أسس
OSOS

OSOS DOCUMENTATION

Administration

A section of the OSOS Documentation and Help Center

English edition

Platform release 1.10
Published 23 November 2026

Guides · How-to

Contents

Start here: Administration 3

Guides

Managing users 5
Managing groups 11
Managing roles 15
Organization units 18
Permissions & access 21
Sensitive Permissions 27
Delegation 36
Organization settings 41
Working hours & holidays 46
Spaces & domains 50
Human verification (bot protection) 53
Look and feel: themes, fonts and icons 56
Monitoring: logs & audit 59
Notifications & email templates 64
Your license & plan limits 68
Users from your company directory 69
Sign-in methods & single sign-on 73

How-to

Manage permissions 81
Let the HR group see only HR requests 84
Create an API token for an integration 87

Videos

Video: Manage permissions (incl. sensitive fields) 90

Administration FAQ 91

Start here: Administration

This section is for **tenant administrators** — the people who run OSOS for their organization day to day. You decide how people sign in, how your organization is structured, who can do what, and how the platform looks and communicates. You do this in the product, under **Administration**. You do not need to install or configure servers.

The examples throughout this section follow an HR department at a mid-size company: an **HR** group, an **HR Manager** role, and HR services such as leave requests.

Before you begin

Most administration screens need the system **Admin** level: **Organization** and **Access Management** do not appear in the menu without it. If you cannot see them, ask whoever set up your platform to give you Admin. See Permissions & access.

Your first-day setup checklist

Work through these in order. Each step builds on the one before it.

1. **Choose how people sign in.** Decide on passwords, passkeys and two-step verification, and whether people sign in with your company account (single sign-on). → Sign-in methods & single sign-on
2. **Set up your spaces and organization structure.** Create your spaces, then your departments and units in the org chart. → Spaces & domains · Organization units
3. **Create roles and groups.** Roles for positions (*HR Manager*), groups for teams (*HR*). → Managing roles · Managing groups
4. **Add your users.** Add people one by one, or ask your operator to import them from your company directory. Then assign them to roles and add them to groups. → Managing users · Users from your company directory
5. **Give access.** Give every person a base level, then give each group and role what its work needs. → Permissions & access · Manage permissions
6. **Make it look like your organization.** Logo, colours, fonts, languages, time zone and date formats. → Organization settings · Interfaces & look and feel
7. **Set up notifications.** Email and message templates, sender details and channels. → Notification configuration
8. **Know where to look when something goes wrong.** Message logs, the audit trail and automation logs. → Monitoring: logs & audit

Common tasks

- Give a group access to a module, entity, view or report
- Let the HR group see only HR requests
- Check what someone can access
- Let only HR Managers see salaries
- Let someone act for a colleague while they are away
- Set working hours and holidays
- Check your license and plan limits

Questions not answered here? Try the Administration FAQ.

What is not your job

Some things are set up outside **Administration**, by other people:

Task	Who does it
Installing, upgrading and backing up the platform; servers, networking and certificates	Your **operator** (the team that runs the platform)
Configuring the identity server, registering single sign-on providers, running directory imports	Your **operator** — you decide what is needed and give them the details
Your license and plan	Your **vendor** or operator
Designing applications, data entities, views, pages, services and workflows	Your **app builders** — see the Development section [Build Apps › Build apps — start here]

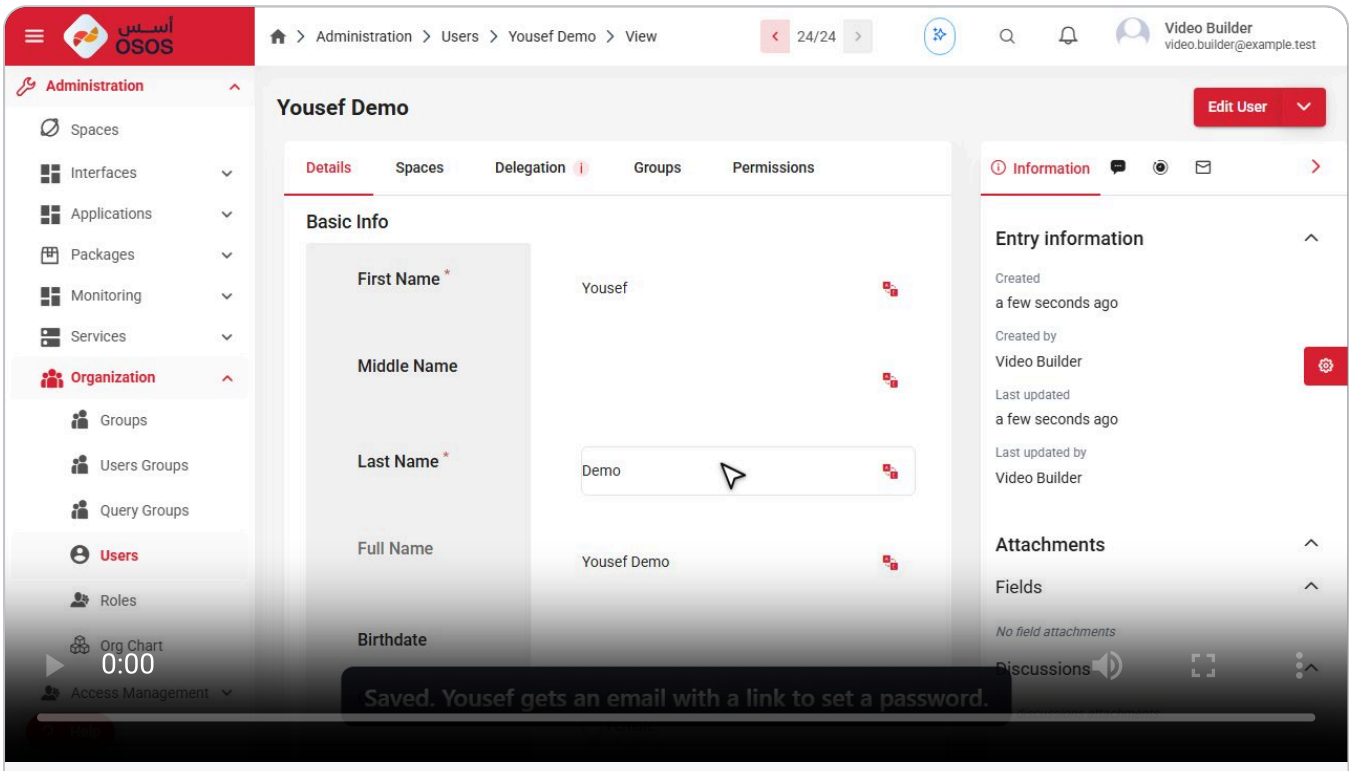
When you need one of these, tell the right person what you need in business terms — for example, “staff should sign in with their Microsoft work account” or “import everyone in the HR department”.

Go deeper

For operators: Setup overview · Deployment · Tenant provisioning

Managing users

This guide is for tenant administrators. It shows you how to add people to OSOS, get a new user to the point where they can actually work, look after their account day to day, and close it down cleanly when they leave. It also covers **application users**, the accounts that external systems use to call OSOS.



HOW-TO **Add a user** A new colleague, ready to sign in · 0:40 · Recorded on OSOS 1.11.0-alpha.1

The examples follow an HR department: Lina joins HR as an HR officer, and later Omar leaves the company.

How it works

Every person in OSOS is a record in the **Users** list. When you save a new user, OSOS creates their sign-in account for you and emails them a link to set their own password. You never see or set a password yourself.

A user record on its own lets someone sign in, but not much more. What they can see and do comes from:

- their **system level** (every person needs at least **Restricted**),
- the **groups** they are in and the **roles** they hold in the org chart,
- the **spaces** they are a member of.

So adding a person is always two jobs: create the user, then set them up.

There are two kinds of user:

User type	Who it is for	How it signs in
Normal	A person.	With a password, passkey or single sign-on, depending on your sign-in settings.

User type	Who it is for	How it signs in
Application	An external system, such as an HR or finance system that sends data to OSOS.	With an API token. It never uses the sign-in page.

People can also come from your company directory instead of being added by hand. See Users from your company directory.

Before you begin

- You need the system **Admin** level. Without it the **Organization** menu does not appear.
- The person needs a unique username and a unique email address. The email address is where the set-password link goes.

Add a user

Example: add Lina, the new HR officer.

1. Open **Administration › Organization › Users**.
2. Select **Add User**.
3. In **Basic Info**, enter **First Name** and **Last Name** (and **Middle Name** if there is one). **Full Name** fills in by itself. Optionally add **Birthdate**, **Gender** and a **Profile Picture**.
4. Set **User type** to **Normal**.
5. In **Account Information**:
    - Enter a **Username** and an **Email Address**. Both must be unique.
    - Set **Organization Account** to **Yes** for your own staff. Only organization accounts can be given a role in the org chart.
6. Select **Save**. A green message confirms that the user was saved. A red message tells you what is missing or wrong.

Lina receives an email with a link to set her password. If she does not get it, see Troubleshooting.

Set up a new user so they can work

Do these steps for every new person, in this order. Skipping step 1 is the most common reason a new user “can’t see anything”.

1. **Give them a system level of at least Restricted.** A person with no system level is refused almost everywhere, even when a group gives them access to something. The easiest way is to give **Restricted** to a group everyone belongs to, once; then step 2 covers this automatically. See Give everyone a base level.
2. **Add them to their groups.** For example, add Lina to the **HR** group. See Managing groups.
3. **Give them their role in the org chart.** For example, *HR Officer* in the HR department, as their **Main** assignment. This also sets their org unit and their direct manager, which workflows use to route approvals. See Assign people to a role.
4. **Add them to the spaces they work in**, if your organization uses more than one space. See Spaces & domains.
5. **Check the result.** Open **Administration › Access Management › User Permissions**, choose the person, and check that they have what their job needs. See Check what someone can access.

Find and open a user

1. Open **Administration › Organization › Users**.
2. Search for the person, or filter the list. You can switch between table and card layouts.
3. Select the person's row to open their record.

The record shows their details, account information, org unit, roles and direct manager. Related lists on the record show their groups and other linked records. The **Discussion** area lets you leave notes for other administrators and mention them with **@**, and the **History** shows every change made to the record.

Edit a user

1. In the users list, select the three dots (**...**) at the end of the person's row and choose **Edit**.
2. Make your changes and select **Save**.

Changes to the username and email address apply to the person's sign-in account straight away. Editing a user never changes their password. If you leave the username or email empty, the sign-in account keeps its current value.

Reset a user's password

When someone has forgotten their password and cannot use **Forgot password?** on the sign-in page themselves:

1. In the users list, select the three dots (**...**) on the person's row and choose **Reset Password**.
2. Confirm with **Reset**.

You see "An email with a reset password link was sent to the user". The person follows the link to choose a new password. Only the newest link works.

Do not use this for people who sign in through single sign-on or who were imported from your company directory. They have no OSOS password to reset: their password is managed in your identity provider.

Deactivate or reactivate an account

Deactivate an account to stop someone signing in while keeping everything about them: their record, history, groups and roles. Use it for a leaver, a long absence or a suspected compromise.

1. In the users list, select the three dots (**...**) on the person's row and choose **Deactivate Account**.
2. Confirm with **Deactivate**.

The person is signed out everywhere at once, on every device, and cannot sign in by any method. If they try, they see "The account is deactivated, Please contact with your system administrator".

To let them back in, choose **Activate Account** on the same menu. They sign in with their existing password.

You cannot deactivate your own account.

Archive and restore a user

Archive a user when the account should no longer exist at all, for example a duplicate or a test account. Archiving deletes the person's sign-in account, including their password, passkeys and two-factor settings. The user record stays in the archive.

1. In the users list, select the three dots (**...**) on the person's row and choose **Archive**.
2. Confirm.

To bring an archived user back, filter the users list to **Archived**, select the three dots (**...**) on the row and choose **Restore Archived Records**. OSOS creates a new sign-in account and emails the person a link to set a new password. Their old password does not work any more. `Since 1.6`

> **Deactivate or archive?**
>
> For people who leave, **deactivate**. It is instant, reversible, and keeps the account linked to everything the person did. Archive only accounts that should never have existed.

Offboard someone who leaves

Example: Omar leaves the company on Friday. Do this on his last day, or straight after.

1. **Deactivate his account.** See Deactivate or reactivate an account. He is signed out everywhere immediately.
2. **Hand over his open work.** Requests that are waiting on Omar stay with him until someone moves them. For each one, open the request, open **Admin Actions**, choose **Change Owner**, and pick the colleague who takes over.
3. **End his delegations.** Open any delegation where Omar is the **Delegator** or the **Delegated To** person and set its **To Date** to today, or set **Is Active** to **Inactive**. See Delegation.
4. **Remove him from his groups and roles.** Deactivating does not take him out of groups or the org chart. If a workflow step is owned by a group, a deactivated member can still be picked to own it, so remove him. See Managing groups and Managing roles.
5. **Remove access given to him directly.** In **Administration › Access Management › User Permissions**, choose Omar and remove any grants marked **Direct**.
6. **Find a new manager for his team.** If Omar managed an org unit, give the unit a new manager so approvals keep flowing. See Organization units.

If Omar was imported from your company directory, also make sure he is removed or disabled there. The directory import does not deactivate people in OSOS when they leave the directory.

Application users

An application user lets another system call OSOS with an **API token**. OSOS treats each call as made by that application user, so it can do only what you give the application user access to, exactly like a person.

Add an application user

1. Open **Administration › Organization › Users** and select **Add User**.
2. In **Basic Info**, enter a name and set **User type** to **Application**.
3. Fill in **Account Information** as for a person.
4. In **Application information**, enter the application's **Name**. Choose whether the token should never expire, or enter an **Expiry date**.
5. Select **Save**.

The token is emailed to **you**, the administrator who created the application user. It is never shown on screen. Hand it to the team that runs the other system through a secure channel, and give the application user a system level and the access it needs, as for a person.

Regenerate an application's token `Since 1.8`

Regenerate the token when it may have leaked, or when it is about to expire.

1. In the users list, select the three dots (**...**) on the application user's row and choose **Regenerate Token**.
2. Confirm.

The new token is emailed to you. The old token keeps working for a short grace period, so the other system has time to switch over. The application keeps its identity, access and history. You cannot regenerate the token of an archived application user. You can for a deactivated one.

Token expiry `Since 1.10`

An application user's **Expiry date** is shown and editable on the form when the token is not set to never expire. The token stops working after that date. It is optional: leave it empty, or choose no expiry, for a token that does not expire.

If an application user's token stops working after an upgrade to 1.10, open the application user, enter the **Expiry date** again and save.

> **Go deeper**
>
> For integrators: API tokens [Integrate & Extend › Use an API token] · Users API & identity accounts [Integrate & Extend]

Good to know

- **Passwords** must be at least 8 characters, with upper- and lower-case letters, a digit and a symbol. When people change their password, the new one cannot be the same as the old one.
- **Locked accounts.** After 5 wrong passwords in a row an account is locked for a few minutes, and the lock grows longer if it happens again. People see "Account is locked". There is no unlock button: the lock ends by itself, and a successful sign-in clears it.
- **Usernames and email addresses are not case-sensitive.** `Lina@example.com` and `lina@example.com` are the same address.
- **Two-factor settings, passkeys and personal preferences** (language, theme, time zone, date format) belong to the person. They manage them on their own **Security** and **User Preferences** pages. See Account security [Using OSOS › Account Security & Revealing Protected Values].
- **Adding fields to users.** The Users entity is a data entity like any other, so an app builder can add fields, views and relations to it. See Data modeling: entities [Build Apps › Create and design an entity].

Troubleshooting

What you see	What to check
A new user never got the set-password email.	Check the email address on the record. If another account already uses that address, no sign-in account is created for the new user: change one of the addresses. Also check the email log in **Administration › Monitoring › Logs**. Then use **Reset Password** to send a new link.

What you see	What to check
A new user signs in but sees nothing.	They have no system level. Give them at least **Restricted**, usually through a group. See step 1.
"User must be an organization account" when you assign a role.	Set **Organization Account** to **Yes** on the user, then assign the role again.
A single sign-on or directory user asks for a password reset.	They have no OSOS password. Send them to whoever manages your company identity provider.
Deactivate Account is missing.	You opened your own row, or the account is already deactivated (look for **Activate Account**).
An application user's token stopped working.	Check its **Expiry date**, and whether it was deactivated or its token was regenerated.

Go deeper

For integrators: Users API & identity accounts [Integrate & Extend] · For operators: Identity server · For end users: Account security [Using OSOS › Account Security & Revealing Protected Values]

Managing groups

This guide is for tenant administrators. A group lets you treat a team as one: you give access, assign work and send notifications to the group, and every member gets it. You will learn which kind of group to use, and how to create one, fill it and look after it.

The example is the **HR** group, which holds everyone in the HR department.

How it works

Instead of giving each person the same access one by one, you give it once to a group. Everyone in the group gets it, and when someone joins or leaves the team you only change the membership. A person's access is the highest of their own access and the access of every group they are in.

Groups are also used outside permissions. A builder can make a group the owner of a workflow step, the recipient of a notification or the audience of a comment, and OSOS works out the people in the group at that moment.

Kinds of group

The **Groups** list shows every kind of group in one place. You create and edit only **Users Groups** and **Query Groups** yourself. The others fill in automatically.

Kind	Who is in it	How you manage it
Users Group	The people you add.	**Administration › Organization › Users Groups**. This is the usual kind.
Query Group	Every user who matches a saved query, for example "everyone whose job title contains Manager". People join and leave automatically as their details change.	**Administration › Organization › Query Groups**.
Role Group	Everyone who holds a role, such as *HR Manager*. Every role has one.	Assign people to the role. See Managing roles.
Org Unit Group	Everyone placed in a unit of your org chart. Every unit has one.	Assign people to the unit. See Organization units.

The system also keeps a few internal groups of its own. You cannot create or change them.

A department: group, role, org unit or query group?

Example: you want "the HR department" to see the HR reports and receive HR notifications.

- **The HR department is already a unit in your org chart?** Use its **Org Unit Group**. Membership follows the org chart, so you maintain it in one place.
- **It is a position, not a team** (every *HR Manager*, wherever they sit)? Use the **Role Group**.
- **It cuts across the org chart** (an HR committee with people from several departments)? Create a **Users Group** and add the people.
- **It can be described by a rule** (everyone whose department field is *HR*)? Create a **Query Group** so nobody has to maintain the list.
- **The people come from your company directory?** Use a **Users Group** that the directory fills. See Users from your company directory.

A group is a set of *people*. Do not confuse it with a **list**, which is a set of records or values. See Collections & lists [Using OSOS › Lists: Keep Records Together].

Before you begin

- You need the system **Admin** level to see the **Organization** menu.
- The people you want to add must already exist as users. See Managing users.

Create a group

1. Open **Administration › Organization › Users Groups**.
2. Select **Add Users group**.
3. Enter the **Group Name** (for example *HR*), and optionally a **Group Description** and a **Group Icon**.
4. Select **Save**.

Then add the members, as below.

Add or remove members

1. Open **Administration › Organization › Users Groups** and open the group.
2. Open the group's **Users** tab.
3. Add the people who belong to the group, or remove the ones who have left.

You can also work from the person's side: open the user in **Administration › Organization › Users** and use their **Groups** tab.

A person is in a group only once, however many times they are added. Only active users count as members: an archived user drops out of every group automatically.

Create a query group

A query group needs a saved query on the **Users** entity that picks the right people. If there is none yet, ask an app builder to create one. See Queries [Build Apps › Queries & filters].

1. Open **Administration › Organization › Query Groups**.
2. Select **Add User Query Group**.
3. Enter the **Group Name**, and optionally a **Description** and an **Icon**.
4. In **query**, choose the saved query.
5. Select **Save**.

OSOS fills the group from the query, and keeps it up to date as people's details change. You do not add or remove members by hand.

Find and view groups

- **Every kind of group:** open **Administration › Organization › Groups**. The **Group Type** column tells you which kind each one is.
- **Only the groups you created:** open **Users Groups** or **Query Groups**.

Select a group to see its details. In each list you can search by name, switch between table and card layouts, sort, and filter by status, for example to show archived groups.

People who are not administrators can see the groups linked to their own account and roles, if you or a builder gives them a view that shows them.

Edit or archive a group

Edit. In **Users Groups** (or **Query Groups**), select the three dots (**...**) at the end of the group's row, choose **Edit**, make your changes and select **Save**.

Archive. Select the three dots (**...**) on the group's row and choose **Archive**, then confirm.

Before you archive a group, check what still uses it: its permissions (**Administration › Access Management › Group Permissions**), and any workflow step or notification a builder pointed at it. Point those at another group first.

Role groups and org unit groups cannot be edited or archived. They change when you change the role or the unit.

Give a group access

Give access to a group on the **Access Management** screens. See Permissions & access and, for a worked example, Let the HR group see only HR requests.

Good to know

When a group owns a workflow step

If a builder makes a group the owner of a workflow step, the step usually goes to **one** member of the group, not to all of them. Unless the builder chose otherwise, it is the first member in the group's member order. A builder can instead send the step to the manager of the group's org unit, let someone pick the owner by hand, or give it to every member in parallel. If work keeps landing on the same person, ask the builder how the step is set up. See Workflows [Build Apps › Workflows & approvals].

Because of this, remove people who leave from their groups promptly, even after you deactivate them. See Offboard someone who leaves.

Groups do not travel in packages

Groups, and the users in them, belong to each environment. When an app builder moves an application from a test environment to production as a package, its workflows and views keep their references to groups, but the groups themselves and their members are not copied. Make sure a group with the same name exists in the target environment, with the right people in it, before the package is installed.

Troubleshooting

What you see	What to check
A member of the group does not get the group's access.	The person may have no system level at all. Give them at least **Restricted**. See Give everyone a base level.
A person is missing from a query group.	Their details do not match the query. Open their user record and compare it with the query's conditions.

What you see	What to check
You cannot edit or archive a group.	It is a role group or an org unit group. Change the role or the org unit instead.
After a package install, a step or view points at a group that does not exist.	Create the group in this environment, or ask the builder to point the step or view at an existing group.

Go deeper

For integrators: Groups, org units & membership [Integrate & Extend] · For builders: Workflows: step owners [Build Apps › Workflows & approvals]

Managing roles

This guide is for tenant administrators. You will learn what roles are for, how to create and change them, how to assign people to a role, and how to give everyone in a role the same access in one step. The examples use an **HR Manager** role in an HR department.

What roles are for

A role describes a position or a responsibility — *HR Manager*, *Payroll Officer*, *Head of Department*. Roles help you in three ways:

- **Access.** Give a role access once, and everyone assigned to it has that access. When someone moves into or out of the role, their access follows automatically.
- **Workflows.** A workflow step can be owned by a role, so a request goes to whoever currently holds it.
- **Organization.** Roles are assigned within your org chart, so the platform knows who holds which position in which unit, and who manages each unit.

Behind every role there is a group that always contains the people assigned to the role. That is the group you choose when you give the role access.

> ⓘ **Two meanings of "role"**
>
> The roles on this page are the ones you create and name. They are not the same as a person's **system level** (Restricted, View, Developer, Edit, Admin), which you set in **Administration › Access Management › Admin Permissions**. See Permissions & access.

Find the roles list

Open **Administration › Organization › Roles**.

The list shows each role's name, description, how it is linked to the org chart, its type and its assignment type. You can sort, search and filter the list, and switch between table, cards and board views.

Create a role

1. Open **Administration › Organization › Roles**.
2. Select the add button at the top of the list.
3. Fill in the form:
    - **Role Name** — for example, *HR Manager*.
    - **Role Description** — optional; say what the role is responsible for.
    - **Linked With Org Unit** — where the role can be held:
        - **Not Linked with org unit** — the role is not tied to a place in your org chart, such as *Fire Warden*.

        - **Linked with one Org Unit** — the role exists in a single unit, such as *HR Manager* in the HR department.
        - **Linked with many Org Units** — the same role exists in several units, such as *Department Secretary*.

    - **Role Type**:

        - **job position** — a real position in the organization chart.
        - **mere role** — a responsibility someone holds alongside their position.

      A role that is not linked to any org unit cannot be a job position.

    - **Role Assignment Type**:

        - **Unique** — only one person can hold the role at a time, such as *HR Manager*.
        - **Multiple** — many people can hold it, such as *HR Officer*.

4. Select **Save**. A green message confirms the save; a red message tells you what to fix.

View, edit or archive a role

View. Open a role from the list. Its tabs show the role's **Details**, a **Discussion** where people can comment and attach files (the allowed file types and size — about 1 MB by default — are set per view), and the **History** of every change.

Edit.

1. In the roles list, select the three dots (**...**) at the end of the role's row and choose **Edit**.
2. Change the details and select **Save**.

Archive. Archive a role you no longer need.

1. In the roles list, select the three dots (**...**) at the end of the role's row and choose **Archive**.
2. Confirm by selecting **Archive**.

An archived role can no longer be assigned. Check that nobody still needs it before you archive it.

Assign people to a role

A role is always held in a unit of your org chart, so each assignment names the person, the role and the unit. Assignments are records of the **Assignment** data entity; the **Org Chart** page lists your units but does not show who holds which role.

1. Open **Administration › Applications › Data Entities**, open **Assignment** and select **Add Assignment**. The **New Assignment** form opens.
2. Fill in:
    - **Role** — for example, *HR Manager*.
    - **Assign Level** — **Main** for the person's main role, or **Additional** for a role they hold on top of it.
    - **Assign Type** — **Original** for a normal assignment, or **Delegation** when the person covers the role for someone else.
    - **User** — the person.

- **Org Unit** — the unit, for example the HR department.
- **Is Manager** — turn on if this person manages the unit.

3. Save.

The platform checks each assignment before it saves it — for example, it refuses a second holder for a **Unique** role. When assignments change, the unit's manager is updated for you. More on units and managers: Organization units.

Give a role access

Give access to a role exactly as you would to any group: choose the role's group wherever the screen asks for a group.

Example: let every HR Manager open the "Headcount by department" report.

1. Open **Administration › Access Management › Group Permissions**.
2. Under **Select Group**, choose the **HR Manager** role's group.
3. Select **Add Permission**, choose the report in **Object**, and save.

A person's access is the highest level they get from all their grants — their own, their groups', and their roles'. For the access levels, how grants add up, and more tasks, see Permissions & access.

Customize the roles entity

Roles are a data entity like any other, so an app builder can add fields, views and relations to it — for example, a *Grade* field. This follows the same steps as any other entity; see Data modeling: entities [Build Apps › Create and design an entity].

Troubleshooting

I can't assign a second person to a role. The role's **Role Assignment Type** is **Unique**. Change it to **Multiple** if several people should hold it.

The role cannot be saved as a job position. A job position must be linked to at least one org unit. Change **Linked With Org Unit**, or choose **mere role**.

Someone assigned to the role does not have its access. Check that the assignment is saved and active, and look at the person in **Administration › Access Management › User Permissions**: **Acquired Through** shows which roles and groups give them each permission.

> ⓘ **Go deeper**
>
> For integrators: Permissions API — Roles data model [Integrate & Extend › Permissions API] · For platform engineers: Engine: Permissions & Voters

Organization units

This guide is for tenant administrators. It shows you how to build your organization's structure, the **org chart**, in OSOS: create the units, arrange them under each other, place people in them and choose who manages each one. Workflows and notifications use this structure to find the right manager, so a correct org chart means approvals reach the right person without anyone picking them by hand.

The example is a mid-size company: **Executive Management** at the top, an **HR** department under it, and a **Recruitment** section under HR.

How it works

An **org unit** is one box of your org chart: a directorate, a department, a section or a team. Each unit has a **parent**, the unit it sits under, which turns the units into a tree with one unit at the top.

People are placed in units through **assignments**. An assignment says "this person holds this role in this unit". For example: *Lina* is *HR Officer* in *Recruitment*. Each person has one **Main** assignment, which is their home unit, and can have **Additional** ones on top. An assignment can also mark the person as the unit's **manager**.

The org chart drives three things:

What	How the org chart is used
A person's direct manager	OSOS starts at the person's home unit and walks up the tree to the nearest unit that has a manager other than the person.
Approvals that go "to the manager"	A builder can route a workflow step to the requester's manager, or to the manager two or more levels up, and OSOS finds that person in the org chart. Notifications can likewise go to a step owner's manager.
"People near me"	Some people pickers offer the people in your own unit, its parent unit, and every unit below yours.

Example. Lina's home unit is *Recruitment*, which has no manager of its own. Its parent, *HR*, is managed by Sara. So Sara is Lina's direct manager, and a leave request "to the direct manager" goes to Sara.

> ⓘ **Org units and your organization's branding**
>
> The **Organization** settings that hold your name, logo, header and footer are something else: they brand the platform and have no place in the org chart. See Organization settings.

Before you begin

- You need the system **Admin** level to see the **Organization** menu. Only administrators can change the org chart; others can at most view it.
- Create the **roles** you will assign people to first, for example *HR Officer* and *HR Manager*. See Managing roles.
- A new installation already has one unit at the top, **Executive Management**. Build the rest of the chart under it, or rename it.

Plan your levels

Levels are labels that say how deep a unit sits: *Directorate*, *Department*, *Section*. They help people read the chart and filter it. They do not change how the tree works; only **Parent** does.

A new installation has one level, **First level**. To add more, open **Administration › Applications › Lists**, open **Org Unit levels**, and add the levels you need.

Create a unit

Example: create the HR department under Executive Management.

1. Open **Administration › Organization › Org Chart**.
2. Select **Add Org unit**.
3. Enter the **Name** in each language, for example *HR*. Names must be unique: two units cannot have the same name.
4. Optionally add a **Description** and an **Icon**.
5. In **Parent**, choose the unit it sits under: *Executive Management*. Leave **Parent** empty only for the unit at the very top.
6. In **Org unit level**, choose the level, for example *Department*.
7. Turn on **Level Assignment is main** only if people may hold just their **Main** assignment in this unit, and no **Additional** ones.
8. Select **Save**.

Repeat for each unit, working from the top down: a unit's parent must exist before you can choose it.

To move a unit, with everything below it, to another part of the chart, edit the unit and change its **Parent**. OSOS refuses a parent that would put a unit under itself or under one of its own sub-units.

Place people in units

You place a person in a unit by giving them a role there.

1. Open **Administration › Organization › Org Chart** and open the unit.
2. Add an assignment, and fill in:
    - **Role**: for example *HR Officer*.
    - **Assign Level**: **Main** for the person's home unit, **Additional** for any other role they hold.
    - **Assign type**: **Original** for a normal assignment, or **Delegation** when the person covers the role for someone else.
    - **User**: the person.
    - **Is Manager**: turn on if this person manages the unit.
3. Save.

Give each person exactly **one Main** assignment. OSOS refuses a second one, and a Main assignment must use a role whose type is **job position**, not **mere role**. The user's record then shows their **Org unit**, **Main assignment role** and **Direct manager** for you. Full details, including the rules for unique roles, are in Assign people to a role.

Only users whose **Organization Account** is **Yes** can be assigned. See Add a user.

Set a manager

A unit's manager is whoever has a **Main** assignment in the unit with **Is Manager** turned on.

- **To choose a manager:** add or edit that person's Main assignment in the unit and turn on **Is Manager**.
- **To change the manager:** turn **Is Manager** off on the old manager's assignment and on for the new one.

The unit's **Manager** field updates by itself from the assignments. You never set it by hand.

A unit does not need a manager. People in a unit without one report to the nearest manager above them.

Good to know

- **Check approvals after you change the chart.** A builder may have set a step to go "two levels up". If the chart has fewer managers above the requester than that, the step is left with no owner. After a reorganization, test a request from someone in each changed unit.
- **When a manager leaves**, give their unit a new manager straight away, or approvals move up to the next manager above. See Offboard someone who leaves.
- **Each unit has a group.** Everyone placed in a unit is automatically in its org unit group, which you can use for access and notifications. See Managing groups.
- **Adding fields.** The org unit entity can be extended by an app builder like any other entity, for example with a cost-centre code. See Data modeling: entities [Build Apps › Create and design an entity].

Troubleshooting

What you see	What to check
"User must be an organization account" when you add an assignment.	Set **Organization Account** to **Yes** on the user.
You cannot add a second Main assignment for someone.	Each person has one home unit. Make the other one **Additional**, or change their existing Main assignment.
"role must be of type job position if the the assignment level is main"	The role is a **mere role**. Use **Additional** for it, or choose a job-position role for the Main assignment.
You cannot add an **Additional** assignment in a unit.	The unit has **Level Assignment is main** turned on.
A unit cannot be saved with a name.	Another unit already uses that name.
A request went to the wrong manager, or to nobody.	Check the requester's Main assignment and walk up the chart: the first unit above them with a manager is where it goes.

Go deeper

For integrators: Groups, org units & membership [Integrate & Extend] · For builders: Workflows [Build Apps › Workflows & approvals]

Permissions & access

This guide is for tenant administrators. It explains how access works in OSOS, where you manage it in **Administration**, and how to give people exactly the access their job needs — no more. The examples follow an HR department at a mid-size company: an **HR** group, an **HR Manager** role, and HR services such as leave requests.

How access works

Every permission answers three questions:

- **Who** gets access — a user, a group, a role or an organizational unit.
- **To what** — the whole platform, a module, a data entity, or a single view, page, report or service.
- **How much** — the access level: Restricted, View, Developer, Edit or Admin.

A person's access is everything they were given directly plus everything given to each group, role and org unit they belong to.

Who you can give access to

You grant to	Use it when
A user	One person needs something nobody else in their team needs. Use sparingly — it is hard to review later.
A group	A team needs the same access, such as the HR group. This is the usual choice.
A role	Everyone who holds a position needs it, such as every HR Manager. Each role has a group behind it that always contains the people assigned to the role. See Managing roles.
An org unit	Everyone in a part of your org chart needs it.
Anonymous visitors	People who are not signed in may use a public form or page. Use with care.

Query groups, whose members are picked by a saved rule, also appear as groups. See Managing groups.

What you can give access to

- **The whole platform (system level).** Every person needs a system level. It decides what they can do everywhere and which **Administration** menus they see.
- **A module** — and, through it, every data entity inside it.
- **A data entity** — one kind of record, such as Employees or Leave Balances. Someone with access to one entity can open its module, but they see only that entity inside it.
- **A view, page or report** — a published screen. You give View access to the people who should be able to open it.
- **A service** — who can submit it and who can watch its requests. Services work a little differently: see Services and their requests.

How much: the access levels

Level	What it lets someone do	Where you can give it
Restricted `Since 1.8`	Nothing on its own. It marks the person as a member, so that the access you give them elsewhere takes effect. On a module or entity, choosing Restricted takes away the access given there.	System level. On a module or entity it means "back to the base level".
View	See the design of a module or entity, and open the views, pages and reports shared with them. **It does not show the records themselves** in Administration.	System, module, entity, view, page, report
Developer	Build and change the design — fields, views, pages, themes — without seeing the records.	System, module, entity
Edit	Everything Developer can do, plus browse, open and change the records in Administration.	System, module, entity
Admin	Full control of the platform, including managing permissions.	System level only

> ⚠ **Seeing records needs Edit**
>
> To list, or to open, the records of an entity in **Administration**, a person needs **Edit** on that entity or its module, or Edit or Admin at the system level. View is not enough, not even for a single record. Most people never need this: they work with records through the views, pages and services you share with them, and those have their own access settings.

How grants add up

- **The highest level wins.** If Lina has View on the Employees entity directly and Edit through the HR group, she has Edit.
- **There is no "deny".** One grant never takes away access another grant gives. To take access away, remove the grant that gives it — and check the person's groups and roles too.
- **Removing a grant only removes that one grant.** If the person still gets access through a group, role or the system level, they keep it.
- **Admin counts only at the system level.** An Admin grant on a module or entity gives nothing; give View, Developer or Edit there instead.
- **Higher levels cover lower ones.** Edit or Admin at the system level covers every module and entity. Access to a module covers the entities inside it.

Services and their requests

Who sees a service's requests is set on the service itself, not with access levels:

Who	What they see
The service's Audience	They see the service in the **Services Catalog** and can submit it. The audience can be everyone, signed-in users only, or a list of users and groups.
The requester	Their own requests, in **Outbox**.
Step owners	Requests that reach a workflow step they own, or their group owns, in **Inbox**.
The service's Watchers	Every request of that service, in **Watching**, without having to act on them.

So to let the HR group follow all leave requests, make the HR group a **Watcher** of the leave service. The full worked example is Let the HR group see only HR requests.

Sharing views, pages and reports

A published view, page or report is either open to everyone who can reach it, or **Restricted** to a list of users and groups. When you add a user or group to a view, page or report, it switches to Restricted, and only the people and groups on its list can open it.

Where you manage access

All access screens are under **Administration › Access Management**. You need the system **Admin** level to see this menu.

Screen	Use it to
Admin Permissions	Set each user's and group's **system level** (Restricted, View, Developer, Edit, Admin).
User Permissions	Pick one person and see everything they can reach, and where each permission comes from. Add or remove their access to views, pages, reports and services.
Group Permissions	Pick one group and manage its access to views, pages, reports and services, see its admin permissions, and its sensitive fields.
Object Permissions	Pick one view, page, report or service and see who can open it.
Sensitive Permissions	Decide who may see masked fields such as salaries. See Sensitive fields.
Anonymous Permissions	Decide what people who are not signed in may open.

Access to a single module or data entity is managed on that module's or entity's own **Permissions** tab (**Administration › Applications › Modules** or **Data Entities**).

Tasks

Give everyone a base level

A person with no system level at all is refused almost everywhere. Give every person at least **Restricted**, usually through a group everyone belongs to.

1. Open **Administration › Access Management › Admin Permissions**.
2. Select the **Groups** tab, then select **Add**.
3. In **Select a group to add**, choose the group. In **Select permission level**, choose **Restricted**. Select **Confirm**.

Raise individual teams from there: for example, give your IT administrators **Admin**, and your app builders **Developer**.

Give a group access to a module or data entity

Example: the HR group maintains the Employees records.

1. Make sure the HR group has a system level (see Give everyone a base level). The module and entity screens list only users and groups that already have one.
2. Open **Administration › Applications › Data Entities** and open **Employees**. (For a whole module, open **Administration › Applications › Modules** and open the module instead.)
3. Open the **Permissions** tab and switch to **Groups**.
4. On the **HR** row, set **Permission Level** to **Edit** and confirm the change.

The levels offered are Restricted, View, Developer and Edit, and never less than what the group already has from its system level or, on an entity, from the entity's module. Groups with system **Edit** or **Admin** are not listed — they already reach everything.

Give a group access to a view, page, report or service

Example: let the HR group open the "Headcount by department" report.

1. Open **Administration › Access Management › Group Permissions**.
2. Under **Select Group**, choose **HR**. Stay on the **Group Permissions** tab.
3. Select **Add Permission**.
4. In **Object**, pick the report. For a service, also choose **Audience** (can submit it) or **Watcher** (sees all its requests).
5. Save.

The dialog warns you when the object will switch to Restricted mode: from then on only the people and groups on its list can open it.

> ⚠ **Known issue: a view granted here may not appear in the sidebar**
>
> When you give a group or a person access to a view from **Group Permissions** or **User Permissions**, the view (and its module) can be missing from their sidebar until the view is next opened in the view designer. To make it appear, give the access from the view's own **Permissions** dialog instead, or open the view once in the view designer (no need to save). The person may need to reload their window to see it.

Change or remove access

- **System level:** in **Admin Permissions**, change the **Permission Level** on the row, or select the trash icon to remove it, and confirm.
- **A module or entity:** on its **Permissions** tab, change the **Permission Level** and confirm. To take the access away, set it back to **Restricted**: the grant on that module or entity is removed and the person or group drops back to their system level.
- **A view, page, report or service:** in **Group Permissions** or **User Permissions**, use the row's **Edit** action, or its delete action to remove it.
- **Everything one group holds:** in **Group Permissions**, choose the group and open its **Admin Permissions** tab to change or remove each grant.

Removing a grant deletes it. It does not block the person: if they still get access through a group, a role or their system level, they keep it.

If you remove someone's only system level, they lose access to almost everything. To take away all of a person's access when they leave, deactivate or archive their account instead. See Managing users.

Check what someone can access

1. Open **Administration › Access Management › User Permissions**.
2. Under **Select User**, choose the person.
3. Read the **End User Permissions**, **Admin Permissions** and **Sensitive Fields** tabs.

For each permission, **Assignment Type** tells you how the person got it — **Direct**, **Indirect (Group)**, **Indirect (Authenticated)**, **Indirect (Delegation)** or **Indirect (Anonymous)** — and **Acquired Through** names the group or

role. **Available Actions** shows what the permission lets them do.

Worked example: an HR department

Need	What to set
Everyone can sign in and use the views and pages shared with them	**Restricted** at the system level for an all-staff group.
Everyone can also submit requests and act on them in their **Inbox**	**View** at the system level for that group instead. A person whose system level is **Restricted** has no **Services** menu (no **Services Catalog**, **Outbox** or **Inbox**). **View** is the lowest level that shows it, and it also shows a reduced **Administration** menu.
HR staff keep the Employees records up to date	**Edit** on the Employees entity for the **HR** group.
HR Managers can open the headcount report	Add the report for the **HR Manager** role's group in **Group Permissions**.
HR follows every leave request	**HR** group as a **Watcher** of the leave service — see the how-to.
Only HR Managers see salaries	A sensitive-field grant on the Salary field for the HR Manager role — see Sensitive Permissions.
Nobody in HR can change the platform's design	Do not give HR **Developer**, **Edit** or **Admin** at the system level.

Good to know

You cannot grant more than you have `Since 1.8`

Unless you are a system **Admin**, you cannot give a level above your own on that module or entity, and you cannot change or remove a grant that is already above your level. The platform shows "You cannot grant, change or revoke access above your own level."

Automations cannot hand out more than their author has `Since 1.10`

An automation (reaction) can give or take away permissions. When someone saves one, the platform checks that they already hold the access it would give; removing access needs **Admin** on that object. If they do not, the automation is not saved.

Delegations and administration rights `Since 1.10`

A delegation that allows everything passes on all the delegator's access except system **Admin** — unless administration is explicitly allowed on the delegation. See Delegation.

Changes apply straight away

A permission change applies to the person's next action across the whole platform. If their menus still look the same, ask them to refresh the page.

Anonymous access

Anonymous permissions and anonymous share links show data to anyone, with no sign-in. Check them carefully before you use them on anything sensitive.

Two people editing the same record

When someone is editing a record, the platform can lock it so that others do not overwrite their changes. Others see that it is locked, and the record keeps a lock history.

A second authorization check your operator may run `Since 1.10`

Your operator can switch on a background check that compares the platform's access decisions with a second system. It only measures. It never changes who can do what, and there is nothing to set up in Administration.

Sensitive fields

Some fields, such as Salary, can be marked **Sensitive** or **Encrypted** by the app builder. Their values stay hidden even from people who can open the record — including Admins — until you give a person or group a sensitive-field grant, and even then the person must confirm their identity before the value is shown. You manage these grants in **Administration › Access Management › Sensitive Permissions**, or on the **Sensitive Fields** tab of a user or group. A grant can have an end date, after which it is removed automatically. The full procedure is in Sensitive Permissions.

Troubleshooting

"A user can't see a module or entity." Open **User Permissions** for the person and look for it. If it is missing, give access to one of their groups (or to them) on the module or entity **Permissions** tab. Also check that they have a system level — at least **Restricted**.

"A user can see the entity but not its records." Viewing records in **Administration** needs **Edit**. Either give Edit, or — better for most people — share a view or page that shows the records they need.

"A user can't see a service in the Services Catalog." They are not in the service's **Audience**, or the service is not active. Add their group as **Audience** in **Group Permissions**, or ask the service's builder to check its visibility.

"I removed access, but the person can still see it." They get it another way. In **User Permissions**, look at **Assignment Type** and **Acquired Through**, and remove the grant from that group or role — or remove the person from the group.

"The level I want is not in the list." Admin exists only at the system level. On an entity, you cannot go below what the group has on the module. And unless you are a system Admin, you cannot give more than you have yourself.

"A user can't see a masked value." They need a sensitive-field grant and must confirm their identity each time. See Sensitive Permissions.

> **Go deeper**
>
> For integrators: Permissions API [Integrate & Extend] — grant records, levels, endpoints and the rules each call enforces · For builders: Sensitive fields [Build Apps › Protect a field] · For platform engineers: Engine: Permissions & Voters

Sensitive Permissions

> **Availability**
>
> **Since 1.8.**

This page is for tenant administrators. It shows you how to decide who may read protected values such as salaries or national ID numbers: how to give that right, limit it in time, review who holds it, and take it back.

An entity designer can mark an individual field — a salary, a national ID, a bank account — as **Sensitive**. OSOS then hides that field's value behind a `****` mask for everyone, and only hands the real value back to people you have explicitly granted it to.

`Since 1.10` A field can alternatively be marked **Encrypted**, which stores the value scrambled in the database rather than masking it on the way out. It is handed out from these same screens, using the same grant — see Encrypted fields use the same grant.

One field is out of reach by design: the entity's **main title field cannot be marked Sensitive**, and a field that already is Sensitive is not offered as a main-title candidate. The main title is rendered unmasked wherever an entry is referenced by name, so allowing it there would leak the value. `Since 1.8`

Administration › Access Management › Sensitive Permissions is where you hand that right out and take it back. It lists every live sensitive-field grant in the space on one screen, and the same grants are also reachable from a single user's or a single group's access-permissions screen. This guide covers all three screens and their mobile-app counterparts.

A separate dimension from ordinary permissions

Ordinary permissions answer "what may this person do with this *object*" — a module, an entity, a view, a report. A sensitive permission answers a much narrower question: "may this person read the *value* of this one field?" The two are evaluated independently:

- A user with **Admin** on an entity still sees `****` in its sensitive fields unless they hold a sensitive permission for those fields.
- **No role bypasses it.** A full system administrator is masked exactly like everyone else. There is no "see everything" level — the only way to read a sensitive value is to hold a grant for it.
- Granting a sensitive permission confers **read** access to that field's value and nothing else. It never widens what the person can reach elsewhere.
- Removing one takes effect immediately, on the next read.

Managing sensitive permissions is itself privileged: the **Sensitive Permissions** menu entry is offered only to **system administrators**. A sensitive-field grant and an ordinary permission for the same person live side by side; giving someone an ordinary permission never changes their sensitive-field grants.

Technical details

Behind these screens, reading a grant list needs system **View** access, and creating, changing or removing a grant needs system **Developer** access or above.

Sensitive field scopes

A grant does not point at a field by its display name — it carries a **scope**, and the scope is what decides how wide the grant reaches. It is worth understanding before you add your first grant, because choosing an entity or a module in the picker grants far more than choosing a single field.

Every sensitive field is identified across the whole space by a single canonical **scope** string. A scope is built from normalized (machine) names, separated by `/`:

```
<module>/<entity>/<field>
```

For a field that sits inside a **block**, the last segment is the dotted `block.subfield` normalized name — for example `hr/employee/payroll.iban`.

A grant carries a scope and a field carries a scope, and the two are compared **segment by segment**. Both must have the same number of `/`-separated segments to match at all. Any segment may be the wildcard `*`, which matches whatever sits in that position:

Grant scope	What it covers
`hr/employee/salary`	Exactly that one field — a **direct** grant.
`hr/employee/payroll.iban`	The `iban` sub-field of the `payroll` block on `employee`.
`hr/employee/*`	Every sensitive field on the `employee` entity — an **entity-wildcard** grant.
`hr/*/*`	Every sensitive field anywhere in the `hr` module — a **module-wildcard** grant.
`hr/*/salary`	The `salary` field on every entity in the `hr` module.

A grant whose scope names a field exactly is a **direct** grant on that field; a grant that reaches the field through a wildcard segment is an **inherited** grant. Both confer exactly the same read access — the difference is *where* the grant lives, and therefore where you go to change or revoke it.

Scopes are not typed by hand. The scope picker presents the space as a **module → entity → field** tree: selecting a leaf field authors the concrete three-segment scope, while selecting an entity or a module node authors the corresponding wildcard scope covering everything beneath it.

Encrypted fields use the same grant `Since 1.10`

A designer can mark a field **Encrypted** instead of Sensitive. The difference is where the protection sits: a Sensitive field is stored normally and masked on the way out to a reader, while an **Encrypted** field is stored as ciphertext in the database itself. A field is one or the other — the two options are mutually exclusive, and switching one on in the field editor switches the other off.

For you as the person handing out access, almost nothing changes:

- **It is the same grant.** An encrypted field is authorized by the same sensitive permission, on the same module → entity → field scope, from the same three screens. There is no second grant type and no separate “decrypt permission” to manage.
- **Encrypted fields are in the same picker.** A field that is encrypted but not sensitive is offered under **Choose sensitive field** alongside the sensitive ones, in every add dialog — the space-wide page, the per-user tab, and the per-group tab.
- **Removing the grant removes decrypt access immediately**, exactly as it does for a sensitive field, and an expiry window works the same way.

What *is* different is what the holder then does with it, which is covered under What the person with the grant actually sees: revealing an encrypted value asks for a single-use, short-lived identity check on every reveal, where revealing a sensitive one opens a longer standing reveal session instead.

The space-wide grants list

The Sensitive Permissions page lists **every** active grant in the space, one row per (scope, user or group) pair:

Column	What it shows
Scope	Two lines — the module/entity context above, and below it the field (or, for a wildcard grant, the entity or module the grant covers).
Scope Type	**Field**, **Entity** or **Module** — how much of the scope the grant covers, so you can see at a glance how broad a row is.
User/Group	The person or group the grant is for.
Expiry	The end of the validity window, rendered in your own date format and timezone. Empty means the grant never expires.
Actions	Edit the expiry, or remove the grant.

A search box above the table filters live across every displayed column, ignoring case and surrounding whitespace.

The list is deliberately **not** filtered by the validity window: a grant whose start date is still in the future appears here too, so you can see and edit it before it becomes effective.

Adding a grant

1. Open **Administration › Access Management › Sensitive Permissions** and select **Add Permission**.
2. Under **Scope**, open **Choose sensitive field** and pick from the module → entity → field tree. Pick a field for a single field; pick an entity or module to grant everything beneath it.
3. Under **Select Users** and **Select groups**, choose who the grant is for. You can pick several of each in one go; each person or group gets their own grant for the same scope.
4. Optionally set an **Expiry** window (see Expiring grants). Leave it empty for a grant that lasts until you remove it.
5. Save.

If one of the principals you picked already holds an active grant for that scope, the dialog warns you before saving and that principal is skipped — you never end up with two competing grants for the same pair.

Editing a grant's expiry

The row's edit action opens the grant with **only the expiry editable**. The scope and the entitled user or group are shown but locked: a grant is a statement about one principal and one scope, so re-pointing it would silently duplicate an existing grant. To change either, remove the grant and add the one you want.

Clearing the date range and saving removes the window entirely, turning the grant back into one that never expires.

Removing a grant

The row's trash action asks you to confirm — *"the user or group will no longer be able to access this sensitive field"* — and then **permanently deletes** the grant record. The access is gone from the next read onward; there is no undo and no archived copy on this screen. If you may need the access back later, it is cheaper to re-add the grant than to try to recover it.

Removing a grant in the mobile app does exactly the same — see Managing grants from the mobile app.

Sensitive permissions for one user

A user's access-permissions screen carries a **Sensitive Fields** tab alongside their ordinary user and admin permissions. It answers "what sensitive data can this person actually read?" — which is not the same as "what have I granted this person directly".

The tab lists **both** kinds of grant:

Assignment type	Meaning
Direct	The grant names this user as its entitled principal.
Group	The grant belongs to a group the user is a member of, and reaches them through that membership. The **Acquired Through** column names the group.

A scope the user holds both ways appears twice, once per route, because the two rows have different lifetimes — removing the group membership kills one and leaves the other standing.

Only **direct** rows carry edit and remove actions here. An inherited row has none: the grant is not the user's to change. To alter it, open the group named in Acquired Through and manage it there — or accept that it applies to every member of that group.

When you add a grant from this tab, the field picker **excludes scopes this user already effectively holds**, directly or through any of their groups, so you cannot accidentally stack a second grant on top of access they already have.

The tab has a **refresh** control, so you can re-read the list without leaving the screen. The validity window is optional: you can save a grant with no expiry, but a half-filled window or one that has already ended blocks **Save**.

Sensitive permissions for one group

A group's access-permissions screen carries the same panel for the group's own grants. Everything listed here is **direct** — a group does not inherit sensitive permissions from another group — and every grant applies to every current member of the group.

- **Add** opens the same dialog, with the picker excluding scopes the group already holds. The validity window is optional here.
- **Remove deletes** the grant outright on the web client, with the same confirmation as the space-wide list. Every member of the group loses access to that field immediately. It is not an archive and there is no restore. (The mobile client's equivalent tab behaves the same way — see Managing grants from the mobile app.)
- A refresh control re-reads the list, and loading, empty, and error states are shown inline with a retry.

Granting to a group is the maintainable choice when several people need the same field: membership changes then move the access with them, and you have one row to revoke instead of many.

Managing grants from the mobile app `Since 1.8`

All three screens above are also in the mobile app, so you can give, extend or take back a grant from a phone:

Surface	Web portal	Mobile app
Space-wide grants list	**Sensitive Permissions** administration page	**Sensitive Data Access Permissions** screen, the last entry in the Access Permissions menu

Surface	Web portal	Mobile app
One user	**Sensitive Fields** tab on the user's access-permissions screen	**Sensitive Fields** tab on the user access screen
One group	**Sensitive Fields** tab on the group's access-permissions screen	**Sensitive Fields** tab on the group access screen

Each mobile surface offers an add sheet and an expiry-only edit sheet, and picks the scope through the same **module → entity → field** tree-select the web picker uses, so the grants you author on a phone are the same records the web pages list. The mobile space-wide screen also warns you, before you save, if a principal you selected is already covered by an existing grant for that scope.

- **Removing a grant means the same on both apps:** the grant is deleted permanently, with no undo.
- **A validity window needs both dates or neither.** Every add dialog, on the web and on mobile, blocks **Save** until both dates are set or both are empty.
- On the mobile group screen, use the screen's floating add button while the **Sensitive Fields** tab is open; it opens the sensitive-field add sheet.

Inherited Permissions

A field's own configuration only shows the people granted that **exact** field. Anyone reaching it through an entity- or module-wildcard grant is invisible there — which is exactly the case most likely to surprise you during a review.

The **Inherited Permissions** dialog, reachable from the field's advanced settings next to the Sensitive toggle, closes that gap. It lists the users and groups that can read the field through a wildcard grant rather than a grant on the field itself.

- The affordance appears only once the field is marked Sensitive **and** at least one inherited holder actually exists for it.
- Opening it re-reads the list first, so you never act on a stale answer.
- The dialog itself is **read-only** — users with their name and email, groups by name, no edit/remove actions of its own. An inherited grant has to be changed where it lives, at its entity- or module-wildcard scope on the space-wide list.

Read a field's direct list and its Inherited Permissions dialog together; only the two combined tell you who can see that value.

Drilling into a holder's full grant record `Since 1.8`

Select a user or group in the Inherited Permissions list to open a read-only panel with that holder's grant that reaches this field: **Scope**, **Scope Type**, (for users) **Assignment Type** and **Acquired Through**, and **Expiry**. When a holder reaches the field through more than one grant, the panel shows the entity-wide grant rather than a module-wide one, ignoring grants that have already expired.

If the holder has lost the grant by the time you open the panel, it says *This user or group no longer has access to this sensitive field.* If the lookup fails, select **Retry**.

An **expand** button on the panel jumps straight to that holder's full, editable record — their Sensitive Fields tab on the per-user or per-group view above, deep-linked directly to it — and, once the navigation actually lands, closes both the panel and the Inherited Permissions dialog behind it, so you are not left with two stacked panels over the tab you just opened.

Expiring grants

A grant can carry a **validity window** instead of lasting until someone remembers to revoke it:

What you set	Effect
Neither date	The grant never expires — the default.
End date only	Effective from now until that date.
Both dates	The full window. Before the start date the grant is listed but confers nothing.

An end date must be in the future when you create the grant — the platform rejects a grant that would already be expired.

When the end date passes, OSOS **archives the grant automatically** through a pre-configured timer, and the revocation takes effect at once: the user's cached permissions are cleared, the field's restricted state is recomputed, and the archived grant is excluded from every later permission query. Nothing manual is required, and nobody has to remember the date.

Removing a grant by hand before its end date has exactly the same immediate effect. Either way, re-granting the same principal afterwards creates a fresh grant rather than quietly reviving the old one.

Use a window for access that is genuinely temporary — showing a reviewer a salary field for the length of a task, for instance — and leave it open-ended only for access that is part of someone's standing job.

What the person with the grant actually sees

A grant is necessary, but on its own it is **not** enough to reveal a value. Set expectations accordingly when you hand one out:

- Without a grant, the value reads `****` everywhere it can appear — list columns, the record detail, entry history, reports, and exports. The user can still type a new value into the field "blind", without seeing the current one.
- **With** a grant, the value still reads `****` by default. To reveal it, the user opens the **Unmask** control in the top bar, chooses how long the reveal should last — 2, 5, 10, 30, or 60 minutes, or 3, 4, 6, or 8 hours — and re-verifies their identity — a code sent to their email or phone, or one from their authenticator app. If their account has no verification method available and your organization has enabled the password fallback, they confirm their account password instead.
- While that session is active, every sensitive field they hold a grant for comes back unmasked. The session belongs to that specific sign-in — signing out, or opening the app on another device, starts masked again there, even for the same person. **Disable unmask** ends it early; otherwise it lapses on its own at the end of the chosen window.
- Enabling a session, disabling it, and every actual reveal of a value are all recorded in the audit trail, so a grant is accountable as well as revocable.

A user who tells you they have "no access" while holding a grant has almost always simply not started an unmask session.

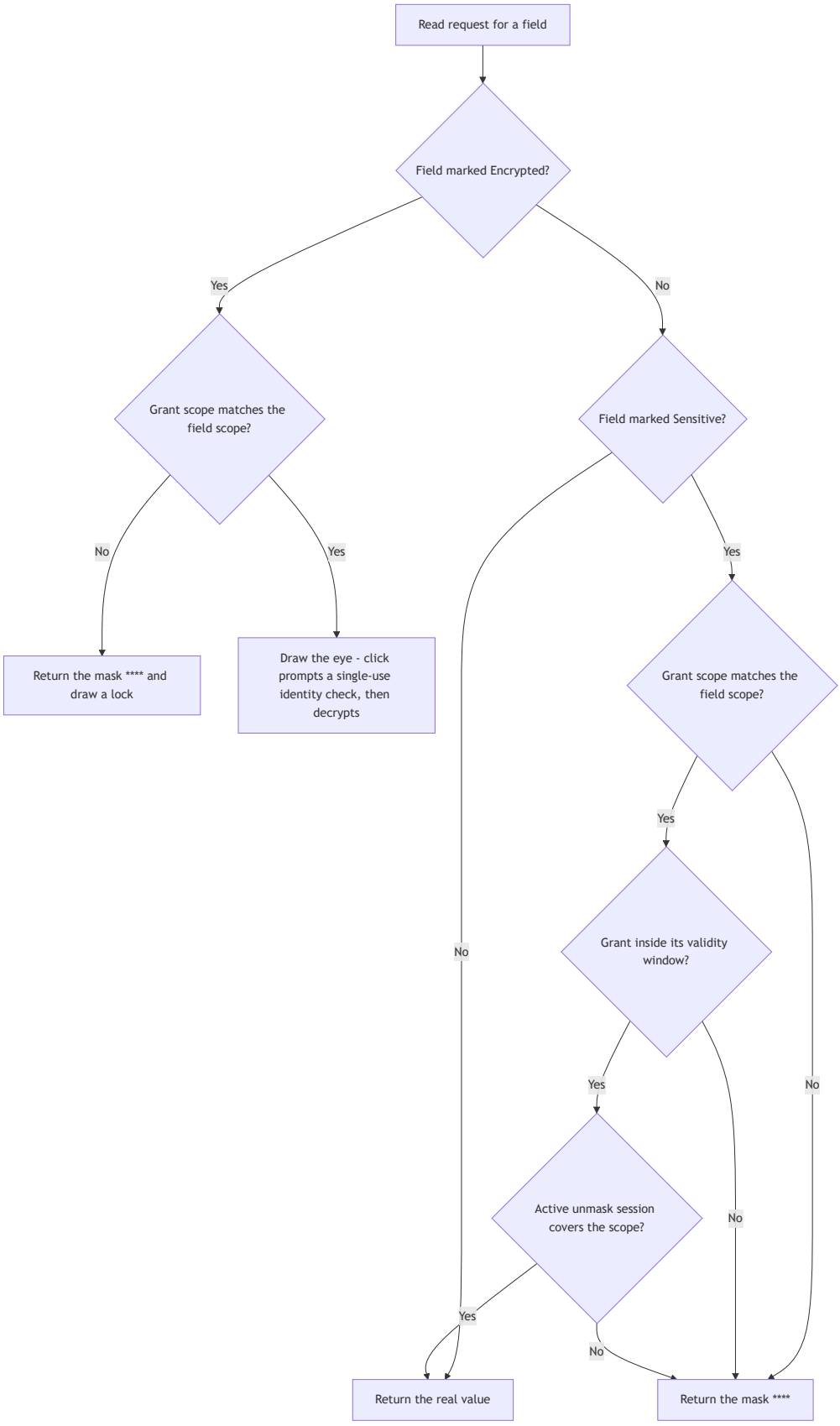
`Since 1.10` **The icon next to the field tells them which of those states they are in.** A padlock means *no grant at all* — there is nothing to click, and the app deliberately does not offer a reveal it would only have to refuse. An eye means *you hold a grant*: click it to start a reveal. This works the same for **encrypted** fields, so granting or revoking one visibly changes the icon that person sees.

`Since 1.10` **On an encrypted field, the eye asks the user to prove their identity before it reveals the value**, with the same methods as the classic unmask: an authenticator-app code, a text-message or email code, a passkey, or the password fallback. Unlike the classic **Unmask** session above, this check is single-use and short-lived (about two minutes) — there is no duration to choose, and it does not open a standing session that unmasks every sensitive field the person holds; it verifies once and shows that one value. The value stays visible only while that field is on screen; navigating away, scrolling it out of view, or refreshing re-hides it, and the next reveal asks again. Every successful reveal is still recorded in the audit trail — as *Decrypted Data*, its own event, separate from the unmask events above — and the value itself is never written there. Every step-up check (successful or not) is also recorded as its own audit event. Set expectations accordingly: the grant still does most of the work — it decides *who could ever* reveal the field — but a holder is asked to prove it's really them, every single time, the same discipline as a sensitive field.

How the platform decides

Every read of a sensitive field runs the same decision, per reader, per field. In short: no grant, or a grant outside its validity window, always means `****`; a grant on a sensitive field also needs an active unmask session; a grant on an encrypted field also needs a single-use identity check for each reveal.

Technical details: the decision chart



Any branch that cannot be answered — an unreachable permission lookup, an unavailable session store — resolves to the mask. The platform fails closed on this path by design.

Related pages

- Permissions & Access Control — the object-level permission ladder these grants sit beside.
- Groups — managing the groups you grant sensitive permissions to.
- Users — where a single user's Sensitive Fields tab lives.
- Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values] — the end-user side of a grant: the verification methods and reveal wizard a grant holder uses to unmask a value.
- Monitoring, Logs & Audit — where unmask and reveal events are recorded.

Delegation

This guide is for tenant administrators. A delegation lets one person act on behalf of a colleague for a set period, typically to cover their work during leave. You will learn how to set one up, how to limit it to just the work that needs covering, and how to end it.

The example: Sara, the HR manager, is on leave for two weeks. Her deputy, Karim, should approve leave requests in her place, but must not see the salary reports.

How it works

A delegation names two people and a period:

- the **Delegator**, the person whose work is covered (Sara),
- the **Delegated To** person, who does the work (Karim),
- a **From Date** and an optional **To Date**.

While the delegation is in effect, Karim can switch into Sara's account from his own and work as her: see her data, fill in and edit her records, add comments, and approve or submit requests in her workflows. Everything he does is completed in Sara's name, and OSOS also records that Karim was the one who actually did it. He never needs her password.

Karim can never get more than Sara has. And you decide how much of what Sara has he gets, with the delegation's **scope**:

Base	What Karim gets	Use it when
Allow All	Everything Sara can do, except the items you list under **Exception**.	Karim covers almost everything. List the few things he must not touch.
Deny All	Nothing, except the items you list under **Exception**.	Karim covers one or two specific things. List exactly those.

Administration rights are separate: Karim gets Sara's **Administration** access only if you allow it explicitly.

Before you begin

- Both people must have active user accounts.
- Know what the delegate should cover: which services (submitting requests, seeing requests, the inbox) and which views, pages and reports.
- You can offer only what you can reach yourself in the **Exception** list. See Good to know.

Create a delegation

Example: Karim covers Sara's leave-request approvals for two weeks, and nothing else.

1. Open **Administration › Applications › Data Entities** and open the **Delegation** entity. Don't start from the **Delegation** tab on Sara's user record: there, **Delegator** is filled in with you, the signed-in administrator, and can't be changed.
2. Add a new delegation.

3. Fill in the two people:
    - **Delegator**: Sara.
    - **Delegated To**: Karim.
4. Set the period:
    - **From Date**: when Karim may start acting for Sara.
    - **To Date**: when it ends. Leave it empty for a delegation with no end date, which then lasts until you end it.
    - **Is Active**: **Active**.
5. In **Delegation Scope**, set **Base** to **Deny All**.
6. In **Exception**, pick the leave-request service's inbox, so Karim can act on leave requests waiting for Sara. See Choose what the delegation covers.
7. Leave **Administration** empty unless Karim must also do Sara's administration work. See Administration rights.
8. Optionally turn on **Forward Notifications** and **Forward Emails**. See Forward notifications and emails.
9. Select **Save**.

The delegation takes effect at the **From Date**.

Choose what the delegation covers

The **Exception** list is a tree of everything that can be delegated:

- **Applications**: the views, pages and reports, grouped by module. (Reports can be delegated for the web only.)
- **Services**: for each service, three separate parts you can pick on their own or together:
    - submitting new requests,
    - seeing the requests,
    - the inbox, that is the requests waiting for the delegator to act.

What the tree means depends on **Base**:

- With **Allow All**, everything you pick is **excluded**. Example: Allow All, with the *Salary reports* module picked, lets Karim do everything Sara does except open the salary reports.
- With **Deny All**, everything you pick is **included**, and nothing else. Example: Deny All, with the leave-request service's inbox picked, lets Karim only act on leave requests waiting for Sara.

Karim's **Inbox**, **Outbox** and **Watching** lists, and the services catalog, follow the same scope while he acts for Sara: they show only the services the delegation covers.

Administration rights

The **Administration** option in **Delegation Scope** decides whether Karim also gets Sara's access to **Administration**:

- **Allowed**: Karim gets all of Sara's administration access, including Admin if she has it.
- **Not allowed** (empty), with **Allow All** and no exceptions: Karim keeps Sara's system-level access below Admin, but not Admin itself. `Since 1.10`
- **Not allowed**, with **Deny All**, or with any exceptions: Karim gets none of Sara's system-level access.

Only allow administration when the cover really needs it. To take it away later, edit the delegation and clear **Administration**. You can also remove it from **Administration › Access Management › User Permissions**: Karim's

administration access through the delegation is listed there as **Indirect (Delegation)**.

Forward notifications and emails `Since 1.10`

Forward Notifications and **Forward Emails** are both off by default. Turn them on when the delegate should also receive what is sent to the delegator, which is typical for leave cover.

What is forwarded

While the delegation is in effect, the delegate gets a copy of the notifications and emails that automations send to the delegator, and of scheduled report emails, but only for work the delegation covers:

- With **Allow All**, a copy is sent unless the message's service, module, report or view is excluded under **Exception**.
- With **Deny All**, a copy is sent only when the message's service, module, report or view (or one of the entity's views) is included under **Exception**. A message about one particular view needs that view itself to be included.

What the delegate receives:

- **Notifications**: a separate, unread copy that shows whose it is. Opening it offers to switch into the delegator's account (or opens it directly if the delegate is already acting for them). After the delegation ends, the copy shows as no longer available. Push notifications carry the same marking.
- **Emails**: the delegate is added in **CC** when the delegator was a To or CC recipient, and in **BCC** when the delegator was a BCC recipient. Nobody is added twice, and the delegator still receives the email.

No copy is sent to a delegate who is deactivated, to the delegator themselves, or, for notifications, to a delegate who already receives the same message directly. A copy is never forwarded again. If forwarding fails for any reason, the original message is still delivered.

See also Notification configuration.

When a delegation is in effect

A delegation works only while all of these are true:

1. **Is Active** is **Active**.
2. The **From Date** has passed.
3. The **To Date** is empty or has not passed yet.
4. The delegation has not been archived.

If the same delegator has delegated to the same person more than once, the delegate sees that delegator only once.

End a delegation

Delegations with a **To Date** end by themselves. To end one early, or to end one that has no end date, open it and do one of these:

- set the **To Date** to now, or
- set **Is Active** to **Inactive**, or
- archive the delegation.

The change applies to the delegate's next action. When the **To Date** has passed, a delegate who is still working as the delegator is switched back to their own account. When you set **Inactive** or archive the delegation, they lose the delegator's access.

When the delegator or the delegate leaves the organization, end all their delegations. See Offboard someone who leaves.

What the delegate sees

You may need to explain this to the person covering:

1. They open their personal menu (their picture or name at the top of the screen). A **Delegations** section lists everyone who has delegated to them.
2. They select the delegator's name. OSOS switches to the delegator's account, and the top of the screen shows their own name, an arrow and the delegator's name, so it is always clear who they are acting as.
3. To go back to their own account, they select **Exit**.

Signing out on the web also ends the delegation they were working under, so the next sign-in starts in their own account. `Since 1.6` On mobile, choosing a delegation reloads the app as the delegator.

Only the person named in **Delegated To** can act under a delegation. Nobody else can use it, even with a link to it. `Since 1.6`

Good to know

- **The Exception list shows what you can reach.** `Since 1.8` If you have a system level (View, Developer, Edit or Admin), the list offers every published module, view, page, report and service. Otherwise it offers only what you yourself have access to. This only limits what you can pick: the delegate never gets more than the delegator has, whatever you pick.
- **Changes reach the delegate straight away.** Editing or archiving a delegation, or changing the delegator's own access, applies to the delegate's next action.
- **Audit trail.** Every action taken under a delegation is recorded in the delegator's name, together with the delegate who actually did it, including button clicks in a record's history. See Monitoring: logs & audit.
- **Delegation is for people.** To let an external system act in OSOS, create an application user instead. See Application users.
- **Visitors who are not signed in** have no delegations.

Troubleshooting

What you see	What to check
The delegate does not see the delegator under **Delegations**.	Check the four conditions in When a delegation is in effect, especially **Is Active** and the dates.
The delegate is suddenly switched back to their own account.	The delegation has ended, or has not started yet. Check **From Date** and **To Date**.
The delegate can switch in but sees nothing.	With **Deny All**, only what is under **Exception** is covered. Add the views, pages or services they need.
The delegate can do too much.	With **Allow All**, everything not excluded is covered. Switch to **Deny All**, or add exclusions. Check **Administration**.

What you see	What to check
Acting for the delegator, the delegate sees a count on **Inbox**, but the list is empty.	This is a known product issue, and no workaround is confirmed yet.
The delegate gets no forwarded notifications or emails.	Turn on **Forward Notifications** / **Forward Emails**, and check that the message's service or report is covered by the scope.

Go deeper

For integrators: Delegation API & enforcement [Integrate & Extend] · For operators: Gateway operations: user-delegation

Organization settings

This page is for tenant administrators. It shows you how to set the defaults everyone in your organization works with — language, time zone and date formats — how to add a language or a time zone, and how to connect WhatsApp. It also explains which parts of your organization's identity, such as the product name and logo, are set by your operator rather than in **Administration**.

Themes, fonts and icons have their own page: Look and feel.

How it works

A **settings record** bundles the choices that shape what people see: the themes on offer and the default theme, the primary language, the time zone, and the date and time formats. Each record applies to one **scope**:

Scope	Who it applies to
Organization	Everyone in your organization. This is your baseline.
Space	Everyone working in one space.
User	One person.

When several records could apply, the most specific one wins: a person's own record, then their space's record, then the organization's record. Visitors who are not signed in get only the space's record: if the space has none, they get the built-in defaults, not the organization's record.

Example. Your organization works in English with dates shown day first. The HR space serves an Arabic-speaking team, so it gets its own settings record with Arabic as the primary language. One HR manager who works with the London office gets a personal record with the London time zone. Everyone else keeps the organization's defaults.

You manage all of this under **Administration › Interfaces**, which has the **Languages**, **Time Zones** and **Settings** screens used on this page (and **Themes** and **Fonts**, covered in Look and feel, and **Templates**, covered in Notification configuration). **Interfaces** appears in the menu for people with at least **Developer** access at the system level; adding or changing entries needs **Edit**. See Permissions & access.

Set your organization's defaults

1. Open **Administration › Interfaces › Settings**, then select **Add Setting**.
2. Under **Scope**, choose **Organization**, **Space** or **User**. A field appears where you pick the organization, the space or the person.
3. Choose the look and the formats:
    - **Available Themes** — the themes people can switch between.
    - **Default Theme** — the theme people get until they choose another.
    - **Primary Language** — the default language, offered by its code, for example `ar` or `en`.
    - **Time zone** — the time zone dates and times are shown in.
    - **Date Time Format**, **Date format** and **Time format** — how dates and times are written.
4. Select **Save**.

Start with one **Organization** record. Add **Space** or **User** records only where someone really needs something different. Each space, and each person, can have only one settings record.

To change a default later, open the record from the **Settings** list, edit it and save. The change applies the next time people load the app.

> ⚠ **Known issue: the Settings list shows only the Organization record**
>
> The **Settings** list opens with a **Scope** filter that shows only the **Organization** record. **Space** and **User** records still apply, but they are not listed, and clearing the filter may not bring them back, so you may not be able to open one from the list to change or archive it. This is a known product issue.

> 🚀 **The default theme is always offered**
>
> You do not have to list the default theme under **Available Themes** as well. It is always offered. If no settings record applies to someone at all, they get the built-in default theme.

Add a language

OSOS stores every name and label in each of your languages, so adding a language means people can work in it everywhere.

1. Open **Administration › Interfaces › Languages**, then select **Add Language**.
2. Fill in the form:
   - **Local Name** — the language's name as its speakers write it, for example *Français*.
   - **flag** — the flag shown next to it.
   - **Language Iso Code2** — the two-letter code, for example `fr`. Each language needs its own code.
   - **direction** — **Left to Right** or **Right to Left**.
   - **Default** — turn this on for the one language that should be your default.
   - **web translation files** and **mobile translation file** — the translations of the product's own labels, for the web app and the mobile app.
3. Select **Save**.

Mark exactly one language as **Default**.

> ⚠ **Removing or renaming a language leaves gaps**
>
> If you delete or rename a language, names and labels that were written in it are left empty for that language. OSOS does not fill them in for you. Afterwards, check your important records, views and services and fill in the missing names.

Add a time zone

1. Open **Administration › Interfaces › Time Zones**, then select **Add Time zone**.
2. Fill in **Time zone** (its name), **ID**, **Time Difference** (the offset from UTC) and, if it applies, **Daylight Saving Time (DST)**.
3. Select **Save**.

The new time zone can then be chosen in a settings record. Time zones also matter for deadlines: workflow time limits are counted in working hours, in the time zone that applies. See Working hours & time monitoring.

Connect WhatsApp `Since 1.10`

OSOS can send WhatsApp messages from your organization's WhatsApp business number: messages from automations, and sign-in codes. You connect your number in **System Settings**.

How it works. System Settings holds one organization-wide record, which applies to every space, and can hold one more record per space. A space that has its own WhatsApp details uses only its own details; it does not mix them with the organization's.

> **No dedicated screen yet**
>
> In 1.10 there is no System Settings screen. You edit the records as ordinary records, as described below.

Before you begin

Ask whoever manages your company's WhatsApp Business account in Meta for:

- the **phone number ID** and the **business account ID** of the number to send from;
- an **access token** and the Meta app's **app secret**;
- for sign-in codes, the name and language of an **approved sign-in code message template**.

Also choose a **webhook verify token**: any secret phrase. You will enter it here and in Meta.

Connect your number

1. Make sure you are working in your organization's main (default) space. The organization-wide record can only be changed from there.
2. Open **Administration › Applications › Data Entities** and open the **System Settings** entity.
3. Open the record that has no **Space** set. This is the organization-wide record.
4. Select **Edit** and fill in the **WhatsApp** section:

Field	What to enter
Provider	Leave it on Meta, the only provider.
Phone Number ID	The ID of the number messages are sent from.
Business Account ID	Your WhatsApp Business Account ID. Used to list your approved templates.
Access Token	The Meta access token.

Field	What to enter
Webhook Verify Token	The secret phrase you chose.
App Secret	The Meta app secret.
OTP Allowed Channels	Where sign-in codes may be sent: text message, WhatsApp or both. Leave WhatsApp out if you do not want WhatsApp sign-in codes.
OTP Default Channel	Which channel is offered first when both are possible.
OTP Template Name and **OTP Template Language**	Your approved sign-in code template and its language.

5. Select **Save**.

6. In your Meta app, register the delivery-status webhook so that the message log shows whether messages were delivered and read. Ask your operator for the webhook address of your space, and enter your webhook verify token there.

7. Ask your operator to switch on real WhatsApp sending for your organization. Until they do, nothing is sent, and every message shows **Not sent (sending disabled)** in the message log.

To give one space its own number, add a System Settings record, set its **Space**, and fill in every WhatsApp field that space needs.

Good to know

- **Sending needs the phone number ID and the access token.** Sign-in codes by WhatsApp also need WhatsApp in **OTP Allowed Channels** and a sign-in code template.
- **WhatsApp sign-in codes affect everyone with a verified phone number.** Once they are set up, people with a verified phone are asked for a code when they sign in. Read WhatsApp codes before you add WhatsApp to **OTP Allowed Channels**.
- **Secrets stay hidden.** After you save, the access token, app secret and webhook verify token are only shown as set or not set.
- **Changes apply straight away**, to the next message sent.
- **One number, one record.** A phone number ID can be used in only one active record across your organization.
- **The organization-wide record cannot be archived.** A space's record can be.
- **Your license must include administration.** Changing System Settings needs it. See Your license.

Where WhatsApp is used next: the WhatsApp channel, the WhatsApp message log, and WhatsApp sign-in codes. Builders send WhatsApp messages from automations; see Reactions & automation [Build Apps].

Your organization's name, logo and sign-in page

The product name, logo, favicon and base theme that people see in the web app and on the sign-in page are **not** set in **Administration**. They come from the configuration of your deployment and, on a shared deployment, from your organization's **brand**: a set of name, logos, theme and email sender address that your organization was registered with. `Since 1.10`

The brand shows in:

- the web app: the product name, logos, favicon, theme and the loading screen;

- the sign-in pages, the sign-in code messages and the name shown in authenticator apps;
- emails that do not name a sender of their own, which are sent from the brand's address.

To change any of these, ask your operator. An organization's brand is fixed once it is registered; changing it means your operator moves your organization.

The sign-in page `Since 1.10`

Your operator also sets the sign-in page's own look: its logo, colours and styling, an optional **Sign up** link, and the emails people get to set or reset their password. Two things you may notice:

- **A sign-up link keeps the sign-in page visible.** When a domain's only way in is one single sign-on provider, people normally go straight to that provider. If a sign-up link is configured, the sign-in page is shown instead, so people can choose. See Sign-in methods & single sign-on.
- **The sign-in page always has a Cancel link.** It cannot be switched off.

If your sign-in page's accent colour changed to red after the 1.10 upgrade, ask your operator to set your colour explicitly.

Currencies

The list of currencies is the same for every organization and cannot be edited. Every currency is available to every money field, and people who can see a money field can always see its currency.

> **Go deeper**
>
> For operators and integrators: Tenant settings & look and feel: technical reference · Identity server

Working hours & holidays

This guide is for tenant administrators. It shows you how to define your organization's **working hours** and **holidays**. App builders use them when they give a workflow step a time limit: the countdown then counts only working time, so a "two-day" approval does not run out over a weekend or a public holiday.

How it works

A builder can give any workflow step a time limit, for example "HR must approve a leave request within 16 working hours". To know what "working hours" means, the step points at one of your **working-hours schedules**. When a request reaches the step, OSOS calculates its deadline by counting only the time inside that schedule, and skips every day that is a **holiday**.

Example. Your office works Sunday to Thursday, 8:00 to 16:00. A leave request reaches HR on Thursday at 14:00 with a 16-working-hour limit:

- Thursday 14:00–16:00 counts 2 hours.
- Friday and Saturday are not in the schedule, so they count nothing.
- Sunday is a public holiday you entered, so it counts nothing.
- Monday 8:00–16:00 counts 8 hours, and Tuesday 8:00–14:00 the last 6.

The deadline is **Tuesday at 14:00**. If HR has not acted by then, the step becomes **Overdue**, and any escalation the builder set up starts.

You set up the schedules and holidays. The builder decides which steps have time limits, how long they are, and what happens when one is missed.

Before you begin

- You need access to **Administration › Applications › Data Entities**. Working hours and holidays are kept there, as records of the **Working Hours** and **Holidays** entities.
- Decide how many schedules you need: usually one per distinct working pattern, for example *Office hours*, *24×7 support desk*, *Branch weekend shift*.

Create a working-hours schedule

Example: *Office hours*, Sunday to Thursday, 8:00 to 16:00.

1. Open **Administration › Applications › Data Entities** and open the **Working Hours** entity.
2. Select **Add Working Hours**.
3. Enter a **Name** in each language, for example *Office hours*.
4. Under **Working Hours**, add one row for each working day:
    - **Day**: for example *Sunday*.

- **Starting Time**: 8:00.
- **Ending Time**: 16:00.

Each day can appear only once. A day with no row is a day off.

5. Select **Save**.

Create a separate schedule for each working pattern.

Add holidays

Example: Eid al-Fitr, three days, for everyone.

1. Open **Administration › Applications › Data Entities** and open the **Holidays** entity.
2. Select **Add Entry**.
3. Enter the **Name**, for example *Eid al-Fitr*.
4. In **Start and End Date**, choose the first and the last day of the holiday. Both days are included, and every day in between is a day off.
5. In **Related Working Hours**, choose the schedules the holiday applies to. Leave it empty to apply the holiday to **every** schedule.
6. Select **Save**.

Use **Related Working Hours** when a holiday applies to some teams only, for example a holiday the 24×7 support desk works through.

Change or remove a schedule or holiday

Edit or archive the record as you would any other. Changes apply straight away to every step that a request reaches from then on. `Since 1.10`

A step that is already counting down keeps the deadline it was given when the request reached it. Adding a holiday today does not move the deadline of a request that arrived yesterday.

Before you archive a schedule, ask your builders whether any step still uses it. A step whose schedule has no working days gets no deadline at all.

How builders use your schedules

This is what a builder sets on a step, so you know what to expect. The step's **Timing** settings are:

Setting	What it does
Timing	Turns the time limit on for the step.
Timing Schedule	The working-hours schedule to count against: one of yours.
Time Unit	**Minutes**, **Hours** or **Seconds**.
Timing method	**Fixed Value** (the same limit every time), **Field Value** (taken from a field of the record) or **Expression** (calculated, for example a shorter limit for urgent requests).
Value / **Fallback Value**	The limit itself, and the value used when a field or expression gives no number.

Each monitored step is tracked as **Pending** while it waits, **Achieved** when it is completed in time, and **Overdue** when the deadline passes first. OSOS checks for missed deadlines every minute, so a step can turn **Overdue** up to a minute after its deadline.

The builder's side is described in Workflows: time limits and escalation [Build Apps › Workflows & approvals].

When a step runs late

A missed deadline on its own only marks the step **Overdue**. To act on it, a builder adds a reaction to the entity, in the **Time Monitoring** section of its **Reactions** tab, with the event **On Time Exceeded**:

- **Choose Step**: the step the reaction watches.
- **Wait for**: how long after the deadline to act the first time, **in seconds**.
- **Repeat after**: how long to wait between repeats, **in seconds**.
- **Stop after**: how many times to repeat.
- The action, for example notify the step owner's manager, or move the step to someone else.

The waits are counted in working time too, against the same schedule and holidays. A typical escalation: remind the owner at the deadline, then notify their manager every four working hours, three times. For the manager's reaction that is **Wait for** 14400 (four hours in seconds), **Repeat after** 14400 and **Stop after** 2: the first run, then two repeats.

See Reactions & automation [Build Apps].

Time zones

A schedule has no time zone of its own. OSOS takes the **Starting Time** and **Ending Time** as times of day and applies them to every day it counts, and treats holidays as whole calendar days.

After you set up your first schedule, check one deadline: send a test request through a monitored step and compare its deadline with what you expect. If deadlines are shifted by a few hours, ask your operator to check the time zone the platform's time calculation runs in. See Working time & SLA engine.

Good to know

- **Every schedule needs at least one working day.** A schedule with no working time can never produce a deadline.
- **Archived holidays are ignored.**
- **Holidays are whole days.** There are no half-day holidays.
- **Time-limit data feeds reporting.** Builders can report on which steps ran late and where, to spot bottlenecks.

Troubleshooting

What you see	What to check
A monitored step never becomes **Overdue**.	The step's **Timing Schedule** is empty, archived, or has no working days, or the step's limit came out as zero. Ask the builder to check the step's **Timing** settings.
A deadline falls on a holiday.	The holiday is not linked to that schedule. Check **Related Working Hours**, or leave it empty to apply the holiday everywhere.

What you see	What to check
A new holiday did not move a deadline.	That request reached the step before you added the holiday. Deadlines are fixed when a request reaches the step.
Deadlines are a few hours off.	See Time zones.
Timing cannot be turned off on a step.	A reaction uses it. The builder must remove or change that reaction first.

Go deeper

For operators and engineers: Working time & SLA engine · For builders: Workflows: time limits and escalation

[Build Apps › Workflows & approvals]

Spaces & domains

This page is for tenant administrators. It shows you how to create a space, add people to it, put it into maintenance or take it offline, and how the web addresses people use — the domains — fit in. For what people see when they switch between spaces, see Signing in & choosing a space [Using OSOS › Signing In & Choosing a Space].

How it works

A **space** is a separate working area inside your organization, with its own address. Each space has its own records, applications, views and members. People see only the spaces they are members of, and inside a space they see only that space's data.

A **domain** is the web address (or addresses) people use to open OSOS, together with the sign-in options offered there: passwords, passkeys, second-factor codes and single sign-on buttons.

Example. Contoso runs OSOS at `contoso.example.com`. Its default space holds the company-wide applications. The HR department gets its own space, `hr.contoso.example.com`, where only HR staff are members. The leave request service lives in the default space, so everyone can use it; the salary review application lives in the HR space, so nobody outside HR even sees that it exists.

Every organization starts with one **default space**. It is served on the main address, holds your organization-wide settings, and cannot be deactivated.

You manage spaces under **Administration › Spaces**. You need **Admin** access at the system level to see it.

Create a space

1. Open **Administration › Spaces**, then select **New Space**.
2. Fill in the form:
   - **Type** — **Standalone** for an independent space, or **Sub Space**.
   - Under **Space Information**: the **Name** (in each of your languages), a **description**, and an **Icon**.
   - Under **Domain**: the **Domain Identifier**, which becomes the first part of the space's address. Use only lowercase letters, digits and underscores, for example `hr`. Each space needs its own identifier. The **URL** (your main address) is filled in for you and cannot be changed, and the **Full URL** — such as `hr.contoso.example.com` — is worked out from the two.
   - **Default Page** — optional: the page people land on when they open the space.
3. Select **Save**.

The new space appears as a card on the **Spaces** screen, with its address. OSOS prepares the space's default pages for you.

Add people to a space

People see a space only when they are members of it.

1. On **Administration › Spaces**, open the space's card menu and select **View Space**.
2. Open the **Users** tab. It lists the space's members.

3. Add a member: choose the **User**, and turn on **Admin** if they should be recorded as an administrator of the space.
4. Select **Save**.

The new member can switch to the space from **My Spaces** in the top bar. To remove someone from a space, remove their entry from the **Users** tab.

Find a member `Since 1.10`

You can search a space's **Users** tab by the person's name, and filter it by user. After an upgrade, searching existing memberships works once the platform has finished re-indexing them.

Put a space into maintenance, or take it offline

Each space card on **Administration › Spaces** has a menu with two independent switches:

Action	What happens	When to use it
Deactivate End User Interfaces	People can no longer use the space's applications, pages and services. Administration keeps working, so you can keep configuring the space.	Maintenance: you are changing an application and do not want people to use it half-finished.
Deactivate Space	The whole space goes offline. Its address stops working for everyone.	The space is no longer needed, or must be closed at once.

Each one asks you to confirm, and has a matching **Activate End User Interfaces** or **Activate Space** action to turn it back on.

- The default space cannot be deactivated.
- Turning a space's end-user interfaces off or on is recorded in **Administration › Monitoring › System Logs**. See Monitoring: logs & audit.

Domains

Each domain lists one or more **Host** addresses and the sign-in options for them. A new installation has one domain, **Default Domain**. Domain records are kept in the **Domain** entity: open **Administration › Applications › Data Entities** and open **Domain**.

A space's own address must be covered by a domain, so that people can sign in there. If you add a space on a new address, ask your operator to make sure the address is set up for the platform.

You cannot archive a domain that views or reports still use. OSOS tells you *You can't archive this domain because it is being used.* and lists them. Point those views and reports to another domain first.

Sign-in options per Domain

Each Domain decides what its sign-in page offers. To change it, open the Domain record, select **Edit**, set these fields, and then select **Save**:

Field	When it is on
Support Local Sign in	People can sign in with a username or email and a password.
Support Passkey	People can sign in with a passkey (fingerprint, face or device PIN) and add passkeys to their account.
SMS 2FA	After their password, people must type a code sent to their phone by text message.

Field	When it is on
Email 2FA	After their password, people must type a code sent to their email address.
Allow Password Step-Up	People without a second-factor method can confirm with their password before they see a protected value.

Single sign-on buttons, such as **Sign in with Microsoft**, are linked to a Domain by your operator. For what each option means, how second factors combine, and how to set up single sign-on, see Sign-in methods & single sign-on.

Good to know

- **Each address belongs to one space.** An address that no space uses shows a “not found” error instead of opening a space. `Since 1.10`
- **Private designs stay in their space.** Entities and views shared privately with one space do not appear in another space’s lists. `Since 1.10`
- **Membership changes apply straight away.** People added to a space can switch to it at once; people removed lose access at once.

Troubleshooting

What you see	What to check
Someone was added to a space but cannot see it.	Check that their entry is on the space’s **Users** tab and that the space and its end-user interfaces are active. Ask them to reload the page or sign in again.
A space’s address shows “not found”.	The space may be deactivated, or the address is not set up yet. Check the space on **Administration** › **Spaces**, then ask your operator.
People cannot use a space’s applications, but Administration works.	Its end-user interfaces are deactivated. Select **Activate End User Interfaces**.
You cannot deactivate a space.	The default space, and spaces the platform created for itself, cannot be deactivated.
You cannot archive a domain.	Views or reports still use it. Move them to another domain first.

Go deeper

For operators and integrators: Spaces & domains: technical reference · API gateway operations

Human verification (bot protection)

`Since 1.8`

This page is for tenant administrators. It explains how OSOS checks that a real person, not a bot, is using your public forms and your sign-in pages, how to turn the check on for a public form, and what to do when people report that they cannot get through.

How it works

OSOS protects two different places, and they behave differently:

	Public forms	Sign-in pages
What is protected	Forms and searches that visitors use **without signing in**.	Signing in, the second-factor step, and resetting a forgotten password.
Who turns it on	Your app builders, one form at a time. Off until they turn it on.	Your operator, for the whole platform. On by default.
What people see	An **I'm not a robot** box. They select it, wait a moment for the tick, and submit.	Nothing. The check runs by itself when they submit; at most they notice a short pause.
If the check service is down	Visitors cannot submit protected forms. Everything else keeps working.	**Nobody can sign in**, complete the second step, or reset a password.

There is no image puzzle and no distorted text to type. The visitor's browser solves a small calculation in the background, which is easy for a real browser and expensive for a bot doing it thousands of times.

People who are signed in are never asked to verify on forms. Verification on forms is only for visitors.

Example. Your HR department publishes a public job application form. Without protection, bots could submit thousands of fake applications. With human verification on, each applicant selects **I'm not a robot** before they submit, and bots are turned away.

Turn it on for a public form

This is done in the form's view, usually by the app builder who published it.

1. Open the view in the builder and go to its **Access Permission** settings.
2. Make sure **Anonymous** is on. Verification only applies to views that visitors can open without signing in.
3. Select **Human Verification**.
4. Save and publish the view.

Making a view public does not turn verification on by itself: it is a separate choice. For how to make a form public in the first place, see Data views [Build Apps › Data Views & Faces].

What it protects

On a protected view, a visitor must pass the check to:

- submit a new entry;
- change an entry;

- run a search, on views that ask visitors to search before they see any results.

Simply opening a public list does not need a check.

Sign-in pages are protected too `Since 1.8`

Separately from forms, the sign-in pages check that a real person is signing in. This applies to everyone, on every domain, and it has no settings in **Administration**: only your operator can switch it off, for the whole platform.

Because this check is invisible, people only notice it when it fails. When it fails, they see the same message as for a wrong password, so a user's "I can't sign in" can have either cause.

> ⚠ **If nobody can sign in**
>
> If suddenly **nobody** can sign in, complete the second step or reset a password, ask your operator to check the human-verification service first. This check does not let people through when the service is down; it blocks sign-in until the service is back or your operator switches the check off.

The end-user view of this step is on Signing in & choosing a space [Using OSOS › Signing In & Choosing a Space].

Good to know

- **Each check is used once.** A visitor who leaves a form open for more than about five minutes is asked to verify again.
- **No settings to tune.** How hard the calculation is and how long a check lasts are set for the whole platform by your operator.
- **Very old or locked-down browsers** that block the check cannot submit protected forms.
- **The sign-in check needs a secure address.** On a sign-in page served without HTTPS the check cannot run, so your operator must switch it off there.

Troubleshooting

What you see	What to check
A visitor gets an error saying human verification is required when submitting.	They did not verify, or their check expired or was already used. Ask them to select **I'm not a robot** again and submit.
The **I'm not a robot** box shows *Verification failed, please try again*.	Ask them to try again. If it keeps failing for everyone, ask your operator to check the human-verification service.
Turning verification on seems to do nothing.	The view must be open to **Anonymous** visitors. Signed-in users never see the check.
No visitor can submit any protected form.	The human-verification service is probably down. Ask your operator.
Nobody can sign in at all.	See If nobody can sign in. Ask your operator to check the human-verification service.
Sign-in fails only for the first person after a quiet period, then works.	The verification service was starting up after being idle. If it happens often, ask your operator to look at its timeout.
Signed-in users report a "CAPTCHA".	It is not from OSOS: signed-in users are never asked to verify on forms.

Go deeper

For operators: Human verification: technical reference · Helm chart — identity human verification

Look and feel: themes, fonts and icons

This page is for tenant administrators. It shows you how to make OSOS look like your organization: add your own fonts and themes, choose which themes people can use, and publish your own icons. Languages, time zones and date formats are on Organization settings.

How it works

- A **font** is a typeface you upload, such as your corporate font.
- A **theme** is a named look: its colours, the fonts it uses, and whether it is a dark theme.
- A **settings record** decides which themes people can choose from and which one they get by default, for your whole organization, for one space or for one person. How settings records work is explained on Organization settings.

So the order is: upload fonts, build themes that use them, then offer the themes through a settings record.

Example. Your brand uses the font *Contoso Sans*. You upload it, create a light theme *Contoso Day* and a dark theme *Contoso Night* that both use it, and then offer both to everyone in your organization's settings record, with *Contoso Day* as the default.

Fonts and themes are under **Administration › Interfaces**. **Interfaces** appears for people with at least **Developer** access at the system level; adding or changing entries needs **Edit**.

Add a font

1. Open **Administration › Interfaces › Fonts**, then select **Add Font**.
2. Fill in the form:
    - **Font name** — the name people will see when they pick the font.
    - **Language** — the languages the font is meant for.
    - **WOFF**, **WOFF2**, **TTF** and **OTF** — upload the font in each of the four file formats. All four are required, so that every browser and device can use it.
3. Select **Save**.

Add a theme

1. Open **Administration › Interfaces › Themes**, then select **Add Theme**.
2. Fill in the form:
    - **Theme Name** — the name people will see.
    - **Dark mode** — turn this on for a dark theme.
    - **Used font** — the fonts the theme uses.
    - **General CSS** — optional. A style sheet with the theme's colours and styling for the web app.
    - **Global Mobile Override** and **Mobile Selected Font** — the colours and the font the theme uses in the mobile app.
3. Select **Save**.

Choose which themes people can use

A theme is not offered to anyone until a settings record lists it.

1. Open **Administration › Interfaces › Settings** and open the record for your organization, a space or a person (or add one). The list may show only the **Organization** record: see the known issue in Set your organization's defaults.
2. Under **Available Themes**, select the themes to offer. Under **Default Theme**, select the one people get until they choose another.
3. Select **Save**.

For the full list of fields in a settings record, see Set your organization's defaults.

Visitors who are not signed in

Public pages and forms use only the settings record of the space they belong to. Your organization's record and personal records do not apply to visitors. If the space has no settings record, visitors get the built-in default look. To give public pages your look, add a settings record for that space.

Publish your own icons `Since 1.8`

You can add your own icons, for example your department logos, and use them wherever OSOS lets people pick an icon.

1. Open **Administration › Applications › Data Entities** and open the **Icons** entity.
2. Add an entry: give it a **Name**, paste the icon's **SVG** code, add **Tags** to make it easy to find, and turn on **Directional** if the icon should be mirrored for right-to-left languages, such as an arrow.
3. Select **Save**.

People who are using OSOS at that moment get the new icon set automatically. You do not need to refresh anything.

Application shells `Since 1.10`

A theme has an optional **Application Shell** field, and you may see an **Application Shell** entity. They are reserved for a visual shell builder that is not part of 1.10. Attaching a shell to a theme stores the choice but changes nothing people see.

Where your name and logo come from `Since 1.10`

The product name, logos, favicon, base theme and loading screen of the web app are not set on this page. They come from your deployment's configuration and, on a shared deployment, from your organization's brand. Your themes change the look of the app around them. To change the name or logo, ask your operator. See Your organization's name, logo and sign-in page.

A change your operator makes to your brand reaches people's browsers within an hour, including installed copies of the web app.

Troubleshooting

What you see	What to check
A new theme is not offered to anyone.	Add it to **Available Themes** (or make it the **Default Theme**) in the settings record that applies to those people.
One person sees a different look from everyone else.	They have a personal settings record, or their space has one. Personal and space records win over the organization's.
A font does not show.	Check that the theme uses it under **Used font**, and that the theme is offered in the right settings record.
Nobody has a settings record and people still see a theme.	That is the built-in default theme, used when no settings record applies.

Go deeper

For operators and integrators: Tenant settings & look and feel: technical reference

Monitoring: logs & audit

This page is for tenant administrators. It shows you how to answer three everyday questions from the **Monitoring** screens: *Did this message go out?*, *Did this automation run, and what did it do?* and *Who changed this?* It also explains who can see these screens and how long records are kept.

How it works

OSOS keeps two kinds of history:

- **Logs** record what the platform *did*: every email, text message and WhatsApp message it tried to send, every automation run, and every time a space's end-user interfaces were switched off or on.
- **The audit trail** records who *changed* what: records, entity designs, views, permissions, spaces, user accounts and more.

You read both under **Administration › Monitoring**:

Screen	Answers	Who can open it
Logs	Did a message go out, and did it arrive? Tabs **Email**, **Sms** and **WhatsApp**.	System administrators
System Logs	Who switched a space's end-user interfaces off or on, and when?	System administrators
Audit Logs	Who changed what, and when?	System administrators
Reaction Logs `Since 1.10`	Did an automation run, and what did each of its actions do?	System administrators

Example. An employee says she never got the email confirming her approved leave. You open **Reaction Logs** and find that the *Leave approved* automation ran for her request and its **Send Email** action succeeded. You then open **Logs › Email**, find the email with the same **Reaction Execution ID**, and see its **Status** is **Success**: the email left the platform, so the next place to look is her mailbox's spam folder.

Check whether a message was sent

1. Open **Administration › Monitoring › Logs**.
2. Choose the **Email**, **Sms** or **WhatsApp** tab.
3. Sort by **Sent at** to see the latest messages first, and page through to the one you are looking for.
4. Select a row to see its details.

For an email, the list shows **Reaction Execution ID**, **To**, **From**, **CC**, **BCC**, **Subject**, **Generated at**, **Sent at** and **Status**. The details add the **Response code** from the mail server. **Status** is **Success** when the mail server accepted the email, and **Failed** when it did not.

The **Reaction Execution ID** links a message to the automation run that sent it. If you know which automation should have sent the message, start in Reaction Logs and match the ID; this is often quicker than paging through the message log.

> **Copies of emails do not contain secrets**
>
> The copy of an email kept in the log never contains the password-reset link or an API token: those are replaced by `[redacted]`. Anything else in the email is kept, so keep secrets out of templates. See Notifications & email templates.

WhatsApp messages `Since 1.10`

The **WhatsApp** tab has one row per message, newest first: the reaction run, the recipient, the template and its language, when it was sent, and its status.

Status	Meaning
Accepted, **Sent**, **Delivered**, **Read**	Meta accepted the message, then reported how far it got. These show as successful.
Failed	The message did not go out, or Meta reported a failure. Open the row to see Meta's error.
Not sent (sending disabled)	Real WhatsApp sending is switched off for your deployment. Ask your operator to switch it on.

Open a row to see its **Status history**. The values that were filled into the template stay hidden until you select **Show template variables**. Sign-in codes are never stored.

Delivery and read statuses appear only if your WhatsApp app in Meta sends delivery reports to OSOS. See Connect WhatsApp.

Check whether an automation ran `Since 1.10`

Administration › Monitoring › Reaction Logs lists every automation (reaction) run in the current space, newest first. Use it to answer "did my reaction run, and what did it do?"

1. Open **Administration › Monitoring › Reaction Logs**.
2. Narrow the list with the quick filters **Reaction**, **Event**, **Action**, **Status** and **Period** (**Last day**, **Last week**, **Last month**, **Last year**), or type in the search box. Search matches the reaction name, its entity or module, the person who triggered it, or an exact run or entry ID.
3. Select a run to open its details: every field of the run, each action's result, and **View JSON** for the full record. Use the previous and next buttons to walk through the filtered list.

The columns are **Reaction Name**, **Status**, **Time**, **Entry**, **Actor**, **Event Name**, **Action Type**, **Entity** and **Module**. You can reorder them for your session; **Details** stays last.

- **Status.** A run has failed when any of its actions failed, and succeeded when at least one action ran and none failed. **No actions** means the run did nothing.
- **Only runs that executed are listed.** A reaction whose filter did not match the event leaves no row. If you expected a run and there is none, check the reaction's filter.
- **Entry.** *Deleted or unknown entry* means the record no longer exists. *N/A* means it exists but has no title.
- **The filters stay in the address**, so the browser's back button and a shared link show the same list.
- **From the entity designer.** On an entity's **Reactions** tab, a system administrator can select a reaction's failed or succeeded count to see the runs behind it, with links to each run and to this screen, already filtered.

Secrets are hidden here too. Values a builder marked **Redact in history**, values from sensitive fields, and well-known secrets such as passwords, tokens and API keys are stored as `****`, so not even a system administrator sees them. `Since 1.6`

Review who changed what

1. Open **Administration › Monitoring › Audit Logs**.
2. Use the list's search and filters to narrow it down — for example by **Event Name**, **User** or **Record Timestamp**.
3. Open a record to see its **Details**: what was changed and, for a record change, which fields changed and their old and new values.

Event Name uses plain names such as *User Signed In*, *Password Reset By Admin*, *User Account Locked*, *Data Entry Updated*, *Data Entity Updated*, *Access Right Changed*, *Space Deactivated*, *Reaction Updated*, *Package Imported* and *Data Entry Permanently Deleted*.

What is recorded

- **People:** sign-ins, password changes and resets, accounts activated, deactivated or locked.
- **Records:** created, updated, archived, restored and deleted (including drafts and bulk actions).
- **Design:** entities and their fields, views, pages, modules, lists, relations, queries and reactions created, changed or deleted.
- **Access:** every permission change.
- **Spaces:** created, updated, activated or deactivated, and end-user interfaces switched off or on.
- **Tokens and packages:** application tokens generated, packages exported or imported.
- **Sensitive data:** see below.

What is not recorded

- Changes the platform makes to itself, such as upgrades and data migrations.
- Changes to notifications, requests and the audit log itself.
- Values of sensitive fields: they are stored as `***`, for everyone, for ever. You cannot recover them from the audit trail.

In rare cases, such as a brief outage, an individual audit record can be lost. The change itself is never blocked or undone because its audit record could not be written.

Sensitive data reveals `Since 1.10`

Four events tell you who looked at protected values:

Event	Recorded when
Sensitive Data Unmask Enabled / *Sensitive Data Unmask Disabled*	Someone starts or ends a reveal session.
Sensitive Data Unmasked	A sensitive value is actually shown to someone during a reveal session, including in a report.
Decrypted Data	Someone reveals an **encrypted** field's value.

The values themselves are never written to the audit trail. Exports and scheduled report emails carry real values by design and are not recorded as reveals. Who can produce these events is decided on Sensitive Permissions.

Identity checks before a reveal

Before someone reveals an encrypted value, or runs a workflow action that asks them to confirm who they are, OSOS asks for a fresh identity check. Every attempt, successful or not, is recorded in the platform's sign-in security log, not on

the **Audit Logs** screen. A wrong code therefore leaves a trace even though nothing was revealed. Ask your operator if you need these records.

Entities deleted for good `Since 1.10`

When a system administrator permanently deletes an entity, the audit trail records *Data Entity Permanently Deleted* with the entity, its module and who deleted it. You can select this event in the **Event Name** filter.

Application tokens

When an administrator asks for an application user's API token to be regenerated, the request is recorded. The token itself is never written to the audit trail. `Since 1.8`

A function's token rotation `Since dev`

On the development line every function has its own application user. When you regenerate that user's token, the new token goes straight to the function, and the audit trail shows your request plus a *Function App Token Regenerated* record.

Check space maintenance changes (System Logs)

Administration › Monitoring › System Logs lists each time a space's end-user interfaces were switched off or on, with **User**, **Date**, **Space Name** and **Status** (**Activated** or **Deactivated**). See Spaces & domains.

Who can see these screens

Monitoring appears for people with **Admin** access at the system level. To read the message logs a person also needs system-wide **Edit** or **Admin** access, and **Reaction Logs** is for system administrators only, because a run's details include what each action read and wrote. What people see on **Audit Logs** also follows their access to the audit records. See Permissions & access.

How long records are kept

Sign-in and second-factor security records are kept for 180 and 90 days.

Troubleshooting

Question	Where to look
"I never got the email."	**Logs › Email**. **Success** means it left the platform — check spam. **Failed** — open the row for the response code and pass it to your operator. No row at all — the automation did not send it: check **Reaction Logs**.
"Nobody gets text messages."	**Logs › Sms**. If messages fail or there are none, ask your operator whether an SMS provider is connected.
WhatsApp shows **Not sent (sending disabled)**.	Real sending is off for your deployment. Ask your operator.
"Did my automation run?"	**Reaction Logs**, filtered by reaction. No row means its filter did not match.
"Who changed this record / permission / view?"	**Audit Logs**, filtered by **Event Name** and **Record Timestamp**.

Question	Where to look
You cannot see **Monitoring** at all.	You need **Admin** at the system level, and your license must include administration. See Your license.

Go deeper

For operators and integrators: Logs, audit & retention: technical reference · Observability

Notifications & email templates

This page is for tenant administrators. It explains how OSOS reaches people — in the app, by email, by text message and by WhatsApp — what you manage yourself (shared templates and your WhatsApp connection), what your app builders and your operator manage, and what to check when someone says a message never arrived.

How it works

Most messages are sent by **automations** (reactions) that your app builders add to their applications: "when a leave request is approved, email the employee". The automation decides **when** a message goes out and **to whom**. It can write the message itself or use one of your shared **templates**.

Channel	What people get	Who sets it up
In-app	A notification under the bell in the web app, and a push notification on their phone if they use the mobile app.	Builders, in automations.
Email	An email.	Builders, in automations. You can provide shared templates. Your operator connects the mail server.
Text message (SMS)	A text message.	Builders, in automations. Your operator connects the SMS provider.
WhatsApp `Since 1.10`	A WhatsApp message built from a template Meta has approved.	You connect your WhatsApp number. Builders send messages from automations. Your operator switches on real sending.

The platform also sends some messages itself, with no automation:

- the email that lets a new user set their password, and the password-reset email;
- the email with a newly generated API token for an application user;
- a notification when a file someone exported in the background is ready to download;
- sign-in codes, by text message, email or WhatsApp, when second-factor sign-in is on;
- warnings to system administrators when your organization nears its entries limit. See Plan limits.

Example. HR wants employees to get a branded email when their leave is approved. You create an email template **Leave approved** with your HR header and footer. The app builder adds an automation to the leave request service that sends that template to the requester when the request is approved.

Manage shared templates

A shared template is written once and used by many automations. Changing it changes every message that uses it.

Template Type	What it is for
Email	A complete email: subject and body, optionally with a header, a footer and attachments.
Email Part	A reusable header or footer, such as your logo and address, that email templates include.
Notification	An in-app notification: a title and content.

Create an email template

1. Open **Administration › Interfaces › Templates**. (You can also open **Administration › Applications › Data Entities** and open the **Templates** entity.)
2. Add an entry and fill in:
   - **Template Name** and **Template Description** — so builders can find the right template.
   - **Template Type** — **Email**.
   - Under **Template Details**: the **Subject** and the **Body**, and optionally a **Template Header** and a **Template Footer** (pick **Email Part** templates).
   - **Attachments** — optional files sent with every email that uses the template.
3. Select **Save**.

Create your header and footer first, as templates of type **Email Part**, so you can pick them here. Changing a shared header or footer updates every email that includes it.

Placeholders

Write `{{ name }}` where a value should go, for example `Dear {{ employee_name }}, your leave from {{ start_date }} was approved.` Names can use letters, digits and underscores. The builder who uses the template decides which value fills each placeholder.

Changing a template that is in use

A template opens read-only. To change it, select **Edit Template**, make the change and select **Save**.

When you save a template that automations already use, OSOS asks you to confirm: *This template is used inside reactions. Are you sure you want to save the changes?* The change applies to every message sent from then on.

Password emails

The emails that let people set or reset their password can use your own templates. Create them as **Email** templates, then ask your operator to use them for your organization's sign-in pages.

> ⚠ **Keep secrets out of templates**
>
> Copies of sent emails are kept in the email log, which other administrators can read. OSOS removes password-reset links and API tokens from those copies, but nothing else. Never put passwords, codes or other secrets in a template.

Who receives a message

Builders choose the recipients in each automation. They can send to specific people, to groups, to lists of people, to organization units, to typed-in addresses, or to whoever the record points to — for example the requester, the current step owner, or that person's manager.

- People whose accounts are deactivated do not receive emails.
- Someone who would get the same email twice — for example as a person and as a group member — gets it once.
- If one recipient cannot be found, the others still get the message.

When someone is away: delegate copies `Since 1.10`

If a person has an active delegation with **Forward Notifications** or **Forward Emails** turned on, their delegate gets a copy:

- **In the app**, the delegate gets their own unread notification, marked as forwarded from the person they cover for.
- **By email**, the delegate is added in copy. The original recipient still gets the email.

A delegate who was already a recipient does not get a second copy, and a copy is never forwarded again. See Delegation — Forwarding notifications and emails.

In-app and push notifications

In-app notifications appear under the bell in the web app. People who use the mobile app and allow notifications also get them as push notifications on their phone. There is nothing to set up per person: the app registers each phone when the person signs in.

Clicking a notification `Since 1.10`

Clicking a notification marks it as read and opens what it is about. Notifications sent by automations open the record they refer to, through the view the builder chose. Some notifications from the platform itself, such as package review notifications from the marketplace, open a specific page directly.

Phones that stop getting push notifications `Since 1.10`

Each phone gets one push notification per message. OSOS forgets phones that the mobile app has not checked in from for 90 days, and phones the push service reports as no longer valid. A person whose phone was forgotten gets push notifications again after they next sign in to the mobile app with notifications allowed.

Email

Emails are sent from your organization's sender address, unless the automation names a different sender. Your operator sets the sender address and connects the mail server. When an automation adds a link to the record, the link is shortened in the email.

Password-reset links and API tokens never reach the email log. The email the person receives is complete, but the copy kept in the email log has those two secrets replaced by `[redacted]`. `Since 1.4`

Text messages (SMS)

Text messages go through an SMS provider that your operator connects. Until a provider is connected, text messages are not sent. If no text messages arrive at all, ask your operator whether a provider is set up.

WhatsApp `Since 1.10`

WhatsApp messages always use a message template that Meta has approved for your WhatsApp Business account. OSOS never sends free text by WhatsApp.

- **To set it up**, connect your WhatsApp number in System Settings. See Connect WhatsApp.

- **Builders send messages** with the **Send WhatsApp** action in an automation, choosing one of your approved templates. If your number is not connected, the action warns *WhatsApp isn't configured for this space. Fill in the WhatsApp section of System Settings first.* See Reactions & automation [Build Apps].
- **Sign-in codes** can also be sent by WhatsApp. See WhatsApp codes.

Every WhatsApp message gets a row in the WhatsApp message log:

Status	What it means	What to do
Accepted, **Sent**, **Delivered**, **Read**	Meta accepted the message, then reports how far it got.	Nothing.
Not sent (sending disabled)	Real WhatsApp sending is switched off for your deployment. Nothing reached Meta.	Ask your operator to switch on real sending. It is off until they do.
Failed	The message did not go out, or Meta refused it. Open the row to see why.	Common reasons: your number is not connected for this space; the recipient's number is not in international form (`+` and the country code); Meta refused the message.

A message Meta did not accept is retried automatically only when it is certain it was not sent, so nobody gets the same message twice. Sign-in codes are never stored in the log.

Troubleshooting

Someone says…	What to check
"I never got the email."	Open **Administration › Monitoring › Logs**, tab **Email**. Find the message and check **Status**. If there is no row, the automation did not send it: check the automation in **Administration › Monitoring › Reaction Logs**. See Monitoring: logs & audit.
"Nobody gets text messages."	Ask your operator whether an SMS provider is connected.
"WhatsApp says *Not sent (sending disabled)*."	Real sending is off. Ask your operator.
"My delegate did not get my emails."	Check that the delegation is active, covers that application, and has **Forward Emails** on.
"My phone stopped getting push notifications."	Sign out of the mobile app and back in, and allow notifications.
"The email went out with an old text."	The automation may write its own text instead of using your template. Ask the builder.

Go deeper

For app builders: Reactions & automation [Build Apps] · For operators and integrators: Notification delivery: technical reference

Your license & plan limits

What needs a license

A license is required to activate **Administration**, where your team builds apps and configures the platform. The OSOS vendor issues it. Operators install and renew it as described in Install and renew the license.

Where to see your license

OSOS has no screen that shows your license. Your vendor holds its details: what it includes, its limits and its expiry date. Ask your vendor, or whoever runs your deployment.

Without a license

- **No valid license.** OSOS does not open. Instead of the sign-in page, people see an error page with a **Retry** button. Contact your vendor. For operators: When the license is refused.
- **A license without administration.** The **Administration** menu does not appear, opening an Administration address shows an access-denied page, and changes to the platform's configuration are refused. People can still read and work with records as usual.

Plan limits `Since 1.10`

Your plan can limit how many records your organization may have. Records in the entities your team built count; an archived record counts as a quarter.

- **Warnings.** On a shared installation, when your records reach 80% of the limit and again when they reach it, system administrators get a notification in OSOS and by email. The notification says from which number new records are refused: 10% above the limit, unless your plan allows going over it.
- **Installing an app.** Installing a package that would take you past that number is refused before anything is written, with a message saying so.
- **What always works.** Reading, editing and exporting records.

To make room, archive or delete records you no longer need. To raise the limit, ask your vendor.

Users from your company directory

This page is for tenant administrators whose organization keeps its staff in a company directory: **Active Directory** on your own servers, or **Microsoft Entra ID** (formerly Azure AD). It explains what the directory import does, which decisions are yours, what to hand to your operator, and how to check the result. You do not run the import yourself: there is no screen for it in **Administration**.

How it works

Instead of adding hundreds of people by hand, your operator can schedule an import that reads your directory and keeps OSOS in step with it:

- **People.** Everyone in the directory who has an email address becomes a user in OSOS, or is updated if they are already there. Names and email addresses follow the directory.
- **Group membership.** For each group you prepared in OSOS, the people in the matching directory group are added to it.

The import runs on a schedule your operator sets, for example every hour. Each run picks up new joiners and changes.

Example. Your HR department has a directory group *HR-Staff* with 20 people. You create an **HR** group in OSOS and link it to *HR-Staff*. After the next run, the 20 people exist as users and are members of **HR**. When a new HR officer joins *HR-Staff* in the directory, they appear in OSOS, in the **HR** group, after the following run.

What the import does not do

Plan for these, because they stay your job:

- **It does not remove anyone.** A person who leaves the directory, or a directory group, stays in OSOS and in the group. Offboard leavers yourself. See Offboard someone who leaves.
- **It does not create groups.** It only fills groups you created and linked.
- **It does not give access.** Imported people get no system level and no permissions until you give them, usually through their groups.
- **It does not bring back archived users.** If someone was archived in OSOS, the import leaves them archived.
- **It skips people without an email address** in the directory.

How imported people sign in

Imported people are marked as directory accounts. They do **not** get a set-password email, and they cannot set or reset a OSOS password. They sign in with their company account through **single sign-on**. So set up single sign-on before you import, or at the same time. See Sign-in methods & single sign-on.

Decide what to import

Agree these with your operator before the first run. They go into the hand-off checklist.

1. **Which directory.** Active Directory on your own servers, Microsoft Entra ID, or both.
2. **Which people.** Everyone, or only some:
    - With Active Directory, the import can be limited to the members of one directory group.

    - With Microsoft Entra ID, it can be limited by a condition, for example *department is Finance*.
3. **Which groups to mirror.** List the directory groups whose membership should fill a OSOS group, and which OSOS group each one fills.
4. **How often.** For example every hour, or once a night.
5. **Names.** Check with your operator how first and last names are mapped. With Active Directory, the standard set-up can swap first and last names; ask for it to be corrected before the first run.

Prepare the groups

For each directory group you want to mirror:

1. **Create the OSOS group**, for example **HR**, under **Administration › Organization › Users Groups**. See Create a group.
2. **Link it to the directory group.** The group must be marked as a directory group and carry the directory group's identifier: its distinguished name in Active Directory (for example `CN=HR-Staff,OU=Groups,DC=example,DC=com`), or its object ID in Microsoft Entra ID.

Out of the box, the group form has no fields for this link. The fields are added once to the **Users Group** entity, either by an app builder or by your operator, and then appear on every group. Ask your operator for this when you hand over the checklist.

3. **Give the group its access**, so that people get what they need as soon as they arrive. Include a system level of at least **Restricted**. See Permissions & access.

Hand-off checklist

Copy this table, fill it in and send it to your operator. Send passwords and secrets separately, through a secure channel.

#	Item	Your value
A	**Directory**	
A1	Active Directory, Microsoft Entra ID, or both	
A2	Active Directory: server address and the part of the directory to read (base)	
A3	Active Directory: a read-only account that can read users and groups *(password sent separately)*	
A4	Microsoft Entra ID: directory (tenant) ID	
A5	Microsoft Entra ID: an app registration with **application** permission to read all users and groups, its client ID and a client secret *(secret sent separately)*, and the secret's expiry date	
B	**Scope**	
B1	Import everyone, or only: one directory group (Active Directory) / a condition such as a department (Entra ID)	
B2	Directory groups to mirror → OSOS group each one fills	
B3	Add the directory-link fields to the **Users Group** entity (needed for B2)	yes / already done
C	**Running it**	
C1	Schedule for people (for example, every hour)	
C2	Schedule for group membership (run after the people import)	
C3	Name mapping checked (first and last name not swapped)	yes / no

#	Item	Your value
C4	Who to tell when a run fails	

The app registration in A5 is not the same as the one used for single sign-on: the import reads the directory on its own, without anyone signing in, so it needs application permissions. Your Entra administrator chooses the exact Microsoft Graph permissions.

> **Tip**
>
> Put the Entra client secret's expiry date in your calendar. When it expires, the import stops. Create a new secret before that date and send it to your operator.

Check the result

After the first run:

1. Open **Administration › Organization › Users** and check that the expected people are there, with their names the right way round.
2. Open each mirrored group under **Users Groups** and check its members on the **Users** tab.
3. Pick one imported person and check their access in **Administration › Access Management › User Permissions**.
4. Ask one imported person to sign in through single sign-on.

Good to know

- **The directory wins for names and email addresses.** If you change an imported person's name in OSOS, the next run changes it back. Change it in the directory instead.
- **People already in OSOS** are matched by email address (and, with Entra ID, also by their directory ID), so they are updated, not duplicated.
- **Running the import again does not add people to a group twice.**
- **Large Entra ID directories.** Group membership is read only from the first page of groups, and the first page of each group's members, that Microsoft returns. If you mirror many groups or very large groups, ask your operator to confirm that everyone arrived.

Troubleshooting

What you see	What to check
Someone is missing.	They have no email address in the directory, are outside the scope you chose (B1), or were archived in OSOS.
A group stays empty.	The group is not marked as a directory group, its directory identifier does not match exactly, or the people import has not run yet.
Someone who left is still in a group.	The import never removes people. Remove them yourself.
First and last names are swapped.	Ask your operator to correct the name mapping (C3).
Imported people cannot sign in.	Single sign-on is not set up for their domain, or their directory account does not match the email address in OSOS.

What you see	What to check
New people stopped arriving.	The import is failing. With Entra ID, the client secret may have expired. Ask your operator to check the import's logs.

Go deeper

For operators: Directory sync operations · Identity server & external providers

Sign-in methods & single sign-on

This page is for tenant administrators. It explains the ways people can sign in to OSOS, which of them you switch on yourself in **Administration**, and which need your operator. It ends with a complete, step-by-step task for setting up single sign-on with **Microsoft Entra ID** (formerly Azure AD), or with any other identity provider that supports OpenID Connect.

How it works

Everyone signs in on the same sign-in page. What that page offers depends on the **domain**, the web address people use to open OSOS (for example `portal.example.com`). Each domain has its own sign-in settings, so an organization can, for instance, let staff sign in with their company account on its staff address while external partners use a password on a second address.

A domain can offer any combination of these:

Sign-in option	What people do	Who switches it on
Password	Type a username or email and a password.	You, per domain.
Passkey	Use the device's fingerprint, face or PIN instead of a password.	You turn passkeys on per domain; each person then creates their own passkey.
Second-factor code	After the password, type a one-time code sent by text message, email or WhatsApp, or shown in an authenticator app.	You, per domain (text message, email). WhatsApp depends on your messaging settings. People turn on an authenticator app themselves.
Single sign-on (SSO)	Select a button such as **Sign in with Contoso** and sign in with the company account they already have.	You prepare it with whoever manages your company identity provider; your **operator** connects it to OSOS.

Example. Contoso uses Microsoft Entra ID for all staff. Its administrator registers OSOS in Entra, hands the connection details to the operator, and then switches off passwords on the staff domain. From then on, staff open the portal, are taken straight to the Microsoft sign-in page, and arrive back in OSOS already signed in. Microsoft's own security rules (such as Entra's multi-factor authentication) protect the account.

Choose how people sign in on a domain

You change the sign-in options of a domain by editing the domain's record. There is no separate screen for sign-in settings.

1. Open **Administration › Applications › Data Entities** and open the **Domain** entity.
2. Open the domain you want to change. A new installation has one, called **Default Domain**. Its **Host** list shows the web addresses that belong to it.
3. Select **Edit**, set the options below, and then select **Save**.

Field	When it is on
Support Local Sign in	The sign-in page shows the username and password form.
Support Passkey `Since 1.6`	The sign-in page shows **Use Passkey**, and people can add passkeys to their account.
SMS 2FA `Since 1.6`	After their password, people must type a code sent to their phone by text message.

Field	When it is on
Email 2FA `Since 1.6`	After their password, people must type a code sent to their email address.
Allow Password Step-Up `Since 1.8`	People who have no second-factor method can confirm with their password before they see a protected value. See Revealing a masked sensitive value [Using OSOS › Account Security & Revealing Protected Values].

The change applies to the next sign-in on that domain. People who are already signed in are not signed out.

> ⚠ **Keep a way in**
>
> Switch off **Support Local Sign in** only after you have tested single sign-on with a real account and confirmed that at least one administrator can sign in through it. If no working sign-in option is left on a domain, nobody can reach it until your operator restores one.

Technical details

A domain's sign-in options are copied to the sign-in service whenever the domain record is saved. The domain that applies to a sign-in is the one whose **Host** list contains the address of the app the person is signing in to. Domains that are still in use by views or reports cannot be archived.

Second-factor options

A second factor asks people to prove who they are a second time after they type their password. The sign-in page offers every method that applies to the person, and shows first the one they used last time on that account. `Since 1.8` That choice is remembered for each account, not for each browser, so people who share a computer each get their own default.

- **Authenticator app.** People turn this on themselves on their **Security** page (**Two-factor authentication › Enable authenticator app**). After that they are always asked for a code, whatever the domain settings. See Account security [Using OSOS › Account Security & Revealing Protected Values].
- **Text message (SMS) and email codes.** Turned on per domain with **SMS 2FA** and **Email 2FA**. A person who asks for too many codes in a short time has to wait a few minutes before asking again. A code that was typed wrong too many times stops working, and the person can select **Resend Code**.
- **Passkeys.** When **Support Passkey** is on, people can add a passkey on their **Security** page and use it either to sign in or as their second step.

WhatsApp codes `Since 1.10`

OSOS can also send sign-in codes by WhatsApp. WhatsApp is offered when both of these are true:

1. **Your organization has WhatsApp sign-in codes set up.** This is part of the WhatsApp settings in your System Settings: a connected WhatsApp business number, WhatsApp listed as an allowed code channel, and an approved sign-in code message template. In 1.10 there is no dedicated screen for these settings. See WhatsApp settings block.
2. **The person has a verified phone number** on their account.

> ⚠ **WhatsApp turns on a second factor for everyone with a verified phone**
>
> WhatsApp does not depend on the **SMS 2FA** switch. Once WhatsApp codes are set up, **everyone with a verified phone number is asked for a code at sign-in**, even on a domain where **SMS 2FA** and **Email 2FA** are both off.

When someone could use both text message and WhatsApp, your WhatsApp settings decide which is shown first. A WhatsApp code works only on WhatsApp, and a text-message code works only by text message. If WhatsApp cannot be used for a sign-in, the person is never let in without a second factor. Someone whose only method was WhatsApp cannot sign in until it works again.

Messages that are not sent show up in the WhatsApp message log. See WhatsApp channel.

> ⓘ **Go deeper**
>
> For operators: Identity server: second factors and codes

Set up single sign-on with Microsoft Entra ID

Single sign-on lets people sign in with the account your company already manages in Microsoft Entra ID. Setting it up is shared work:

Who	Does what
You (tenant administrator)	Decide the rules, fill in the worksheet, choose the domains, and test.
Your Entra administrator (often you)	Registers OSOS as an application in Microsoft Entra and creates a client secret.
Your operator	Adds the provider to the OSOS sign-in service using the worksheet. In 1.10 this cannot be done from **Administration**.

Before you begin

- You need access to the Microsoft Entra admin center, with permission to register applications, or someone who has it.
- You need a contact with your OSOS operator, and a secure way to send them a secret, such as a password manager or encrypted message. Do not send a client secret in plain email or chat.
- Decide what happens to people who already have a OSOS account, and to new people, as described in step 1.

Step 1: Decide the rules

Agree these with your operator before anyone registers anything. They go into part C of the worksheet.

1. **A short provider name.** Use lowercase letters only, for example `entra`. It becomes part of the sign-in address registered in Entra, so changing it later means updating Entra too.
2. **The button text**, for example *Contoso*. The sign-in page shows **Sign in with Contoso** under **Sign in to your account using**. You can supply a logo image address instead of the text.
3. **The domains** where the button appears. Identify them by name and **Host**.
4. **People who already have an account.** OSOS recognizes a person by the email address their Entra account reports. Choose one option:

- **They were imported from Entra by directory sync.** They are recognized automatically, with no extra step. See Directory sync.
- **Ask them to link their account once.** The first time they use the button, OSOS asks for their current password and shows **Link my account**. After that, the button signs them in directly.
- **Do neither.** People with an existing password account are refused when they use the button.

5. **People who have no account yet.** Either refuse them (you create their accounts first), or **create their account automatically**. If you create accounts automatically, choose the **User Group** and the **Space** that new people join, and optionally a rule such as "only email addresses that end in @contoso.com".
6. **Keep profiles up to date.** Choose whether OSOS should copy the person's name and email from Entra every time they sign in, or only when the account is created.

Step 2: Get your sign-in address

Entra needs to know where to send people back after they sign in. That address is built from your **sign-in server address** and the provider name:

```
https://<sign-in server address>/ext-auth/<provider name>/redirect
```

For example: `https://identity.contoso.com/ext-auth/entra/redirect`.

Ask your operator for the sign-in server address. You can also check it yourself: open the sign-in page and look at the browser's address bar, which shows the sign-in server's address. Use exactly that host, with `https://` and no trailing slash.

Step 3: Register OSOS in Microsoft Entra ID

Microsoft changes the admin center from time to time. If a label below does not match, see Microsoft's guide to registering an application.

1. In the Microsoft Entra admin center, open **App registrations** and select **New registration**.
2. Give it a name people will recognize, such as *OSOS portal*.
3. Under **Supported account types**, select **Accounts in this organizational directory only**, unless you deliberately want accounts from other organizations to sign in.
4. Under **Redirect URI**, choose the **Web** platform and paste the address from step 2. Do not choose *Single-page application*.
5. Select **Register**. On the overview page, copy the **Application (client) ID** and the **Directory (tenant) ID** into the worksheet.
6. Open **Certificates & secrets**, select **New client secret**, and choose an expiry. Copy the secret's **Value** immediately, because Entra shows it only once. Record its expiry date.
7. Open **API permissions**. Make sure the Microsoft Graph delegated permissions `openid`, `profile`, `email` and `User.Read` are present. If your organization requires it, select **Grant admin consent**.

> **Tip**
>
> Put the secret's expiry date in your calendar. When it expires, single sign-on stops working. Create a new secret in Entra before that date and send it to your operator the same way.

Step 4: Hand the worksheet to your operator

Send the completed worksheet to your operator, with the client secret sent separately through your secure channel. The operator adds the provider to the sign-in service and links it to your domains. It takes effect at the next sign-in, and no restart is needed.

> **Go deeper**
>
> For operators: Identity server: add an external identity provider

Step 5: Test it

1. Open the portal on one of the chosen domains in a private browser window.
2. Check that the sign-in page shows **Sign in to your account using** with your button, for example **Sign in with Contoso**.
3. Select the button, sign in with a test Entra account, and confirm that you return to OSOS signed in as the right person.
4. Test the cases you chose in step 1. Use an existing account (it should be recognized, or asked to link once), and a brand-new person (they should get an account in the chosen group and space, or be refused).
5. Sign out and check that you are also signed out of Microsoft.

If something fails, see Troubleshooting.

Step 6 (optional): Make single sign-on the only way in

When everyone on a domain can sign in through Entra, you can turn off the other options:

1. Edit the domain (see Choose how people sign in on a domain) and turn off **Support Local Sign in** and **Support Passkey**.
2. `Since 1.10` If single sign-on is now the only option on that domain, and your sign-in page has no sign-up link, people skip the sign-in page completely and go straight to Microsoft. The page, and its **Cancel** link, are no longer shown. Your operator can turn this skip off if you prefer to show the page every time. A sign-up link also keeps the page visible; see Organization settings.

Other identity providers

The same steps work with other identity providers that support OpenID Connect or OAuth 2.0 sign-in with a client secret, such as Okta, Google Workspace or a national digital ID. Each provider uses its own terms, but you collect the same things: the provider's sign-in and token addresses, a client ID and secret, the scopes to ask for, the address of the user's profile, and which profile fields hold a stable ID, the email address and the name. Register the same redirect address from step 2.

Single sign-on worksheet

Copy this table, fill it in, and send it to your operator. Send the client secret separately.

#	Item	Your value
A	**Sign-in server**	
A1	Sign-in server address (from your operator)	`https://`

#	Item	Your value
A2	Redirect address registered in Entra	`https://<A1>/ext-auth/<C1>/redirect`
B	**From Microsoft Entra ID**	
B1	Directory (tenant) ID	
B2	Application (client) ID	
B3	Client secret	*(sent separately)*
B4	Client secret expiry date	
B5	Permissions granted (`openid profile email User.Read`)	yes / no
B6	Accounts without a mailbox: use the sign-in name (user principal name) as the email?	yes / no
C	**Your decisions**	
C1	Provider name (lowercase, for example `entra`)	
C2	Button text, or logo image address	
C3	Domains where the button appears (name and host)	
C4	Existing accounts: imported by directory sync / link once with password / neither	
C5	New people: refuse / create automatically	
C6	If created automatically: User Group, Space, and any rule (for example, the email domain)	
C7	Update name and email from Entra at every sign-in?	yes / no
C8	Optional: your own wording for error messages	
C9	After testing, turn off passwords on these domains?	yes / no

What people experience with single sign-on

Security page for single sign-on users `Since 1.8`

People who sign in through an identity provider manage their password and second factor there, not in OSOS. On their **Security** page, in place of the two-factor and **Passkeys** tabs, they see: *"Two-factor authentication and passkeys for this account are managed by your organization's identity provider."* They also cannot change or reset a OSOS password.

OSOS does not ask single sign-on users for its own second-factor code. If you want a second factor for them, set it up in your identity provider (for example, with Microsoft Entra Conditional Access).

Signing out

Signing out of OSOS also sends people to the identity provider's sign-out page, so they are signed out there too.

Opening a specific provider directly `Since 1.10`

A link into OSOS can name the provider to use, so that people skip the provider choice and go straight to it. This is useful on an intranet page, for example. The portal's sign-in link supports this. See Signing in & choosing a Space [Using OSOS › Signing In & Choosing a Space]. The link uses the provider name (C1 in the worksheet), not the button text. If the name does not match a provider on that domain, the normal sign-in page appears.

Good to know

Cancel on the sign-in page `Since 1.4`

The sign-in page has a **Cancel** link. It ends the sign-in and returns the person to the app they came from, which for the web portal is its home page. If the sign-in page is left idle for 10 minutes, it cancels itself the same way.

Password reset links `Since 1.4`

When someone selects **Forgot password?**, OSOS emails them a reset link:

- The link works for 24 hours by default. Your operator can change this.
- Asking again sends a **new link and the old one stops working**. Only the newest link is ever valid.
- OSOS keeps only a scrambled fingerprint of the link, so the working link exists only in the email. The copy in the email log shows `[redacted]` in its place. See Reset and API tokens stay out of the email log.
- People who sign in through single sign-on cannot reset a OSOS password.

After an upgrade to a release with this protection, reset links that were sent before the upgrade stop working. People simply request a new one.

Locked and deactivated accounts

Repeated wrong passwords lock an account for a while. A deactivated user cannot sign in by any method. To protect against password guessing, the password pages also limit how many attempts can come from one place in a short time. People who hit the limit see an error and can try again after about 15 minutes.

Sign-in history

OSOS records every sign-in attempt, both successful and failed, including how the person signed in and whether a second factor was used. People can review their own recent sign-in attempts on their **Security** page. See Sessions & sign-in attempts [Using OSOS › Account Security & Revealing Protected Values].

When your organization shares a deployment `Since 1.10`

On a deployment that serves several organizations, each organization has its own separate sign-in service, with its own sign-in address, accounts, sessions and providers. Nothing you configure is visible to another organization. A few things you may notice:

- **The first administrator** of a new organization gets an email to set their password once the organization is active. The link returns them to the organization's own address.
- **Your brand** appears on the sign-in pages, in sign-in code messages and in authenticator apps, if your organization was registered with one. See Per-tenant brand.
- **Passkey prompts** show your organization's display name.
- **During maintenance**, for example while your operator restores a backup or moves your organization, nobody can sign in. People see a temporary error until the maintenance ends.

Troubleshooting

What you see	What to check
The button does not appear on the sign-in page.	The domain you opened is not one of the domains the provider is linked to (C3), or the operator has not activated the provider yet. Check the address in the browser against the domain's **Host** list.
Microsoft shows an error about the redirect address (for example *AADSTS50011*).	The redirect address in Entra does not exactly match A2. Check `https`, the host, the provider name and that there is no trailing slash. It must be registered under the **Web** platform.
People come back to the sign-in page with "Something went wrong during the login, please try again later!"	Most often the client secret has expired or was copied wrongly, or a permission is missing. Ask your operator to check the sign-in service log for the reason, and to add your own wording for that error if you want one (C8).
An existing user is refused.	Their OSOS account has the same email but is not linked yet, and account linking is off. Turn on "link once with password" (C4), or import the user from Entra.
A new person is refused.	Automatic account creation is off, or the person does not meet your rule (C5–C6).
New people sign in but have no access.	Check the User Group and Space chosen for new accounts (C6), and the group's permissions.
Nobody can sign in after you turned passwords off.	If you are still signed in, turn **Support Local Sign in** back on yourself. Otherwise ask your operator to turn it back on for that domain. Then test single sign-on again.
Someone could not sign in on a shared computer after another person's account was removed. `Since 1.10`	No action is needed. Sign-in now replaces the stale session automatically.
Everyone is suddenly asked for a code.	WhatsApp codes were set up, so everyone with a verified phone is asked. See WhatsApp codes.

Go deeper

> **Go deeper**
>
> For operators: Identity server & external providers · For integrators: Authentication & OAuth [Integrate & Extend] · For end users: Account security [Using OSOS › Account Security & Revealing Protected Values]

Manage permissions

Use this how-to when you need to give a team access to part of OSOS, change what they can do, or take access away. The examples use an HR department: an **HR** group and an **HR Manager** role. For how access works in general, read Permissions & access.

Before you begin

- You need the system **Admin** level to see **Administration › Access Management**.
- Unless you are a system Admin, you can only hand out levels up to your own.
- Give access to groups and roles rather than to individual people. It is easier to review and it follows people when they change teams.

The access levels

From least to most:

- **Restricted** — nothing on its own; it marks the person as a member so the access you give them elsewhere takes effect. Every person needs at least this at the system level.
- **View** — see the design of a module or entity, and open the views, pages and reports shared with them.
- **Developer** — build and change the design, without seeing the records.
- **Edit** — everything Developer can do, plus browse, open and change records in Administration.
- **Admin** — full control of the platform. System level only.

A person's access is the highest level they get from all their grants — direct, and through every group, role and org unit. No grant takes access away from another. To open records of an entity in **Administration** — even a single record — a person needs **Edit**; View is not enough.

Give a group a system level

1. Open **Administration › Access Management › Admin Permissions**.
2. Select the **Groups** tab, then select **Add**.
3. In **Select a group to add**, choose the group, for example an all-staff group.
4. In **Select permission level**, choose the level — usually **Restricted** — and select **Confirm**.

Give a group access to a module or data entity

1. Make sure the group has a system level (see above); only those groups are listed.
2. Open the module (**Administration › Applications › Modules**) or the data entity (**Administration › Applications › Data Entities**).
3. Open the **Permissions** tab and switch to **Groups**.
4. On the group's row, choose a **Permission Level** — **View**, **Developer** or **Edit** — and confirm. For example, give **HR Edit** on the Employees entity.

Access to a module covers every entity inside it. On an entity you cannot go below what the group already has on the module.

Give a group access to a view, page, report or service

1. Open **Administration › Access Management › Group Permissions**.
2. Under **Select Group**, choose the group.
3. Select **Add Permission**, and in **Object** pick the view, page, report or service.
4. For a service, choose **Audience** (they can submit it) or **Watcher** (they see all of its requests).
5. Save.

Adding a group to a view, page or report switches it to **Restricted**: only the people and groups on its list can open it.

> ⚠ **Known issue: a view granted here may not appear in the sidebar**
>
> When you give a group or a person access to a view from **Group Permissions** or **User Permissions**, the view (and its module) can be missing from their sidebar until the view is next opened in the view designer. To make it appear, give the access from the view's own **Permissions** dialog instead, or open the view once in the view designer (no need to save). The person may need to reload their window to see it.

Change or remove access

- **System level:** in **Admin Permissions**, change the row's **Permission Level**, or select the trash icon to remove it, and confirm.
- **Module or entity:** on its **Permissions** tab, change the **Permission Level** and confirm; set it back to **Restricted** to take away the access given there.
- **View, page, report or service:** in **Group Permissions** or **User Permissions**, edit or delete the row.

Removing a grant deletes it. It does not block anyone: a person keeps any access they still get through another group, a role or their system level. To find out where someone's access comes from, open **User Permissions**, choose the person, and read **Assignment Type** and **Acquired Through**.

Protect sensitive fields

A field such as **Salary** can be made **Sensitive** by the app builder in the entity designer (see Sensitive fields [Build Apps › Protect a field]). Its value is then hidden — shown as `****` with a lock icon — from everyone without a sensitive-field grant, Admins included. The platform never sends the hidden value to their device.

To let the HR Manager role see salaries:

1. Open **Administration › Access Management › Sensitive Permissions**.
2. Add a grant: choose the HR Manager role's group, and choose the **Salary** field as the scope. You can also set an end date.
3. Save.

People with the grant see an eye icon instead of the lock. Selecting it asks them to confirm their identity before the value is shown. The full procedure, including expiring grants and the per-user and per-group tabs, is in Sensitive Permissions.

Related

- Let the HR group see only HR requests
- Permissions & access
- Managing roles

Let the HR group see only HR requests

This how-to is for tenant administrators. You will set things up so that the HR team sees every request made through the HR services — leave requests, salary certificates, and so on — and does not see requests made through any other department's services.

How request visibility works

Who sees a request is not decided by access levels. It is decided by each **service**:

Who	Sees	Where they see it
The person who submitted the request	Their own requests	**Outbox**
The owners of a workflow step	Requests that reach a step they, or their group, own	**Inbox**
The service's **Watchers**	Every request of that service	**Watching**
The service's **Audience**	The service itself, so they can submit it	**Services Catalog**

So "HR sees only HR requests" means: the HR group is a **Watcher** of every HR service, and of no other service. You set watchers per service, so other departments' requests stay out of HR's view automatically.

> **Note**
>
> Access levels still matter in one place. Anyone with **Edit** or **Admin** at the system level can read the records of every service in **Administration**. Keep the HR team at **Restricted** or **View** at the system level.

Before you begin

- You need the system **Admin** level.
- The HR services must already exist. Designing a service is a builder's task; you only decide who watches it, and optionally who can submit it.
- The **HR** group exists and contains the HR team. See Managing groups.
- Every HR team member has a system level — at least **Restricted**. See Give everyone a base level.

Step 1: Make the HR group a watcher of each HR service

1. Open **Administration › Access Management › Group Permissions**.
2. Under **Select Group**, choose **HR**.
3. On the **Group Permissions** tab, select **Add Permission**.
4. In **Object**, choose the first HR service, for example **Leave Request**.
5. Under **Access Level**, choose **Watcher**.
6. Save.
7. Repeat for every other HR service.

The HR team now sees all requests of these services in **Watching**, including requests made before you added them.

Step 2: Check that HR watches nothing else

1. Still on **Group Permissions** for **HR**, review the list.
2. If HR is listed for a service that is not an HR service — for example the IT service desk — delete that row.
3. Open **Administration › Access Management › Admin Permissions** and check that the HR group is at **Restricted** or **View**, not **Edit** or **Admin**.

Step 3 (optional): Decide who can submit each HR service

By default a service can be open to all signed-in users — usually right for leave requests. For a service only HR should use, such as **Salary Adjustment**:

1. In **Group Permissions** for **HR**, select **Add Permission**.
2. In **Object**, choose the service, choose **Audience**, and save.

Adding a group to a service's audience limits the service to the people and groups on its audience list. Other staff no longer see it in the **Services Catalog**. If the service is also open to visitors who are not signed in, ask its builder to change that first — a service cannot be both open to visitors and limited to a list.

Step 4: Route the work to HR (builder setting)

Watching is read-only. If HR also has to act on requests — approve, reject, return for changes — the HR group must own the relevant workflow steps. Then the requests arrive in their **Inbox**. Step owners are part of the service design, so ask the service's builder to set the **HR** group (or the **HR Manager** role) as the owner of those steps.

Check the result

1. Open **Administration › Access Management › Object Permissions** and choose an HR service. The HR group is listed, marked **Watcher**.
2. Ask a member of HR to open **Watching**. They see the HR services' requests, and no others.
3. Ask someone outside HR to open **Watching**. The HR requests do not appear.

Troubleshooting

HR does not see requests in Watching. Check that the group is a watcher of that exact service (step 1), and that the person is a member of the HR group. Ask them to refresh the page.

HR sees requests from another department. They may be a watcher of that service, or they own a step in its workflow. Check **Group Permissions** for HR, and ask the other service's builder who owns its steps. Also check the person's own access in **User Permissions**: they may get it through another group or role.

A member of HR sees every request in Administration. They have **Edit** or **Admin** at the system level, or **Edit** on the module that holds the services. Lower it in **Admin Permissions**, or on the module's **Permissions** tab.

Go deeper

For builders: the service's **Watchers** and visibility settings are also in the service design, and step owners are covered in Workflows — Owner assignment [Build Apps › Workflows & approvals] · What end users see: Services and requests [Using OSOS › Services & Submitting Requests] · For integrators: Permissions API — Service requests [Integrate & Extend › Permissions API]

Create an API token for an integration

This how-to is for tenant administrators. When another system — a payroll tool, a reporting script, an automation platform — needs to read or write your data in OSOS, you give it an **API token**. You will create the token, give it exactly the access the integration needs, hand it to the developer safely, and later replace or revoke it.

How API tokens work

An API token belongs to an **application user**: a user record that stands for a system rather than a person. An application user cannot sign in to the portal. The system that holds its token acts as that user, so it can do exactly what you allow the application user to do — no more.

You never create a token on its own. You add an application user, and the platform creates the token for it and **emails it to you**, the person who added the user. The token is not shown on screen, and nobody can read it again later. If it is lost, you regenerate it.

Example: the finance team's reporting tool needs to read approved purchase requests. You add an application user called **Finance reporting**, share with it a view of the purchase requests that shows what the tool needs, and nothing else, and send its token to the developer who builds the connection. The tool reads through that view.

Before you begin

- You need permission to add users in **Administration › Organization › Users**. To regenerate a token later, you need the system **Admin** level.
- Your own account must have a working email address: the new token is emailed to you.
- Agree with the developer which data the integration reads or changes. Give each integration its own application user, so you can revoke one without breaking the others.

Step 1: Add the application user

1. Open **Administration › Organization › Users**, then select **Add User**.
2. In **Basic Info**, enter a name that says what the integration is, for example **Finance reporting**.
3. Set **User type** to **Application**.
4. Fill in **Account Information** as your organization requires.
5. In **Application information**:
    - In **Name**, enter the application's name.
    - Decide how long the token stays valid. Turn on **unexpired** for a token that never expires, or leave it off and set an **Expiry date**. The token stops working at the end of that day.
6. Select **Save**.

Within a short time you receive an email with the new API token. Your organization may have renamed some of these fields; the user type is the setting that matters.

> ⚠ **Treat the token like a password**
>
> Anyone who holds the token can act as this application user. Do not paste it into chat, tickets or shared documents. Pass it to the developer through your organization's password manager or secret store, then delete the email.

Step 2: Give the application user only the access it needs

A new application user can do almost nothing until you give it access. Grant it access exactly as you would for a person:

1. Give it a system level, usually **Restricted**. See Give everyone a base level.
2. Give it access to the data the integration works with. To **read** records, share a view of the entity with it — for example a Purchase Requests view — and the developer reads through that view. See Give a group access to a view, page, report or service; for one user, add the permission under **User Permissions** instead. To write records, or to read them straight from the entity, give it **Edit** on the entity. See Permissions & access.

View on the entity is not enough to read its records. View shows the entity's design, not its records, so when the integration asks the entity for its records it gets an empty list, with no error. For an integration that only reads, prefer a shared view to **Edit**: Edit would also let it change records. Do not give an application user the system **Edit** or **Admin** level unless the integration really administers the platform.

Step 3: Hand the token to the developer

Send the developer:

- the token, through a secure channel;
- your organization's OSOS address (the one people use to sign in);
- which data the application user can reach.

The developer's side — how the token is sent with each request — is in Use an API token [Integrate & Extend].

Replace a token (regenerate)

Regenerate a token when it may have leaked, when the developer lost it, or on your organization's rotation schedule. The application user, its access and its history stay the same.

1. Open **Administration › Organization › Users** and find the application user.
2. Open its menu, then select **Regenerate Token**. (It is also in the menu at the top of the application user's own page.)
3. Select **Regenerate** to confirm.

The new token is emailed to **you**, the administrator who regenerated it — not to whoever created the application user. The old token keeps working for a grace period (24 hours unless your operator changed it), so the developer has time to switch to the new one.

You cannot regenerate the token of an archived application user. Restore it first, or add a new application user.

Revoke a token

- **To pause an integration**, open the application user and select **Deactivate Account**. Its token stops working. Select **Activate Account** to allow it again.
- **To end an integration for good**, archive the application user. Its token stops working and cannot be regenerated.

Either way, allow up to about five minutes before the token is refused everywhere.

Good to know

- **The token is only ever emailed.** Creation sends it to the person who added the application user; regeneration sends it to the administrator who regenerated it. If the platform cannot find the account of the person who added the application user, no token is created — regenerate it from the user's menu.
- **Expiry is enforced.** After the **Expiry date**, the token stops working and the integration is refused. Regenerating does not change the expiry date; edit the application user to extend it.
- **Applications installed from the store and server-side functions** get their own application users automatically. You do not create tokens for them, and the token of a function is not emailed when you regenerate it: it goes straight to the function.

Troubleshooting

I never received the token email. Check your spam folder, and check that your own account has the right email address. If the email is lost, regenerate the token.

The integration is refused after it worked before. Check that the application user is not deactivated or archived, and that its **Expiry date** has not passed. If you recently regenerated the token, the old one stopped working when the grace period ended.

The integration gets an empty list, with no error. The application user cannot list that entity's records. View on the entity is not enough: share a view with the application user and read through it, or give it **Edit** on the entity.

The integration is refused for one kind of record only. The application user lacks access to that data. Check it in **Administration › Access Management › User Permissions**.

> ### Go deeper
>
> For integrators: Use an API token [Integrate & Extend] · Choosing a credential [Integrate & Extend › Choosing a Credential] · Token lifecycle [Integrate & Extend › Token Lifecycle]

Video: Manage permissions (incl. sensitive fields)

A short video (about two minutes) for tenant administrators: give a group access to an entity, see why records need **Edit**, then protect a sensitive field such as *Salary* so only the people you choose can read it.

> **The recording is not available yet**
>
> Until the video is published, follow the written steps in Manage permissions and Sensitive Permissions.

Storyboard

This is the script the recording will follow.

Scene	On-screen action	Narration (script)	Duration
1. Intro	The **Employees** entity in Administration.	"We'll decide who can see and change employee records, then protect the salary field."	8s
2. Open permissions	Open the entity's **Permissions** tab.	"Each module and entity has a Permissions tab."	8s
3. The levels	Show the levels: Restricted, View, Developer, Edit, Admin.	"Access comes in levels: Restricted, View, Developer, Edit and Admin."	12s
4. Grant View	Give the **HR** group **View**.	"View lets the HR group see the entity's design and open the views shared with them."	10s
5. Records need Edit	Point out that View does not show records in Administration.	"To browse or open the records themselves in Administration, a person needs Edit. Most people use shared views instead."	12s
6. Grant Edit	Give the **HR Managers** group **Edit**.	"HR Managers get Edit, so they can work with the records."	8s
7. Access adds up	Show a person in both groups.	"Access adds up: someone in both groups gets the higher level. To take access away, remove the grant that gives it."	12s
8. Where else	Show **Administration › Access Management**.	"You can also manage access per user, per group, and for views, pages and reports."	12s
9. Sensitive field	In the entity designer, mark **Salary** as **Sensitive**.	"Now we protect the salary. We mark the field as sensitive."	12s
10. Masking	Show the value as `****` with a lock icon.	"Everyone now sees stars — even administrators — because the value is hidden by the platform, not just the screen."	12s
11. Grant the field	In **Sensitive Permissions**, give **HR Managers** a grant for **Salary**; show the eye icon.	"We give HR Managers a sensitive permission. They see an eye icon, and can reveal the value after confirming who they are."	12s
12. Outro	The configured Permissions tab.	"That's it: the right people see the right records, and salaries stay private."	8s

Administration FAQ

Short answers to the questions tenant administrators ask most often. Each answer links to the page that shows you how. For questions about using OSOS day to day, see the Using the platform FAQ [Using OSOS › Questions from people using the portal].

People and accounts

Can people create their own accounts?

No. An administrator creates each account. If your organization uses single sign-on, your operator can let people's accounts be created automatically the first time they sign in with their company account. See Managing users and Sign-in methods & single sign-on.

How does a new user get their password?

When you create an account, the person gets an email with a link to set their own password. People who sign in with their company account (single sign-on) do not need one.

Can I see or set someone's password?

No. You can only send them a password-reset email; they choose the new password themselves. See Managing users.

Someone is locked out after too many wrong passwords. How do I unlock them?

There is no unlock button. The lock lifts by itself after a while, and it is longer each time it happens again. To let them in straight away, send them a password-reset email.

What is the difference between deactivating and archiving a user?

Deactivating blocks the account at once and signs the person out everywhere; activating it again lets them sign in with their old password. **Archiving** removes the account's sign-in details completely. If you restore an archived user, they must set a new password from the email they receive. To offboard someone who may come back, deactivate. See Managing users.

Can I import our users from Active Directory or Microsoft Entra ID?

Yes, but your operator runs the import; there is no import screen in **Administration**. Tell them which people and groups to bring in. See Users from your company directory.

Access

Someone is in several groups. Which access do they get?

The highest. Access from the person's own grants, their groups and their roles adds up, and nothing takes access away. See How grants add up.

How do I check what someone can access?

Open **Administration › Access Management › User Permissions** and choose the person. See Check what someone can access.

How do I take access away?

Remove the grant that gives it. Then check the person's groups and roles: if one of them still gives access, they keep it. See Change or remove access.

Someone can open a view but not the records in Administration. Why?

Seeing an entity's records in **Administration** needs **Edit** on that entity or its module. Most people do not need this: they work through the views, pages and services you share with them. See Permissions & access.

Can a system administrator see sensitive fields?

Not without a grant. Nobody, not even a system administrator, sees a sensitive value unless they hold a sensitive permission for it, and even then they must start a reveal session. See Sensitive Permissions.

How do I find everyone who can see a particular sensitive field?

Look at the field's own list of grants **and** its **Inherited Permissions** dialog: only the two together show everyone. See Inherited Permissions.

Can access be given for a limited time?

Yes, for sensitive fields: give the grant an **Expiry** and it ends by itself. See Expiring grants.

Delegation

How do I let someone act for a colleague who is away?

Create a delegation from the colleague to their stand-in, with start and end dates and what it covers. See Delegation.

Can the stand-in receive the colleague's notifications and emails?

Yes, if you turn on **Forward Notifications** and **Forward Emails** in the delegation. The colleague still receives everything. `Since 1.10` See Delegate copies.

How do I end a delegation that has no end date?

Set it to inactive, give it an end date, or archive it.

Spaces and organization

What is the difference between a space and a domain?

A **space** is a separate working area with its own data and members. A **domain** is a web address people use, with its sign-in options. See Spaces & domains.

I added someone to a space but they cannot see it. Why?

Check that they are on the space's **Users** tab and that the space is active. Then ask them to reload the page or sign in again. See Spaces & domains.

What is the difference between deactivating a space and deactivating its end-user interfaces?

Deactivate Space takes the whole space offline. **Deactivate End User Interfaces** stops people using its applications while you keep working on it in Administration. See Put a space into maintenance, or take it offline.

Where do I change the language, time zone and date format?

In a settings record, under **Administration › Interfaces › Settings**: one for the whole organization, and optional ones for a space or a person. See Organization settings.

Can I change our product name or logo?

Not in **Administration**. They come from your deployment or your organization's brand; ask your operator. Your own themes and fonts you can add yourself. See Organization settings and Look and feel.

Can I edit the list of currencies?

No. The currency list is the same for everyone.

How do I set working hours and holidays?

See Working hours & time monitoring.

Sign-in

Can people sign in with their Microsoft or other company account?

Yes. You prepare it with whoever manages your company's identity provider, and your operator connects it. See Sign-in methods & single sign-on.

Suddenly nobody can sign in. What do I do?

Ask your operator to check the human-verification service first: when it is down, sign-in stops for everyone. See Human verification.

Why is everyone with a phone number now asked for a sign-in code?

WhatsApp sign-in codes have been set up. Once they are, everyone with a verified phone number is asked for a code. `Since 1.10` See WhatsApp codes.

Messages and monitoring

Someone says they never got an email. How do I check?

Open **Administration › Monitoring › Logs**, tab **Email**, and check the message's **Status**. If there is no message at all, check the automation in **Reaction Logs**. See Monitoring: logs & audit.

Nobody gets text messages. Why?

Text messages need an SMS provider, which your operator connects. Ask them whether one is set up.

Our WhatsApp messages say “Not sent (sending disabled)”. Why?

Real WhatsApp sending is switched off for your deployment. Your operator switches it on. `Since 1.10`

Can administrators read password-reset links or API tokens in the email log?

No. They are removed from the copy kept in the log. See Notifications & email templates.

How do I find out who changed a record, a view or a permission?

Open **Administration › Monitoring › Audit Logs** and filter by **Event Name** and date. See Review who changed what.

Did an automation run, and what did it do?

Open **Administration › Monitoring › Reaction Logs** and filter by the reaction. A reaction whose filter did not match leaves no row. `Since 1.10` See Check whether an automation ran.

License

The Administration menu is missing for everyone. Why?

Your license most likely does not include administration. Ask your vendor to add it. See Your license.

We are getting warnings about the entries limit. What now?

Archive records you no longer work on — each then counts a quarter — or delete them, and ask your vendor for a larger limit. New records are refused at 110% of the limit. There is no usage screen in 1.10. `Since 1.10` See Plan limits.

> **Go deeper**
>
> For operators and integrators: Administration: technical questions