Users, Roles & Access

Managing who can sign in and what they can see and do.

- Managing Users
- Roles, Teams & Permissions

Managing Users

What a user is

Anyone who signs into the CRM has a **user** record. Each person should have **their own account** — never a shared login. Individual accounts keep the activity history accurate (who logged which session, who changed what) and let you remove someone's access cleanly when they leave.

Types of user

Type	Who it's for
Regular user	Staff and mentors who sign in and do everyday work.
Administrator	Can reach the Administration panel and change system settings. Grant this to as few people as possible.
API user	Non-human accounts used by integrations (e.g. the website or documentation tooling). You'll rarely create these.
Portal user	External people using a limited portal, if CBM uses one.

Adding a user

1. Administration → **Users** → **Create User**.
2. Enter their name, **User Name**, and **Email Address**.
3. Set the **Teams** and **Roles** that match their job — this is what controls what they can see and do (see Roles, Teams & Permissions).
4. Decide whether they're an administrator (usually **no**).
5. Save — then either set a password or have the CRM email them a setup link (recommended if email is configured).

> **Get teams and roles right at creation time.** They govern the user's access; skip them and the person may see far too much or too little.

Resetting a password

- Open the user → **Actions / ⋮ menu** → **Change Password**, or
- Have the person use **"Forgot password?"** on the login screen (requires outbound email — see Chapter 5 — Integrations).

When someone leaves: deactivate, don't delete

- Open the user and turn **Is Active** off. This blocks sign-in but **keeps all their history** intact and correctly attributed.
- **Avoid deleting users** — deletion can orphan or reassign the records they created. Deactivating is almost always the right move.

> Deactivate first. You can always reactivate; undoing a delete is far harder.

Roles, Teams & Permissions

The two halves of access

EspoCRM controls access with **Roles** and **Teams**, working together:

- **Roles** decide *what a user can do* — which record types they can see, create, edit, or delete, plus any administrative abilities.
- **Teams** decide *whose records a user can see* — records are shared with teams, and members of a team can see them.

A user can hold several roles and belong to several teams; their effective access is the **combination**.

Roles

- Administration → **Roles**.
- A role grants permissions per record type (Contacts, Engagements, Sessions, …) — typically **None / View / Edit / Create / Delete** — and a **scope** (all records, just their team's, or just their own).
- CBM's roles are already set up to match job functions. **Prefer assigning an existing role** over inventing new ones.

> **Least privilege:** give people the access their job needs and no more. Widening access later is easy; clawing it back after data has been over-exposed is not.

Teams

- Administration → **Teams**.
- Teams group users and act as the unit of record sharing.
- Add someone to a team from either the user record or the team record.

How they combine — an example

A mentor holds the *Mentor* role (view/edit engagements for their team, no delete) **and** belongs to the team that owns their engagements → they see and update *their* engagements and session notes, and nothing else.

Changing roles or teams safely

- A role or permission change affects **everyone** with that role at once.
- **Test in the Sandbox first** (the golden rule from Signing In & the Administration Panel): confirm a typical user sees exactly what you intend, then apply in Production.
- After the change, spot-check with a real user in that role.

> Permission changes that **broaden** access can expose data unintentionally. Treat them like configuration changes (Chapter 3), not casual edits.