Admin Setup Handbook

Model your organisation's hierarchy, compose roles from a shared capability catalogue, assign scoped access to people, and set your company profile, branding and shared settings — the platform-wide configuration every module builds on.

Version 1.0 · ixlcore.com

Reference

Admin Setup is where the platform is shaped to your business. Before a single invoice, quote or shift exists, an administrator models the organisation — its legal entities, branches, departments and positions — defines who may do what by composing roles from a shared capability catalogue and granting them at the right level, and records the company's own identity, branding and shared settings. Everything else in IXL CORE reads from this foundation: records are scoped to a level in the hierarchy, every action is checked against the access rules set here, and documents carry the company profile and logo configured here.

This guide is a reference for what Admin Setup does and how the pieces fit together. It describes IXL CORE **version 1.0**.

Overview

Admin Setup covers three connected areas:

- **Organisation structure** — the organisation and the entity !’ branch !’ department !’ position hierarchy beneath it.
- **Roles & permissions** — roles you compose from a shared capability catalogue, each capability carried at a defined data-scope level.
- **Users & role assignments** — granting a role to a member at a chosen scope, with optional start and end dates.
- **Platform settings** — the company profile, branding assets, document templates and shared setting values that the rest of the platform draws on.

Two ideas run through all three. First, a **five-level hierarchy** — organisation, entity, branch, department, position — that both structures your records and scopes access to them. Second, every administrative change is written to the **audit trail**, so who changed what, and when, is always recoverable.

Organisation structure

An **organisation** is the top of the tree — your tenant. It carries a name, a unique slug, an optional legal

name, an external reference and a description, and a status of **draft**, **active**, **inactive** or **archived**. Beneath it sits a strict hierarchy, each level nested inside the one above:

- An **entity** is a legal company within the organisation. It has a name, a **code** unique within the organisation, an optional legal name and registration number, a description, and the same four-way status.
- A **branch** belongs to an entity — a physical or operational location, with a name, a code unique within its entity, a description, a status, and an **is primary** flag to mark the head location.
- A **department** belongs to a branch, with a name, a code unique within its branch, a description and a status. Departments can nest: a department may name a **parent department** within the same branch, giving you a departmental tree.
- A **position** belongs to a department — a named seat with a title, a code unique within its department, a description and a status. A position may name the position it **reports to** (anywhere in the same organisation) and may reference a **job profile** it represents, so the org chart and HR's job library stay in step.

Codes are enforced unique only within their parent, so the same code can recur under different entities or branches without clashing. This tree is not cosmetic: it is the same set of levels — organisation, entity, branch, department, position — that scopes every record and every access grant across the platform.

Typical steps

1. Create the **organisation**, giving it a name and slug.
2. Add each **entity** (legal company) beneath it, then the **branches** for each entity.
3. Within each branch, add **departments** (nesting them where useful) and the **positions** they contain, wiring reporting lines and job profiles as you go.

Roles & permissions

Access is built in three layers: capabilities, permissions and roles.

A **capability** is a single named ability — for example the right to view leads or manage settings — and belongs to a functional **domain**. The capability catalogue is **global**: it is shared by every tenant and authored only by the platform operator, so a tenant administrator composes from it rather than inventing entries. Each capability declares which data-scope levels it may be exercised at.

A **permission** binds a capability to a specific **scope level** (organisation, entity, branch, department or position) with an **allow** effect, an optional set of constraints, and a status of **active**, **inactive** or **reserved**. The permission's scope level must be one the capability allows, and a capability may carry only one active permission per scope level and effect.

A **role** groups permissions under a name and a **code** (lower-case, dot/underscore/hyphen separated, unique within the organisation), with a description, a status, and **is system** and **is assignable** flags. You compose a role by attaching active permissions to it; only active, assignable roles may receive

permissions or be granted to users. Crucially, every access decision is enforced on the request itself, not merely hidden in the interface, and an administrator may only confer capabilities they already hold — you cannot attach a permission, or assign a role, that grants an ability you lack yourself. That ceiling stops privilege escalation by an admin curating around their own limits.

Typical steps

1. Browse the shared **capability catalogue** to find the abilities a role should carry.
2. Create a **permission** for each capability at the scope level you want it exercised.
3. Create a **role**, then **attach** those permissions to build up its rights.

Users & role assignments

Access reaches a person through a **role assignment**. You pick an existing organisation **member**, a role, and a **scope level**, and — for anything below organisation level — the exact entity, branch, department or position the grant applies to. The assignment can carry a **start** and **end** date, so access can be time-boxed.

Assignments are guarded on every side. The chosen user must already be a **member** of the organisation (you provision existing members into roles here; adding a brand-new person is a separate concern). The role must belong to the same organisation and be active and assignable. The scope you name must be internally consistent — a branch must sit under the named entity, a department under that branch, and so on — and each level of the hierarchy is checked against the organisation. Two further protections apply: an administrator may only assign a role whose every capability they already hold, and the platform refuses to expire the **last remaining** access-administration grant in an organisation, so a tenant can never lock itself out of its own access controls. Every assignment, and every later change to one, is written to the audit trail.

Typical steps

1. Choose a **member** of the organisation to grant access to.
2. Select the **role** and the **scope level**, naming the entity, branch, department or position where required.
3. Optionally set **start and end dates** to time-box the grant, then save — the grant is audited.

Platform settings

Settings record the company's own identity and the shared preferences the platform reads.

The **company profile** holds structured identity on the organisation record: name, legal and trading name, registration and tax numbers, company email, phone and website, and a full address (street, city, region, postal code, country). Viewing it needs the `settings.view` capability and editing it needs `settings.manage`, and every save is audited.

Branding stores the company **logo** and **favicon** — uploaded as real raster images (PNG, JPG, WebP;

SVG is deliberately refused to avoid stored-script risk, and there are size and dimension limits). These assets render in-app and are embedded in branded documents such as letterheads, proposals and the public signing page. The brand colour lives alongside as a shared setting.

Document templates provide the branded layouts those documents are built from, and a template can be set as the default. Finally, **shared settings** are definition-and-value pairs: a global catalogue of setting definitions (each declaring the scope levels it may be set at) against which your organisation stores values at the organisation, entity, branch or department level. Because a value can be set at several levels, the platform resolves an **effective value** for any given scope by walking the hierarchy — so a department inherits the organisation's setting unless a more specific value overrides it. All of this is governed by the same `settings.view` / `settings.manage` capabilities and audited on change.

Typical steps

1. Complete the **company profile** with the legal identity and contact details documents will carry.
2. Upload the **logo** and **favicon**, and set a default **document template**.
3. Set **shared setting values** at the level you want them to apply, letting more specific scopes override broader ones.

How Admin Setup connects

Admin Setup is the platform foundation the other modules stand on:

- **Every module** scopes its records to the organisation !’ entity !’ branch !’ department !’ position hierarchy defined here, and checks every action against the roles and permissions set here.
- **The access engine** enforces capabilities on each request across the platform — from `crm.leads.view` to `settings.manage` — with the same anti-escalation and last-admin protections applied everywhere.
- **Branded documents** — proposals, contracts, letterheads and the public signing page — draw the company profile, logo and default template from Settings.
- **HR's job library** links to positions through the job profile a seat represents, keeping the org chart and workforce data aligned.

Configure this foundation once, and the rest of the business inherits its structure, its access rules and its identity by default.

How-to guides

Set up your organisation structure

Model your organisation as a five-level hierarchy — organisation, entity, branch, department and position — that both structures your records and scopes access to them.

Model your organisation as a strict five-level tree — organisation, then entity, branch, department and position — so every record and access grant across the platform can be scoped to the right level.

Before you start

- You need the `settings.manage` capability (or an equivalent administration role) at the level you are working in.
- Have your legal companies, their locations, and the departments and seats within them mapped out before you begin.
- Remember that **codes** are enforced unique only within their parent, so the same code can safely recur under a different entity or branch.

Steps

Create the organisation

1. Open **Admin Setup !' Organisation** and create the organisation — this is your tenant, the top of the tree.
2. Enter the **Name** (required) and a unique **Slug** (required).
3. Choose a **Status** (required): draft, active, inactive or archived.
4. Optionally add a **Legal name**, an **External reference** and a **Description**. Save.[screenshot: Create organisation]

Add an entity

1. Beneath the organisation, add an **Entity** — a legal company within it.
2. Enter the **Name** (required) and a **Code** (required, unique within the organisation).
3. Choose a **Status** (required). Optionally add a **Legal name**, a **Registration number** and a **Description**. Save.

Add a branch

1. Within an entity, add a **Branch** — a physical or operational location.
2. Enter the **Name** (required) and a **Code** (required, unique within the entity).
3. Choose a **Status** (required). Optionally add a **Description** and mark it **Is primary** to flag the head location. Save.

Add a department

1. Within a branch, add a **Department**.
2. Enter the **Name** (required) and a **Code** (required, unique within the branch).
3. Choose a **Status** (required). Optionally add a **Description** and a **Parent department** (within the same branch) to nest departments into a tree. Save.

Add a position

1. Within a department, add a **Position** — a named seat.
2. Enter the **Title** (required) and a **Code** (required, unique within the department).
3. Choose a **Status** (required). Optionally add a **Description**, a **Reports to** position (anywhere in the same organisation) and a **Job profile** the seat represents. Save.[screenshot: Position with reporting line]

Result

Your organisation hierarchy is in place. These same five levels scope every record and every access grant across the platform, and each change is written to the audit trail.

Related

- Admin Setup reference
- Create roles and permissions
- Assign roles to your team

Create roles and permissions

Compose roles from the shared capability catalogue by binding capabilities to a scope level as permissions, then attaching those permissions to named roles.

Build the roles your people will hold by composing them from the shared capability catalogue — you bind capabilities to a scope level as permissions, then group those permissions under a named role.

Before you start

- You need an administration role that already holds the capabilities you intend to confer. You may only attach capabilities you hold yourself — an admin cannot grant an ability they lack.
- The **capability catalogue is global**: it is shared by every tenant and authored only by the platform operator. You compose from it; you cannot create or edit capability entries, and authoring a permission entry is likewise a platform-operator concern.

Steps

Browse the capability catalogue

1. Open **Admin Setup !’ Access !’ Capabilities** and review the abilities available, grouped by domain (for example `crm.leads.view` or `settings.manage`).
2. Note the scope levels each capability may be exercised at — a permission's scope must be one the capability allows.[screenshot: Capability catalogue]

Understand permissions

A **permission** binds a capability to a **scope level** — organisation, entity, branch, department or position — with an **allow** effect, an optional set of **constraints**, and a **status** of active, inactive or reserved. A capability may carry only one active permission per scope level and effect. Tenant administrators compose roles from existing permissions rather than authoring new catalogue entries.

Create a role

1. Open **Admin Setup !’ Access !’ Roles** and create a role.
2. Enter the **Name** (required) and a **Code** (required, unique within the organisation). The code must be lower-case, using dots, underscores or hyphens between segments — for example `sales.manager`.
3. Choose a **Status** (required): active, inactive or reserved.
4. Optionally add a **Description**, and set the **Is system** and **Is assignable** flags. Save.[screenshot: Create role]

Attach permissions to the role

1. Open the role and attach a **Permission** (required — select an existing permission).
2. Only **active** permissions may be attached, and only **active, assignable** roles may receive them.
3. Repeat to build up the role's rights. Every change is audited.

Result

Your role carries the permissions that define what its holders may do, each capability enforced on the request itself rather than merely hidden in the interface. The role is ready to grant to your team.

Related

- Admin Setup reference
- Set up your organisation structure
- Assign roles to your team

Assign roles to your team

Grant a role to an existing organisation member at a chosen scope, with optional start and end dates, guarded by scope consistency, an anti-escalation ceiling and a last-admin protection.

Give people access by assigning a role to an existing member of your organisation at the exact level it should apply. There is no email invitation step — you assign roles to members who already belong to the organisation, not to new people by email.

Before you start

- The person must already be a **member** of the organisation. Membership means they are an employee in the organisation or already hold a role assignment there; assigning a role provisions an existing member, it is not an invite flow.
- You may only assign a role whose every capability you already hold yourself — this anti-escalation ceiling stops an admin curating access around their own limits.
- The role must belong to the same organisation and be **active and assignable**.

Steps

Choose the member and role

1. Open **Admin Setup !’ Access !’ Assignments** and create an assignment.
2. Select the **User** (required) — an existing member of this organisation.
3. Select the **Role** (required).[screenshot: New role assignment]

Set the scope

1. Choose the **Scope level** (required): organisation, entity, branch, department or position.
2.
3. Name the scope targets the level requires:
   The scope must be internally consistent — the branch must sit under the named entity, the department under that branch, and each level must belong to this organisation. A field that is not permitted for the chosen scope level is rejected.

Time-box the grant (optional)

1. Optionally set a **Starts at** date and an **Ends at** date (the end date must be on or after the start date).
2. Save — the assignment, and any later change to it, is written to the audit trail.

Result

The member now holds the role at the scope you chose, for the period you set. Two protections stay in force: the anti-escalation ceiling on what you may confer, and a **last-admin guard** that refuses to expire the final remaining access-administration grant, so the organisation can never lock itself out of its own access controls.

Related

- Admin Setup reference
- Create roles and permissions
- Set up your organisation structure

Set your company profile and branding

Record your company's legal identity and contact details on the organisation record, then upload the logo and favicon that render in-app and on branded documents.

Record the company identity and branding that the rest of the platform reads — the profile documents will carry, and the logo and favicon that render in-app and on letterheads, proposals and the public signing page.

Before you start

- Viewing the profile needs the `settings.view` capability; editing it needs `settings.manage`. Every save is audited.
- Have your legal identity, tax registration and contact details ready, along with logo and favicon image files.

Steps

Complete the company profile

1. Open **Admin Setup !’ Settings !’ Company profile**
2. Enter the **Name** (required).
3. Optionally add the **Legal name**, **Trading name**, **Registration number** and **Tax number**.
4. Optionally add the **Company email**, **Company phone** and **Website**.
5. Optionally add the address: **Street**, **City**, **Region**, **Postal code** and **Country**.
6. Save — the change is audited.[screenshot: Company profile form]

Upload your logo

1. Open **Admin Setup !’ Settings !’ Branding**
2. Upload the **Logo**. It must be a real raster image — **PNG, JPG/JPEG or WebP** (SVG is deliberately refused to avoid stored-script risk). Size and dimension limits apply.[screenshot: Branding assets]

Upload your favicon

1. Upload the **Favicon**. Accepted formats are **PNG or WebP**, within the applicable size and dimension limits.
2. You can remove either asset later; each upload or removal is audited.

Result

Your company profile and branding are set. The profile provides the legal identity and contact details on branded documents, and the logo and favicon render in-app and are embedded in letterheads, proposals and the public signing page. The brand colour is kept alongside as a shared setting.

Related

- Admin Setup reference
- Manage shared settings
- Set up your organisation structure

Manage shared settings

Store values against centrally registered setting definitions at the organisation, entity, branch or department level, letting more specific scopes override broader ones.

Set the shared preferences the platform reads by storing values against centrally registered setting definitions — at the organisation level, or at a more specific scope that overrides it.

Before you start

- You need `settings.view` to read and `settings.manage` to change shared settings. Every change is audited.
- A **setting definition** is a central, global entry that declares its **value type**, the **scope levels** it may be set at, and whether **overrides** below the organisation are permitted. You store values against these definitions; you do not create the definitions themselves.
- Secrets and provider credentials are out of scope for shared settings — values containing keys such as `secret`, `token`, `password` or `api_key` are rejected.

Steps

Pick the setting definition

1. Open **Admin Setup !' Settings !' Shared settings** and choose the **Setting definition** (required). It must be an active, centrally registered definition.
2. Note its value type — the value you enter is checked against it (for example a boolean, an integer, a three-letter currency code, a locale such as `en-US`, an enum from the registered allowed values, or a start/end time range in HH:MM).[screenshot: Shared settings list]

Choose the scope

1. Choose the **Scope level** (required): organisation, entity, branch or department.
2.
3. Name the targets the level requires:
   The scope must belong to your organisation and be internally consistent. If the definition does not permit lower-level overrides, only the organisation level is accepted.

Enter the value

1. Enter the **Value** (required — it must be present) in the type the definition expects.
2. Optionally add an **Override reason** and a **Status**.
3. Save — the change is audited.[screenshot: Set shared setting value]

Result

Your value is stored at the chosen scope. Because a value can be set at several levels, the platform resolves an **effective value** for any scope by walking the hierarchy — a department inherits the organisation's setting unless a more specific value overrides it.

Related

- Admin Setup reference
- Set your company profile and branding
- Set up your organisation structure