



Synapse-Time

Workforce Management — Admin & User Guide

Version 0.1 · 10 July 2026

Standalone · API-first · Self-hosted

Contents

1. About Synapse-Time	3
Key capabilities	3
2. Getting Started	3
2.1 Prerequisites	3
2.2 Run with Docker	3
2.3 First-run administrator setup	4
2.4 Signing in & accessing from other machines	4
3. Roles & Access Control (RBAC)	5
4. Administration	5
4.1 User Management	5
4.2 Access Control	5
4.3 User status values	5
4.4 Sites / Locations	5
4.5 Training Courses	6
5. Managing Employees	7
6. HR & Contracts	7
6a. Contract changes (Maker-Checker)	7
6b. Performance reviews	7
6c. Overtime rates & rules	7
6d. Notifications	8
7. Rostering & Scheduling	9
7a. Auto-generating a roster	9
8. Time & Attendance	9
8a. Absence & Holidays	9
9. Developer / API	10
10. Security & Compliance	10
11. Troubleshooting	11
Appendix A — Demo accounts (optional seed)	11
Appendix B — Quick command reference	11

1. About Synapse-Time

Synapse-Time is a standalone, enterprise workforce-management portal for rostering, time & attendance, and HR contract management. It is API-first and ships as Docker containers for self-hosted (on-premise) or cloud deployment, prioritising data sovereignty and strict access control.

Key capabilities

- **Rostering** — week / month / year views, per-location filtering, colour-coded shifts and absence, and **auto-generation** from a site's opening hours with an approval workflow.
- **Working Time compliance** — overlap, 11-hour rest, contracted-hours and 48-hour weekly-maximum checks.
- **Time & attendance** — web clock in/out, standard vs. overtime calculation, and weekly/monthly booked hours.
- **Absence** — holiday, sickness and training with balances, approvals, and manager/HR entry on behalf of staff.
- **HR** — an HR dashboard, employee records (pay grade, working location, leave entitlement), Maker-Checker contract changes, performance reviews with notifications, and overtime rates & rules.
- **Training** — a course catalogue with allocation to staff and employee self-booking.
- **Identity & access** — granular RBAC, optional multi-factor authentication (MFA), and immutable audit logging.
- **Developer / integration** — scoped API keys and signed webhooks for HR, payroll and finance systems.

2. Getting Started

2.1 Prerequisites

Before installing, make sure you have:

- **Docker Engine** and **Docker Compose v2** (the `docker compose` command) on the host. Compose v1 works but needs the down-before-up workaround shown in Section 2.3.
- A **host machine** (64-bit Linux, macOS or Windows) with roughly 2 vCPUs, 4 GB RAM and a few GB of free disk (more as data grows).
- The **Synapse-Time project files** (this repository) available on the host.
- Ability to create and edit the `.env` file to set the database, Redis and JWT secrets (openssl is handy for generating them).
- Network access to the host on **ports 3000** (web portal) and **8000** (API) from the machines that will use it — reach the host by its IP address, not `localhost`, from other computers.
- A modern **web browser** (Chrome, Edge, Firefox or Safari).

2.2 Run with Docker

From the project root (the folder containing `docker-compose.yml`):

```
cp .env.example .env          # then set the secret values
docker compose up -d --build
```

The web portal is served by the frontend container. If you use Docker Compose v1 (docker -compose), always bring the stack down before up to avoid a known recreate bug:

```
docker-compose down --remove-orphans
docker-compose build
docker-compose up -d
```

After an update: if the change added database tables, columns or permissions, recreate the database with `docker compose down -v` before `up -d --build` (dev only — this deletes data). A code-only change just needs `up -d --build`. See the separate one-page *Docker Quick Start* for the full command reference.

2.3 First-run administrator setup

On first launch no users exist. Open the portal and you will be taken to the **Set up administrator** screen (or go directly to `/setup`). Enter a name, email and password to create the first System Admin — you are then signed in automatically. This screen only appears until an administrator exists.

2.4 Signing in & accessing from other machines

Open the portal at the host's address on the frontend port, for example `http://<host-ip>:3000`. Use the host machine's IP (not `localhost`) when accessing from another computer. The portal proxies API calls internally, so no additional configuration is required.

3. Roles & Access Control (RBAC)

Every user holds one or more roles. Roles grant permissions; permissions gate every screen and action, and are re-checked on the server for every request.

Role	Typical user	Can do
System Admin	IT / system owner	Everything: manage users, roles, sites, training and API keys; all HR, roster, time and absence functions; view the audit log.
HR / Payroll	HR and payroll staff	Add/amend employees; approve contract changes and absence; manage training and overtime rules; generate and approve rosters; manage timesheets; read audit.
Manager	Line / service managers	Build and auto-generate rosters; approve rosters and absence; propose contract changes (maker); read timesheets.
Employee	Front-line staff	View their own roster and timesheet; book absence; self-book training courses.

The full permission matrix is visible in-app under **Administration** → **Access Control**. Assign roles to users under **User Management**.

4. Administration

4.1 User Management

Administration → **User Management** (requires the user : manage permission) lists all users with their roles, status and MFA state. From here an administrator can:

- **Add a user** — set name, email, an initial password, status, and one or more roles.
- **Edit a user** — change name, status, roles, or reset the password (leave password blank to keep it).
- **Delete a user** — remove the account. You cannot delete your own account, and the system prevents deleting or demoting the last remaining System Admin.

4.2 Access Control

Administration → **Access Control** shows the role-to-permission matrix and a summary of how many permissions each role grants. Use it to confirm what a role can do before assigning it.

4.3 User status values

Status	Meaning
active	Can sign in and use the portal.
invited	Created but not yet activated.
suspended	Temporarily blocked from signing in.
disabled	Deactivated (retained for audit).

4.4 Sites / Locations

Administration → **Sites / Locations** lists the sites where shifts are rostered. Add or edit a site to set its **name, time zone, opening hours and open days**. Opening hours drive auto-rostering (Section 7a); a site is also chosen as an employee's working location and when creating shifts. For an overnight site, set the closing time earlier than the opening time. Sites not referenced by any shift can be removed.

4.5 Training Courses

Administration → **Training Courses** is the course catalogue. Administrators and HR (permission `training:manage`) can:

- **Add / edit / remove courses** (title, provider, duration, active flag).
- **Allocate** a course to a specific employee, optionally with a scheduled date.
- Track all **enrolments** (allocated or self-booked) and mark them complete or cancel them.

Active courses can be **self-booked** by staff under **Absence** → **Training**, where they also see the courses allocated to them.

5. Managing Employees

An employee is a user with an employment profile (employee number, job title, contracted hours, hire date). To add one, go to **HR & Contracts** and choose **Add employee** (requires `employee:manage`, held by HR and Admin).

- Enter the person's name, login email and a temporary password.
- Set the employee number, job title, **pay grade**, **working location** (a site from Administration → Sites), contracted weekly hours and hire date.
- Set the annual-leave entitlement and public-holiday allowance.
- The account is created with the Employee role and an initial effective-dated contract record.

Contracted-hours values are the source of truth for the rostering fatigue checks and are only changed afterwards through the Maker-Checker workflow (next section).

6. HR & Contracts

The HR area opens on an **HR Dashboard** summarising tasks needing attention — pending contract approvals, absence requests across all types, and scheduled performance reviews — with quick approve/reject actions. It also hosts **Overtime rates & rules**, where HR defines named rules pairing a trigger (weekly/daily over a threshold, weekend, public holiday or night) with a pay multiplier such as 1.5× or 2.0×. The **Employees & Contracts** tab holds the employee directory and the Maker-Checker workflow below.

6a. Contract changes (Maker-Checker)

Changing an employee's contracted hours is a two-person control:

- **Propose (Maker)** — a Manager opens the employee in the directory and proposes new hours with a reason.
- **Approve/Reject (Checker)** — an HR/Payroll user reviews the request in the Approvals inbox and either approves it with an **Effective From** date or rejects it with a note.

On approval the change is written as an effective-dated contract version and cascades into the rostering and fatigue engines. Safeguards enforced by the system:

- The approver cannot be the person who proposed the change (segregation of duties).
- An Effective From date is mandatory to approve.
- Concurrent approvals are prevented; every action is recorded in the immutable audit log.

6b. Performance reviews

Open an employee from the HR directory (**Details**) to see their profile and schedule a **performance review**. Choose the review type, the manager to conduct it, and a due date. On creation both the **employee and the manager are notified** (see the bell icon in the top bar) that the review needs to be done. All reviews are listed under the **Performance Reviews** menu, where they can be marked complete.

6c. Overtime rates & rules

The HR Dashboard's **Overtime rates & rules** panel (permission `pay:manage`) lets HR/Admin define named pay rules, each pairing a trigger with a pay multiplier:

- **Trigger** — weekly over a threshold, daily over a threshold, weekend, public holiday, or night.
- **Multiplier** — e.g. 1.5× (time-and-a-half) or 2.0× (double time); threshold triggers also take an hours value.
- Rules can be activated or deactivated, edited and removed.

6d. Notifications

The **bell** in the top bar shows in-app notifications (for example, a performance review that needs doing) with an unread count. Open it to read items and mark them read.

7. Rostering & Scheduling

The rosters view offers **Current Week**, **Month** and **Year** views, with a week picker (week-commencing date and week number), previous/next navigation, and a **location filter** to show the rota for a single site (or all sites). The week view shows the rota by employee; the month view is a calendar of shifts and hours; the year view summarises shifts and hours per month. The rostering engine evaluates each assignment and flags:

- **Overlaps** — an employee booked into two intersecting shifts.
- **Rest-period breaches** — less than 11 hours between consecutive shifts.
- **Weekly-hours breaches** — scheduled hours above the employee's contracted hours, or above the 48-hour legal maximum.

Shifts and absences are **colour-coded** on the roster — shifts (green), holiday (amber), sickness (red) and training (blue) — with a legend beneath the grid, so cover and absence are visible at a glance. Each shift pill also carries a small **morning / day / night** icon in its top-right corner.

7a. Auto-generating a roster

Auto-generate (Manager / HR / Admin) builds a draft rota for a site from its **opening hours** and its **contracted staff** (employees whose working location is that site). The scheduler respects Working Time rules — 11h minimum rest between shifts, each person's contracted weekly hours, and the 48-hour legal maximum — and leaves blocks it cannot fill as open shifts.

- Set each site's opening hours and open days under Administration → Sites first.
- The generated roster is created as **draft** and must be **Approved or Rejected** by a Manager, HR or Admin from the banner on the roster.
- Draft shifts are shown with a dashed outline and remain fully **editable** (add, reassign or remove) before or after approval.

8. Time & Attendance

Staff clock in and out from the web portal. The service computes worked hours (break-aware) and splits them into standard and overtime against the weekly standard. The page also shows the **hours booked by employee** for the current week and month, drawn from the published rota.

8a. Absence & Holidays

The **Absence** menu contains **Holiday**, **Sickness** and **Training**. Employees book their own absence from these pages; the Holiday page also shows their entitlement and remaining balance. Each employee record holds an annual-leave-days entitlement and a public-holiday allowance (set when adding or amending the employee).

- Managers and HR (permission `Leave: approve`) see all requests and approve or reject them.
- Managers and HR can also **enter an absence on behalf of an employee** (recorded as approved) from the Book dialog.
- Pending **holiday** requests also appear on the Dashboard under the **Holidays Approval** card for quick sign-off.
- Approved holidays reduce the employee's remaining balance for the year.

9. Developer / API

Developer / API (requires `apikey:manage`) lets administrators provision scoped API keys for external integrations.

- **Generate** a key with specific read/write scopes (e.g. `roster:read`). The secret is shown once — copy it immediately.
- **Revoke** a key at any time; it stops working instantly.
- Keys are stored hashed, never in plain text; every use is timestamped.

Outbound events (such as an approved contract change) are delivered to registered webhook endpoints with an HMAC-SHA256 signature so receivers can verify authenticity.

10. Security & Compliance

- **RBAC** — least-privilege roles; permissions checked server-side on every request.
- **MFA** — optional time-based one-time-password (TOTP) via authenticator apps.
- **Immutable audit log** — all state changes (users, contracts, keys) are append-only and tamper-evident.
- **Secrets** — passwords hashed with Argon2; API keys and MFA secrets stored hashed/encrypted.
- **Data sovereignty** — self-hosted; data never leaves your environment.

11. Troubleshooting

Symptom	Likely cause & fix
"Cannot save..." errors, or empty dropdowns (locations, etc.) after an update	The database is missing new columns/tables. Recreate it: <code>docker compose down -v</code> then <code>docker compose up -d --build</code> (dev only — this deletes data). Schema changes only apply on a fresh database volume.
Setup screen doesn't appear; goes to login	An admin already exists. Go to /set up directly, or reset the database with <code>docker compose down -v && up -d</code> .
"Cannot reach the API"	The gateway isn't running or isn't reachable. Check <code>docker compose ps</code> and <code>docker compose logs api-gateway</code> .
Gateway exits immediately	A required value in <code>.env</code> is missing, or a dependency isn't installed. Check the gateway logs for the traceback.
'ContainerConfig' error on up	Legacy docker-compose v1 bug. Use Docker Compose v2 (<code>docker compose</code>), or always run <code>down</code> before <code>up</code> .
App works locally but not from another PC	Use the host machine's IP address in the URL, not <code>localhost</code> .

Appendix A — Demo accounts (optional seed)

The optional demo seed (`python -m app.seed`) creates sample staff for trying the Maker-Checker workflow. It does **not** create an administrator — create that via the setup screen.

Role	Email	Password
HR / Payroll	hr@synapse.test	Passw0rd!hr
Manager	manager@synapse.test	Passw0rd!mgr
Employee	emp@synapse.test	Passw0rd!emp

Change or remove these before any real deployment.

Appendix B — Quick command reference

```
docker compose up -d --build      # start the stack
docker compose exec api-gateway \
  python -m app.seed              # optional demo data
docker compose ps                 # container status
docker compose logs api-gateway   # gateway logs
docker compose down -v            # stop + wipe data
```