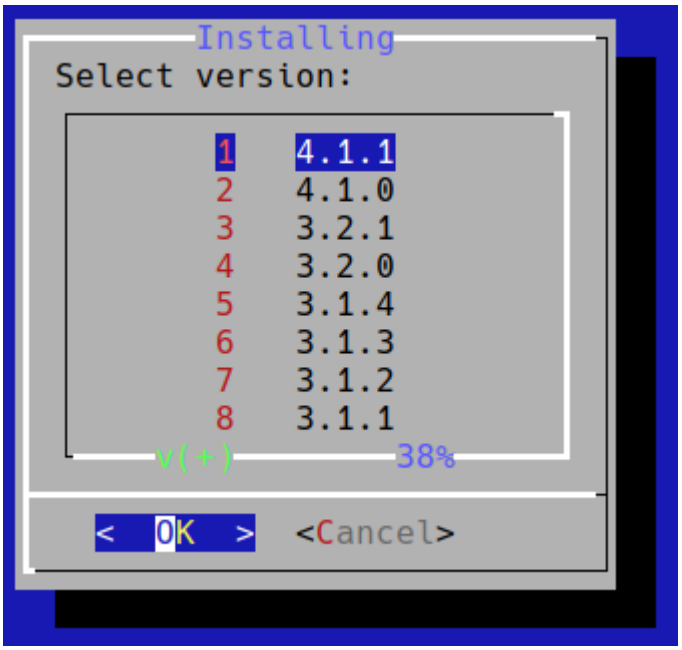
Product Phoenix Journal

After calling `installproduct` command the product selection window appears where `phoenix_journal` must be selected to install `Smilart Phoenix Journal`.



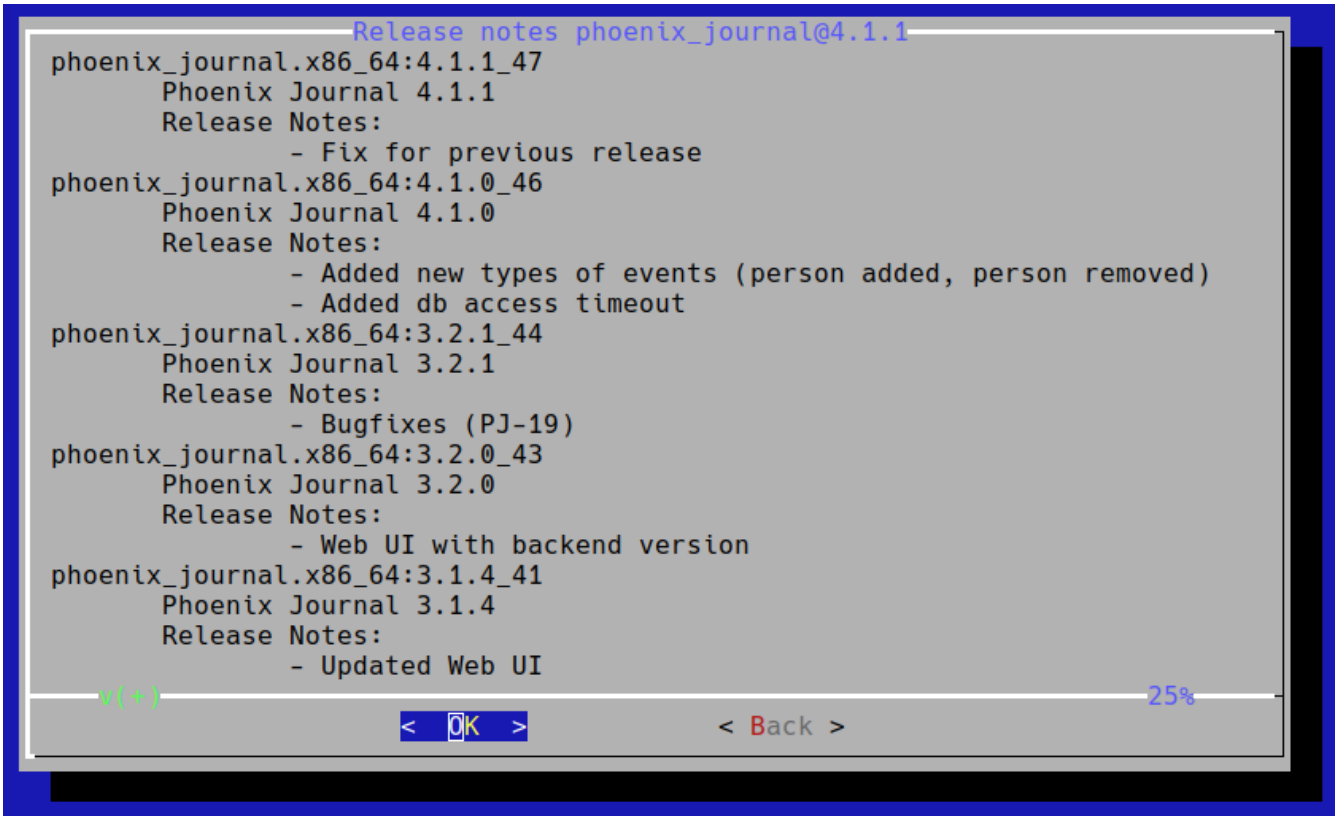
Product selection window

After that preferred version of `Smilart Phoenix Journal` can be selected.



Product version select window

Screen with information about selected version appears.



Product information window

After confirmation system will collect information about the installation, download necessary files and offer to install collected images into the system.

```
Created tmp directory /data/share/phoenix_journal@2.0.0/tmp.iI2KJ5wJ7m.
Check applications availability for installation.


2.0.0_20: Pulling from phoenix_journal.x86_64
045db372f3ec: Already exists
7ce2b960e7a8: Already exists
07fc1462b747: Already exists
1e2c5810f0ab: Already exists
b9e27d958282: Already exists
a3ed95caeb02: Already exists
5975f4160c66: Already exists
7bc0fdb4ce1e: Already exists
005810534f15: Already exists
79e62d2e82bc: Already exists
b3f2da010c12: Already exists
2c3925aa658e: Pull complete
0c03360aed38: Pull complete
48bb9ac94cc7: Pull complete
33707848e54d: Pull complete
8ab9f90d184d: Pull complete
36054e2cf5df: Pull complete
6a56ad8e6bec: Pull complete
Digest: sha256:37725abdcecf378bd9d47f76f003b5c60f88377330d066c06334c2af4dba0f9d
Status: Downloaded newer image for int.docker.smilart.com/phoenix_journal.x86_64:2.0.0_20

Creating tag 'smilart/phoenix_journal.x86_64:2.0.0_20'...

List removed applications:
phoenix_journal.x86_64:development

List added applications:
phoenix_journal.x86_64:2.0.0_20

Continue the import? [y/N] y
```

Install confirmation

After confirmation system will complete the installation of `Smilart Phoenix Journal`.

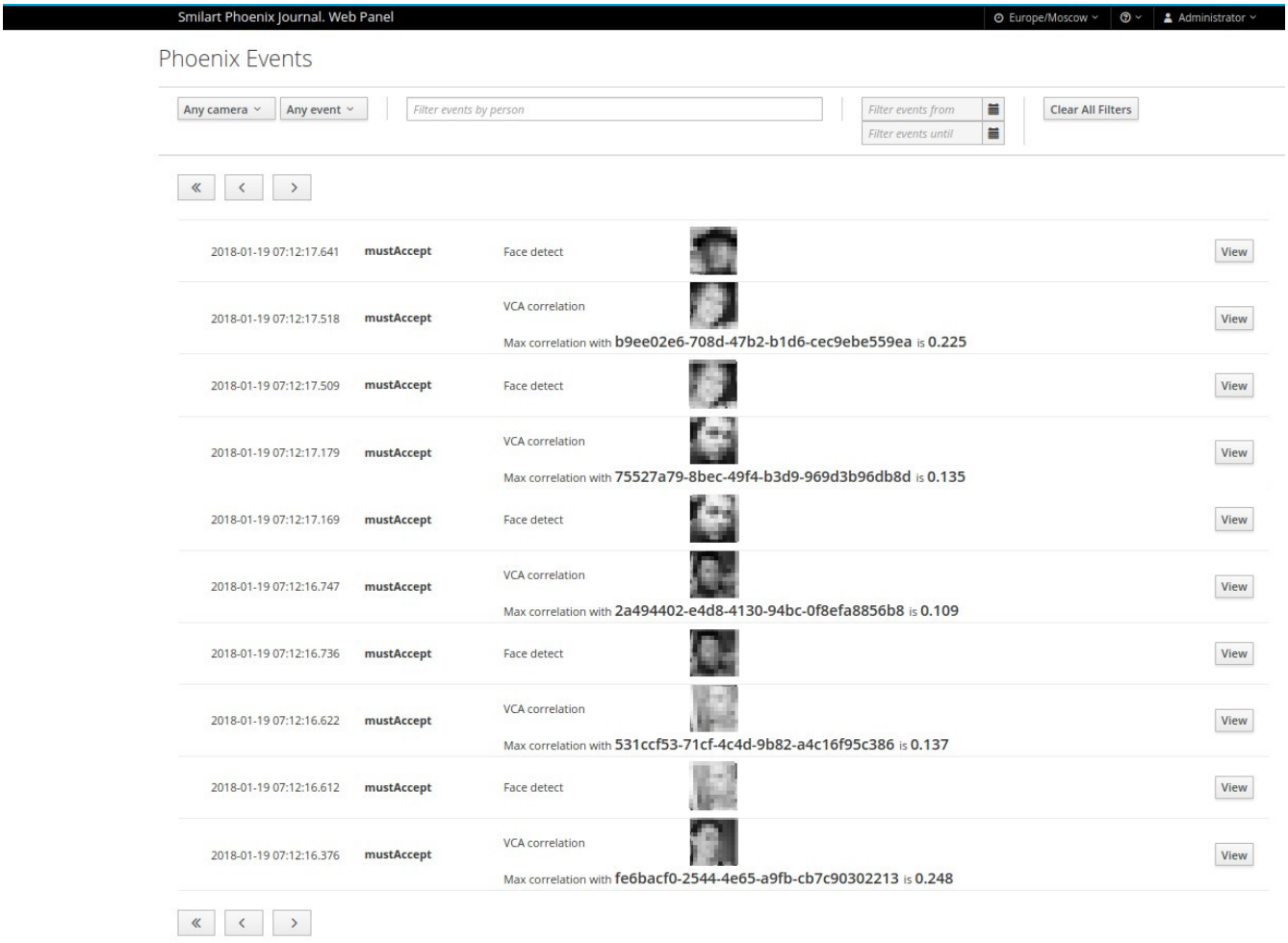
Successful installation can be verified by using

```
$ sam list

      List apps:

Name                                | Version       | Repository
-------------------------------------------------------------------
phoenix_journal.x86_64              | 4.5.1_55      | int
...
```

If any errors occur during the installation contact technical support.

User interface

Smilart Phoenix Journal provides a basic user interface which can be found at http://<hostname>:8084/, where <hostname> is a hostname or ip-address of the server where journal is installed. This interface can be used to find and display events stored in journals' database. It also provides capabilities to filter displayed events by date and time, type of event, camera or person id. All events are sorted from newest to oldest.



User Interface

Configuration

Configuration file for `Smilart Phoenix Journal` is located at `/etc/phoenix_journal/current/sys.config`.

Text after `%` is a comment.

```
[
  {sml_phoenix_journal, [
    {web_panel, [
      %% {authentication, none | {keycloak, #{...}} }
      {authentication,
        none
        % {keycloak, #{
        %   host=> "${keycloak_host}",
        %   role=> "phoenix_journal_auditor",
        %   port=> 8080,
        %   client_id=> "phoenix_journal_web_panel",
        %   realm=> "smilart"
        % }}
      },
      {port, "${WEB_PANEL_PORT}"}
    ]},
    {phoenix_connection, [
      {node, 'sml_phoenix@phoenix'}, %% Erlang node name
      {cookie, 'sml_phoenix'} %% Cookie
    ]},
    {max_db_size_gb, 20},
    {max_retained_correlation_photos, 4},
    {db_partition_backup_timeout_sec, 180},
    {user_request_timeout_ms, 15000},
    {path, "/data/phoenix_journal/journal_db/"},
    {db_access_timeout_sec, 15},
    {save_frames, [
      {verification, [
        {enabled, false},
        {frames_throttle_time_ms, 100}
      ]}
    ]}
  ]},
  {cache, [
    {event_cache, [{n, 10}, {ttl, 60}]},
    {detect_cache, [{n, 10}, {ttl, 300}]}, %ttl in sec
    {read_cache, [{n, 10}, {ttl, 300}]},
    {person_cache, [{n, 10}, {ttl, 1200}]}
  ]},
  {lager, [
    {async_threshold, 100},
    {async_threshold_window, 25},
    {crash_log_size, 104857600},
```

```
      {crash_log_date, ""},
      {crash_log_count, 4},
      {handlers, [
        %{lager_file_backend, [{file, "log/debug.log"}, {level, debug}, {size,
104857600}, {date, ""}, {count, 4}]},
        {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}]}
      ]}
    ]},
    {sasl, [
      {sasl_error_logger, {file, "log/sasl_errors.log"}},
      {errlog_type, error}
    ]}
  ].
```

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.

Web panel settings

The section `web_panel` contains Web Panel settings.

Name	Type	Description
authentication	`none` or `{keycloak, #{…}}`	Describes what authentication method will be used by Web Panel to access Phoenix Journal.
port	string	On which port Web Panel will be available.

Web panel authentication settings

See Secure User Interface with Keycloak

Phoenix Journal Db settings

This section contains settings for Phoenix Journal Database.

Name	Type	Description

max_db_size_gb	integer	Maximum size of journal database in gigabytes. Minimum size allowed is 3 gb.
max_retained_correlation_photos	integer	Maximum number of top correlated photos per event to store in Journal database. You will reduce storage size by decreasing this parameter.
db_partition_backup_timeout_sec	integer	Timeout in seconds to backup one partition of Journal database. You should increase this timeout if the copy speed on the hard drive is slow.
user_request_timeout_ms	integer	Timeout in milliseconds to obtain data for UI from Journal database. You should increase this timeout if querying the database takes a long time.
db_access_timeout_sec	integer	Timeout in seconds to access Journal database. If your hdd writing speed is less than 1 GB/s then double the timeout.
path	string	Where in filesystem Journal database files will be stored.

Any settings not described above should be changed only by developers.

Phoenix Events settings

This section contains settings for some external Phoenix Events.

Name	Type	Description
save_frames	object	Settings to save frames from cameras to Journal database.
save_frames.verification	object	Settings to save camera frames during verification process to Journal database.
save_frames.verification.enabled	boolean	Setting whether to save frames camera frames during verification process to Journal database. Default is `false`.
save_frames.verification.frames_throttle_time_ms	integer	Minimum time interval in milliseconds between saving camera frame events to Journal database. Default is `100`.

`save_frames.verification.enabled` can be `true` only for the period of debugging the quality of the system.

Low value of `save_frames.verification.frames_throttle_time_ms` can affects system performance and accelerates events rotation in Journal database.

User interface settings

By default there is no authorization for **User interface**, but it is possible to add some.

1. By using **Nginx** basic authorization.
2. By using **Keycloak** (https://www.keycloak.org/).

Secure User interface with Nginx basic access authentication

Install and setup **Nginx** (https://smilart.atlassian.net/wiki/spaces/PS/pages/173670407/Install+nginx+for+basic+access+authentication).

For instance we want to have access to **User Interface** by http://nginx-address/journal/. And assume that **Nginx** can get **User Interface** by http://192.168.1.39:8084

To achieve that you must modify /etc/nginx/nginx.conf by adding `location` rules to your `http server` which is on port 80.

```
#...
http {
#...
server {
    listen       80;
#...

# Next rule is to get User Interface resources files
location /journal/ {
    proxy_pass http://192.168.1.39:8084/; # note the trailing slash!
}

# Next is used for internal Journal API
location /journal/api/csi {
    proxy_pass http://192.168.1.39:8084/api/csi;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_read_timeout 2073600; #maximum supported is 24 days
}

#...
}
}
#...
```

To Apply changes in the config file **Nginx** must be restarted by `systemctl restart nginx`.

After that follow next steps to close **Journal User Interface** port:

1) Create file `/var/lib/iptables/rules-save` and edit like in example below.

```
*filter
-A INPUT -p tcp -m tcp --dport 8084 -j DROP
COMMIT
```

2) Enable iptables-restore `systemctl enable iptables-restore`.

3) Reboot.

Secure User Interface with Keycloak

Install and setup **Keycloak** (https://smilart.atlassian.net/wiki/spaces/PS/pages/164102281/Install+keycloak).

Change **Journal** config parameter `web_panel` (/etc/phoenix_journal/current/sys.config.orig)

```
% - makes comment until end of line
% ...
    {web_panel, [
      %% {authentication, none | {keycloak, [ ...]} }
      {authentication,
        {keycloak, [
          {host, "${keycloak_host}"},
          {port, 8080},
          {role, "phoenix_journal_auditor"},
          {client_id, "phoenix_journal_web_panel"},
          {realm, "smilart"}
        ]}
      },
      {port, "${WEB_PANEL_PORT}"}
    ]},
% ...
```

Properties

Name	Type	Description
web_panel	object	Settings for **User Interface**.
web_panel.port	integer	**User Interface** port . By default will be applied from env. `${WEB_PANEL_PORT}`, which is 8084 by default.
web_panel.authentication	object	Can be `none` or `keycloak`.
web_panel.authentication.keycloak.host	string	Host name of **Keycloak**. By default will be applied from env. `${keycloak_host}`, which is hostname of station with the **Journal**.
web_panel.authentication.keycloak.port	integer	Port of **Keycloak**. By default `8080`.

web_panel.authentication.keycloak.role	string	Required role for **Keycloak** user to access **Admin Panel**. By default `phoenix_journal_auditor`.
web_panel.authentication.keycloak.client_id	string	Name of **Keycloak** 'Client' for **Admin Panel**. By default `phoenix_journal_web_panel`.
web_panel.authentication.keycloak.realm	string	**Keycloak** realm to use. By default `smilart`.

If you change **User Interface** port you also need to change this port forwarding in systemd unit file (/etc/systemd/system/phoenix_journal.service).

Host name and port of **Keycloak** must be reachable from `phoenix_journal` docker container and from client side of **User Interface**.

To Apply changes in the config **Journal** must be restarted by `systemctl restart phoenix_journal`.

Backup and Restore

Backuping journal database can be done by invoking `sml_phoenix_journal_backup` from `/opt/bin`

```
/opt/bin/sml_phoenix_journal_backup --path <path>
```

Here `<path>` is the path to where database files will be dumped. This path must be attached as a volume to Phoenix Journal docker container.

Restoring journal database can be done by invoking `sml_phoenix_journal_restore` from `/opt/bin`

```
/opt/bin/sml_phoenix_journal_restore --path <path>
```

Here `<path>` is the path from where database files will be copied. This path must be attached as a volume to Phoenix Journal docker container.

The path `/data/share` is attached to Phoenix Journal container by default and can be used for backup and restore purposes.

Tips

To drop database stop journal, delete database files in `/data/phoenix_journal` and start journal back. It will create clean database on startup.

To change database size stop journal, change database size in sys.config and drop database before starting journal.