

DAV & database commands



- [DAV commands](#)
 - [Manage addressbooks](#)
 - [dav:list-addressbooks](#)
 - [dav:create-addressbook](#)
 - [Manage calendars](#)
 - [dav:list-calendars](#)
 - [dav:list-calendar-shares](#)
 - [dav:create-calendar](#)
 - [dav:delete-calendar](#)
 - [dav:move-calendar](#)
 - [dav:clear-calendar-unshares](#)
 - [Manage calendar subscriptions](#)
 - [dav:list-subscriptions](#)
 - [dav:create-subscription](#)
 - [dav:delete-subscription](#)
 - [Manage absences \(out-of-office\)](#)
 - [dav:absence:set](#)
 - [dav:absence:get](#)
 - [Sync](#)
 - [dav:sync-system-addressbook](#)
 - [dav:sync-birthday-calendar](#)
 - [Maintenance](#)
 - [dav:fix-missing-caldav-changes](#)
 - [dav:remove-invalid-shares](#)
 - [dav:retention:clean-up](#)
 - [dav:send-event-reminders](#)
 - [dav:clear-contacts-photo-cache](#)
 - [Export and import calendars](#)

- [calendar:export](#)
 - [calendar:import](#)
 - [Disable creation of example events](#)
 - [Database commands](#)
 - [Schema maintenance](#)
 - [db:add-missing-indices](#)
 - [db:add-missing-columns](#)
 - [db:add-missing-primary-keys](#)
 - [Schema inspection](#)
 - [db:schema:check](#)
 - [db:schema:export](#)
 - [db:schema:expected](#)
 - [Data conversions](#)
 - [db:convert-filecache-bigint](#)
 - [db:convert-mysql-charset](#)
 - [db:convert-type](#)
-

DAV commands

The `dav` commands manage addressbooks, calendars, calendar subscriptions, absences, and related data:

dav	
dav:absence:get	get the out-of-office absence settings for a user
dav:absence:set	set the out-of-office absence settings for a user
dav:clear-calendar-unshares	clear calendar unshares for a user
dav:clear-contacts-photo-cache	clear cached contact photos
dav:create-addressbook	create a dav addressbook
dav:create-calendar	create a dav calendar
dav:create-subscription	create a dav subscription
dav:delete-calendar	delete a dav calendar
dav:delete-subscription	delete a calendar subscription for a user
dav:fix-missing-caldav-changes	insert missing calendarchanges rows for existing events
dav:list-addressbooks	list all addressbooks of a user

dav:list-calendar-shares	list all calendar shares for a user
dav:list-calendars	list all calendars of a user
dav:list-subscriptions	list all calendar subscriptions for a user
dav:move-calendar	move a calendar from one user to another
dav:remove-invalid-shares	remove invalid dav shares
dav:retention:clean-up	delete CalDAV trash elements that are due for removal
dav:send-event-reminders	send event reminder notifications
dav:sync-birthday-calendar	synchronize the birthday calendar
dav:sync-system-addressbook	synchronize users to the system addressbook
calendar	
calendar:export	export a calendar of a user
calendar:import	import a calendar to a user

Manage addressbooks

dav:list-addressbooks

List all addressbooks for a user:

```
sudo -E -u www-data php occ dav:list-addressbooks layla
```

Database ID	URI	Owner principal	Owner displayname	Writable
contacts	Contacts	principals/users/layla	Layla Smith	✓

dav:create-addressbook

Create an addressbook for a user:

```
sudo -E -u www-data php occ dav:create-addressbook layla work
```

Manage calendars

dav:list-calendars

List all calendars for a user:

```
sudo -E -u www-data php occ dav:list-calendars layla
```

URI	Displayname	Owner principal	Owner displayname	Writable
personal	Personal	principals/users/layla	Layla Smith	✓

dav:list-calendar-shares

List all calendar shares for a user:

```
sudo -E -u www-data php occ dav:list-calendar-shares layla
User layla has no calendar shares
```

Use `--calendar-id` to filter results to a specific calendar.

dav:create-calendar

Create a calendar for a user:

```
sudo -E -u www-data php occ dav:create-calendar layla holidays
```

dav:delete-calendar

Delete a named calendar for a user:

```
sudo -E -u www-data php occ dav:delete-calendar layla holidays
```

Use `--birthday` to delete the birthday calendar instead of specifying a name. Use `-f` / `--force` to delete immediately instead of moving to the trashbin:

```
sudo -E -u www-data php occ dav:delete-calendar --force layla holidays
sudo -E -u www-data php occ dav:delete-calendar --birthday layla
```

dav:move-calendar

“ ”

Moving a calendar changes its existing share URLs.

Move a calendar from one user to another:

```
sudo -E -u www-data php occ dav:move-calendar personal layla fred
```

Without `--force`, the command fails if the destination user already has a calendar with the same name, or if the calendar is shared with a group the destination user is not a member of.

Use `-f` / `--force` to proceed anyway: conflicting group shares are dropped, and if the calendar name is already taken at the destination a new name is tried automatically (`personal-1`, `personal-2`, up to ten attempts). The command fails if no free name is found within those attempts.

dav:clear-calendar-unshares

Clear stale calendar unshare records for a user. Useful when a user's calendar sharing state has become inconsistent:

```
sudo -E -u www-data php occ dav:clear-calendar-unshares layla  
User layla has no calendar unshares
```

Manage calendar subscriptions

dav:list-subscriptions

List all calendar subscriptions for a user:

```
sudo -E -u www-data php occ dav:list-subscriptions layla  
User layla has no subscriptions
```

dav:create-subscription

Create a subscription calendar for a user:

```
sudo -E -u www-data php occ dav:create-subscription layla "Astronomy Calendar" \  
webcal://cantonbecker.com/astronomy-calendar/astrocal.ics
```

Optionally pass a HEX color code for the calendar:

```
sudo -E -u www-data php occ dav:create-subscription layla "Astronomy Calendar" \  
webcal://cantonbecker.com/astronomy-calendar/astrocal.ics "#ff5733"
```

If not set, the theming default color is used.

dav:delete-subscription

Delete a calendar subscription for a user:

```
sudo -E -u www-data php occ dav:delete-subscription layla "Astronomy Calendar"
```

Manage absences (out-of-office)

dav:absence:set

Set the out-of-office absence for a user. `first-day` and `last-day` are inclusive and formatted as `YYYY-MM-DD`.

- **short-message** — a brief one-liner displayed as the user's status next to their name across the UI while they are absent (e.g. in contacts, chat, and user lists).
- **message** — the full out-of-office text shown to users who try to reach the absent user.
- **replacement-user-id** — optional; a colleague to contact instead. Their display name is stored alongside the absence and shown to others.

```
sudo -E -u www-data php occ dav:absence:set layla 2026-05-01 2026-05-15 \  
"On leave" \  
"I am on leave until May 15. For urgent matters please contact my colleague."
```

To set an absence with a replacement user:

```
sudo -E -u www-data php occ dav:absence:set layla 2026-05-01 2026-05-15 \  
"On leave" \  
"I am on leave. Please contact fred for urgent matters." \  
fred
```

dav:absence:get

Show the current absence settings for a user:

```
sudo -E -u www-data php occ dav:absence:get layla  
Start day: 2026-05-01  
End day: 2026-05-15  
Short message: On leave  
Message: I am on leave until May 15. For urgent matters please contact my colleague.  
Replacement user: none  
Replacement display name: none
```

Sync

dav:sync-system-addressbook

Synchronize all users to the [system address book](#):

```
sudo -E -u www-data php occ dav:sync-system-addressbook  
Syncing users ...
```

dav:sync-birthday-calendar

Add birthdays from shared addressbooks to a user's calendar:

```
sudo -E -u www-data php occ dav:sync-birthday-calendar layla  
Start birthday calendar sync for layla
```

Maintenance

dav:fix-missing-caldav-changes

Restore missing calendar sync change records. If the `calendarchanges` table has lost data, clients may not sync correctly. Run for a single user, or omit the user ID to fix all users (may take some time on large instances):

```
sudo -E -u www-data php occ dav:fix-missing-caldav-changes layla  
sudo -E -u www-data php occ dav:fix-missing-caldav-changes
```

dav:remove-invalid-shares

Remove invalid DAV shares created by a bug in a previous version:

```
sudo -E -u www-data php occ dav:remove-invalid-shares
```

dav:retention:clean-up

Delete elements from the CalDAV trash that are past their retention period:

```
sudo -E -u www-data php occ dav:retention:clean-up
```

dav:send-event-reminders

Send pending calendar event reminder notifications. By default, Nextcloud sends reminders via the background job. To use this command instead, switch the mode first:

```
sudo -E -u www-data php occ config:app:set dav sendEventRemindersMode --value occ
```

Then call the command regularly via a dedicated cron job to ensure reminders are sent on time:

```
sudo -E -u www-data php occ dav:send-event-reminders
```

See [../groupware/calendar](https://nextcloud.com/groupware/calendar) for the recommended cron schedule.

dav:clear-contacts-photo-cache

Clear all cached contact photos. Useful after a migration or if contact avatars are not displaying correctly:

```
sudo -E -u www-data php occ dav:clear-contacts-photo-cache  
No cached contact photos found.
```

Export and import calendars

calendar:export

Export a calendar to a file or standard output:

```
sudo -E -u www-data php occ calendar:export layla personal --location /tmp/personal.ics
```

`--format` selects the output format (default: `ical`):

- `ical` — iCalendar (RFC 5545)
- `xcal` — XML iCalendar (RFC 6321)
- `jcal` — JSON iCalendar (RFC 7265)

`--location` sets the output file path. If omitted, output is written to standard output:

```
sudo -E -u www-data php occ calendar:export layla personal --format xcal \  
  --location /tmp/personal.xcal  
sudo -E -u www-data php occ calendar:export layla personal --format jcal
```

calendar:import

Import calendar entries into a user's calendar:

```
sudo -E -u www-data php occ calendar:import layla personal /tmp/personal.ics
```

`--format` selects the input format (default: `ical`). Same values as `calendar:export`. If `location` is omitted, reads from standard input:

```
sudo -E -u www-data php occ calendar:import --format xcal layla personal \  
    /tmp/personal.xcal  
cat /tmp/personal.ics | sudo -E -u www-data php occ calendar:import layla personal
```

`--supersede` forces override of existing objects with the same UID.

Validation behaviour is controlled with `--validation`:

- `0` — no validation
- `1` — validate, skip invalid entries
- `2` — validate, fail on invalid entries

Error handling is controlled with `--errors`:

- `0` — continue on error (default)
- `1` — fail on first error

Use `--show-created`, `--show-updated`, `--show-skipped`, or `--show-errors` to see the UIDs of affected objects after import.

Disable creation of example events

Disable the automatic creation of example events for new calendar users:

```
sudo -E -u www-data php occ config:app:set dav create_example_event \  
    --value=false --type=boolean
```

Database commands

The `db` commands manage database schema, indices, and data conversions:

<code>db</code>	
<code>db:add-missing-columns</code>	add missing optional columns to the database tables
<code>db:add-missing-indices</code>	add missing indices to the database tables
<code>db:add-missing-primary-keys</code>	add missing primary keys to the database tables

db:convert-filecache-bigint	convert the ID columns of the filecache to BigInt
db:convert-mysql-charset	convert charset of MySQL/MariaDB to utf8mb4
db:convert-type	convert the Nextcloud database to a different type
db:schema:check	compare the live database schema with the expected schema
db:schema:expected	export the expected database schema for a fresh installation
db:schema:export	export the current database schema

Schema maintenance

db:add-missing-indices

Add any indices to the database that are defined in the schema but missing from the current installation. Run this after upgrades if performance has degraded or if prompted by the admin overview:

```
sudo -E -u www-data php occ db:add-missing-indices  
Done.
```

Use `--dry-run` to preview the SQL statements without executing them:

```
sudo -E -u www-data php occ db:add-missing-indices --dry-run
```

db:add-missing-columns

Add optional columns that are defined in the schema but missing from the current installation:

```
sudo -E -u www-data php occ db:add-missing-columns  
Done.
```

Use `--dry-run` to preview the SQL statements without executing them.

db:add-missing-primary-keys

Add primary keys that are defined in the schema but missing from the current installation:

```
sudo -E -u www-data php occ db:add-missing-primary-keys  
Done.
```

Use `--dry-run` to preview the SQL statements without executing them.

Schema inspection

db:schema:check

🔧 New in version 35.0

Compare the live database schema with the schema expected for the currently installed Nextcloud version:

```
sudo -E -u www-data php occ db:schema:check
```

The live database schema matches the expected schema.

The expected schema is generated from the migrations provided by Nextcloud core, all enabled apps, and any disabled apps whose code is still present. The command reports missing or unexpected tables, columns, and indexes. It also reports column properties that differ, such as the type, length, default value, or whether the column is nullable.

The command only inspects the schema and does not modify the database. It exits with status `0` if there are no findings for core or an enabled app, and status `1` otherwise. This makes it suitable for use in monitoring scripts. Findings that only concern a disabled app never affect the exit status. They are excluded from the plain output unless you add `-v`:

```
sudo -E -u www-data php occ db:schema:check -v
```

oc_example: column 'size' differs in: type, default

Disabled apps (not affecting exit code):

If the schema for a disabled app differs from what is expected, this might indicate the app was updated since it was disabled. Missing migrations will be applied once the app is enabled again.

files_external:

- oc_storages: missing column 'backend'

Specify a table name to limit the comparison to that table:

```
sudo -E -u www-data php occ db:schema:check oc_filecache
```

Use the complete table name, including the configured database table prefix. Use `--output=json` or `--output=json_pretty` to produce machine-readable output. Unlike plain output, JSON output always includes every finding, including those for disabled apps. For example:

```
$ sudo -E -u www-data php occ db:schema:check --output=json_pretty
[
  {
    "table": "oc_filecache",
    "type": "modified_column",
    "name": "size",
    "changes": [
      "type",
      "default"
    ],
    "app": null,
    "enabled": true
  },
  {
    "table": "oc_storages",
    "type": "missing_column",
    "name": "backend",
    "app": "files_external",
    "enabled": false
  },
  {
    "table": "oc_example",
    "type": "unexpected_table",
    "app": null,
    "enabled": true
  }
]
```

Each finding contains:

- `table` — the complete database table name.
- `type` — one of `missing_table`, `unexpected_table`, `missing_column`, `unexpected_column`, `modified_column`, `missing_index`, or `unexpected_index`.
- `name` — the affected column or index. This field is omitted for table findings.
- `changes` — the changed column properties. This field is only present when `type` is `modified_column`. Possible values include `type`, `length`, `precision`, `scale`, `nullable`, `default`, `autoincrement`, `unsigned`, `fixed`, and `comment`.
- `app` — the ID of the disabled app that owns the affected table, or `null` if the table belongs to core or an enabled app.
- `enabled` — `false` if the finding only concerns a disabled app and does not affect the exit status, `true` otherwise.

When the schemas match, JSON output contains an empty array.



The expected schema does not account for all database-wide settings or defaults, so some reported differences may be valid. A table or other object still counts as unexpected if it belongs to an app that was disabled and then fully removed. Review each finding before changing the database.

The schema check also runs automatically after `occ upgrade`. Findings for core or an enabled app are displayed as warnings but do not cause the upgrade to fail; findings that only concern a disabled app are not shown here. For example, the following output may appear near the end of an upgrade:

```
The database schema does not match what is expected for the installed version:
```

- oc_filecache: column 'size' differs in: type, default
- oc_filecache: missing index 'fs_storage_path_hash'
- unexpected table 'oc_example'

```
Run "occ db:schema:check -v" for details.
```

Run `occ db:schema:check` separately to inspect the differences in plain (add `-v` to include disabled-app findings) or machine-readable format.

db:schema:export

Export the current database schema. Useful for debugging or comparing against the expected schema:

```
sudo -E -u www-data php occ db:schema:export
```

- oc_filecache:
 - columns:
 - checksum:
 - name: checksum
 - type: string

Limit output to a single table:

```
sudo -E -u www-data php occ db:schema:export oc_filecache
```

Use `--sql` to output SQL `CREATE TABLE` statements instead of the default structured format. Use `--output=json_pretty` for machine-readable output.

db:schema:expected

Export the expected schema for a fresh installation of the current version. Compare this against `db:schema:export` to identify divergences caused by incomplete upgrades or manual changes:

```
sudo -E -u www-data php occ db:schema:expected
sudo -E -u www-data php occ db:schema:expected oc_filecache
```

Supports the same `--sql` and `--output` options as `db:schema:export`.

Data conversions

db:convert-filecache-bigint

Convert the `filecache` table's ID columns from integer to BigInt. Required on large installations where file IDs have exceeded the integer limit. This command is non-destructive but may take time on large databases:

```
sudo -E -u www-data php occ db:convert-filecache-bigint
```

db:convert-mysql-charset

Convert all MySQL or MariaDB tables to use `utf8mb4` (full Unicode, including emoji). Required for proper Unicode support. Run this once after switching to or upgrading MySQL/MariaDB:

```
sudo -E -u www-data php occ db:convert-mysql-charset
```

db:convert-type

Convert the Nextcloud database from SQLite to MySQL, MariaDB, or PostgreSQL. SQLite is suitable for testing and single-user setups, but production servers with multiple users should use one of the other supported databases.

Requirements:

- The target database and its PHP connector must be installed.
- Login credentials for a database admin user.
- The database port number, if non-standard.

This example converts from SQLite to MySQL/MariaDB:

```
sudo -E -u www-data php occ db:convert-type mysql oc_dbuser 127.0.0.1 oc_database
```

For a detailed walkthrough see [./configuration database/db conversion](https://github.com/cocobooks/cocobooks/blob/master/docs/development/development.md#database-conversion).

#3

cocobook-sync 2026-10-28 16:27:21 UTC

cocobook-sync 2026-10-29 01:09:17 UTC