univention
be open.

UCS Intercom Service app

Release 2.25.0

Univention GmbH

Oct 16, 2026

The source of this document is licensed under GNU Affero General Public License v3.0 only.

CONTENTS

1 Installation 3

2 Configuration 7

3 Architecture 15

4 Requirements and limitations 21

5 Troubleshooting 23

6 Changelog 25

Bibliography 33

Index 35

Welcome to the documentation about the Univention app **UCS Intercom Service**. The app installs the ICS (UCS Intercom Service), an intermediary for communication between applications like *Nextcloud*, *OX App Suite*, *XWiki* and *Matrix*. The functionalities *File-picker*, *Video conference*, create and accessing the *Univention-Portal navigation* endpoint from other apps and *Univention-Portal* news feed require the app **UCS Intercom Service**.

This documentation is for system administrators who operate the app **UCS Intercom Service** from Univention App Center connected to the LDAP directory in UCS (Univention Corporate Server). It covers the following topics:

1. *Architecture* (page 15)
2. *Installation* (page 3)
3. *Configuration* (page 7)
4. *Requirements and limitations* (page 21)
5. *Troubleshooting* (page 23)

This documentation doesn't cover the following topics:

- Usage of UCS, see *UCS 5.0 Manual* [1].

To understand this documentation, you need to know the following concepts and tasks:

- Use and navigate in a remote shell on Debian GNU/Linux derivative Linux distributions like UCS. For more information, see Shell and Basic Commands[1] from *The Debian Administrator's Handbook*, Hertzog and Mas [2].
- Manage an app through Univention App Center[2] in *UCS 5.0 Manual* [1].

The app **UCS Intercom Service** supports *Keycloak* as *IdP*. In the context of this document, the term *IdP* is synonymous for *Keycloak* in the context of an *IdP* in OIDC (OpenID Connect).

Your feedback is welcome and highly appreciated. If you have comments, suggestions, or criticism, please send your feedback[3] for document improvement.

[1] https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands
[2] https://docs.software-univention.de/manual/5.0/en/software/further-software.html#computers-softwareselection
[3] https://www.univention.com/feedback/?intercom-service-app=generic

CHAPTER

ONE

INSTALLATION

You can install the app **UCS Intercom Service** like any other app with Univention App Center.

UCS offers two different ways for app installation:

- With the web browser in the UCS management system
- With the command-line

For general information about Univention App Center and how to use it for software installation, see Univention App Center[4] in *UCS 5.0 Manual* [1].

1.1 Prerequisites

Installing this app has various prerequisites:

1. ICS supports OIDC. *Nextcloud (>=23.0)* and *OX App Suite (>=7.10.6)* must authenticate through OIDC, as well.
2. For working CSRF (Cross-Site-Request-Forgery) protection all other apps need to be up to date.
3. ICS requires the *Nordeck* bot up and running for Matrix.
4. ICS requires its three secrets before installation, see *Secrets* (page 7) for details.
5. ICS requires a configured *Keycloak* (>=12.0), including a valid Intercom OIDC client before installation.

1.2 Add ICS client to IdP

To prepare the existing *IdP* for the installation of the app **UCS Intercom Service**, use the following steps:

1. Enter the *Keycloak Admin Console*.
2. Go to your realm where you want to create the ICS client.
3. Create a OIDC Client. Recommendation is to use the default value `intercom` for the *Client ID* and leave the *Root URL* empty and save it.

   During app installation, **UCS Intercom Service** asks for the *Client ID*.
4. Set *Client authentication* to `On`.
5. Set *Authorization*, *Service Accounts Enabled*, and *Standard Token Exchange* to `On`.

> **Note**
>
> *Standard Token Exchange* corresponds to Keycloak Token Exchange v2 (RFC 8693). It is enabled by default in Keycloak 26 and later and is the recommended implementation. To use it, also set *intercom-service/settings/token-exchange-v2* (page 13) to `True`.

[4] https://docs.software-univention.de/manual/5.0/en/software/app-center.html#software-appcenter

Keycloak Token Exchange v1 (legacy) is deprecated and disabled by default in Keycloak 26+. If you are running an older Keycloak version that does not support v2, leave *Standard Token Exchange* off and keep *intercom-service/settings/token-exchange-v2* (page 13) at its default value of `False`.

When using Token Exchange v2, verify that the *Client Scopes* of the `intercom` client include the audiences for each enabled backend service (for example `xwikioidc` and `ncoidc`). Keycloak Token Exchange v2 resolves audiences through client scopes rather than granting them directly based on the target client.

6. Set *Front channel logout* to `Off`.
7. Set *Backchannel logout URL* to your intended domain for ICS with protocol and append the `backchannel-logout` path, for example:

```
https://ics.example-domain.example-tld/backchannel-logout
```

   This step requires *Keycloak* >=12.0.0.

8. Set *Backchannel logout session required* to `On`.
9. Set the valid redirect URL to your intended ICS domain with protocol and append the `/callback` path, for example:

```
https://ics.example-domain.example-tld/callback
```

10. Go to the tab *Credentials*, copy the secret and save it to `/etc/intercom-client.secret`.
11. Go to the tab *Client Scopes* and add change `offline_access` to *Default*.
12. Make sure the *Access Token* includes a mapper for both the username and the user unique identifier. The documentation for these claims can be found in the *UCR* variables *intercom-service/settings/username-claim* (page 8) and *intercom-service/settings/user-unique-mapper* (page 8).

1.3 Installation with the web browser

To install ICS from the UCS management system, use the following steps:

1. Use a web browser and sign in to the UCS management system.
2. Open the *App Center*.
3. Select or search for *Intercom Service* and open the app with a click.
4. To install *Intercom Service*, click *Install*.
5. Leave the *App settings* in their defaults or adjust them to your preferences. For a reference, see *Settings* (page 8).
6. To start the installation, click *Start Installation*.

Note

To install apps, the user account you choose for login to the UCS management system must have domain administration rights, for example the username `Administrator`. User accounts with domain administration rights belong to the user group `Domain Admins`.

For more information, see Delegated administration for UMC modules[5] in *UCS 5.0 Manual* [1].

[5] https://docs.software-univention.de/manual/5.0/en/central-management-umc/delegated-administration.html#delegated-administration

1.4 Installation with command-line

To install the app **UCS Intercom Service** from the command-line, use the following steps:

1. Sign in to a terminal or remote shell with a username with administration rights, for example `root`.
2. Choose between default and custom settings and run the appropriate installation command.

   For installation with default settings, run:

   ```
   $ univention-app install intercom-service
   ```

   To pass customized settings to the app during installation, run the following command:

   ```
   $ univention-app install --set $SETTING_KEY=$SETTING_VALUE intercom-service
   ```

   Caution

   Some settings don't allow changes after installation. To overwrite their default values, set them before the installation. For a reference, see *Settings* (page 8).

   Example: To define a different Keycloak-realm in ICS, run:

   ```
   $ univention-app install intercom-service \
     --set intercom-service/keycloak/realm-name=master
   ```

CHAPTER

TWO

CONFIGURATION

The app **UCS Intercom Service** offers various configuration options. Some settings don't allow changes after installation. Therefore, you must set them **before** installation. You find those settings marked with *Only before installation* in *Settings* (page 8). You can change all other settings at any time after the installation.

To change settings after installation, sign in to the UCS management system with a username with administration rights and go to *App Center* ‣ *UCS Intercom Service* ‣ *Manage Installation* ‣ *App Settings*. On the appearing *Configure UCS Intercom Service* page, you can change the settings and apply them to the app by clicking *Apply Changes*.

The App Center then *re-initializes* the Docker container for the app **UCS Intercom Service**. *Reinitialize* means the App Center throws away the running ICS Docker container and creates a fresh ICS Docker container with the just changed settings.

2.1 Intercom Service

The app **UCS Intercom Service** provides the backend for inter-app communication of *Nextcloud*, the *UCS Portal*, *Matrix* through the *Nordeck* bot, *OX App Suite* and *XWiki*.

Warning

This app doesn't configure any *Keycloak* settings. It requires an existing client and realm setup in *Keycloak*.

2.2 Secrets

The app **UCS Intercom Service** requires secrets, that aren't automatically generated. Those secrets are:

/etc/intercom-client.secret
: The client secret for authenticating with the *IdP*. You can retrieve the client secret from the *Keycloak Admin Console* in the *Authorization* tab of the *intercom-client*.

/etc/matrix.secret
: The secret for backend communication with the *Matrix* server. You can retrieve it from the automatic join app service on the system running *Matrix*.

  The following command shows how to retrieve the secret for the backend communication with the *Matrix* server:

```
$ kubectl exec --stdin --tty synapse-0 -n matrix-000-prod -- \
    /bin/bash -c "cat /data/autojoin-appservice.yaml \
    | grep as_token \
    | sed -e 's/as_token. \(.\+\)/\1/'"
```

/etc/intercom-portal.secret
: The secret to communicate with the UCS Portal navigation service. You can retrieve the secret from `/etc/portal-navigation-service.secret`.

/etc/intercom-redis.secret
: The secret to communicate with the app *Redis*, that stores the sessions. It is only needed if you use an external *Redis* server. Otherwise, it will be generated during installation.

2.3 Settings

The following references show the available settings within the app **UCS Intercom Service**. Univention recommends to keep the default values.

intercom-service/settings/client-id
: Defines the OIDC client name of ICS in *Keycloak*. The file `/etc/ics_client.secret` stores the secret of this client.

Required	Default value	Set
Yes	`intercom`	Only before installation

intercom-service/settings/user-unique-mapper
: Defines the *Keycloak* ICS client token claim name mapper configured. The field must uniquely identify the user across *Matrix*, *Nextcloud*, *OX App Suite*, and the *Nubus*. Only the value `entryuuid` is guaranteed to be unique in Nubus.

  ICS reads the claim from the OIDC ID token, both at login and when it renews the *Matrix* token of the user. The mapper of the ICS client in *Keycloak* must add the claim to the ID token.

Required	Default value	Set
Yes	`entryuuid`	Only before installation

intercom-service/settings/username-claim
: Defines the *Keycloak* ICS client token claim name mapper configured. The field must contain the username of the user.

Required	Default value	Set
Yes	`phoenixusername`	Only before installation

intercom-service/settings/enable-session-cookie
: Enable session cookie (transient cookie). If enabled, the session cookie will be deleted at the end of the browser session. Otherwise, the session will be a rolling session (reset every time the user is active).

Required	Default value	Set
No	`False`	Only before installation

intercom-service/settings/session-rolling-duration
: Rolling session duration in seconds. The session will be reset if the user is active within the duration. Otherwise, the user will be logged out, requiring a silent login. If *intercom-service/settings/enable-session-cookie* (page 8) is set to true, this setting will be ignored.

Required	Default value	Set
No	`86400`	Only before installation

intercom-service/settings/intercom-url

Defines the URL where you can reach ICS. This needs to be a externally reachable address as it's used by the browser to connect to ICS.

Required	Default value	Set
Yes	`https://ics.@%@domainname@%@`	Only before installation

intercom-service/settings/base-url

Defines the base URL used to identify with the *IdP*. This URL must match the base URL defined in the OIDC client used on the *IdP*. The value should be the same as in *intercom-service/settings/intercom-url* (page 8).

Required	Default value	Set
Yes	`https://ics.@%@domainname@%@`	Only before installation

intercom-service/keycloak/url

URL of the *Keycloak* instance that ICS uses as *IdP*. ICS ignores this value, if *intercom-service/settings/issuer-base-url* (page 9) is defined.

Required	Default value	Set
Yes	`https://id.@%@domainname@%@`	Only before installation

intercom-service/keycloak/realm-name

Name of the realm containing the configured OIDC ICS client. ICS ignore this value, if *intercom-service/settings/issuer-base-url* (page 9) is defined.

Required	Default value	Set
Yes	`UCS`	Only before installation

intercom-service/settings/issuer-base-url

Defines a full base URL for the OIDC token issuer. Usually, the *IdP Keycloak* issues OIDC tokens.

This variable overwrites *intercom-service/keycloak/url* (page 9) and *intercom-service/keycloak/realm-name* (page 9).

Only set this variable, if you really need to change the default URL generated from the before mentioned variables.

Required	Default value	Set
No	`None`	Only before installation

intercom-service/settings/origin-regex

Defines the origin *CORS* regular expression. Normally this will be the shared domain name. Changing this value may have security implications.

Required	Default value	Set
Yes	`@%@domainname@%@`	Only before installation

intercom-service/settings/log-level

Logging level for the standard output, as well as, log file at `intercom-service.log`.

Required	Default value	Set
Yes	`info`	Only before installation

intercom-service/settings/proxy

This setting is passed to *node-axios* within the container. It allows or disallows connections through a proxy server between ICS and apps like *Matrix*, *Nextcloud*, or *OX App Suite*, instead of a direct connection to the backends.

Required	Default value	Set
Yes	`False`	Before installation or application settings afterwards.

intercom-service/redis/host

Defines the host name of the *Redis* server. By default ICS uses the bundled *Redis* server.

Required	Default value	Set
Yes	`redis-intercom`	Before installation or application settings afterwards.

intercom-service/redis/port

Port where the *Redis* server is available.

Required	Default value	Set
Yes	`6379`	Before installation or application settings afterwards.

intercom-service/redis/user

The user name for the *Redis* server. By default ICS uses the *default* user.

Required	Default value	Set
Yes	`default`	Before installation or application settings afterwards.

intercom-service/redis/ssl

Activates the SSL connection to the *Redis* server. Set to `False` by default.

Required	Default value	Set
Yes	`False`	Before installation or application settings afterwards.

intercom-service/redis/mtls

Activates the mTLS authentication for the connection to the *Redis* server. Set to `False` by default.

Required	Default value	Set
Yes	`False`	Before installation or application settings afterwards.

intercom-service/redis/certificates/mtls/cert

Defines the host path to the certificate to identify the ICS client during the mTLS authentication to the *Redis* server. The file must be in PEM (Privacy Enhanced Mail) format.

Required	Default value	Set
No	`Empty`	Only before installation.

intercom-service/redis/certificates/mtls/key

Defines the host path to the certificate private key for the mTLS authentication of the ICS client to the *Redis* server. The file must be in PEM format.

Required	Default value	Set
No	`Empty`	Only before installation.

intercom-service/matrix/url

Defines the URL, where you can reach the *Matrix* server. The file `/etc/ics_matrix_as.secret` stores the Matrix secret.

Required	Default value	Set
Yes	`https://matrix.@%@domainname@%@`	Only before installation

intercom-service/matrix/enabled

Defines if the *Matrix* proxy functionality is enabled. Set to `False` to disable *Matrix*.

Required	Default value	Set
Yes	`True`	Only before installation

intercom-service/matrix/server-name

Defines the server name of the *Matrix* server, that is a unique identifier configured in *Matrix*. The server name must match the configured server name in *Matrix*.

It isn't necessarily the server name defined in *intercom-service/matrix/url* (page 11).

Required	Default value	Set
Yes	`matrix.@%@domainname@%@`	Only before installation

intercom-service/matrix/login-type

Defines the login type that ICS uses for the *Matrix* server.

Refer to the Matrix[6] documentation for more information about login types.

Required	Default value	Set
Yes	`uk.half-shot.msc2778.login.application_service`	Only before installation

intercom-service/matrix/nordeck-mode

Defines the connection mode of the *Nordeck* bot.

Possible values: `test`, `live`, `test proxies`.

Required	Default value	Set
Yes	`test`	Only before installation

intercom-service/matrix/nordeck-url

Defines the URL, where you can reach the *Nordeck* bot.

Required	Default value	Set
Yes	`https://meetings-widget-bot.@%@domainname@%@`	Only before installation

intercom-service/portal/portal-url

Defines the URL for the UCS portal. The file `/etc/ics_portal.secret` stores the Portal API key.

Required	Default value	Set
Yes	`@%@ucs/server/sso/fqdn@%@`	Only before installation

intercom-service/xwiki/url

Defines the URL where you can reach *XWiki*. Set to empty to disable *XWiki*.

Required	Default value	Set
Yes	`https://xwiki.@%@domainname@%@`	Only before installation

intercom-service/xwiki/enabled

Defines the *XWiki* proxy functionality is enabled. Set to `False` to disable *XWiki*.

Required	Default value	Set
Yes	`True`	Only before installation

intercom-service/xwiki/audience

Defines the *OIDC audience* setting for *XWiki* that *XWiki* uses in the *IdP Keycloak*.

Required	Default value	Set
Yes	`xwiki`	Only before installation

intercom-service/nextcloud/audience

Defines the *OIDC audience* setting for *Nextcloud* that *Nextcloud* uses in the *IdP Keycloak*.

Required	Default value	Set
Yes	`ncoidc`	Only before installation

[6] https://matrix.org/docs/

intercom-service/nextcloud/url

Defines the URL where you can reach *Nextcloud*. Set to empty to disable *Nextcloud*.

Required	Default value	Set
Yes	`https://fs.@%@domainname@%@`	Only before installation

intercom-service/nextcloud/enabled

Defines the *Nextcloud* proxy functionality is enabled. Set to `False` to disable *Nextcloud*.

Required	Default value	Set
Yes	`True`	Only before installation

intercom-service/nextcloud/origin

Defines the *Nextcloud CORS* setting. Usually this value is the same as *intercom-service/nextcloud/url* (page 12).

Required	Default value	Set
Yes	`https://fs.@%@domainname@%@`	Only before installation

intercom-service/certificates/external/root-ca-pem

Defines the host path to self-signed external certificates, allowing for secure communication with *Nextcloud*, *Matrix*, *Redis* and *OX App Suite*. The file must be in PEM format, allowing for multiple certificates in one file.

Required	Default value	Set
No	`Empty`	Only before installation

intercom-service/certificates/external/root-ca-crt

Defines the host path to self-signed external certificates, allowing for secure communication with *Nextcloud*, *Matrix* and *OX App Suite*. The file must be in CRT (Certificate) format.

Required	Default value	Set
No	`Empty`	Only before installation

intercom-service/settings/token-exchange-v2

Enables Keycloak Token Exchange v2 (standard, RFC 8693) instead of the legacy Token Exchange v1.

Token Exchange v2 is enabled by default in *Keycloak* 26 and later and is the recommended implementation going forward. Token Exchange v1 is deprecated and disabled by default in *Keycloak* 26+.

To use Token Exchange v2, the following additional *Keycloak* configuration is required on the `intercom` client:

- Enable the *Standard Token Exchange* toggle in the *Keycloak Admin Console* under the client's *Settings* tab.
- Ensure the *Client Scopes* of the `intercom` client include the audiences for each enabled backend service, for example `xwikioidc` for *XWiki* (see *intercom-service/xwiki/audience* (page 12)) and `ncoidc` for *Nextcloud* (see *intercom-service/nextcloud/audience* (page 12)). Token Exchange v2 resolves audiences through client scopes rather than granting them directly based on the target client.

See *Installation* (page 3) for the full client setup procedure.

Required	Default value	Set
No	`False`	Any time

CHAPTER

THREE

ARCHITECTURE

The **UCS Intercom Service** app architecture consists of the following elements:

- The operating environment UCS with the App Center and the Docker engine running the ICS container.
- The ICS software based on *Node.js®* running in the ICS container.
- The *Keycloak Identity Access Management*, used by ICS to authenticate sessions and retrieve login tokens for applications.
- A *Redis* container to store OIDC sessions.

Note

The following sections may refer to the *browser* rather than to the *client* to avoid confusion with OIDC clients configured in *Keycloak*.

Back end communication as referred to in the following section isn't related to the *Backchannel Logout* path, which is a specific part of the OIDC protocol.

3.1 Design decisions

The app **UCS Intercom Service** aims to provide a way to facilitate *CORS* conform communication to different backends directly from the browser. It can proxy, modify, and authenticate requests and use *Keycloak* as *IdP* and its own session storage to hold OIDC sessions. It acquires those sessions through a silent background login, provided a valid OIDC cookie is already available in the browser.

3.2 Overview

Starting from the basics, ignoring everything related to login, authentication, and sessions for now, this is how ICS works on a basic level.

1. The browser opens the intended app.
2. The app contains ICS related JavaScript code as part of its responses.
3. This code instructs the browser to send a requests to ICS, after the browser needs communication to a separate app.
4. ICS acts as a middleware to modify and forward those requests appropriately to the relevant, second app using a backend communication channel.
5. ICS receives the response and finally sends an appropriately modified response to the browser.

For a visual representation, refer to Fig. 3.1.

The following list describes how this fits into the wider OIDC authentication scheme. See also Fig. 3.2.

1. The browser starts unauthenticated at the login endpoint of an app that supports ICS, for example *Matrix*, *Nextcloud*, *XWiki* or *OX App Suite*.

2. The browser follows the OIDC login procedure. The app redirects the browser to the *IdP Keycloak* and upon successful login assigns an OIDC session for the app to the browser.
3. A silent login happens in the background, that uses the information stored in the browser to authenticate ICS with the *IdP Keycloak* through a hidden IFrame[7].
4. The browser requests an action, for example to create a video conference, as part of a calendar entry. The browser requests an interaction from *OX App Suite* to *Matrix*. In detail, the browser requests the *Nordeck* bot that runs in the Matrix user frontend *Element*.
5. The functional interaction begins as displayed in Fig. 3.1.
6. ICS sends a requests to the backend, usually another app on UCS. ICS acts as a middleware between the browser and backends, for example apps.

Note

ICS may use shared secrets rather than relying on OIDC authentication when communicating with backends.

Warning

Back end communication is only safe, if done through HTTPS or a secure network. Otherwise, attackers may eavesdrop secrets on application layer.

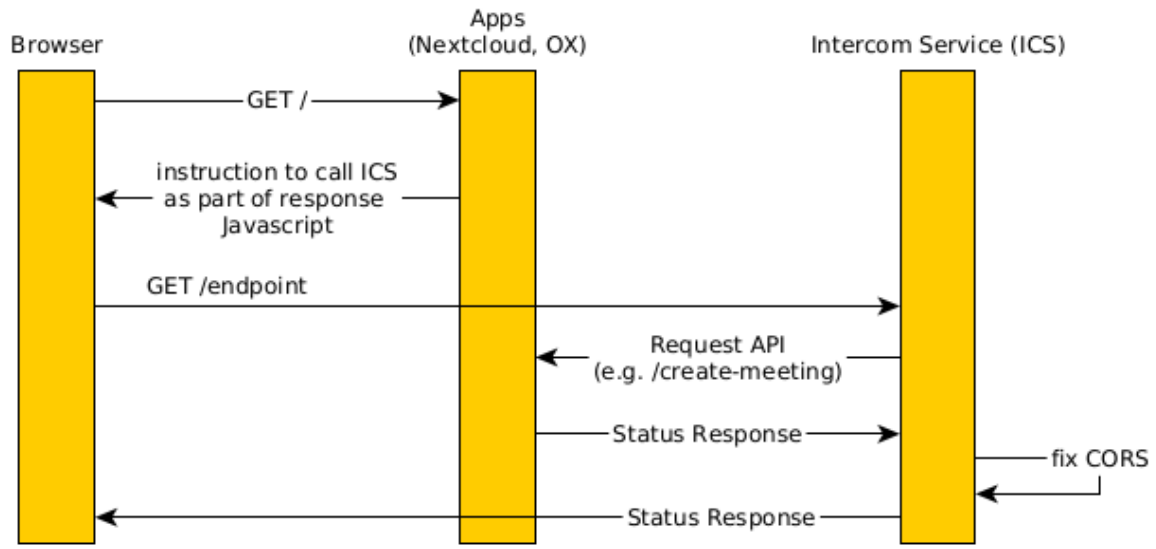

Fig. 3.1: Interactions of ICS without OIDC

[7] https://en.wikipedia.org/wiki/HTML_element#Frames

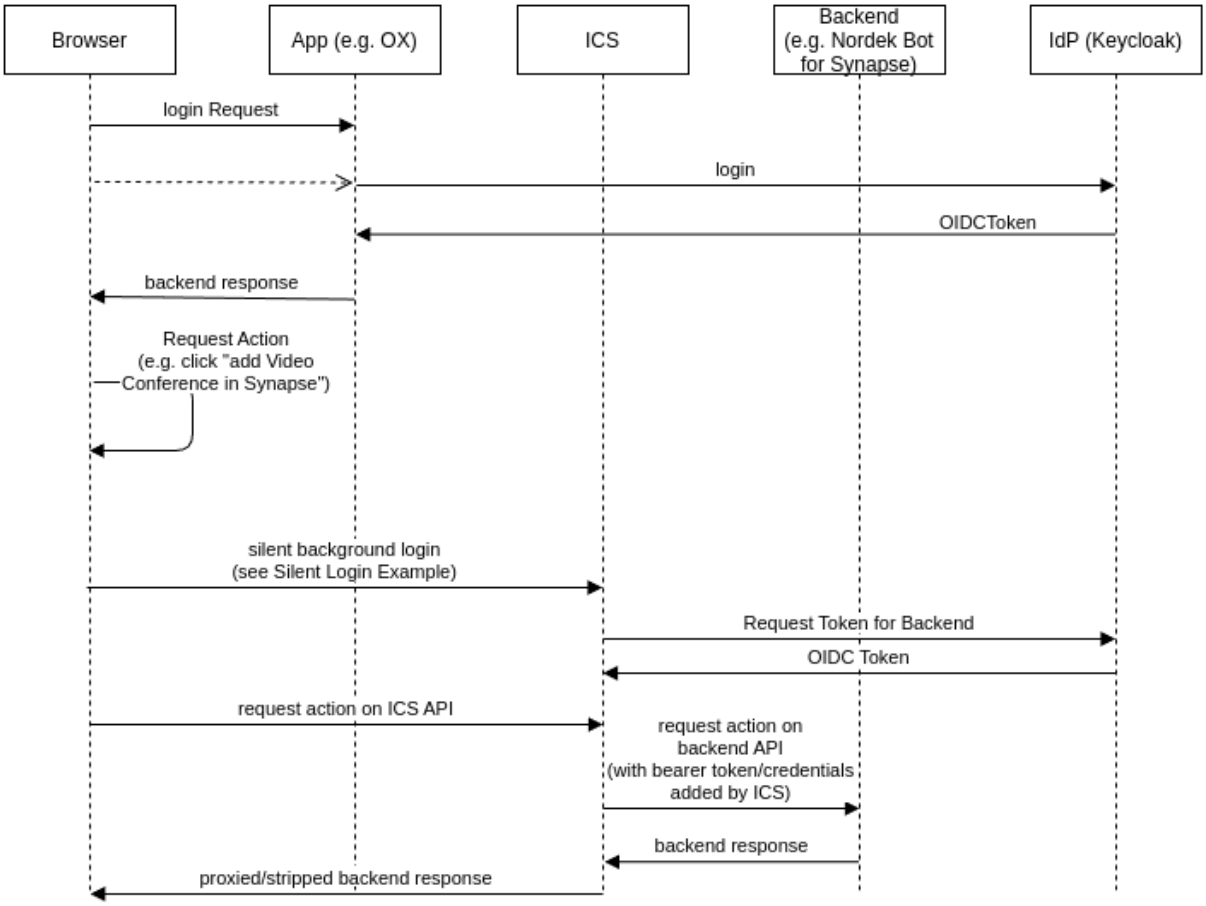

Fig. 3.2: Interactions of ICS with OIDC, *OX App Suite* and *Nordeck*

3.3 Portal navigation

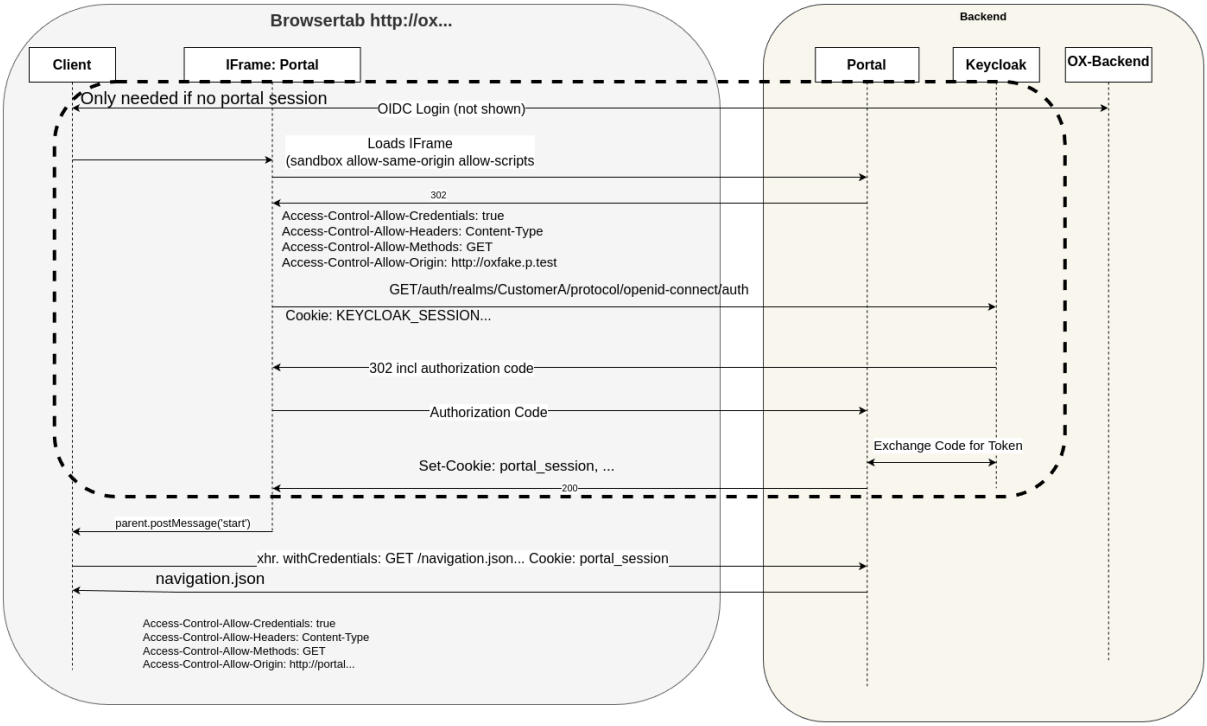

Fig. 3.3: Communication overview for the `Central Navigation` capability, which requires cross app communication between *OX App Suite* and the *UCS Portal*.

3.4 Filepicker

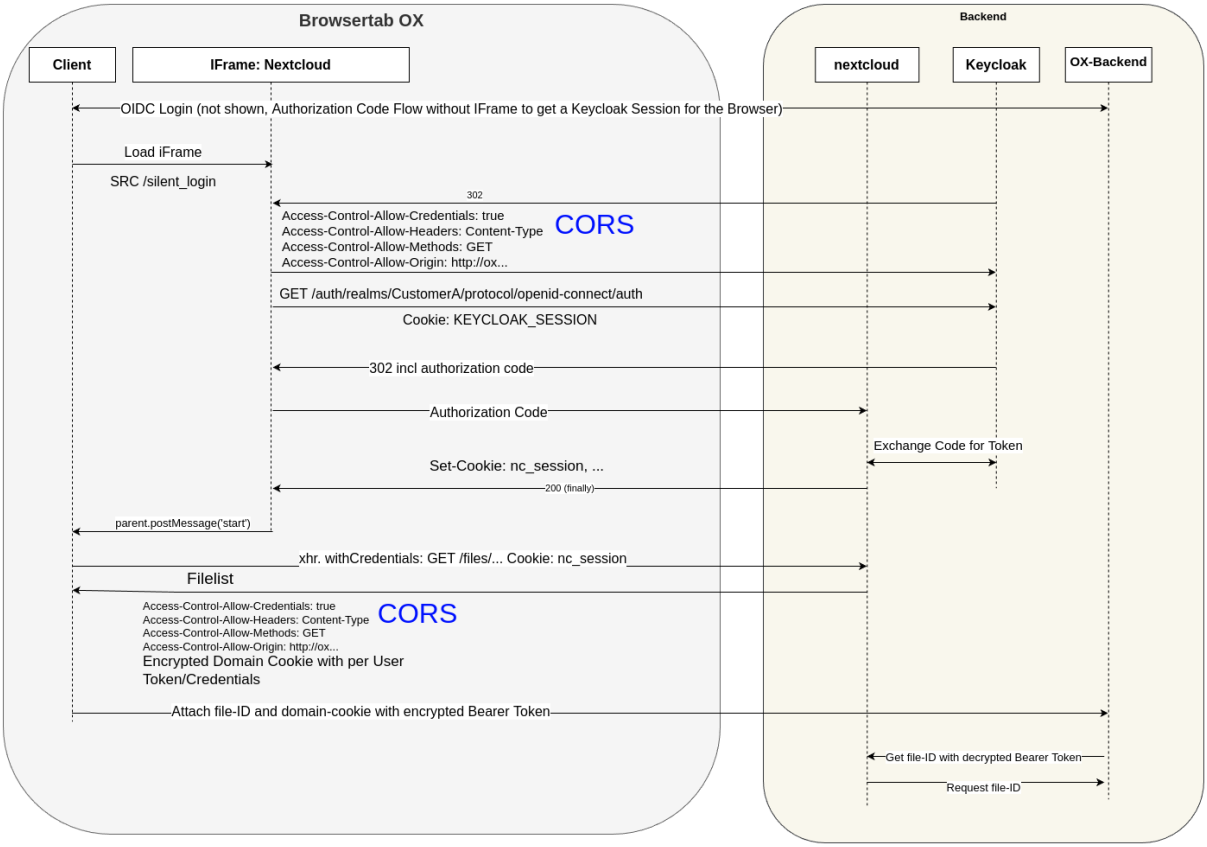

Fig. 3.4: Communication overview for the `Filepicker` capability, which requires cross-app communication between *OX App Suite* and *Nextcloud*.

3.5 Endpoints

The app **UCS Intercom Service** offers the API endpoints listed below.

3.5.1 General

/
: Alive test only

/silent
: Silent OIDC login endpoint

/backchannel-logout
: Endpoint for OIDC back channel logout requests

3.5.2 App specific

/fs
: Proxy for *Nextcloud*

/navigation.json
: Proxy to Univention-portal for central navigation data

/nob
: Proxy for the *Nordeck* bot. In a testing environment, developers can use this endpoint to requests to the plain *Matrix* `UserInfo` service.

3.6 Terms

The document uses the terms that may not be clear to the reader. The following list provides context and explanation.

CORS
: CORS stands for *Cross-Origin Resource Sharing* and is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.

  For more information about CORS (Cross-Origin Resource Sharing), refer to Wikipedia: Cross-origin resource sharing[8].

IdP
: stands for *Identity Provider*. An IdP offers user authentication as service. In the context of the app **UCS Intercom Service** the software *Keycloak* offers the IdP service to ICS and its app backends.

OIDC audience
: The OIDC audience is a required claim within the ID Token for all OAuth 2.0 flows used by OIDC. According to the specification, it must contain the OAuth 2.0 `client_id` of the relying party as audience value.

  For more information, see section ID Token[9] in *OpenID Connect Core 1.0 incorporating errata set 1* [3].

[8] https://en.wikipedia.org/wiki/Cross-origin_resource_sharing
[9] https://openid.net/specs/openid-connect-core-1_0.html#IDToken

CHAPTER

FOUR

REQUIREMENTS AND LIMITATIONS

To ensure a smooth operation of the app **UCS Intercom Service** on UCS, administrators need to know the following requirements and limitations.

4.1 Cross-Site-Request-Forgery protection

Warning

CSRF protection wasn't extensively tested and may break at any time.

CSRF protection may not be working for *Matrix*, *Nextcloud*, and *OX App Suite* versions released before September 2022.

CHAPTER

FIVE

TROUBLESHOOTING

When you encounter problems with the operation of the app **UCS Intercom Service**, this section provides information where you can look closer into and to get an impression about what's going wrong.

5.1 Log files

The app **UCS Intercom Service** produces different logging information in different places.

/var/log/univention/appcenter.log
: Contains log information around activities in the App Center.

  The App Center writes ICS relevant information to this file, when you run app lifecycle tasks like install, update and uninstall or when you change the app settings.

/var/log/univention/join.log
: Contains log information from join processes. When the App Center installs ICS, the app also joins the domain.

ICS Docker container
: The app uses a custom built node image. The App Center runs the container. You can view log information from the ICS Docker container with the following command:

```
$ univention-app logs intercom-service
```

/var/log/apache2/*.log
: Reverse proxy logs may contain relevant information for queried URLs by **UCS Intercom Service**, for example the status of middleware queries to other components.

> **Note**
>
> For externalized setups, like for example the *BMI-UX* setup, ICS proxies the queries through the external *HA-Proxy*. Therefore, you can find the proxy log files in `/var/log/haproxy.log` on the *HA-Proxy* server.

5.2 Common problems

Failing to provide the protocol, for example HTTP or HTTPS, for middleware relevant URLs like *intercom-service/nextcloud/url* (page 12), `intercom/portal/portal-url`, and `intercom/matrix/nordeck-url` leads to an error during the request in the form of:

```
TypeError: Cannot read properties of null (reading 'split')
  at required (/app/node_modules/requires-port/index.js:13:23)
```

CHAPTER

SIX

CHANGELOG

This changelog documents all notable changes to the ICS app. Keep a Changelog[10] is the format and this project adheres to Semantic Versioning[11].

6.1 v2.25.0

Released: 16. October 2026

6.1.1 Changed

- The app version now follows the version of the ICS container image and Helm chart. The version after 1.12 is 2.25.0. You can upgrade from 1.12 as usual.
- ICS now gets the *Matrix* token for a user from the *Matrix* OpenID `request_token` endpoint and no longer uses the deprecated `m.login.application_service` login. It forwards the token to the Meetings bot, so that the meeting creation in the *OX App Suite* calendar works with the *Matrix* Authentication Service.
- ICS answers unauthenticated requests to its API routes with `401` instead of a redirect to the identity provider. Only `/silent` and `/login` start a login.
- ICS logs to the console only and no longer writes the file `intercom-service.log` in the container, which grew without limit.

6.1.2 Security

- The back-channel logout now checks the audience and the `jti`, `events` and `nonce` claims of the logout token. Before, any token that the realm signed in the last 10 seconds ended the session of the user, for example an ID token for another client in the same single sign-on session.
- ICS no longer logs the request data of failed requests to *Matrix* and *Keycloak*. On the debug log level, the log contained the application service secret, the client secret and the token of the user.
- ICS runs in production mode. Error responses no longer contain stack traces, the `X-Powered-By` header is gone, and ICS sets the `X-Content-Type-Options` and `Referrer-Policy` headers.
- CORS no longer exposes the `Authorization` header to other origins.
- Update the base image and dependencies to fix CVEs.

6.1.3 Fix

- An invalid back-channel logout request no longer terminates ICS. It answers missing, invalid or expired logout tokens with `400`.
- Requests to `/nob` no longer terminate ICS when the session has no *Matrix* token, for example a session from before the upgrade.

[10] https://keepachangelog.com/en/1.0.0/
[11] https://semver.org/spec/v2.0.0.html

- The renewal of the *Matrix* token takes the *Matrix* user ID from the ID token, the same as the login. Before, the renewal failed if the claim was only in the ID token, and ICS forwarded the request with the old token. Requests to `/nob` without a *Matrix* token now get `502`.
- ICS tries the silent login again after a failed attempt. Before, one failed attempt turned off the silent login for the rest of the browser session.
- A *Redis* password with the characters `@`, `:`, `/`, `#` or `%` no longer makes ICS crash at startup.
- The mapping of the session state for the back-channel logout now expires in *Redis*. Before, the keys of all sessions that didn't end with a back-channel logout stayed in *Redis* forever.
- ICS forwards URL-encoded request bodies to the backends. Before, the backends got no body and waited until they timed out.

6.2 v1.12

Released: 17. July 2026

6.2.1 Add

- *UCR* variables to enable support for Keycloak Token Exchange v2.
  - *intercom-service/settings/token-exchange-v2* (page 13) to enable support for Keycloak Token Exchange v2 in the Intercom service

6.2.2 Changed

- The app's container image is now pulled from `artifacts.software-univention.de` instead of `docker.software-univention.de`. Operators who restrict outbound traffic with a firewall or proxy must, before upgrading, allow both `artifacts.software-univention.de` and `s3.de-west-1.psmanaged.com`: the registry answers image blob downloads with an HTTP 307 redirect to the latter (its S3 backing storage), so pulls fail if only the registry host is reachable. See Allowing artifacts.software-univention.de in firewalls and proxies[12] for guidance.

6.3 v1.11

Released: 20. March 2026

6.3.1 Security

- Update dependencies to fix critical and high CVEs.

6.3.2 Fix

- Enforce lowercase usernames for *Matrix* login as per Matrix specification[13].

6.4 v1.10

Released: 9. December 2025

6.4.1 Add

- *UCR* variables to allow configuration of session cookies.
  - *intercom-service/settings/enable-session-cookie* (page 8) to enable session cookie (transient cookie).

[12] https://help.univention.com/t/allowing-artifacts-software-univention-de-in-firewalls-and-proxies/25333
[13] https://spec.matrix.org/v1.17/appendices/#user-identifiers

- *intercom-service/settings/session-rolling-duration* (page 8) in seconds. The duration for which the session will be reset if the user is active within this time.

6.4.2 Security

- Critical CVE CVE-2025-49844[14] in **redis-server**.

6.5 v1.9

Released: 28. January 2025

6.5.1 Add

- Access Tokens are now attempted to be fetched from an integration if the integration is enabled but a current session does not have a token for the service. This allows for *XWiki* and *Matrix* integrations to become healthy in a user session once a user has used those services once, thus the account exists in the respective service.
- The *Redis* connection now supports TLS. Also, mTLS authentication can be configured.
- *UCR* variables to configure SSL connection to *Redis*.
  - *intercom-service/redis/ssl* (page 10) to activate or deactivate the SSL connection to the *Redis* server.
  - *intercom-service/redis/mtls* (page 10) to activate or deactivate the mTLS authentication to the *Redis* server.
  - *intercom-service/redis/certificates/mtls/cert* (page 11) to specify the path to certificate used during mTLS.
  - *intercom-service/redis/certificates/mtls/key* (page 11) to specify the path to certificate secret key used during mTLS.

6.5.2 Change

- Better log formatting in JSON for proxied routes.

6.5.3 Fix

- Missing imports in the *XWiki* proxy caused the service to crash under certain circumstances.
- Wrong logger configuration caused some log messages to not be correctly formatted.

6.6 v1.8

Released: 15. December 2024

6.6.1 Add

- Endpoint to securely proxy requests from *Univention-Portal* to *XWiki* on `/wiki`, allowing to load *RSS* feeds from *XWiki*.
- *UCR* variables to configure *XWiki*.
  - *intercom-service/xwiki/enabled* (page 12) to activate or deactivate *XWiki* integration.
  - *intercom-service/xwiki/url* (page 12) to set the URL of the *XWiki* instance.
  - *intercom-service/xwiki/audience* (page 12) to configure the OIDC client.

[14] https://www.cve.org/CVERecord?id=CVE-2025-49844

- *UCR* variable `intercom-service/settings/username-claim` (page 8) to configure the claim in the identity token that contains the username.
- *UCR* variable `intercom-service/nextcloud/enabled` (page 13) and `intercom-service/matrix/enabled` (page 11) to activate or deactivate the respective services.

6.6.2 Change

- Load all the configuration during initialization to improve asynchronous performance.
- Updated base image to *UCS 5.2-0* December 2024 build.

6.6.3 Fix

- *Univention-Portal* navigation crashed the application if the `intercom-service/settings/username-claim` (page 8) Keycloak mapping was missing in the access token.

6.7 v1.7

Released: 15. November 2024

6.7.1 Add

- *UCR* variable `intercom-service/settings/user-unique-mapper` (page 8) to configure the unique user claim name for the *Keycloak* OIDC client.

6.7.2 Change

- Order of middlewares for *Nextcloud* proxying caused authentication prompts in some cases.

6.8 v1.6

Released: 22. April 2024

6.8.1 Add

- Reconnect to *Redis* after the service was down.
- *UCR* variables to allow users to configure an external *Redis* server.
  - `intercom-service/redis/host` (page 10) to set the hostname of the *Redis* server.
  - `intercom-service/redis/port` (page 10) to set the port of the *Redis* server.
  - `intercom-service/redis/user` (page 10) to set the password of the *Redis* server.

6.8.2 Change

- Image is now based on *UCS* base image.

6.8.3 Fix

- AppCenter settings are now correctly applied during after installation.

6.9 v1.5

Released: 20. June 2023

6.9.1 Add

- *UCR* variables to set the path to external self-signed certificates.

6.9.2 Security

- Updated dependencies.

6.10 v1.4

Released: 13. December 2022

6.10.1 Change

- Refresh *Nextcloud* token when expired before proxying the request.
- Improved logging messages with JSON formatting.

6.10.2 Add

- *UCR* variable to set the log level.
- Logging to files and standard output.

6.10.3 Security

- Intercom service requests *Nextcloud* tokens with the *Nextcloud* audience, instead of the audience of *OX App Suite*.

6.11 v1.3

Released: 28. October 2022

6.11.1 Change

- Treat expired refresh tokens as no token, triggering a silent login attempt.
- Matrix login type set to *m.login.application_service* and is not configurable any more.
- Switch to *v3* Matrix client API.

6.12 v1.2

Released: 29. September 2022

6.12.1 Add

- Various debug logs

6.12.2 Change

- Apply firewall rules during installation to make ICS accessible from outside of UCS.
- Set Docker DNS based on the *UCR* variables *nameserver1*, *nameserver2* and *nameserver3*.

6.12.3 Security

- The *Filepicker* functionality of ICS now fetches a separate token for authenticating with the file hosting application *Nextcloud*. The *OX* OIDC-client in the IdP must be allowed, to fetch a token for the *Nextcloud* OIDC-client. This was always intended, but not correctly enforced in earlier versions.

6.12.4 Fix

- Update deprecated usage of *express.urlencoded*.
- ICS health check failed because of *Nordeck* URL returning *404*.
- Video conferences created as the wrong user.
- Central navigation returning *navigation.json* for the wrong user under certain circumstances.

6.13 v1.1

Released: 16. September 2022

6.13.1 Add

Stability

- ICS split the cookie headers by a logic that didn't consider certain cases. Now, ICS uses a standard cookie library for the handling cookie headers.
- During app installation, ICS tests the URLs of the required services **`Keycloak`**, *Nextcloud*, *Nordeck*, and *UCS Portal*, if it can reach them. The installation shows a warning, if the test can't reach the services. Additionally, ICS runs a health check within the Docker container every 60 seconds to test, if it can reach the services.

Refreshing Access Tokens
A middleware that automatically refreshes access tokens when they expire.

6.13.2 Change

- Improve the readability of user documentation.

6.13.3 Security

- The *Redis* database provides persistence for app sessions. The update applies the following security fixes to *Redis*:
  - Password protection provided in `/etc/intercom-redis.secret`.
  - The Redis container is only accessible from the **`docker-compose`** internal network (`external: false`).
- Verify the JWT (JSON Web Token) access or ID token with the public key of the *Keycloak* issuer.
- Enable `backchannel-logout` and remove the appropriated app-session from ICS.

6.13.4 Fix

- Convert the uppercase value for the environment variable `PROXY` to lowercase. Using the variable in JavaScript requires the value in a lowercase string.

6.14 v1.0

Released: 22. August 2022

6.14.1 Add

- Endpoint for OIDC silent login against **`Keycloak`** on `/silent`.
- Endpoint to securely proxy requests from *Open-Xchange* to *Nordeck* on `/nob`, allowing the creation of Element videoconferences from *Open-Xchange*.
- Endpoint to securely proxy requests from *Open-Xchange* to *Nextcloud* on `/fs`, allowing to use the email *Filepicker* with *Nextcloud*.
- Endpoint to securely proxy requests from *Open-Xchange* to *UCS Portal* `/navigation.json`, allowing for use of *UCS Portal* central navigation from *Open-Xchange*.
- Session storage with *Redis*.

BIBLIOGRAPHY

[1] *UCS 5.0 Manual*. Univention GmbH, 2021. URL: https://docs.software-univention.de/manual/5.0/en/.

[2] Raphaël Hertzog and Roland Mas. *The Debian Administrator's Handbook*, chapter Shell and Basic Commands. Freexian SARL, First edition, 2020. URL: https://www.debian.org/doc/manuals/debian-handbook/short-remedial-course.en.html#sect.shell-and-basic-commands.

[3] *OpenID Connect Core 1.0 incorporating errata set 1*. OpenID Foundation, 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html.

INDEX

C

CORS, **20**
CVE
    CVE-2025-49844, 27

E

environment variable
    intercom/matrix/nordeck-url, 23
    intercom/portal/portal-url, 23
    intercom-service/certificates/external/root-ca-crt, 13
    intercom-service/certificates/external/root-ca-pem, 13
    intercom-service/keycloak/realm-name, 9
    intercom-service/keycloak/url, 9
    intercom-service/matrix/enabled, 11, 28
    intercom-service/matrix/login-type, 11
    intercom-service/matrix/nordeck-mode, 12
    intercom-service/matrix/nordeck-url, 12
    intercom-service/matrix/server-name, 11
    intercom-service/matrix/url, 11
    intercom-service/nextcloud/audience, 12, 13
    intercom-service/nextcloud/enabled, 13, 28
    intercom-service/nextcloud/origin, 13
    intercom-service/nextcloud/url, 12, 13, 23
    intercom-service/portal/portal-url, 12
    intercom-service/redis/certificates/mtls/cert, 11, 27
    intercom-service/redis/certificates/mtls/key, 11, 27
    intercom-service/redis/host, 10, 28
    intercom-service/redis/mtls, 10, 27
    intercom-service/redis/port, 10, 28
    intercom-service/redis/ssl, 10, 27
    intercom-service/redis/user, 10, 28
    intercom-service/settings/base-url, 9
    intercom-service/settings/client-id, 8
    intercom-service/settings/enable-session-cookie, 8, 26
    intercom-service/settings/intercom-url, 8, 9
    intercom-service/settings/issuer-base-url, 9
    intercom-service/settings/log-level, 9
    intercom-service/settings/origin-regex, 9
    intercom-service/settings/proxy, 10
    intercom-service/settings/session-rolling-duration, 8, 27
    intercom-service/settings/token-exchange-v2, 3, 4, 13, 26
    intercom-service/settings/username-claim, 4, 8, 28
    intercom-service/settings/user-unique-mapper, 4, 8, 28
    intercom-service/xwiki/audience, 12, 13, 27
    intercom-service/xwiki/enabled, 12, 27
    intercom-service/xwiki/url, 12, 27
    PROXY, 30

I

IdP, **20**
intercom/matrix/nordeck-url, 23
intercom/portal/portal-url, 23
intercom-service/keycloak/realm-name, 9
intercom-service/keycloak/url, 9
intercom-service/matrix/enabled, 28
intercom-service/matrix/url, 11
intercom-service/nextcloud/audience, 13
intercom-service/nextcloud/enabled, 28
intercom-service/nextcloud/url, 13, 23
intercom-service/redis/certificates/mtls/cert, 27
intercom-service/redis/certificates/mtls/key, 27
intercom-service/redis/host, 28
intercom-service/redis/mtls, 27
intercom-service/redis/port, 28
intercom-service/redis/ssl, 27
intercom-service/redis/user, 28
intercom-service/settings/enable-session-cookie, 8, 26
intercom-service/settings/intercom-url, 9
intercom-service/settings/issuer-base-url, 9

intercom-service/settings/session-rolling-duration, 27
intercom-service/settings/token-exchange-v2, 3, 4, 26
intercom-service/settings/username-claim, 4, 28
intercom-service/settings/user-unique-mapper, 4, 28
intercom-service/xwiki/audience, 13, 27
intercom-service/xwiki/enabled, 27
intercom-service/xwiki/url, 27

O

OIDC audience, **20**

P

PROXY, 30