Software by Design
Nextcloud apps

Tours · Visits · Care documentation

CareCheck

User manual

End-user documentation for the web app

Ambulatory care on your Nextcloud: tours, clients, tamper-evident log.

App version	Language	Date
0.4.3	**English**	**November 15, 2026**

Contents

Introduction 1
1 Roles and permissions 2
2 Getting started 3
3 Coverage board 5
4 Tours and tour templates 7
5 Clients and the client record 8
6 The care plan 11
7 Client log, photos and voice notes 12
8 Exports, billing and imports 14
9 Members, teams and devices 15
10 Inbox, review queue and notices 17
11 The task catalog 19
12 Settings 20
13 Access, allow lists and teams 25
14 The companion app (CareCheck Mobile) 26
15 Errors, troubleshooting and FAQ 28
16 Accessibility 29
17 Glossary 30
18 Surface index — where things live 31
About Software by Design 32
Support & how to contribute 32
Index 34

Introduction

CareCheck is the Nextcloud app for ambulatory care documentation on your own instance: tour templates that materialize into daily tours, offline-capable visit bundles with plan snapshots and signatures, a tamper-evident hash-chained client log with external timestamp anchoring, and audited break-glass cover access. The web app is free (AGPL); the companion app CareCheck Mobile needs a licence (CCK2) and is scoped in chapter 14.

Status: app version **0.4.3** (commit `32b30fe`), companion **0.1.1**, November 15, 2026, English · Web AGPL.

Atlas farm dual ACCEPT; screenshots captured against the live web app with seeded demo data.

Audience: head office/dispatch · care staff · billing · app admins.

You need: CareCheck access on the Nextcloud · a browser with JavaScript · admin rights for **‘Settings’**.

Chapters: 1–2 roles/getting started · 3–10 daily work · 11–13 organisation/admin · 14 companion · 15–18 errors, accessibility, glossary, index.

Register: formal; UI examples **‘Coverage board’**, **‘Publish plan’**, **‘Pair code’** from the EN translations.

Quick start (dispatch): **‘Catalog’** and **‘Members’** → **‘Clients’** with a care plan → publish a **‘Tours’** template → **‘Generate tours now’** → **‘Coverage board’**.

1 Roles and permissions

CareCheck separates *who may open the app* (access), *what they may do inside it* (role) and *which care tasks they are allowed to perform* (qualification). These three layers answer different questions, and confusing them is the most common source of “why can’t this person see X?” tickets. Your own role is always printed at the bottom of the left sidebar — for example **‘App admin’** — so you can check at a glance which hat you are wearing.

1.1 The four access roles

App admin Every Nextcloud administrator, plus every account listed under **‘Delegated app admins’** on the **‘Access’** page (chapter 13). App admins see and change everything: client records, tours, members, the catalog and all seven **‘Settings’** sections including **‘License’**, **‘Access’**, **‘Privacy & register’** and **‘Transfer’**.

Head office The dispatch desk (internally the “Zentrale” role). Head office users plan tours, assign uncovered stops, work the **‘Inbox’** and the **‘Review queue’**, and maintain clients, members and the catalog. They see the **‘Settings’** sections that configure day work but not the access control page.

Accounting The billing role (“Buchhaltung”). Accounting reads client records and produces the monthly exports on the **‘Exports’** tab — **‘Service record (CSV)’**, **‘Service record (PDF)’** and the **‘Billing data (JSON)’**/**‘Billing data (XML)’** files used for invoicing (chapter 8).

Member Care staff without a lead function. A member sees their own tours and the clients assigned to them and documents visits through the companion app; the web views they cannot use stay hidden.

	Member	Head office	Accounting	App admin
Own tours & assigned clients	•	•	—	•
All clients, members, catalog	—	•	read	•
Inbox & review queue	—	•	—	•
Monthly exports	—	•	•	•
Settings: day work	—	•	—	•
Settings: **‘License’**, **‘Access’**, **‘Privacy & register’**, **‘Transfer’**	—	—	—	•

1.2 Teams and team leads

Larger providers split their staff into *teams* — for example *Team Nord* for the northern district. A team lead only sees the clients and tours of the teams they lead; the head office keeps full visibility across every team. Teams are created and staffed on **‘Settings’** → **‘Access’** (chapter 13); a client is attached to a team on its **‘Record’** tab and a member through the member list.

1.3 Professional qualification

Every member carries one of three qualifications — **‘Care assistant’**, **‘Caregiver’** or **‘Qualified caregiver’** — and every catalog task and every client may demand a **‘Required qualification’**. The app enforces this at assignment time: a stop that requires a qualified caregiver cannot be given to a care assistant, so a typo in planning cannot become a compliance violation in the field.

1.4 The access door

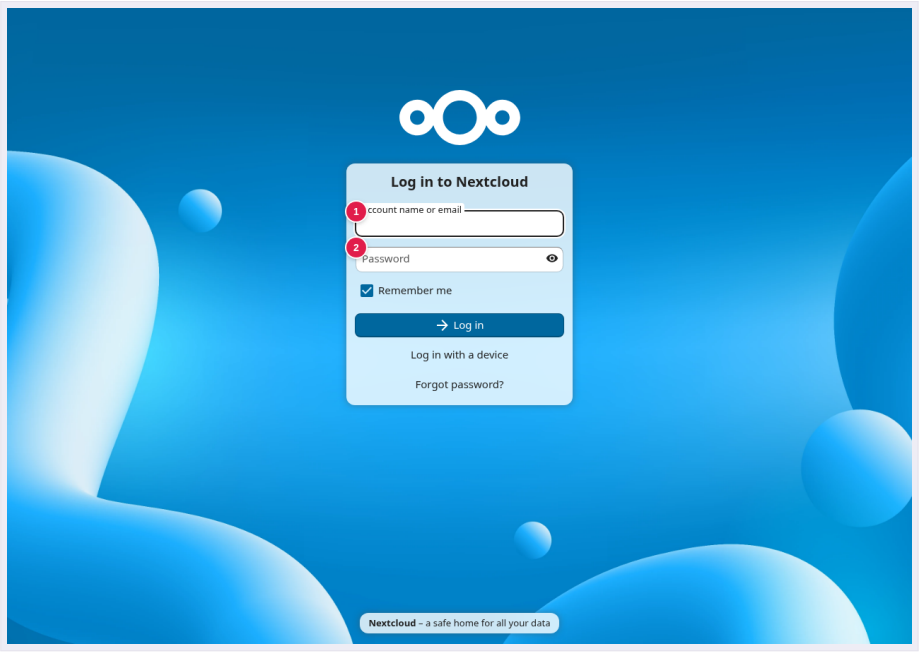
Before any role applies, the *access mode* decides who may open CareCheck at all: **‘Open — every Nextcloud user’** or **‘Restricted — allow list below’**. In restricted mode an account that is not on the user or group allow list sees **‘No access.’** and nothing else. Configuration lives in chapter 13.

2 Getting started

This chapter takes you from the Nextcloud sign-in page to your first planned day. It assumes an empty or freshly seeded workspace; if your dispatchers already set everything up, skim it for orientation and jump straight to the chapter that matches your job.

2.1 Signing in

Open your Nextcloud address in the browser, type your user name into the first field (marker 1) and your password into the second (marker 2), then submit the form. You land on the Nextcloud dashboard; CareCheck lives in the *Apps* menu as *CareCheck*. The interface language follows the language set in your personal Nextcloud profile — the app itself has no separate language switch.

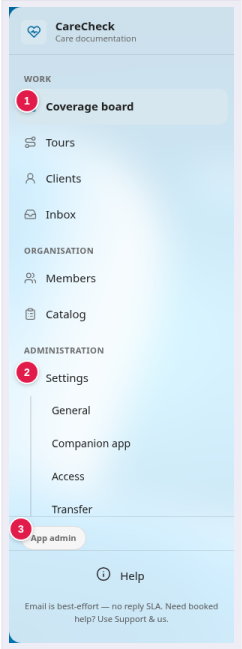


Nextcloud sign-in: the user name field (marker 1) and the password field (marker 2).

If you see **‘No access.’** instead of the app, the workspace runs in restricted mode and your account is not on the allow list. That is not a password problem — ask an app admin to add you under **‘Settings’** → **‘Access’** (chapter 13).

2.2 Finding your way around

The sidebar organises CareCheck into three groups. *Work* holds the daily surfaces: **‘Coverage board’**, **‘Tours’**, **‘Clients’** and **‘Inbox’** — the inbox carries a small badge with the number of items waiting for a decision. *Organisation* holds **‘Members’** and **‘Catalog’**. *Administration* holds **‘Settings’** with its five sub-pages and **‘Get the app’**, the companion-app page.



The sidebar: the work area with **‘Coverage board’** (marker 1), the **‘Settings’** entry with its sub-pages (marker 2) and the footer with your role badge and the help menu (marker 3).

Your role is printed in the sidebar footer (marker 3) so you always know which permissions apply to you. On narrow screens the sidebar collapses behind the **‘Menu’** button at the top. Keyboard users get a **‘Skip to content’** jump link as the very first tab stop — chapter 16 covers the details.

2.3 The four-step setup

While the workspace is still empty, the **‘Coverage board’** shows a **‘Getting started’** checklist instead of the day view. It encodes the only order that works, because each step feeds the next one:

1. **‘Add catalog tasks’** — the service catalogue is the vocabulary every care plan and tour stop is built from (chapter 11).
2. **‘Add staff’** — members carry the qualification that decides who may execute which stop (chapter 9).
3. **‘Add first client’** — a client record with a published care plan (chapters 5 and 6).
4. **‘Create first tour’** — a published weekly template that materialises into daily tours (chapter 4).

2.4 Your first planned day — a worked example

Suppose you run the dispatch desk of a small provider. You want tomorrow morning covered for your client Johanna Meier. The full path looks like this:

1. In **‘Catalog’**, add a task with **‘Code’** `body-care` and **‘Label’** *Personal hygiene*, requiring a **‘Qualified caregiver’**.
2. In **‘Members’**, add *Anna Bergmann* with the qualification **‘Qualified caregiver’**.
3. In **‘Clients’**, add *Johanna Meier* with her street and open her record. On the **‘Care plan’** tab, tick the *Personal hygiene* card and press **‘Publish plan’** — the companion only ever sees published plans.
4. In **‘Tours’**, open the **‘New tour template’** form. Name it *Morgentour Nord*, tick tomorrow’s weekday, set the status to **‘Published’** and add a stop: client Johanna Meier, a morning time window, executor Anna Bergmann. Save with **‘Save tour template’**.
5. Back on the **‘Coverage board’**, press **‘Generate tours now’**. The board fills with tomorrow’s tour card and the **‘Stops planned’** counter moves up — your first planned day exists.

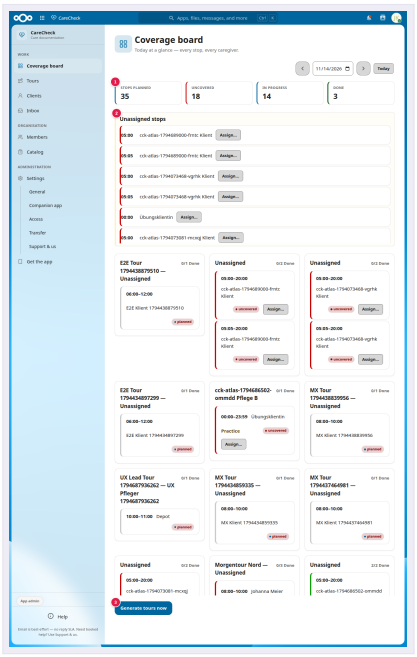
From here the day runs itself: the background job materialises future days overnight, Anna documents the visit on her phone, and the stop on the board walks through **‘planned’**, **‘in progress’** and **‘done’** without you touching it again.

3 Coverage board

The **‘Coverage board’** is the dispatch home page and the answer to the question every morning starts with: *is every visit covered?* It shows one day at a time — every stop that the published templates materialised, who executes it, and how far it has progressed.

3.1 Picking a day and reading the numbers

The date row under the page title offers **‘Previous day’** and **‘Next day’** arrows, a date field labelled **‘Board date’** for jumping to an arbitrary day, and **‘Today’** to return. Directly below, four tiles (marker 1) count the day: **‘Stops planned’**, **‘Uncovered’**, **‘In progress’** and **‘Done’**. The uncovered tile is deliberately red — a number above zero there means someone has to act before the day can be released.

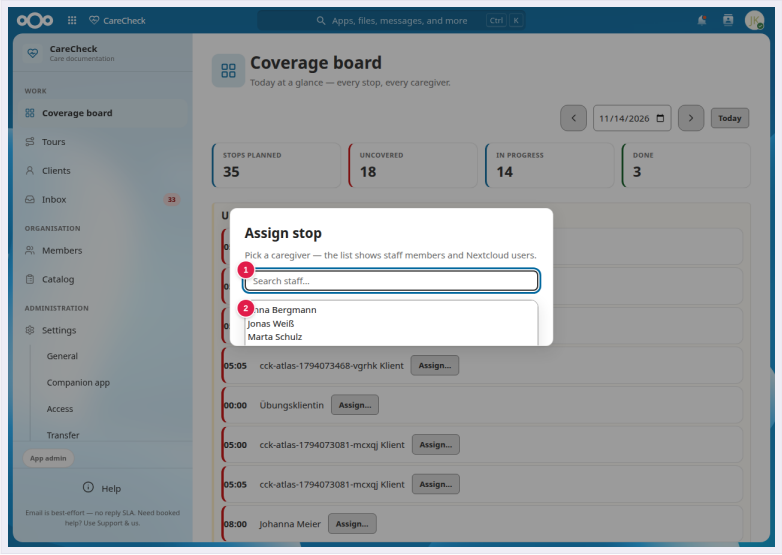


The **‘Coverage board’**: the four counters of the **‘Day overview’** (marker 1), the **‘Unassigned stops’** warning block (marker 2) and **‘Generate tours now’** (marker 3).

Below the counters the board lists each daily tour as a card with its stops. Every stop shows its time window, the client and its status — **‘planned’**, **‘uncovered’**, **‘in progress’** or **‘done’** — so a late-running tour is visible before the phone rings.

3.2 Unassigned stops and the assign dialog

A template stop without an executor lands in the **‘Unassigned stops’** warning block (marker 2). This happens when a template leaves the executor open on purpose — a floating shift — or when someone calls in sick. Each entry offers an **‘Assign…’** button that opens the **‘Assign stop’** dialog (marker 1 in the figure below).



The **‘Assign stop’** dialog: the directory picker listing staff members and Nextcloud accounts (marker 1) and the **‘Assign’** confirmation button (marker 2), which stays disabled until a person is picked.

Work through the dialog like this:

1. Open **‘Assign…’** on the uncovered stop.
2. Type a name in the picker or scroll the list — it contains staff members and Nextcloud accounts, and group entries are marked.
3. Pick a person whose **‘Qualification’** covers the stop’s requirement; the dialog shows the qualification next to each name.
4. Confirm with **‘Assign’**. The button only becomes active once a person is selected (marker 2) — there is no way to assign “nobody”.

The board re-reads itself and the stop moves out of the warning block into its tour card with the new executor. The companion app of that person picks the stop up on the next sync — nothing else needs to be sent.

3.3 The on-duty strip

When the sibling app DutyCheck is installed and its roster sharing is enabled, the board shows an **‘On duty (DutyCheck)’** strip between counters and tour cards with today’s published shifts: name, time window and location. **‘Open DutyCheck roster’** jumps straight into the duty plan. Stops whose executor is a linked Nextcloud account that is not on today’s roster are flagged *not on duty*, so the dispatcher sees the collision before the caregiver does. Without DutyCheck — or with the opt-in switched off — the strip simply does not render and the board works unchanged.

3.4 Generating tours

‘Generate tours now’ (marker 3) materialises every **‘Published’** template for today plus the next seven days — the equivalent of what the nightly background job does automatically. That horizon comes from the server-side `horizon_days` value (seven by default; it is a deployment setting, not a field on the **‘Settings’** pages), and the delete confirmation on a template names the same window. Press it after you publish or change a template and want the coming days updated immediately. If the instance cron is stopped, a red banner warns *Background jobs are not running — generated tours may be missing. Check the instance cron setup.* The fix is on the server side (chapter 15); until then the manual button keeps the days coming.

3.5 Edge cases and limits

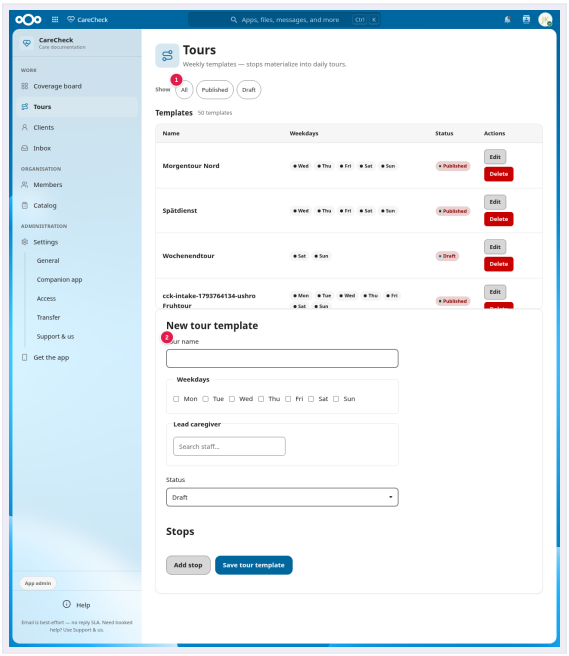
- A person marked absent in **‘Members’** still appears on the board — absence is information, not a lock. Reassign their stops via **‘Assign…’** (chapter 9).
- Past days stay readable: journal and exports keep the evidence; the board is a planning tool, not the audit trail.
- The board only ever shows materialised tours. A **‘Draft’** template produces nothing — check chapter 4 if a day looks suspiciously empty.

4 Tours and tour templates

CareCheck never schedules single visits by hand. You describe the week once as a *tour template* — a set of weekdays plus the ordered stops with their time windows — and the app materialises a concrete daily tour for every matching day. Planning therefore happens in two places that answer different questions: **‘Tours’** holds the templates (“what repeats every week?”), the **‘Coverage board’** holds the days (“who actually goes out tomorrow?”, chapter 3).

4.1 The template list

The **‘Tours’** view opens with the quick filter **‘Show’** (marker 1): **‘All’**, **‘Published’** or **‘Draft’**. Under **‘Templates’** each row lists the template’s **‘Name’**, its **‘Weekdays’** and its **‘Status’**; the **‘Actions’** column offers edit and delete. Deleting a template stops future materialisation but leaves every tour that was already generated untouched — yesterday’s documentation never disappears because someone cleaned up a template. The confirmation names the keep-alive window it spares: tours for today and the next *n* days, where *n* is the configured generation horizon (chapter 3).



The **‘Tours’** view: the status quick filter (marker 1) above the template table, and the **‘New tour template’** form below (marker 2).

4.2 Creating a template

The **‘New tour template’** form (marker 2) collects everything a weekly tour needs:

1. Give the template a **‘Tour name’** — pick something the day staff will recognise on the board, like *Morgentour Nord* or *Spätdienst*.

2. Tick the **‘Weekdays’** the tour should run. A template with Monday and Friday checked produces a daily tour on both days.
3. Pick the **‘Lead caregiver’** responsible for the tour as a whole and set **‘Status’** to **‘Draft’** while you are still shaping it.
4. Add stops with **‘Add stop’**: each stop takes a client, a time window (start and end in minutes past midnight — the form shows the resulting clock times) and an executor. Leave the executor empty to create a floating stop the dispatcher assigns daily on the board.
5. Save with **‘Save tour template’**. While the status stays **‘Draft’** nothing is scheduled; flip it to **‘Published’** when the template is ready to produce tours.

Editing a published template re-opens the same form with the values filled in. Changes apply to days materialised *after* the save — tours already generated keep their snapshot, because the field staff may already be working them.

4.3 Why templates and days are separated

The split protects you from a classic scheduling trap: if editing Monday’s tour silently rewrote every future Monday, a one-off change for a public holiday would corrupt the whole month. Instead, the template is the *rule* and the generated tour is the *instance*. The nightly job (or **‘Generate tours now’**) reads the rule and writes instances; exceptions get fixed on the board, not on the template.

4.4 The practice tour

Every member can receive a *practice tour* — a private sandbox tour created from the **‘Practice tour’** action in the member list (chapter 9). It contains synthetic stops with no real client data, so a new colleague can rehearse the whole companion-app flow — check-in, tasks, signature, sync — without leaving a trace in anyone’s care documentation. Use it for onboarding before you hand out real stops.

4.5 Edge cases and limits

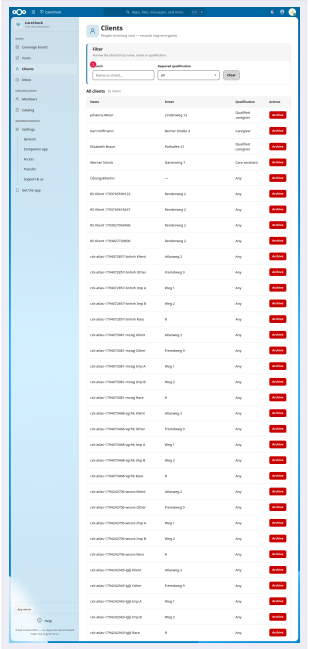
- A stop’s executor must satisfy the client’s **‘Required qualification’**; the form will not let you save an under-qualified assignment.
- Time windows are per-stop, not per-tour: stagger them so a caregiver is never expected in two places at once.
- Archiving a client prunes their stops from every template automatically, and generation skips stops whose client is no longer active either way — a template left without any eligible stop simply produces no tour for that day. Already-generated tours keep their visits for the record (chapter 5).

5 Clients and the client record

Clients are the people you care for. The **‘Clients’** view is their register; each record opens into five section tabs — **‘Record’**, **‘Care plan’**, **‘Client log’**, **‘Photos’** and **‘Exports’** — of which this chapter covers the register and the record. The remaining tabs have their own chapters because they answer different questions (chapters 6, 7 and 8).

5.1 The register and its filter

The filter panel (marker 1) narrows the table of **‘All clients’** by two inputs: a free-text **‘Search’** over name and street (the placeholder **‘Name or street…’** says exactly that) and a **‘Required qualification’** dropdown. **‘Clear’** resets both. The table itself shows **‘Name’**, **‘Street’**, **‘Qualification’** and **‘Actions’** for every row.



The **‘Clients’** register: the filter panel (marker 1) above the list of all clients.

5.2 Adding a client

The **‘Add client’** form asks for three things: **‘Name’**, **‘Street’** and **‘Required qualification’**. That is deliberately minimal — a record exists so the day can be planned, and everything clinical lives on the tabs afterwards. Client records are stored encrypted; the page header states it plainly because it is the kind of promise a reader should not have to take on faith.

5.3 The Record tab

Open a client and the **‘Record’** tab (marker 1) holds the fields dispatch and care staff actually need:

‘Name’, ‘Street’ identity and where the visit happens.

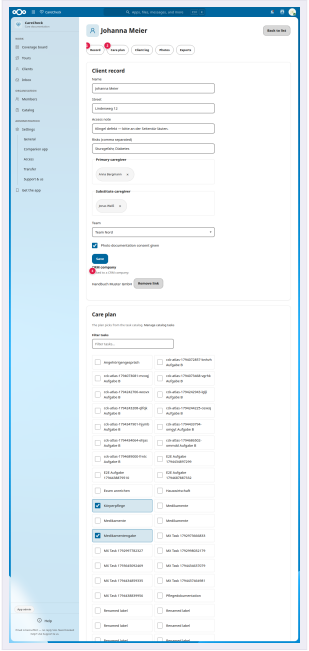
‘Access note’ practical entry help — a broken doorbell, a hidden side door, a key safe code hint. Field staff see this in the companion before they knock.

‘Risks (comma separated)’ flagged hazards such as *Sturzgefahr* or *Diabetes*; they travel with the visit bundle so nobody learns about a risk at the bedside.

‘Primary caregiver’ / ‘Substitute caregiver’ the continuity pair: who normally visits, and who covers.

‘Team’ the team assignment that scopes what team leads see (chapter 9).

‘Photo documentation consent given’ the consent gate that unlocks the **‘Photos’** tab — without it, photo documentation stays locked by design (chapter 7).



A client record with the section chips **‘Record’** (marker 1) and **‘Care plan’** (marker 2) and the **‘CRM company’** link block (marker 3).

‘Save’ writes the form against the revision you loaded. If a colleague saved the same record while you were editing, the app answers with a conflict instead of silently overwriting their work — reload, re-apply your change, save again. This optimistic locking is why two dispatchers can safely work the same register (chapter 15).

‘Archive’ retires a client: the pickers no longer offer them for new stops, while every past tour, journal entry and export stays readable. Archiving also removes the client’s stops from every tour template in the same step, so no fresh visits are planned for someone who left care — and even a stop that somehow stayed behind is skipped at generation time, because only stops with an active client materialise. Already-generated tours keep their visits for the record. Care documentation is never deleted by archiving — that is a legal requirement, not a feature option.

5.4 The CRM company link

When the sibling app CustomerCheck is installed, the record shows a **‘CRM company’** block (marker 3): the billing payer behind the client. App admins attach a company through **‘Link a company’** with the **‘Search companies…’** picker and detach it with **‘Remove link’**. The link is what the invoice-draft push on the **‘Exports’** tab needs (chapter 8); if the linked company disappears from CustomerCheck, the stale link stays visible so nothing silently loses its billing context.

5.5 Importing clients

For an initial migration, CareCheck offers a two-step CSV import on the API: a preview call parses the file and marks duplicate names without writing anything, and a commit call creates the records that survived the check. There is deliberately no drag-and-drop upload box in the web UI — bulk imports are an admin job run once, not a daily surface. Ask your administrator to run the import endpoints (`/api/clients/import/preview`, then `/api/clients/import`) from a script; the preview report names every row it would skip, so nothing enters the register unseen.

5.6 Edge cases and limits

- Duplicate names are allowed — two clients can share a name. Use street and qualification to tell them apart, and the client id inside the URL when in doubt.

- An archived client keeps its team and CRM links for the record; reactivation is a new record, not an un-archive, so history never forks.
- The register search matches substrings case-insensitively — *meier* finds *Johanna Meier*.

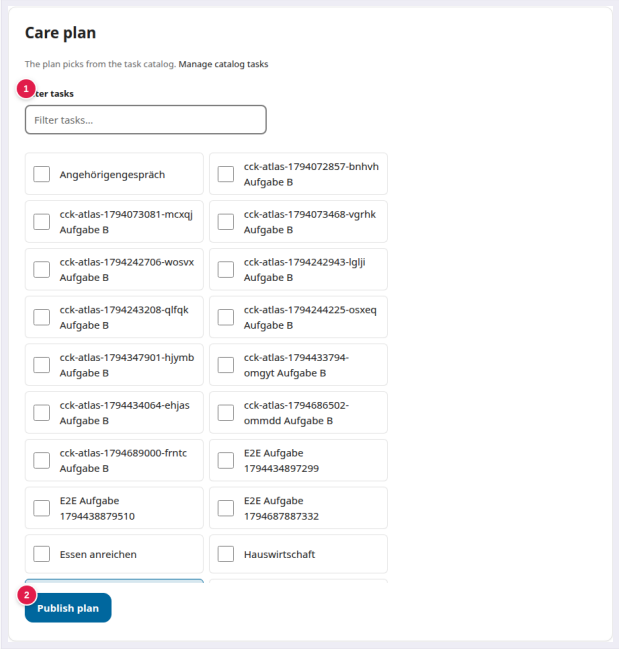
6 The care plan

A client's **'Care plan'** tab defines *what may be documented* on a visit: the set of catalog tasks the field staff sees in the companion, plus whether a task demands a measured value. The plan is not free text — it is a pick-list built from the **'Catalog'** (chapter 11), which is what makes visit documentation comparable across staff and months.

6.1 Building the plan

The tab lists every catalog task as a card, sorted alphabetically. Tick the cards that belong to this client's care; cards that require a measured value (blood pressure, temperature, blood sugar) can be marked so the companion insists on a number instead of a bare checkmark. The hint line above the list — *The plan picks from the task catalog.* — links to **'Manage catalog tasks'** so a missing task is one click away from being created.

From eight catalog tasks on, a **'Filter tasks'** box (marker 1) appears and narrows the card list as you type. The filter only changes what you *see* — ticks outside the current filter stay set, so searching can never silently deselect a task you picked earlier.



The **'Care plan'** tab: the **'Filter tasks'** box (marker 1), the task cards and **'Publish plan'** (marker 2).

6.2 Publishing — and why it is a snapshot

'Publish plan' (marker 2) does not just save the tick list; it creates a new, immutable *plan snapshot* with its own revision number. The companion app always documents against the latest published snapshot, never against a half-edited draft. That split is what makes the documentation defensible later: a visit bundle recorded on Tuesday still names the plan revision it was measured against, even if the plan was re-published on Wednesday.

Concretely:

1. Adjust the tick list and measured-value flags.

2. Press **‘Publish plan’**.
3. The app stores the snapshot and tells the companion on its next sync; field staff simply see the new task list appear.

Because every publish is a new revision, there is nothing to “undo” — publish again with the correct set and the newest snapshot wins. Like the record form, publishing is guarded by optimistic locking: if someone else published while you were editing, you get a conflict to resolve rather than a silent overwrite.

6.3 Edge cases and limits

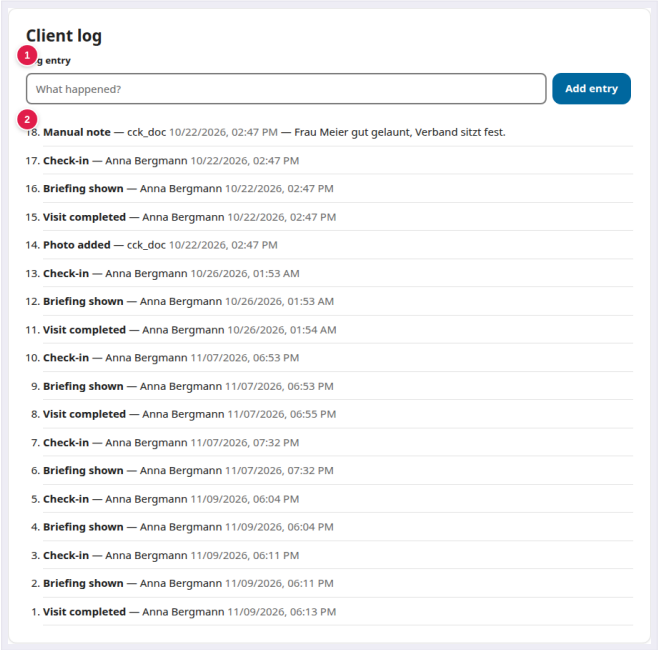
- A client without a published plan shows no tasks in the companion — check this first when a caregiver reports an empty visit screen.
- Removing a task from the plan does not touch past documentation; the journal and exports keep the historical entries under the code that was valid then.
- Measured values are mandatory where flagged: the companion cannot submit the task without a number, which is exactly the point of the flag.

7 Client log, photos and voice notes

The **‘Client log’** tab is the client’s chronology: what happened, when, and who caused it. Together with the **‘Photos’** tab it forms the documentation the whole app exists to protect — so this chapter explains not just how to write entries, but why the log is built the way it is.

7.1 The hash-chained client log

Every entry in the **‘Client log’** is part of a *hash chain*: each new entry stores the fingerprint of its predecessor, so changing or deleting a past entry breaks the chain and is detectable. Entries are append-only — there is no edit button and no delete button, by design. A wrong note is corrected by a new entry, exactly like a paper journal you may not tear pages out of.



The **‘Client log’**: the **‘Log entry’** form for manual notes (marker 1) above the reversed chronology — newest first (marker 2).

Write a manual note by typing into **‘Log entry’** (marker 1) and pressing **‘Add entry’**. The list below (marker 2) shows the newest entry first. Next to your manual notes the chain also records the automatic events the system writes itself: **‘Visit completed’**, **‘Check-in’**, **‘Stop skipped’**, **‘Visit rejected’**, **‘Photo added’**, **‘Voice note added’**, **‘Export created’**, group-session openings and closures, and digest deliveries. Because machine and human entries share one chain, the chronology is complete by construction — nothing “important” can hide in a parallel log.

Optionally, an external timestamp authority (TSA) anchors the chain head on a schedule, producing independent proof that the log existed in this state at that time. The anchor endpoint is configured under **‘Settings’** → **‘Companion app’** (chapter 12); anchoring is additive — the chain works without it, it is just stronger with it.

7.2 Photo documentation

The **‘Photos’** tab holds clinical photo documentation — wound progression being the classic case. It is deliberately locked behind consent: until **‘Photo documentation consent given’** is ticked on the **‘Record’** tab, the panel shows only the hint that photo documentation requires the client’s consent. Once consent is recorded, **‘Add photo’** (marker 1) accepts JPEG, PNG and WebP files, and the grid (marker 2) shows the history. Every upload and every deletion writes its own journal entry — photo evidence never appears or vanishes silently.



The consent-gated **‘Photos’** tab: the upload control (marker 1) and the photo grid (marker 2).

Deleting a photo removes the image but leaves the **‘Photo deleted’** journal entry behind, so the documentation trail admits what happened rather than pretending the photo never existed.

7.3 Voice notes

Care staff can dictate short voice notes in the companion app instead of typing at the doorstep. Voice notes arrive through the same visit bundle, land under the client’s documentation and write a **‘Voice note added’** journal entry; where server-side transcription is available, **‘Voice note transcribed’** follows. They are part of the chain like everything else.

7.4 Edge cases and limits

- The log is chronological evidence, not a chat: keep manual entries factual (“Verband sitzt fest, Frau Meier gut gelaunt”), not conversational.
- There is no per-entry access control — everyone who can open the client can read the whole chain. Scope access at the client or team level instead (chapters 5 and 13).

- Consent covers photo *documentation*; it does not replace your organisation's own consent paperwork and retention policy — **'Retention (years)'** in the settings is the technical half of that.

8 Exports, billing and imports

Documentation only earns its keep when it leaves the app in the shape the next process needs: a signed service record for the care file, structured billing data for accounting, an invoice draft for InvoiceCheck. The **'Exports'** tab on the client record is where all of that happens, scoped to one month at a time.

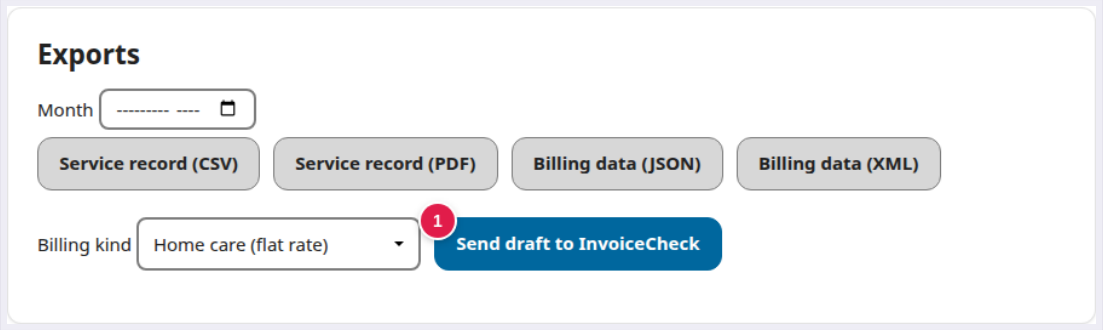
8.1 Monthly exports

Pick a **'Month'** in the period field, then choose the output you need:

'Service record (CSV)' the performed visits and tasks as a spreadsheet-ready table — for archives, audits and ad-hoc evaluation.

'Service record (PDF)' the same month as a formatted, printable record — the document that goes into the care file.

'Billing data (JSON)' / 'Billing data (XML)' the machine-readable month: per-visit tasks, times and signatures in the structure your billing system or the payer's portal expects.



The **'Exports'** tab: the month picker with the four export buttons and, when InvoiceCheck is enabled, the invoice-draft push (marker 1).

Every export writes an **'Export created'** entry into the client log — so the documentation itself records who produced which evidence and when.

8.2 Pushing a draft to InvoiceCheck

When the sibling app InvoiceCheck is enabled, the tab gains a second row: pick a **'Billing kind'** — **'Home care (flat rate)'**, **'Care visit'**, **'Relief amount § 45b SGB XI'**, **'Care assistance SGB XII'** or **'Private additional service'** — and press **'Send draft to InvoiceCheck'** (marker 1). CareCheck packages the month's signed visits into a draft invoice and hands it to InvoiceCheck, where accounting reviews and sends it. The status line under the button reports the result.

The push is idempotent by design: repeating it for the same month and kind never creates a second draft — the answer is *Draft already exists in InvoiceCheck — nothing duplicated.* Two preconditions are checked first and named in the status line when missing: the client needs a linked **'CRM company'** (chapter 5), and that company needs a billing customer in InvoiceCheck.

8.3 Importing clients at go-live

Chapter 5 describes the API-side CSV import — preview first, then commit — which exists for the one-time migration of an existing client register. There is intentionally no recurring import surface: after

go-live, data flows *into* CareCheck through tours and visit bundles, and *out* of it through the exports above.

8.4 Edge cases and limits

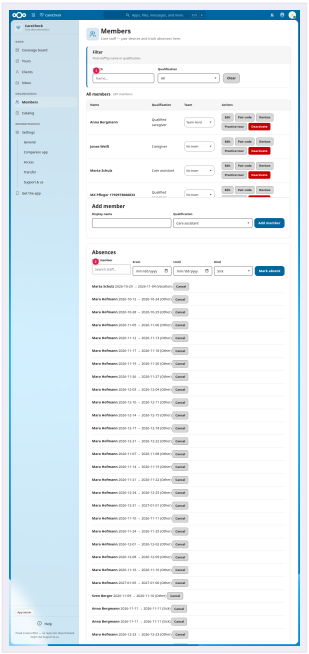
- Exports are monthly by design — a quarter is three exports. The journal keeps each one attributable.
- A month without signed visits produces an empty but valid export; absence of documentation is itself documented.
- Billing data carries exactly what the visits recorded — measured values, task states and signatures — nothing is recalculated or “improved” at export time, because the export is evidence.
- The invoice push does not send money anywhere: it creates a *draft*. A human in InvoiceCheck still reviews every invoice before it goes out.

9 Members, teams and devices

The **‘Members’** view is the staff register. A *member* is a care worker the app knows — with or without a Nextcloud account — and the row for each member is where their devices, absences and practice tours are managed. Teams, which scope what team leads see, are created on the **‘Access’** page and summarised here as well.

9.1 The member list

The filter panel (marker 1) searches by name or narrows to a **‘Qualification’**. Each row offers **‘Edit’**, **‘Pair code’**, **‘Devices’**, **‘Practice tour’** and **‘Deactivate’**. The **‘Add member’** form takes a **‘Display name’** and the qualification — **‘Care assistant’**, **‘Caregiver’** or **‘Qualified caregiver’** — that decides which stops and tasks this person may execute (chapter 1).

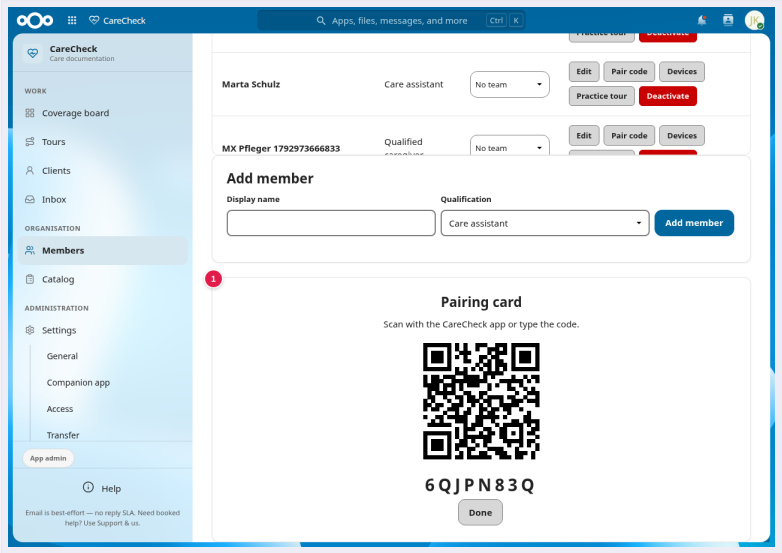


The **‘Members’** view: the filter (marker 1) above the member table with its per-row actions, and the **‘Absences’** form below (marker 2).

‘Deactivate’ ends a member’s access without erasing their history: paired devices are unpaired immediately, and all tours, journal entries and exports they produced stay attributable. Never “delete and re-add” a person who left — deactivation keeps the documentation chain honest.

9.2 Pairing a device

‘Pair code’ opens the **‘Pairing card’** (marker 1 in the figure): a QR image plus a plain-text code. On the phone, the companion app offers *Scan with the CareCheck app or type the code.* — one scan or eight typed characters pair the device to exactly this member. **‘Done’** closes the card.

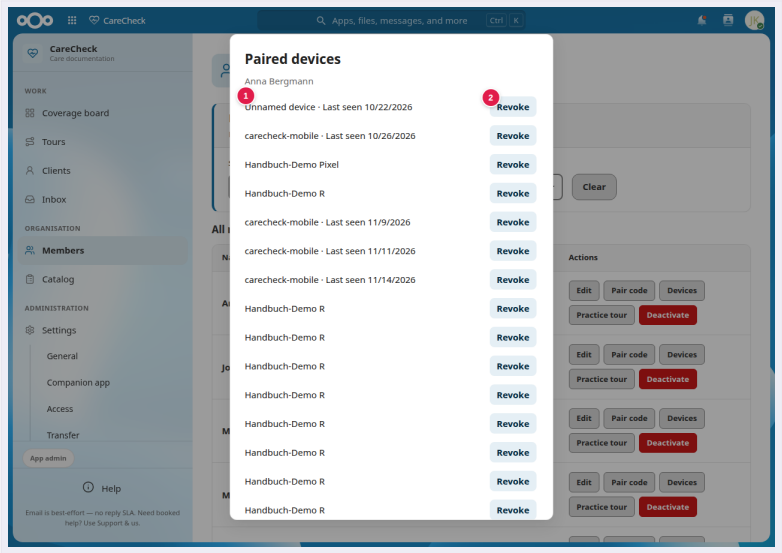


The **‘Pairing card’** with the QR image and the code (marker 1).

Treat a pair code like a password: it is a one-time credential that converts into a device token. A paired device is its own identity — deactivating the member cuts its access immediately, server-side, without needing the phone in your hand. Devices without their own Nextcloud account always pair by code; staff with an account may alternatively sign in inside the app (chapter 14).

9.3 Revoking a single device

‘Devices’ on a member row opens the **‘Paired devices’** dialog (the figure): every live device on that member — its label or **‘Unnamed device’**, plus the **‘Last seen’** date (marker 1). **‘Revoke’** (marker 2) asks *Revoke this device? It loses access on the next call.* and then retires exactly that one token: the member stays active, their other devices keep syncing, and the lost phone fails on its next call. A member who never paired shows **‘No paired devices.’**



The **‘Paired devices’** dialog for Anna Bergmann: each device row (marker 1) carries its own **‘Revoke’** (marker 2).

Two scopes, deliberately separate: **‘Revoke’** cuts one lost phone, **‘Deactivate’** unpairs *every* device at once — reach for revoke when the phone is gone, for deactivation when the person leaves. Revoking is final for that token; the phone needs a fresh **‘Pair code’** to come back.

9.4 Practice tours

‘Practice tour’ creates a private sandbox tour for the member — synthetic stops, no real client data — so new staff can rehearse check-in, tasks and signatures before their first real day (chapter 4).

9.5 Absences

The **‘Absences’** form (marker 2) records who is away: pick the **‘Staff member’**, set **‘From’** and **‘Until’**, choose a **‘Kind’** — **‘Sick’**, **‘Vacation’** or **‘Other’** — and press **‘Mark absent’**. The list under the form shows current and upcoming absences and lets you cancel them.

Absence is information for the dispatcher, not a lock: an absent person’s stops still need reassigning on the **‘Coverage board’** (chapter 3). When ArbeitszeitCheck is installed, its absence sync feeds the same list, so vacation approved there shows up here without double entry.

9.6 Teams

Teams partition both staff and clients — *Team Nord* covers the northern district, and its leads only see their own tours and clients. Teams themselves are created under **‘Settings’** → **‘Access’**, where you also appoint and remove team leads (chapter 13); a member joins a team through the member record, a client through the **‘Team’** field on its record. The head office always keeps full visibility — teams scope the leads’ view, never the dispatcher’s.

9.7 Edge cases and limits

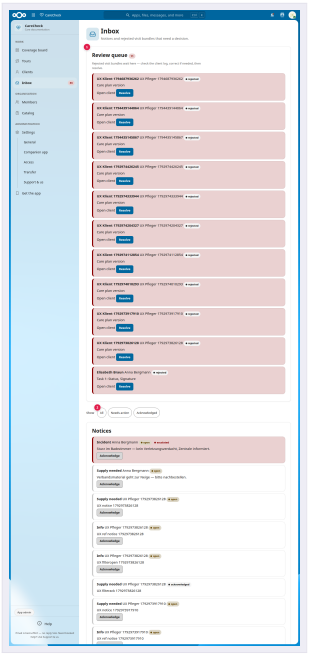
- A member does not need a Nextcloud account — pair-code devices work for staff who exist only as members.
- Re-pairing after a lost phone is a fresh **‘Pair code’** on the same member — but a new code does not retire the old device: revoke the lost one under **‘Devices’** (or deactivate the member to cut all devices at once), so report losses to the office immediately instead of quietly re-pairing.
- Qualification changes apply to future assignments; past documentation keeps the qualification recorded at the time.

10 Inbox, review queue and notices

The **‘Inbox’** is where everything lands that needs a human decision: messages from the field, and visit bundles the server refused to file. The badge next to **‘Inbox’** in the sidebar counts the open items, so a glance at the navigation tells you whether anything is waiting.

10.1 The review queue

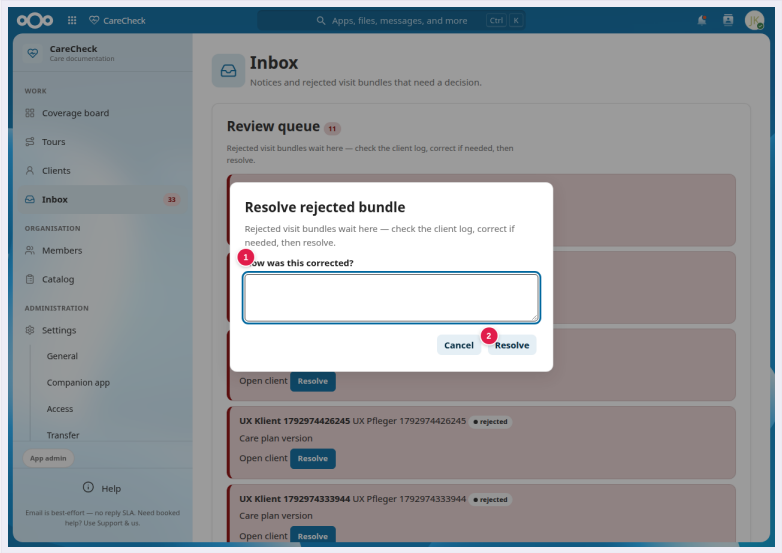
At the top sits the **‘Review queue’** (marker 1): visit bundles the companion sent that the server could not accept — an invalid signature, a task status that does not exist, a bundle for the wrong client. Rejection is a safety feature, not an error to fear: the data is kept here for a human decision instead of being silently dropped or, worse, silently written into the care record.



The **‘Inbox’**: the **‘Review queue’** with its badge (marker 1) above the **‘Notices’** and their filter (marker 2).

Work a queued bundle like this:

1. Read the rejection reason on the queue row.
2. Open the linked client via **‘Open client’** and check the **‘Client log’** — the rejected bundle usually has a twin story to tell.
3. Clarify the facts with the caregiver if needed, then press **‘Resolve’**. The **‘Resolve rejected bundle’** dialog (marker 1 below) asks **‘How was this corrected?’** — a short note that becomes part of the audit trail.
4. Confirm with **‘Resolve’** (marker 2). The case closes auditably; nothing is rewritten, the resolution is appended.



The **‘Resolve rejected bundle’** dialog: the correction note (marker 1) that becomes part of the audit trail, and the **‘Resolve’** confirmation (marker 2).

10.2 Notices

Below the queue, **‘Notices’** collects what the field reports through the companion: **‘Incident’** (a fall, an aggressive dog, a locked door), **‘Supply needed’** (bandages running out), **‘Plan change**

proposal' (the client needs a different task set) and **'Info'**. The quick filter (marker 2) narrows the list to **'All'**, **'Needs action'** or **'Acknowledged'**.

Read a notice, then press **'Acknowledge'** to confirm that the office has seen it. Acknowledgement is the escalation tripwire: an unacknowledged incident escalates automatically after the **'Incident SLA (minutes)'** configured in the settings — so **'Acknowledge'** means "a human took this", not "this is done". A fall reported at 10:14 that nobody acknowledges by the SLA deadline is pushed up the chain rather than aging quietly in a list.

10.3 Edge cases and limits

- The inbox badge counts open items across both lists — a full review queue keeps the badge up even when notices are quiet.
- Resolving a bundle does not rewrite the rejected data; the resolution note is appended so the trail stays complete.
- Notices are intentionally lightweight. Clinical facts belong in the **'Client log'**, not only in a notice that ages out of view.

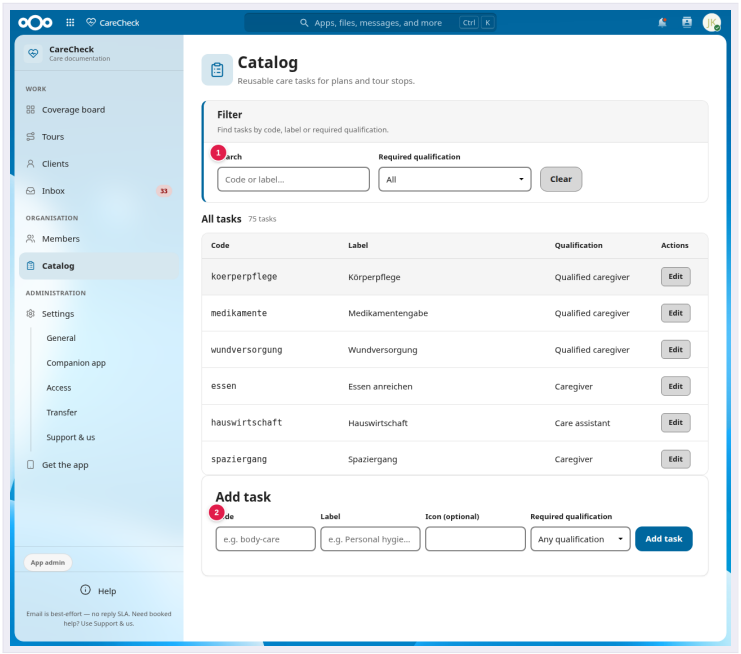
11 The task catalog

The **'Catalog'** is the vocabulary of the whole app: every entry is a reusable care task — *Körperpflege*, *Medikamentengabe*, *Spaziergang* — that care plans pick from and tour stops and visit bundles reference. Building plans from a shared catalog instead of free text is what makes the documentation comparable: two caregivers document the same `body-care`, not two spellings of the same thing.

11.1 Finding and adding tasks

The filter panel (marker 1) searches **'Code'** and **'Label'** or narrows by **'Required qualification'**; **'Clear'** resets. Below the table of **'All tasks'**, the **'Add task'** form (marker 2) creates a new entry:

1. Pick a stable **'Code'** — lowercase identifiers like `koerperpflege` or `body-care`; only `A-Z`, `0-9`, `_`, `.` and `-` are allowed.
2. Give the human-readable **'Label'** the staff will see, e.g. *Personal hygiene*.
3. Optionally set an **'Icon (optional)'** for the companion's task list.
4. Set **'Required qualification'** — **'Any qualification'** or one of the three levels — so under-qualified staff can never be scheduled for the task.
5. Save with **'Add task'**; existing rows can be edited in place.



The **‘Catalog’**: the filter (marker 1) above the task table and the **‘Add task’** form (marker 2).

11.2 Why codes must stay stable

A task’s **‘Code’** is the foreign key of the whole system: visit bundles, journal entries, monthly exports and billing data all reference it. Renaming a **‘Label’** is safe — it is display text — but reusing a code for a different task corrupts history: last March’s `body-care` would silently mean something else. Retire codes you no longer need by removing them from plans; never recycle them.

11.3 Edge cases and limits

- Keep the catalog small and canonical: one task per real service, not one per phrasing. The plan filter handles a long list, but reporting does not merge synonyms.
- The qualification on the task is a floor, not a suggestion — a **‘Qualified caregiver’**-task cannot be assigned to a **‘Care assistant’** anywhere in the system.
- Tasks a client needs but the catalog lacks are created here first — the plan editor links straight into this page (chapter 6).

12 Settings

‘Settings’ holds provider-wide configuration on seven pages — one topic per page, reached through the chip bar at the top or the sidebar sub-items. **‘General’**, **‘Companion app’** and **‘Support & us’** are visible to head office and app admins; **‘License’**, **‘Access’**, **‘Privacy & register’** and **‘Transfer’** are admin-only — anyone else who follows a link to them lands back on **‘General’**.

12.1 General

The **‘General’** page (marker 1 in the figure) sets the working frame for the whole provider:

‘Provider timezone’ e.g. `Europe/Berlin` — every time window, export and materialised tour is computed in this zone. Set it once, before the first template; changing it later shifts every scheduled window.

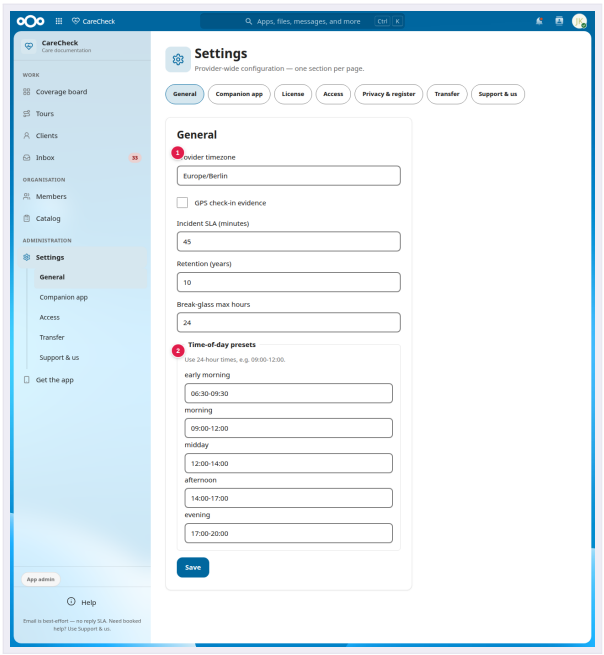
‘GPS check-in evidence’ records the device position at check-in as visit evidence.

‘Incident SLA (minutes)’ how long an unacknowledged **‘Incident’** may sit in the inbox before it escalates (chapter 10).

‘Retention (years)’ how long documentation is kept — the technical half of your retention policy.

‘Break-glass max hours’ the ceiling for emergency cover access in the companion (chapter 14).

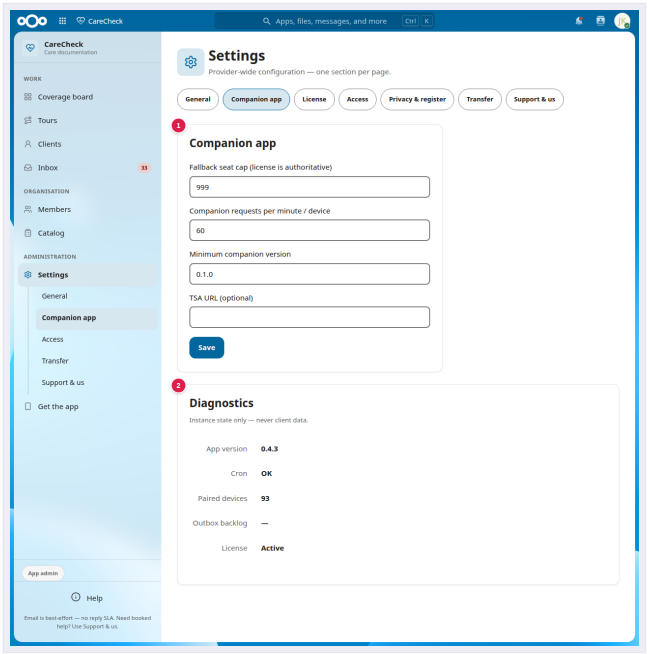
‘Time-of-day presets’ the five named windows — **‘early morning’** to **‘evening’** — as 24-hour ranges like `09:00–12:00`, so *morning* means the same thing on every plan and export.



‘Settings’ → **‘General’**: the provider frame — timezone field (marker 1) and the **‘Time-of-day presets’** (marker 2).

12.2 Companion app

The **‘Companion app’** page manages the mobile side’s technical guards. **‘Fallback seat cap (license is authoritative)’** bounds devices when no licence is applied, **‘Companion requests per minute / device’** rate-limits each phone, **‘Minimum companion version’** forces old app builds to upgrade, and **‘TSA URL (optional)’** points at the timestamp authority that anchors the client-log chain (chapter 7). Below, the **‘Diagnostics’** panel (marker 2) reports instance state — versions, seat counts, cron health — deliberately never client data.



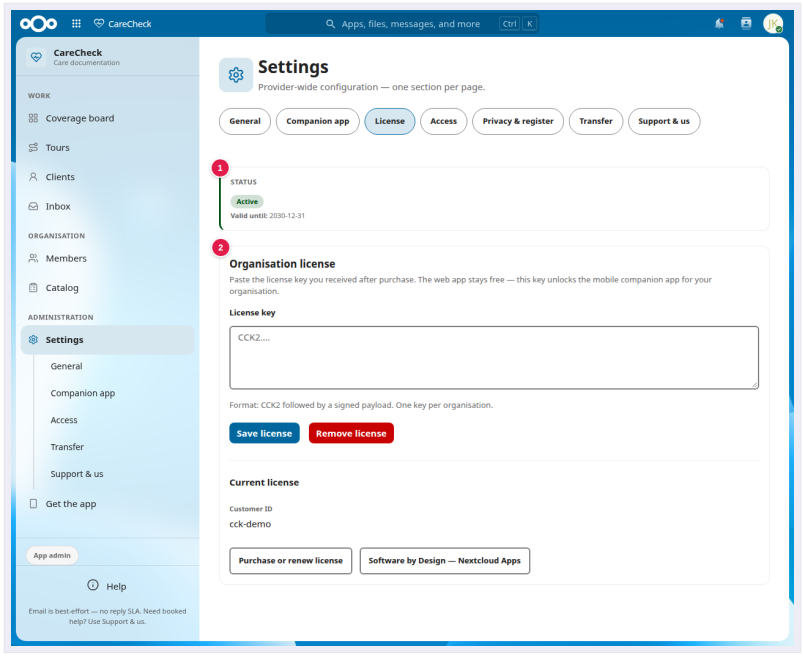
‘Settings’ → **‘Companion app’**: the companion limits (marker 1) and **‘Diagnostics’** (marker 2).

12.3 License

The **‘License’** page (admin-only) is where the `CCK2` key lives. The web app itself stays free (AGPL) — the licence only unlocks companion seats, and companions claim a seat automatically when they pair, so there is no seat-assignment list to maintain here.

The status row opens the page (marker 1): once a key is saved it holds a single **‘Status’** tile carrying the **‘Valid until’** date, and while no key is stored it also shows placeholder tiles for **‘Mobile seats’** and **‘Devices’** plus the **‘No license yet’** callout; with a key stored, the **‘Current license’** block shows the **‘Customer ID’**, and a **‘License expires soon’** callout appears inside the last 30 days with renewal contact links.

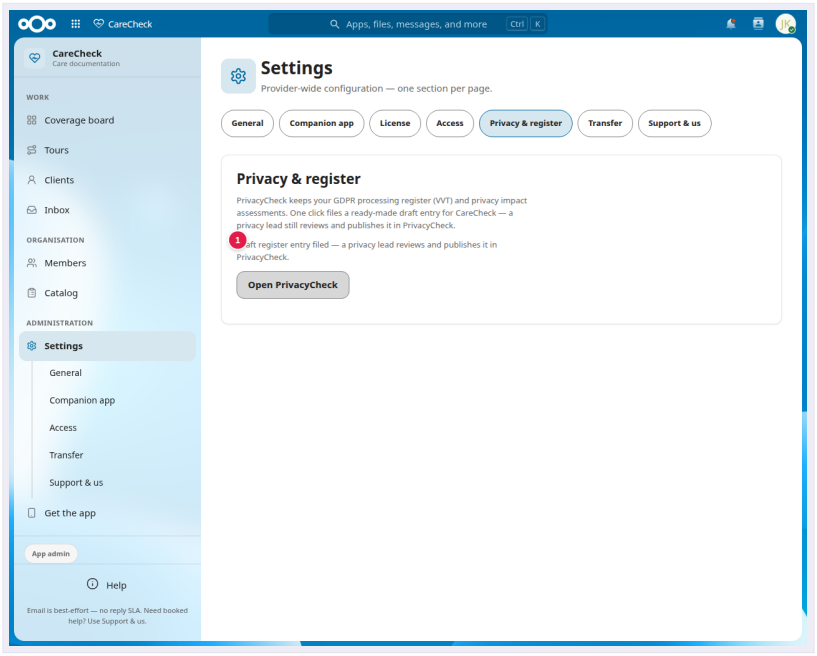
To apply a key, paste it into **‘License key’** on the **‘Organisation license’** card (marker 2) and choose **‘Save license’**. The card also shows the **‘Nextcloud instance ID’** — quote it when ordering so the key can be bound to this server. **‘Remove license’** asks for confirmation and then revokes the seats and devices tied to the key; this cannot be undone, and seat checks fail closed afterwards, so treat **‘Mobile seats’** as the number to watch when renewing.



‘Settings’ → **‘License’**: the status overview (marker 1) above the **‘Organisation license’** card (marker 2).

12.4 Privacy & register

‘Privacy & register’ connects CareCheck to PrivacyCheck’s GDPR processing register. **‘Create draft register entry’** files a ready-made draft describing the app’s data processing — retention follows **‘Retention (years)’** — which a privacy lead still reviews and publishes in PrivacyCheck. Once filed, the status line (marker 1) confirms the draft and the button yields to **‘Open PrivacyCheck’**; without PrivacyCheck installed, the panel says so instead of failing.



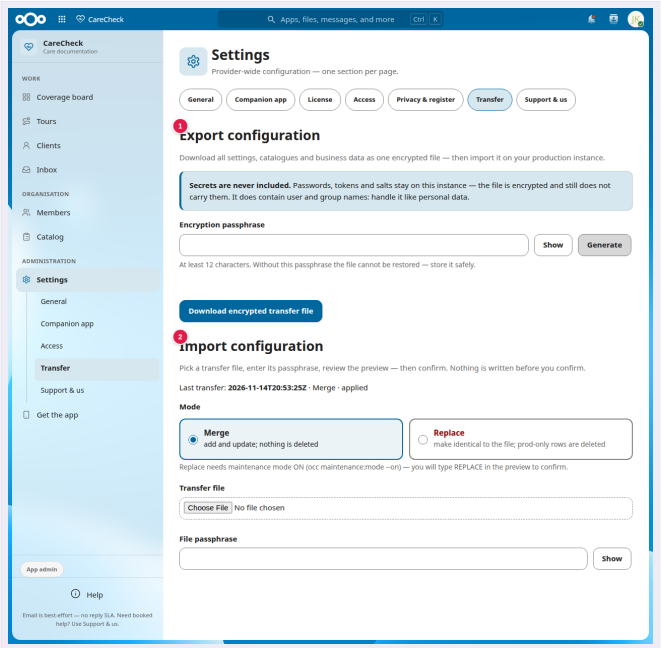
‘Settings’ → **‘Privacy & register’**: the filed register draft — the status line confirms it (marker 1).

12.5 Transfer

The **‘Transfer’** page (admin-only) moves the whole configuration — settings, catalogues and business data — from a test instance to production as one encrypted `.nctransfer` file. It exists so a provider can stage everything quietly, then go live without typing it all again.

- **Export** (marker 1): set an **‘Encryption passphrase’** of at least 12 characters — **‘Generate’** offers a strong one — and choose **‘Download encrypted transfer file’**. **‘Secrets are never included.’** Passwords, tokens and salts stay on the source instance even inside the encrypted file; user and group names do travel, so handle the file like personal data.
- **Import** (marker 2): pick the file, enter its passphrase, and the app runs a mandatory dry run first — the **‘Import preview’** dialog lists exactly what would change, grouped per area, and nothing is written before you confirm with **‘Apply import’**. The **‘Last transfer’** line keeps the receipt (time, mode, result, snapshot id).

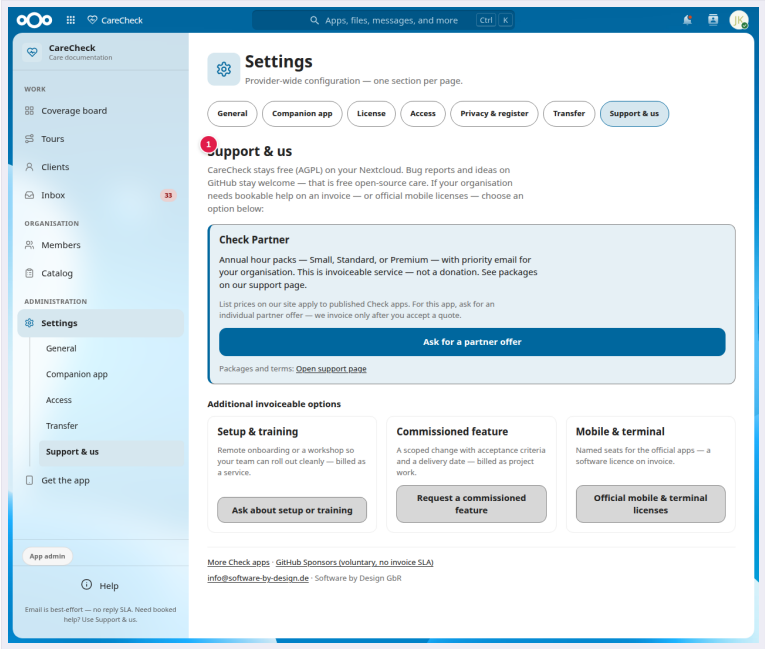
Two modes decide how the file meets existing data: **‘Merge’** adds and updates and deliberately deletes nothing — the safe repeat choice. **‘Replace’** makes the target identical to the file and deletes rows that exist only there; it is restricted to Nextcloud system administrators, needs maintenance mode on, and makes you type `REPLACE` in the preview. A snapshot is taken before applying, so a bad import can be undone. On the shell the same path is `occ carecheck:config:import`; after a raw `config:import` bypass, `occ carecheck:config:reconcile` re-writes the mirrored keys.



‘Settings’ → **‘Transfer’**: the **‘Export configuration’** card (marker 1) and the **‘Import configuration’** card (marker 2).

12.6 Support & us

‘Support & us’ collects the contact points: the free channels (GitHub issues, community) and the invoiceable partner and setup packages for organisations that want booked help. Error banners across the app also carry a **‘Report this problem’** link that pre-fills a diagnostic e-mail (chapter 15).



12.7 Edge cases and limits

- Change **‘Provider timezone’** only before go-live if you can — the setting anchors every time window in the system.
- **‘Access’** has its own chapter because it is also the security boundary: chapter 13.
- Removing the licence does not brick devices instantly — seat checks fail closed over time; treat the **‘Mobile seats’** meter as the number to watch when renewing.
- A transfer file is not a backup — it carries no secrets and no audit log. Keep real backups on the instance level, and keep the passphrase somewhere a fired colleague cannot reach.
- **‘Replace’** on a live instance is a maintenance-window action, never an experiment: maintenance mode keeps members and companions from writing while the rows are swapped.

13 Access, allow lists and teams

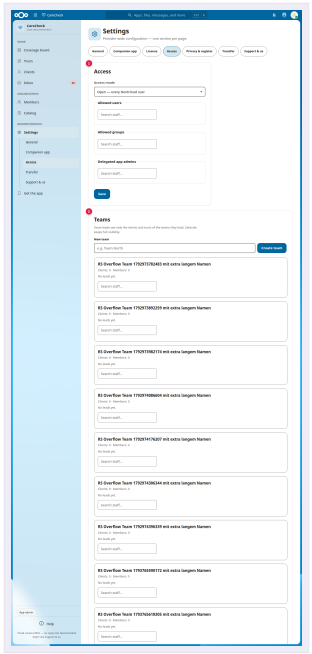
The **‘Access’** page under **‘Settings’** is the security boundary of the app: it decides who may open CareCheck at all, who may administer it, and how team leads’ visibility is scoped. Only app admins see this page.

13.1 Access mode

The **‘Access mode’** dropdown (marker 1) offers two values:

‘Open — every Nextcloud user’ every account on the instance can open the app and gets the base *Member* role. Use this on dedicated instances where the whole organisation is legitimate staff.

‘Restricted — allow list below’ only accounts named in **‘Allowed users’** or member of a group in **‘Allowed groups’** can open the app at all; everyone else sees **‘No access.’** This is the right choice on shared instances — care data exists before roles even apply, so the door is the first control, not the last.



‘Settings’ → **‘Access’**: the **‘Access mode’** dropdown with the allow lists (marker 1) and the **‘Teams’** panel below (marker 2).

Both pickers are directory pickers, not free text: **‘Allowed users’** and **‘Allowed groups’** search the Nextcloud account and group directories, so a typo can never lock a real person out — you either pick an existing entry or none. **‘Delegated app admins’** works the same way and grants full app-admin rights to named accounts without making them Nextcloud administrators — the right tool for a care provider’s own administrator on a hosted instance.

13.2 Teams

The **‘Teams’** panel (marker 2) manages the team layer: type a **‘New team’** name — say *Team Nord* — and press **‘Create team’**. Each team row then lets you appoint and remove *team leads*. A team lead sees only the clients and tours of the teams they lead; the head office keeps full visibility. Assign members to a team in the **‘Members’** list and clients on their **‘Record’** tab (chapters 9 and 5).

Because the scoping happens at read time, a client without a team is invisible to every team lead — not an error, but worth remembering when a lead reports a missing record. Give every client a team if your organisation uses the feature at all.

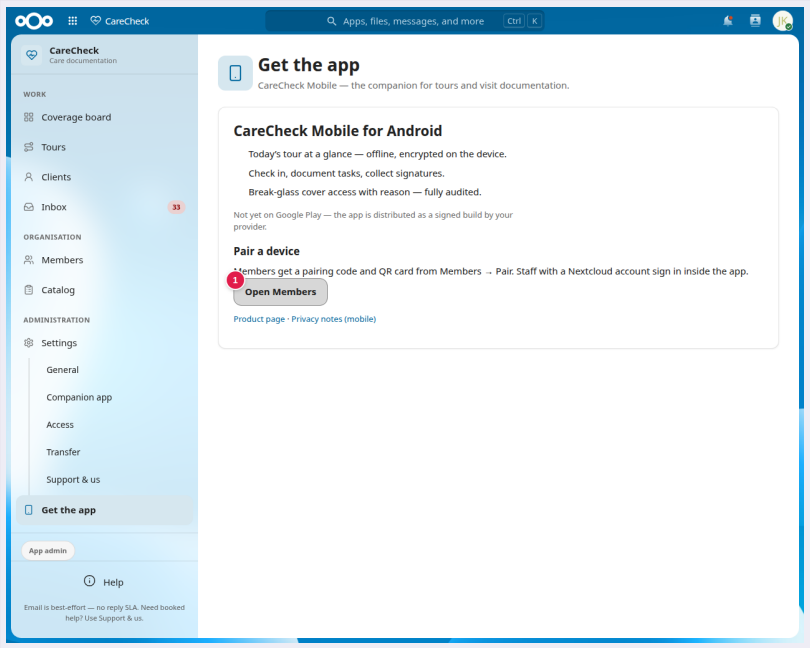
13.3 Edge cases and limits

- Restricted mode without any allow-list entry locks *everyone* out except app admins — add at least the dispatch group before saving.
- Allow lists grant the *door*, not a role: a care worker on the list still needs a member record (and a team, where teams are used) to see real work.
- Nextcloud administrators always pass the door — the app cannot lock out the platform admin, by design.

14 The companion app (CareCheck Mobile)

CareCheck Mobile is the Android companion for the road: it carries today’s tour in a pocket, documents visits offline and syncs when coverage returns. The web app is free — the companion is the licensed

part (CCK2 seats, chapter 12). The **‘Get the app’** page in the sidebar summarises distribution and pairing in one place.



The **‘Get the app’** page: what the companion does, how it is distributed, and the link to **‘Open Members’** for pairing (marker 1).

14.1 Pairing and locking the device

Two ways in, depending on the staff member:

- **Pair code.** For members without a Nextcloud account: create a **‘Pair code’** on their row in **‘Members’** (chapter 9) and let them scan the QR or type the code in the app.
- **Account sign-in.** Staff with a Nextcloud account sign in inside the app directly.

After pairing, a PIN protects the local store — biometrics can stand in on supported devices. Everything the app holds is encrypted on the device; *Unpair* and *Deactivate* cut access server-side, so a lost phone is an inconvenience, not a data leak.

14.2 Working a day

Today lists your stops in order, with time window and a *Navigate* shortcut into the maps app. At the client:

1. *Start visit* checks in — with optional GPS evidence (chapter 12).
2. Work the task list from the published plan snapshot: mark each task done, not done or refused, with a reason where required; tasks that demand a measured value insist on the number.
3. Collect the signature — client or representative — or record why it could not be collected.
4. Finish. The visit becomes a signed *visit bundle* that syncs to the server when the device has connectivity.

Everything works offline: the bundle waits encrypted on the device and uploads itself later — a dead zone in a basement flat costs nothing. *Report something* files a report — **‘Incident’**, **‘Supplies needed’** or **‘Suggest plan change’** — straight to the **‘Inbox’**, where the same reports sit under **‘Supply needed’** and **‘Plan change proposal’**. The same menu also opens **‘Group session’**: name the session, add the participating clients via **‘Add participant’**, press **‘Start session’**, mark absentees on the attendance list and **‘End session’** — closing writes one journaled group-session

entry per participating client. Unlike visit bundles, group sessions need a connection; they do not queue offline. Voice notes and photos round the menu out (chapters 10 and 7).

The *Today* list on the phone shows the day's stops in order; the **'Get the app'** page in the sidebar repeats the pairing paths.

14.3 Cover access — break-glass

Sometimes the right caregiver is not the scheduled one — sick cover, a stop dispatch sends you to, an emergency. The companion's *Cover* flow opens access to a client outside your tours, but it demands a reason first and writes the access into the audit trail — bounded by **'Break-glass max hours'** in the settings (chapter 12). The name is deliberate: like breaking the glass on a fire alarm, it is allowed, logged, and visible afterwards. There is no quiet way around the log.

14.4 Edge cases and limits

- An empty *Today* usually means an unpublished plan (chapter 6) or no materialised tour (chapter 4) — not a sync failure.
- Rejected bundles surface in the **'Review queue'**, never silently on the phone (chapter 10).
- Devices below **'Minimum companion version'** must update before they sync again — the server enforces it.

15 Errors, troubleshooting and FAQ

This chapter collects the failure modes a user can actually meet, in the shape you meet them: what you see, why it happens, and what to do. When a banner offers a **'Report this problem'** link, use it — it pre-fills a diagnostic e-mail with the technical context support needs, without client data.

15.1 Access and sign-in

"'No access.'" The workspace runs **'Restricted — allow list below'** and your account is not listed. Not a password problem — an app admin adds you under **'Settings'** → **'Access'** (chapter 13).

The app does not appear in the apps menu CareCheck is not enabled for your account or is disabled instance-wide; a Nextcloud administrator checks the app management.

15.2 Planning and day work

"Background jobs are not running — generated tours may be missing." The instance cron is stopped, so the nightly materialisation did not run. Server-side fix: restore the Nextcloud cron; meanwhile **'Generate tours now'** on the board keeps the days coming (chapter 3).

A day is unexpectedly empty Only **'Published'** templates on their ticked **'Weekdays'** produce tours — check status and weekday set in **'Tours'**, then generate.

A stop cannot be assigned to a person Their **'Qualification'** is below the client's **'Required qualification'** — pick someone qualified or lower the requirement where clinically acceptable.

15.3 Saving and conflicts

Saving a record or plan fails with a conflict Someone else saved while you were editing. The app refuses to silently overwrite — reload the view, re-apply your change and save again. Nothing is lost; your change just has to be re-typed.

“Photo documentation requires the client’s consent” **‘Photo documentation consent given’** is not set on the **‘Record’** tab. Without consent the **‘Photos’** tab stays locked — that is the point of the gate.

15.4 Inbox and documentation

A visit bundle sits in the review queue The submission was formally invalid. Check the client log, clarify the facts, **‘Resolve’** with a correction note — the queue is a holding pen, not a graveyard (chapter 10).

An incident seems ignored Nobody pressed **‘Acknowledge’**. Acknowledgement is the human receipt that stops the **‘Incident SLA (minutes)’** escalation clock.

15.5 Companion app

The companion is locked or refuses to sync Licence missing, seats exhausted, or the device was unpaired: check **‘Settings’** → **‘License’** (the **‘Status’** and **‘Mobile seats’** tiles) and re-pair the member.

A member’s phone is lost or stolen Open **‘Devices’** on their member row and **‘Revoke’** the lost device — access dies on its next call while the member and their other devices keep working (chapter 9).

An old phone stops syncing It fell below **‘Minimum companion version’** — update the app and it syncs again.

Empty *Today* on the phone The client’s plan is not published or no tour materialised — chapters 6 and 4.

15.6 FAQ

Is the web app paid? No — the web app is AGPL-free. Only companion seats are licensed (`CCK2`).

Can log entries be lost or edited? No. The client log is append-only and hash-chained; corrections are new entries, and anchoring can prove the chain externally.

Who sees a client record? Head office and app admins in full, team leads within their teams, devices only their assigned stops — details in chapters 1 and 13.

Does documentation work offline? Yes — the companion buffers encrypted on the device and syncs on the next connection.

Where do I get help? **‘Settings’** → **‘Support & us’** lists the channels; error banners carry **‘Report this problem’** for a pre-filled diagnostic mail.

16 Accessibility

CareCheck is built to WCAG 2.1 AA: the full workflow — planning, dispatch, documentation, administration — is operable by keyboard alone, readable with a screen reader, and usable at high zoom and in forced-colors mode. This chapter is the map to those affordances.

16.1 Keyboard operation

- **‘Skip to content’** is the first tab stop on every page and jumps past the navigation straight into the main panel.
- The sidebar is a real list of links — Tab walks it in order; on small screens **‘Menu’** toggles it and reports its state via `aria-expanded`.

- Client section chips (**‘Record’** to **‘Exports’**) behave like tabs: they move focus to the section they open, so the reading order stays sane.
- Dialogs — **‘Assign stop’**, **‘Please confirm’**, **‘Resolve rejected bundle’** — trap focus while open, restore it on close, and submit with Enter / cancel with Escape. Confirm buttons that stay disabled until a choice exists (**‘Assign’**) are reachable but honest about being unavailable.
- Filter panels are labelled `role=search` regions with a visible **‘Filter’** heading; quick filters expose `aria-pressed` so the active state is announced, not just painted.

16.2 Screen readers and live feedback

Two off-screen live regions carry the dynamic feedback: a polite region for status changes (list counts like *3 notices*, form results) and an assertive region for errors. Async panels announce themselves through `aria-live` counters rather than repainting silently — the inbox badge, the board’s day counters and the invoice-push status line all speak. Form errors attach an inline message to the named field with `aria-invalid`, so a screen reader lands on the exact input that failed instead of a banner at the top.

16.3 Vision

- Text and controls meet WCAG AA contrast on both light and dark Nextcloud themes; status like **‘uncovered’** is never carried by colour alone — label plus badge colour together.
- The layout is mobile-first: at 320 px nothing scrolls sideways except tables that deliberately offer it, and every touch target meets the 44 px floor.
- Animations respect `prefers-reduced-motion` — the app renders usable with motion suppressed, including on the companion’s capture flow.
- Forced-colors mode keeps controls identifiable: borders and focus rings are drawn, not implied.

16.4 If something is not accessible

Accessibility is a requirement, not a feature toggle. If you hit a barrier — an unreachable control, an unlabelled icon, contrast that fails on your theme — report it via **‘Support & us’** or the **‘Report this problem’** link; naming the surface and your assistive technology gets it fixed fastest.

17 Glossary

Domain terms, in the sense the app uses them — not the UI labels (the chapter text uses those verbatim).

Tour template The weekly rule: weekdays plus ordered stops with time windows. Only *published* templates produce tours.

Tour A concrete daily instance materialised from a template for one date; the unit the coverage board shows.

Stop One scheduled visit inside a tour: client, time window, executor. A stop without an executor is *uncovered*.

Executor The member or account responsible for a stop.

Floating stop A template stop deliberately left without an executor, assigned daily on the board.

Head office / Zentrale The dispatch desk — the role that plans, assigns and works the inbox.

Qualification The three-level professional grade (care assistant, caregiver, qualified caregiver) that floors task and client requirements.

Catalog task A reusable care service with a stable code — the vocabulary of plans and documentation.

Care plan The per-client set of catalog tasks a visit may document, versioned by published snapshots.

Plan snapshot An immutable, numbered revision of a care plan; visit bundles record which snapshot they documented against.

Visit bundle The signed package a companion uploads per visit: check-in, task states, measured values, signature.

Review queue The inbox holding pen for bundles the server rejected — resolved by a human, never auto-fixed.

Notice A lightweight field report to dispatch: incident, supply needed, plan change proposal, info.

Acknowledgement The office's receipt on a notice; the act that stops the incident-SLA escalation clock.

Client log The per-client, append-only, hash-chained journal of manual notes and system events.

Hash chain The entry-to-entry fingerprinting of the log that makes silent edits and deletions detectable.

TSA / anchoring An external timestamp authority that seals the chain head — independent proof of the log's state at a time.

Break-glass Audited emergency access to a client outside your tours, reason required, time-bounded by settings.

Cover The companion flow that uses break-glass for sick cover and dispatch-sent stops.

Practice tour A synthetic sandbox tour for training staff, with no real client data.

Pair code The one-time credential that binds a companion device to a member.

CRM company The CustomerCheck billing entity linked to a client — required for invoice drafts.

Access mode The door setting: open to every Nextcloud user, or restricted to allow lists.

Team lead A lead scoped to their teams' clients and tours; head office always sees everything.

18 Surface index — where things live

Every surface in the app, mapped to the chapter that explains it. Use this as the lookup table when you know the screen but not the chapter.

'Coverage board' day view, counters, unassigned stops, assign dialog, generate tours, DutyCheck strip — chapter 3

'Tours' weekly template list, template form, draft vs published — chapter 4

'Clients' client register, filters, add form — chapter 5

Client record: 'Record' master data, caregivers, team, consent, CRM link, archive — chapter 5

Client record: 'Care plan' task pick-list, measured values, publish snapshot — chapter 6

Client record: 'Client log' hash-chained journal, manual entries — chapter 7

Client record: 'Photos' consent-gated photo documentation — chapter 7

Client record: 'Exports' monthly CSV/PDF, billing JSON/XML, invoice drafts — chapter 8

'Members' staff register, pair codes, practice tours, absences, devices — chapter 9

'Inbox' review queue, notices, acknowledgement — chapter 10

'Catalog' reusable task list and add form — chapter 11

'Settings' → **'General'** timezone, GPS, SLA, retention, break-glass hours, time-of-day presets — chapter 12

'Settings' → **'Companion app'** seat cap, rate limits, minimum version, TSA, diagnostics — chapter 12

'Settings' → **'License'** `CCK2` key, seat tiles, instance ID, renewal — chapter 12

‘Settings’ → **‘Access’** access mode, allow lists, delegated admins, teams — chapter 13

‘Settings’ → **‘Privacy & register’** PrivacyCheck register draft — chapter 12

‘Settings’ → **‘Transfer’** encrypted test-to-production export/import with dry run — chapter 12

‘Settings’ → **‘Support & us’** contact and partner channels — chapter 12

‘Get the app’ companion distribution and pairing overview — chapter 14

Dialogs **‘Assign stop’** (chapter 3), **‘Resolve rejected bundle’** (chapter 10), **‘Paired devices’** (chapter 9), **‘Please confirm’** (destructive actions, throughout)

Companion app pairing, today list, visits, notices, cover — chapter 14

18.1 How do I…?

…plan a new recurring round? Create a template, add stops, set **‘Published’** — chapter 4.

…cover a sick colleague today? Assign their stops on the board, or let the replacement use *Cover* in the app — chapters 3, 14.

…give a new nurse access? Add a member, pair their device (or allow-list their account) — chapters 9, 13.

…document a wound? Consent on the record, then **‘Photos’** or the companion — chapter 7.

…bill a month? **‘Exports’** on the client, or push a draft to InvoiceCheck — chapter 8.

…prove nothing was tampered with? The hash chain plus TSA anchoring — chapters 7, 12.

…fix a rejected visit? **‘Inbox’** review queue, **‘Resolve’** with a note — chapter 10.

…restrict who opens the app? **‘Access mode’** restricted plus allow lists — chapter 13.

…cut off a lost phone? **‘Devices’** on the member row, then **‘Revoke’** — chapter 9.

About Software by Design

Software by Design builds Nextcloud apps for organisations that want to host their own data — from time tracking and projects to inventory, fleet management and customer relations. Every app runs entirely on your Nextcloud instance; the web interfaces are free (AGPL).

Company

- **Website:** https://nextcloud.software-by-design.de/
- **Partner & quotes:** mailto:info@software-by-design.de
- **Report a bug / suggest an improvement:** mailto:dev@software-by-design.de

Support & how to contribute

CareCheck remains free on your Nextcloud (AGPL). Bug reports and sensible ideas via GitHub or email are part of open-source maintenance — without a guaranteed response time.

In the app

Under **‘Settings’** → **‘Help & support’** (navigation chip may still read **‘Support us’**) you find the same options as on our website: partner quote, onboarding, commissioned feature and voluntary support.

Check Partner (invoiceable)

If your organisation needs predictable help with an invoice — setup, hour packages, prioritised email — choose **‘Check Partner’**. That is a professional service with quote and invoice, not a donation.

- Packages and terms on the support page: https://nextcloud.software-by-design.de/en/support/
- Request a partner quote: mailto:info@software-by-design.de?subject=Partner%20inquiry

Other invoiceable options

- **Onboarding & training** — remote setup or workshop for your team.
- **Commissioned feature** — scoped change with acceptance criteria and delivery date.
- **Official mobile & terminal licences** — where offered for CareCheck; named seats with invoice.

GitHub Sponsors (voluntary)

Voluntary support without an invoice SLA — appreciated, but optional:
https://github.com/sponsors/aSoftwareByDesignRepository

Legal note

This manual does not replace legal, tax or compliance advice. Third-party product names and trademarks belong to their respective owners.

Index

Absence, 7, 17
Access, 25
Access mode, 3, 25
Accessibility, 29
Accounting, 2
Add client, 9
Add task, 19
Allow list, 25
App admin, 2
Append-only, 12
Assign, 5
Assign stop, 5
Billing, 14
Billing data, 14
Break-glass, 26
Care plan, 11
CareCheck Mobile, 26
Catalog, 19
Client, 8
Client list, 8
Client log, 12
Client record, 8
Code, 19
Code stability, 20
Companion app, 26
Companion app settings, 21
Conflict, 28
Consent, 13
Contrast, 30
Cover access, 28
Coverage board, 5
Create team, 26
Create template, 7
CRM, 8
CRM company, 10
Cron, 6
CSV import, 10, 14
CustomerCheck, 10
Day overview, 5
Device, 15
  revoke, 16
Diagnostics, 20
DutyCheck, 6
Error, 28
Escalation, 17
Export, 14, 23
FAQ, 28, 29
Filter, 8
First tour, 4
GDPR, 23
General settings, 20
Generate tours, 6
Getting started, 3
Glossary, 30
Hash chain, 12
Head office, 2
Import, 10, 14, 23
Inbox, 17
Incident, 18
Index, 31
Instance ID, 22
Invoice draft, 14
InvoiceCheck, 14
Keyboard, 29
License, 20, 22
Live region, 30
Materialisation, 8
Measured value, 11
Member, 2, 15
Member list, 15
Navigation, 3, 31
No access, 28
Not on duty, 7
Notice, 17
Offline, 26
On duty, 6
Optimistic locking, 28
Pair code, 16
Pairing, 15, 27
Pairing card, 16
Permission, 2
Photo, 12
Photo documentation, 13
PIN, 27
Plan snapshot, 11
Plan tasks, 11
Practice tour, 7, 8, 17
PrivacyCheck, 23
Publish plan, 11
Qualification, 2
Record, 9
Reduced motion, 30
Rejected bundle, 17
Report a barrier, 30
Review queue, 17
Revoke, 16
Role, 2
Screen reader, 29
Seats, 22
Service record, 14
Settings, 20
Setup checklist, 4
Sign in, 3
Signature, 27
Skip link, 29
Staging, 23
Stop, 5
Supply needed, 18
Support, 24
Task, 19
Team, 2, 15, 17, 25
Team lead, 2, 17
Template list, 7
Today, 27
Tour, 7
Tour template, 7
Transfer, 20, 23
Troubleshooting, 28
TSA, 21
Unassigned stops, 5
Visit, 27
Voice note, 12, 13