univention
be open.

School Accounts

Univention GmbH

Oct 28, 2026

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

CONTENTS

1 Introduction **1**
- 1.1 Audience . . . 1

2 Installation **3**
- 2.1 System requirements . . . 3
- 2.2 Install the app . . . 3
- 2.3 Upgrade the app . . . 4
- 2.4 Remove the app . . . 4

3 Configuration **5**
- 3.1 Choose who sees the portal tile . . . 5
- 3.2 Point the portal tile to another system . . . 5
- 3.3 User visibility . . . 5
- 3.4 Sessions . . . 6
- 3.5 App settings . . . 6

4 Usage **9**
- 4.1 List the users of a class or work group . . . 9
- 4.2 Reset passwords . . . 9
- 4.3 Sign out . . . 10

5 Roles and permissions **11**
- 5.1 Permissions of each role . . . 11
- 5.2 Legal guardians . . . 12

6 Troubleshooting **13**
- 6.1 A user doesn't see the School Accounts tile in the portal . . . 13
- 6.2 A password reset fails for some accounts . . . 13
- 6.3 A user sees "Invalid or missing security token" . . . 13
- 6.4 A user sees "not a valid UCS@school user" . . . 13
- 6.5 Logging . . . 14

7 Known issues **15**
- 7.1 Removing the app from one system deletes the portal tile . . . 15

8 Changelog **17**
- 8.1 v1.1.0 (unreleased) . . . 17
- 8.2 v1.0.0 (2026-10-01) . . . 17

Index **19**

CHAPTER

ONE

INTRODUCTION

The School Accounts app lists the users of a UCS@school school and lets authorized users reset their passwords. This manual explains how to install, configure, and use the app, and which roles and permissions decide what each user can see and do.

The Guardian[1] makes every authorization decision of the app. It evaluates the policy files that ship with the app, so a permission depends on the roles of both users and on how they relate to each other, for example a teacher and the students in their class, or a legal guardian and their ward. For more information about how those policies work and where they come from, see Policies[2].

Caution

The School Accounts app is a preview. Its features and configuration may change incompatibly in future releases. Univention supports it on a best-effort basis to help you test it. We're happy to receive your feedback.

1.1 Audience

This manual is for administrators who install and operate the School Accounts app. It assumes that you can run commands on the command line of a UCS system. The chapter *Usage* (page 9) is for the users of the app and uses only a web browser.

[1] https://docs.software-univention.de/nubus-manual/1.x/en/authorization.html#nubus-authorization
[2] https://docs.software-univention.de/nubus-manual/1.x/en/authorization.html#nubus-authorization-policies

CHAPTER

TWO

INSTALLATION

This section is for administrators and covers the installation, upgrade, and removal of the School Accounts app.

2.1 System requirements

If you install the School Accounts app and all its dependencies on the Primary Directory Node[3], that system needs at least the following resources:

- 12 GB RAM
- 150 GB disk space

The dependencies are UCS@school, the Guardian, the Kelvin REST API, and Keycloak, see *Install the app* (page 3).

2.2 Install the app

You can install the School Accounts app on a Primary Directory Node[4] or on a Backup Directory Node[5] that runs Univention Corporate Server (UCS) 5.2-6 with UCS 5.2 erratum 569[6] or later.

Your domain must meet the following prerequisites before you install the app:

- UCS@school[7] is installed and configured in the domain.
- The Guardian (`univention-guardian`) is installed and configured on the same system as the School Accounts app. For more information, see Install the authorization engine[8].
- The Kelvin REST API[9] in version 4.1.0 or later is installed and configured on the same system as the School Accounts app. The installation and the update of the School Accounts app abort otherwise.
- Keycloak is installed and configured in the domain as the identity provider. For more information about installing Keycloak, see the Univention Keycloak app documentation[10].

To install the School Accounts app from the command line, run the following command:

Listing 2.1: Install the School Accounts app from the command line

```
$ univention-app install ucsschool-accounts
```

After the installation, users reach the app through the Univention Portal[11]. To choose who sees the portal tile, see *Choose who sees the portal tile* (page 5). The tile opens the app on the first system in the domain that you install it on. To point it to another system, see *Point the portal tile to another system* (page 5).

[3] https://docs.software-univention.de/manual/latest/en/domain-ldap/system-roles.html#domain-ldap-primary-directory-node
[4] https://docs.software-univention.de/manual/latest/en/domain-ldap/system-roles.html#domain-ldap-primary-directory-node
[5] https://docs.software-univention.de/manual/latest/en/domain-ldap/system-roles.html#domain-ldap-backup-directory-node
[6] https://errata.software-univention.de/#/?erratum=5.2x569
[7] https://docs.software-univention.de/ucsschool-manual/latest/de/index.html#introduction
[8] https://docs.software-univention.de/ucs-operation/5.2/en/iam/authorization-engine.html#iam-authorization-engine-install
[9] https://docs.software-univention.de/ucsschool-kelvin-rest-api/index.html
[10] https://docs.software-univention.de/keycloak-app/latest/installation.html#app-installation
[11] https://docs.software-univention.de/manual/latest/en/central-management-umc/portal.html#central-portal

2.3 Upgrade the app

To upgrade the School Accounts app to a later version, run the following command:

Listing 2.2: Upgrade the School Accounts app from the command line

```
$ univention-app upgrade ucsschool-accounts
```

2.4 Remove the app

To remove the School Accounts app from a system, run the following command:

Listing 2.3: Remove the School Accounts app from the command line

```
$ univention-app remove ucsschool-accounts
```

The removal deletes the portal tile and the policies of the app. It deletes the tile even if the app still runs on other systems, see *Removing the app from one system deletes the portal tile* (page 15). It keeps the Keycloak client, because other systems in the domain can still use it.

2.4.1 Remove the Keycloak client

The School Accounts app signs users in through the Keycloak client `ucsschool-accounts-ui` in the `ucs` realm. After you remove the app from every system in the domain, you can delete that client in the Keycloak administration console at `https://ucs-sso-ng.`*`DOMAIN_NAME`*`/admin/master/console/`. Replace `DOMAIN_NAME` with the domain name of your UCS domain. If you have set the Univention Configuration Registry (UCR) variable `keycloak/server/sso/fqdn`[12], use its value instead of `ucs-sso-ng.`*`DOMAIN_NAME`*.

[12] https://docs.software-univention.de/keycloak-app/latest/configuration.html#envvar-keycloak-server-sso-fqdn

CHAPTER

THREE

CONFIGURATION

3.1 Choose who sees the portal tile

The installation adds a School Accounts tile to the Univention Portal[13] and shows it to everyone, including visitors who aren't signed in. Selecting the tile asks them to sign in first. The tile doesn't grant any permission. The app shows every user only what their roles allow, as described in *Roles and permissions* (page 11).

To show the tile only to the users of the school `DEMOSCHOOL`, add the `Domain Users DEMOSCHOOL` group of that school to the portal entry:

Listing 3.1: Show the portal tile only to the users of one school

```
$ univention-directory-manager portals/entry modify \
    --dn "cn=ucsschool-accounts,cn=entry,cn=portals,cn=univention,$(ucr get ldap/
↪base)" \
    --append allowedGroups="cn=Domain Users DEMOSCHOOL,cn=groups,ou=DEMOSCHOOL,
↪$(ucr get ldap/base)"
```

To add further schools, repeat the `--append` option for each of them.

3.2 Point the portal tile to another system

The domain has one School Accounts tile, no matter on how many systems you install the app. The first installation in the domain creates the tile and links it to the system it runs on. Later installations on other systems leave the tile unchanged.

To let the tile open the app on another system, for example `school-accounts.example.com`, change the link of the portal entry:

Listing 3.2: Point the portal tile to the app on another system

```
$ univention-directory-manager portals/entry modify \
    --dn "cn=ucsschool-accounts,cn=entry,cn=portals,cn=univention,$(ucr get ldap/
↪base)" \
    --set link='"en_US" "//school-accounts.example.com/univention/ucsschool/
↪ucsschool-accounts/ui/"'
```

3.3 User visibility

This preview doesn't yet support configuring which users appear in the list view. The list view shows a user when the Guardian grants the signed-in user at least one permission on them. For the permissions of each role, see *Roles and permissions* (page 11).

[13] https://docs.software-univention.de/manual/latest/en/central-management-umc/portal.html#central-portal

3.4 Sessions

After a user signs in, the School Accounts app keeps the session in an encrypted browser cookie. The server stores no session data, so a session survives both a restart of the app and an app update.

3.4.1 Session lifetime

A session lasts at most 8 hours. Keycloak's own session limits usually end it earlier: the realm's *SSO Session Idle* and *SSO Session Max*, or *Client Session Idle* and *Client Session Max* if the client overrides them. Whichever limit runs out first sends users back to the sign-in page.

Signing out also ends the session in Keycloak. A copy of the session cookie keeps working until the app next renews the sign-in against Keycloak, which happens within the lifetime of an access token.

Each host encrypts its cookies with its own secret, so users who switch to the app on another UCS system pass through Keycloak again. While their Keycloak session is open, they see only a redirect and don't have to enter their credentials.

3.4.2 End all sessions on a host

To end every app session on a host, write a new secret into the `session.secret` file and restart the app:

Listing 3.3: Replace the session secret

```
$ openssl rand -hex 32 > \
    /var/lib/univention-appcenter/apps/ucsschool-accounts/conf/session.secret
$ univention-app restart ucsschool-accounts
```

This doesn't sign users out of Keycloak, so their next request takes them back into the app without a credential prompt. To force users to authenticate again, end their sessions in Keycloak. Don't delete the `session.secret` file: the app generates it only during installation and refuses to start without it.

3.4.3 Change the client secret

The app authenticates to Keycloak with the client secret in `/var/lib/univention-appcenter/apps/ucsschool-accounts/conf/client.secret`. The secret belongs to the domain, and Keycloak holds the authoritative copy. Every installation and every app update adopts the secret that Keycloak holds.

To change the secret, regenerate it for the client `ucsschool-accounts-ui` in the Keycloak administration console. Then write the new secret into the `client.secret` file on every host that runs the app, and restart the app there:

Listing 3.4: Replace the client secret on a host

```
$ printf '%s\n' "NEW_SECRET" > \
    /var/lib/univention-appcenter/apps/ucsschool-accounts/conf/client.secret
$ univention-app restart ucsschool-accounts
```

Replace `NEW_SECRET` with the secret from Keycloak. Until you have updated a host, it can't sign users in.

3.5 App settings

This section is a reference for all app settings of the School Accounts app. You can configure the settings with the **`univention-app`** command or in the app settings dialog of the App Center UMC module[14].

The Univention App Center restarts the app after you change any setting.

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

3.5.1 Kelvin

ucsschool-accounts/kelvin/url

Defines the hostname of the system that runs the Kelvin REST API. If unset, the app uses the fully qualified domain name of the system that it runs on.

3.5.2 Logging

ucsschool-accounts/logging/level

Sets the log level of the app. Possible values are `TRACE`, `DEBUG`, `INFO`, `WARNING`, `ERROR`, and `CRITICAL`, with `INFO` as the default. The values are case-insensitive. `DEBUG` adds the technical context of each request, which helps with a bug report. `TRACE` additionally logs the start and end of every request. A value that names no level doesn't keep the app from starting: the app logs a warning and uses `INFO`.

ucsschool-accounts/logging/structured

If set to `True`, the app writes each log record as a single JSON object, which a log aggregator can parse without a pattern. The default is `False`, which writes the format described in *Log format* (page 14).

You can't configure the format itself. Both variants follow the Univention log format, so that logs stay consistent across the products of a domain.

3.5.3 Portal

ucsschool-accounts/frame-ancestors

Defines the sites allowed to show the application inside a frame, as a space separated list of sources of the `Content-Security-Policy` directive `frame-ancestors`. The portal opens the application inside a frame, so the list must contain every system whose portal links to it. The default `'self'` covers a portal on the UCS system where you installed the application. Add the URL of any other system, for example `https://primary.example.org`. Without a matching entry, the browser refuses to show the application inside the portal.

3.5.4 Authentication

ucsschool-accounts/oauth/keycloak-uri

Defines the base URI of the Keycloak server that signs users in. If unset, the app builds the URI from the hostname in the Univention Configuration Registry (UCR) variable `keycloak/server/sso/fqdn`[15], and if that is unset too, from `ucs-sso-ng.`*`DOMAIN_NAME`*.

[15] https://docs.software-univention.de/keycloak-app/latest/configuration.html#envvar-keycloak-server-sso-fqdn

CHAPTER

FOUR

USAGE

This section describes how school administrators, teachers, and legal guardians use the School Accounts app to find users and reset their passwords. What each of them can see and do depends on their roles, as described in *Roles and permissions* (page 11).

4.1 List the users of a class or work group

To list the users of a school class or a work group, follow these steps:

1. Open the School Accounts tile in the Univention Portal[16]. If Keycloak asks for your credentials, sign in.
2. In the *School* list, select the school. The list contains only the schools that you can filter by.
3. In the *Classes & Work Groups* list, select the class or work group. To find it in a long list, enter part of its name. To list every user of the school, select *All classes and work groups*.
4. Optional: To list only the users with one role, select the role in the *Roles* row above the table, for example *Student*. Selecting another role replaces it. To list every role again, select the role once more or select *All*. A user counts only with the roles they hold at the selected school.

The list view shows the users of your selection with the columns *Last name*, *First name*, *Username*, *Roles*, and *Activated*. It shows only the users whose data you may read.

The browser address contains the selected school, group, and role. You can bookmark it or share it, and it opens the same list for every user who has the same permissions.

4.2 Reset passwords

You can set one new password for one or more users at once. To reset passwords, follow these steps:

1. List the users as described in *List the users of a class or work group* (page 9).
2. Select the users whose passwords you want to reset. The list view shows a checkbox only for the users whose password you can reset. To select every such user on the page, select the checkbox in the table header.
3. Click *Reset password*. A dialog lists the accounts that the reset affects. It warns you when the selection contains users with different roles, or when it affects many accounts at once.
4. In the *New password* box, enter the new password.
5. Optional: To make the users choose their own password when they sign in next, select *User has to change the password at the next login*. The dialog offers this option only when the Kelvin REST API exposes the Univention Directory Manager (UDM) property `pwdChangeNextLogin` of users.
6. Select the checkbox that states how many passwords the reset affects, and click *Set password*.

The app then shows how many passwords it has reset and lists every account whose password it couldn't reset, with the reason. A single reset covers at most 200 users.

[16] https://docs.software-univention.de/manual/latest/en/central-management-umc/portal.html#central-portal

4.3 Sign out

To end your session, click *logout* in the page header. The app also ends your session in Keycloak, so the next visit asks for your credentials again.

CHAPTER

FIVE

ROLES AND PERMISSIONS

This section is for administrators and describes what each role can do in the School Accounts app. The Guardian[17] makes the authorization decisions, using a policy bundle that ships with the School Accounts app. For more information about how the Guardian reaches a decision, see Policies[18].

The policy bundle fixes the roles and their permissions. This preview doesn't yet support creating new roles or changing the permissions of a role.

5.1 Permissions of each role

The following table shows what a user with each role can do on other users in the same school:

Table 5.1: Permissions of each role

Role	Read user data of	Reset passwords of	Own password reset by
School administrator (`school_admin`)	Every user in their school	Every user in their school, except school administrators	Nobody
Teacher (`teacher`)	Students in their school classes	Students in their school classes whose only role is `student`	School administrators
Staff (`staff`)	Nobody	Nobody	School administrators
Student (`student`)	Nobody	Nobody	Teachers of their school classes, school administrators, and legal guardians
Legal guardian	Their wards	Their wards whose only role is `student`	School administrators

The following rules apply to every role:

- A user can read their own data, unless `student` is their only role.
- A user can filter the list view by the schools they're enrolled in and by the school classes and work groups they're a member of. A school administrator can also filter by every school class and work group in their school.
- Filtering the list view by role needs no permission of its own. It only hides users from the list, and it hides every user whose roles the list view doesn't show.
- Nobody can reset their own password.
- A password reset requires permission for *every* role of the user whose password is reset. For example, a teacher can't reset the password of a student who is also a staff member, and nobody can reset the password of a user who holds `school_admin` in any school.
- A role counts only at the schools where a user holds it, and the other user must be enrolled in one of those schools.
- The role `student` takes nothing away from a user who also holds another role.

[17] https://docs.software-univention.de/nubus-manual/1.x/en/authorization.html#nubus-authorization
[18] https://docs.software-univention.de/nubus-manual/1.x/en/authorization.html#nubus-authorization-policies

5.2 Legal guardians

Legal guardianship is a relationship between two users that UCS@school records on both of them. A legal guardian can read the data of their wards because of that relationship alone, whatever roles or schools either user has. Resetting the password of a ward also requires that `student` is the ward's only role. To see a ward in the list view, the legal guardian must be enrolled in the school of the ward.

UCS@school also gives a legal guardian the role `legal_guardian` in the school of their wards. Because of that role, only school administrators can reset the password of a legal guardian, and a legal guardian can read their own data.

CHAPTER

SIX

TROUBLESHOOTING

6.1 A user doesn't see the School Accounts tile in the portal

By default, the portal shows the tile to everyone. If the `allowedGroups` attribute of the portal entry names any groups, the portal shows the tile only to their members. To check or change those groups, see *Choose who sees the portal tile* (page 5).

6.2 A password reset fails for some accounts

After a password reset, the app lists every account whose password it couldn't reset, with one of the following reasons:

Permission denied
: The signed-in user can't reset the password of that account. For the permissions of each role, see *Roles and permissions* (page 11).

User not found
: The account doesn't exist anymore.

Rejected
: UCS refused the new password, for example because it doesn't meet the password policy. The message that follows states the reason.

Could not be confirmed
: An unexpected error occurred during the reset, so the password of that account may or may not have changed. Check the account before you reset its password again. The log contains the error as `A password reset could not be confirmed`.

This account cannot be required to change its password at the next login
: The option *User has to change the password at the next login* was selected, but the Kelvin REST API doesn't expose the UDM property `pwdChangeNextLogin` for that account. The app left the password unchanged. To solve the problem, add `pwdChangeNextLogin` to the user properties in `/etc/ucsschool/kelvin/mapped_udm_properties.json` of the Kelvin REST API.

If no reset succeeds and UCS refuses the password with the same reason every time, the app shows no list. The dialog stays open and shows the reason below *New password*, so that you can enter a different password.

6.3 A user sees "Invalid or missing security token"

The page that the user worked on is older than their current session, for example after they signed in again in another browser tab. To solve the problem, reload the page.

6.4 A user sees "not a valid UCS@school user"

Only UCS@school users can use the School Accounts app. Accounts without a UCS@school role, such as the `Administrator` account, can't use it.

The app logs this situation as a warning, `Actor is not a UCS@school user, so this module is unusable for them.`, with the account in the data section as `actor_id`.

6.5 Logging

When an error occurs, the app logs the error with its details and context. To read the logs, run the following command:

Listing 6.1: Read the logs of the School Accounts app

```
$ univention-app logs ucsschool-accounts
```

Add an excerpt of these logs to a bug report, even when the messages don't tell you much, because they help the developers resolve the error.

Before reproducing a problem, set *ucsschool-accounts/logging/level* (page 7) to `DEBUG`. The app then logs the technical context of each request, such as the school and group that a list view was filtered by, and which backend answered.

6.5.1 Log format

Each log record follows the Univention log format and looks like the following example:

Listing 6.2: Log record in the text format

```
2026-09-10T08:22:57.275+00:00 WARNING  [31f863092a] Actor is not a UCS@school user,
↪ so this module is unusable for them. Assign a UCS@school role, or use a school␣
↪account.  | actor_id="administrator" module="ucsschool_accounts.business_logic"␣
↪line=50 pid=8 request_id="31f863092ade1cb"
```

A record consists of the following five parts:

Table 6.1: Parts of a log record

Part	Description
Timestamp	The time the app wrote the record, in ISO 8601 and always in UTC, so that you can read the records of different hosts of a domain as one sequence.
Severity	The log level, padded to a fixed width.
Request ID	In square brackets, shortened to 10 characters. It identifies the browser request that produced the record, and is a hyphen for records written outside a request, such as during startup.
Message	The event that happened, without any data.
Data	Everything after the separator, which is a tab, a pipe, and a space (`\t| ``). Written as ``key=value` pairs: first the data belonging to the event, then the source code location, the process ID, and the full request ID. The app replaces every character of the request ID outside of `[A-Za-z0-9._:+-]` with _.

To follow a single browser request through the log, search for its request ID. The app returns that value in the `X-Request-ID` response header, and reuses the value of an incoming `X-Request-ID` request header when one is present. In the text format, an ID with replaced characters doesn't match the header value. The structured format keeps the ID as sent.

To write each record as a single JSON object instead, set *ucsschool-accounts/logging/structured* (page 7) to `True`.

CHAPTER

SEVEN

KNOWN ISSUES

This section lists the known issues of the School Accounts app.

7.1 Removing the app from one system deletes the portal tile

The domain has one School Accounts portal tile for all systems that run the app. Removing the app from any of these systems deletes that tile, even if the app still runs on other systems. Users then can't open the app from the portal anymore.

CHAPTER

EIGHT

CHANGELOG

This section lists the changes to the School Accounts app.

8.1 v1.1.0 (unreleased)

- You can filter the list of users by role, one role at a time. The role filter applies to the roles that users hold at the selected school. The browser address contains the selected role, so a bookmark or a shared link opens the same list.

8.2 v1.0.0 (2026-10-01)

- School Accounts 1.0.0 is the first release of the app. School administrators and teachers open it from the Univention Portal and work with the user accounts of their school in a web browser.
- You can list the users of a school and filter the list by school class or work group.
- You can reset the password of one or more users at once. A confirmation dialog shows who is affected before anything changes. Teachers can reset the passwords of the students in their classes. School administrators can reset the passwords of the other users of their school.
- The user interface is available in English and German.
- The app requires the Kelvin REST API 4.1.0 or later and the Guardian on the same system, and Keycloak in the domain.

INDEX

E

environment variable
    keycloak/server/sso/fqdn, 4, 7
    ucsschool-accounts/frame-ancestors, 7
    ucsschool-accounts/kelvin/url, 7
    ucsschool-accounts/logging/level, 7, 14
    ucsschool-accounts/logging/structured, 7, 14
    ucsschool-accounts/oauth/keycloak-uri, 7
Errata updates
    UCS 5.2 erratum 569, 3

K

keycloak/server/sso/fqdn, 4, 7

U

ucsschool-accounts/logging/level, 14
ucsschool-accounts/logging/structured, 14