أســس
OSOS

OSOS DOCUMENTATION

General

A section of the OSOS Documentation and Help Center

English edition

Platform release 1.10
Published 23 November 2026

Contents

General 3
Overview 5
Core concepts 8
Versions 11
Change log 28
Certification paths 41

General

This section is for everyone, whatever you do with OSOS. Use it to find the part of the documentation written for your job, and to look up the things every reader shares: what the platform is, the words it uses, and what each version brought.

What do you want to do?

The rest of the documentation is organised by what you are trying to get done. Pick the description that sounds like you. Each one opens on a short “Start here” page. You can always come back.

Videos →		
Using OSOS →	[Using OSOS › Start Here]	I use apps in the portal, on the web or the mobile app: I submit and track requests, approve work in my inbox, find and update records, and read dashboards.
Administration →	[Administration › Start here: Administration]	I run my organization in the platform: who can sign in and how, users, groups and roles, what each person may see and do, look and feel, and monitoring.
Build Apps →	[Build Apps › Build apps — start here]	I design applications without writing code: data and forms, lists and views, services and approval workflows, automations, reports and dashboards.
Integrate & Extend →	[Integrate & Extend › Integrate & extend — start here]	I connect other systems to the platform: API tokens, reading and writing records through the API, events and webhooks, functions, packages and AI agents.
Setup & Maintenance →		I install, configure, upgrade and monitor the platform itself, and I am called when it is not working.

Not sure which one you are? Most people start as a portal user. If you were asked to “set up” the platform for your organization, you are probably its administrator. If you were asked to install it on servers, you operate it.

Pages for every reader

- **Overview** — what OSOS is, what you can build with it, and who uses it. Read this first if the platform is new to you.
- **Core concepts** — a short glossary of the words the rest of the documentation uses (entity, entry, view, service, workflow, space, role…), each with a link to the page that explains it.
- **Versions** — what each release changed, starting with a one-screen summary for each kind of reader.
- **Certification paths** — structured courses with an assessment at the end, from a two-day practitioner course to the master certification, in English and Arabic.
- **Change log** — corrections and changes to these documentation pages, by date.

Not sure yet?

Read the Overview and then Core concepts. Together they take about ten minutes, and they make every other section easier to read. If you would rather learn the platform as a guided course, start from Certification paths.

Overview

OSOS is a **no-code / low-code platform for building complete information systems** — web and mobile — by assembling them visually instead of writing an application from scratch.

If your organization runs on paper forms, email threads and spreadsheets, the work it does is real but invisible: nobody can say where a request is, who has it, or how long it took. OSOS is how that work becomes a system — one where data has a shape, requests have a route, the right people are notified, and anything can be reported on.

What you actually build

You do not start with a blank editor. You start with the pieces every information system is made of, and you configure them:

- **Entities and entries** — the shape of your data, and the records in it. An entity is a kind of thing (a ticket, an employee, an asset); an entry is one of them.
- **Views and pages** — how people see and edit that data: lists, cards, forms, dashboards, static content.
- **Services and requests** — the things people submit and track, with the route each one takes.
- **Workflows and reactions** — what happens next, automatically: who approves, who is notified, what gets created or updated.
- **Permissions** — who can see and do what. Two people opening the same screen can legitimately see different things.

Everything is multi-tenant and space-aware, so one deployment can serve many organizations without their data ever meeting.

Where the “low-code” comes in

Most of what you build needs no code at all. When the visual tools stop short — a computed value, a condition that needs real logic — you drop in a small **expression** rather than escaping to a full development project. That is the “last mile”, and it is deliberately small.

How it works, in general

Three things are worth knowing before you read anything else.

There is a building side and a using side. Someone designs the entities, views, services and workflows; everyone else works inside what was published. The two experiences are different by design, and this documentation is split along the same line — see the section map.

A space is the boundary. The platform is multi-tenant: one deployment hosts many organizations, and a *space* is where one organization’s data, users and configuration live. You sign in to a space, and what you can reach is scoped to it. Data in one space never meets data in another.

Permissions are evaluated per person, per screen. Two people opening the same view can legitimately see different rows, different fields, and different actions — and a field can be marked sensitive so its value is masked even from people who can otherwise read the entry. If a screen looks different from a colleague’s, that is usually configuration, not a fault.

Both the web portal and the mobile app run against the same platform, though they do not implement every capability identically; where they differ, the documentation says so.

What the platform offers

- **Advanced data design** — designing and organizing the data you work with, whether on the move or at rest.
- **Data management and control** — contexts that keep your data under control and secure.
- **Innovative user interfaces** — advanced interfaces that display and visualize data in a visually appealing way.
- **Advanced technological features** — email and SMS notifications, security, authentication, and access management.
- **Easy page building** — a quick way to build pages such as landing pages.

What organizations use it for

Organizations build applications on OSOS across a wide range of domains, such as:

- **Human resource management** — recruitment, training, performance, and payroll.
- **Communication and messaging** — internal and external email, messages, and notifications.
- **File and document management** — organizing, storing, and sharing files securely.
- **Service and workflow automation** — routing tasks and processes between users.
- **Digital experience platforms** — surveys, polls, and interactive experiences.
- **Project, team, and product management** — tracking progress through their various stages.
- **Medical management** — medical files, appointment schedules, and patient tracking.

Who uses it

Five groups, and the documentation is organized around them:

You are…	You mostly…	Start at
An **end user**	Submit and track requests, act on your inbox, work with records, read dashboards	Using OSOS [Using OSOS › Start Here]
An **app builder** (citizen developer)	Design the data, forms, services, workflows, automations and reports everyone else uses — without code	Build apps [Build Apps › Build apps — start here]
An **administrator**	Set up sign-in, manage users, groups, roles and permissions, and run your organization's settings	Administration [Administration › Start here: Administration]
An **integrator**	Connect other systems through the HTTP API, tokens, events and functions	Integrate through the API [Integrate & Extend › Integrate & extend — start here]
An **operator**	Deploy, configure, monitor, upgrade and recover the platform	Setup & Maintenance

Most people are only ever one of these. Some are several — a small organization's administrator is often also its app builder.

About this documentation

This site is the **documentation and help center** for OSOS. It is published in **English and Arabic**, and every page exists in both: use the language switcher in the header to move between them without losing your place.

The content is organized **by what you are trying to do**, not by how the product is built. Each audience section is self-contained: a page for end users, app builders or administrators describes screens and outcomes, in the words the product shows you. When there is more to know — the API behind a screen, how an operator configures it, how the engine works inside — the page says so in a collapsed **Technical details** block or a **Go deeper** link to the more specialized page. You never need to follow those links to finish your task.

Search covers both languages and indexes the full text of every page, so if you know the term you are looking for, searching is usually faster than navigating.

How to read the docs

A few conventions carry meaning:

- **Availability badges.** A heading or list item marked `` `Since 1.8` `` became available in that release. Milestones are recorded as buckets — `1.2`, `1.4`, `1.6`, `1.8` — and anything present before 1.2 is marked **Long-standing** rather than given a version. If a feature carries no badge, assume it has been there for a long time. Versions lists what each one brought.
- `OSOS` **and brand names.** The product name is substituted when the site is built, so the same content can be published under a different brand. Screenshots may still show another brand's name or logo.
- **Web vs mobile.** Where the mobile app behaves differently from the web portal, it is called out explicitly rather than left for you to discover.
- **What is *not* here.** This site documents the product as it ships. It is not a specification of your organization's own configuration — the entities, services and workflows you actually see were built by someone in your organization, and only they can explain those choices.

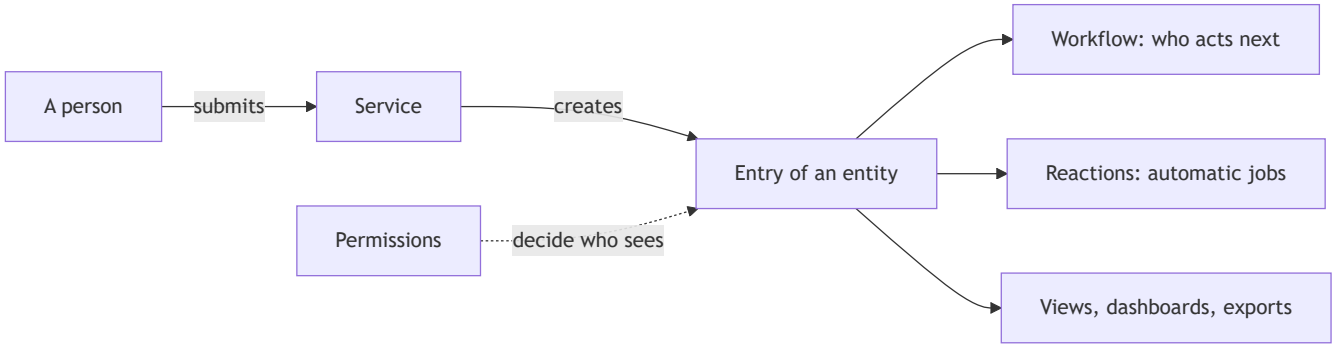
If something here contradicts what the product does, the product is right and the page is stale — the change log records when each part of the site was last brought up to date.

Core concepts

This glossary is for every reader. It explains, in a sentence or two, the words the rest of the documentation uses, and links each one to the page that covers it properly. Where a word matters to more than one kind of reader, there is a link for each: **Use** (people working in the portal), **Build** (people designing applications), **Manage** (administrators) and **Integrate** (developers connecting other systems).

The whole product in one picture

A person submits a **request** through a **service**. The request is stored as an **entry** of an **entity**, inside a **space**. A **workflow** routes it to the people who act on it, **reactions** run automatically along the way, and **permissions** decide who can see it. Everything after that — lists, dashboards, exports, notifications — reads the same entries.



Take a leave request: an employee opens the *Leave request* service, fills in the form and submits it. That creates one entry of the *Leave Request* entity. The workflow sends it to the employee's manager, a reaction emails the employee when it is approved, and HR follows every leave request from a list.

Data

Term	What it means	Learn more
Entity	A kind of thing your organization keeps records of — a ticket, an employee, an asset. It defines which fields each record has. Think of it as a table.	Build: Data Modeling: Entities [Build Apps › Create and design an entity]
Entry	One record of an entity: a single ticket, one employee. Each entry keeps its comments and a history of every change.	Use: Working With Data [Using OSOS › Create and Edit Records]
Field	One piece of information in an entry, such as a name, a date or an amount.	Build: Field Types [Build Apps › Choose a field type]
Field type	What kind of value a field holds and how it is entered: text, number, date, yes/no, a choice from a list, a file, a link to another record, and many more.	Build: Field Types [Build Apps › Choose a field type]
Reference	A field that links an entry to an entry of another entity — an asset request that points to the employee who asked for it.	Build: Relations & References [Build Apps › Link entities]
Query	A saved search or filter over entries, which you can reuse wherever a list is shown.	Use: Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List] · Build: Queries [Build Apps › Queries & filters]

Term	What it means	Learn more
Sensitive field	A field whose value is hidden as `****` unless you have been given access to it, and that may ask you to confirm your identity before it shows the value.	Use: Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values] · Build: Sensitive Fields [Build Apps › Protect a field] · Manage: Sensitive Permissions [Administration]

Screens

Term	What it means	Learn more
Module	One application, or one area of work: it groups the entities, views, pages and services that belong together, such as *Asset Management*.	Build: Modules [Build Apps]
View	A screen that lists and opens the entries of one entity, with its own columns, filters and form.	Use: Working With Data [Using OSOS › Create and Edit Records] · Build: Data Views & Faces [Build Apps]
Face	One way a view can display its entries: a table, cards, a calendar, a Kanban board, a tree, a timeline. A view can offer several.	Use: Switching View Faces [Using OSOS › Browse a List and Change How It Looks]
Page	A screen that is not tied to one entity: a home page or portal built from widgets, such as a list, a chart, text and links.	Build: Pages, Widgets & System Links [Build Apps › Pages & Widgets]
Dashboard and summary	A dashboard is a set of cards; each card (a summary) shows a number, a short list or a chart calculated from entries, such as *open requests by department*.	Use: Viewing Dashboards & Reports [Using OSOS › Dashboards, Counters & Reports] · Build: Dashboards & Summaries [Build Apps › Summaries, dashboards & counters]
Report	A formatted document produced from an entity's entries, which you can download or receive on a schedule.	Use: Viewing Dashboards & Reports [Using OSOS › Dashboards, Counters & Reports] · Build: Reports [Build Apps]
Template	The layout for a generated file, such as a Word, Excel or PDF document filled in from an entry.	Build: Templates [Build Apps › Export templates]
Space	A separate workspace with its own data and its own web address. Your organization may have one or several; you only see the spaces you belong to.	Use: Signing In & Choosing a Space [Using OSOS] · Manage: Spaces & Domains [Administration › Spaces & domains]
Administration	The area of the portal where administrators and builders work: users and access, applications, services, look and feel. You see only the parts you have access to.	Manage: Start here: Administration [Administration] · Build: Build apps — start here [Build Apps]

Work and automation

Term	What it means	Learn more
Service	Something people can ask for from the portal — leave, a laptop, a support ticket. Services are listed in the **Services Catalog**.	Use: Services & Submitting Requests [Using OSOS] · Build: Publish a service with an approval workflow [Build Apps]
Request	One submission of a service. It follows the service's workflow until it is completed.	Use: Submit & Track a Request [Using OSOS]
Inbox, Outbox, Watching	Where requests appear for you: **Inbox** holds the ones waiting for your action, **Outbox** the ones you submitted, and **Watching** the ones you follow without acting on them.	Use: Services & Submitting Requests [Using OSOS]

Term	What it means	Learn more
Workflow	The route a record takes: the steps it moves through, who acts at each step, and the actions they can take, such as approve, return for changes or reject.	Use: Executing Workflow Actions [Using OSOS › Acting on Workflow Steps] · Build: Workflows & approvals [Build Apps]
Reaction	An automatic job that runs when something happens — send an email when a request is approved, update a related record, generate a file.	Build: Reactions & Automation [Build Apps › Reactions & automation]
Pointer	A word that stands for "whoever or whatever is current" in a filter or rule, such as **@me** for the person using the screen or **@today** for today's date.	Build: Pointers and expressions [Build Apps › Builder concepts and glossary]
Expression	An optional formula for the cases the screens do not cover, such as a calculated value or a custom condition. You never need one to build an application.	Build: Pointers and expressions [Build Apps › Builder concepts and glossary]
Package	A bundle of built applications that can be moved to another environment or installed from the in-app Store.	Integrate: Packaging & Distribution [Integrate & Extend › Packaging & distribution]

People and access

Term	What it means	Learn more
User	Someone who can sign in.	Manage: Managing Users [Administration › Managing users]
Group	A set of users who get the same access at once, such as the HR team. It is the usual way to grant access.	Manage: Managing Groups [Administration › Managing groups]
Role	A position people hold, such as *HR Manager*. Each role has a group behind it that always contains the people who hold the role, so you can grant access to the role.	Manage: Managing roles [Administration]
Org unit	A part of your organization chart — a department or a section — with a manager and members. Workflows can send a step to "the requester's manager", and access can follow the chart.	Manage: Organization Units [Administration › Organization units]
Permission	Who may see or do what. Access is given to a user, group, role or org unit, on the whole platform, a module, an entity, or a single screen or service.	Manage: Permissions & access [Administration]
Access level	How much a permission allows: **View**, **Developer** (build and change the design without seeing the records), **Edit** (change the design and work with the records) or **Admin** (full control of the platform). **Restricted** gives nothing on its own; it only makes a person a member so that access given elsewhere takes effect.	Manage: Permissions & access [Administration]
Delegation	Letting someone act on your behalf for a while, for example during your leave. The platform records who really acted.	Manage: Delegation [Administration]
Application user and API token	An account used by another system rather than a person, and the secret that system uses to call the platform.	Integrate: API Tokens [Integrate & Extend › Use an API token]

More words

- How the builder pieces fit together, with a worked example: Build apps — start here [Build Apps] and Information-Structure Concepts [Build Apps › Builder concepts and glossary].
- What the version badges such as `Since 1.10` mean: Versions.

Versions

This page is for everyone who wants to know what a OSOS release changed. Start with the summary for your role below; the full history, version by version, follows it. Each entry links to the page that explains the feature.

1.10 at a glance

1.10 is the current release: **1.10.0** came out on 2026-11-17 and **1.10.1** on 2026-11-18. You have everything below once your organization runs 1.10. Where an item needs a later 1.10 update, the full list says so.

If you use apps

- **Totals and grouping in lists** — group a list by a field, with a subtotal for each group and a grand total over everything the list matches.
- **Quick export to Excel or CSV** from any list that allows it, including just the rows you selected.
- **Better account security** — recovery codes for your authenticator app, a list of your signed-in sessions that you can end one by one, and verification codes over WhatsApp where your organization turns them on.
- **Encrypted fields** — reveal a protected value after a one-time identity check; it hides again as soon as you move away.
- **Fewer surprises** — a warning when someone else saved the record you are editing (where it is turned on), one verification for a protected action on several records at once, and files that are kept only when you save.

Details: Everyday use · Protecting data · Totals and grouping

If you build apps

- **Encrypted fields** as an option next to sensitive fields, and a setting that decides whether an automation, a generated document or a summary may see protected values.
- **Totals** (Count, Sum, Min, Max, Average) defined once on an entity and switched on per view, plus grouping with subtotals.
- **Reactions** gain descriptions, run and failure counts, Set and Remove Permission actions, and a Send WhatsApp action.
- **Housekeeping** — move entities between modules, delete an unused entity or module, add custom indexes, and let a category field hold several values.
- **The in-app Store** — publish your applications as packages, with a second person approving each release, and install, update or fork (copy) apps from the Store.

Details: For builders · Packages and the in-app Store

If you administer your organization

- **The plan's entries limit is enforced**, with warnings to system administrators before anything is refused.
- **Delegation** can forward a person's notifications and emails to their delegate.
- **A Reaction Logs page**, application users with an expiry date, and a searchable list of each space's users.
- **WhatsApp** as a notification and verification channel, and a separate brand (name, logo, theme, email sender) for each organization on a shared installation.

Details: Administration

If you integrate through the API

- **An API description for each view**, which you can browse or use to generate a client.
- **No more silent overwrites** — a write based on an out-of-date copy of a record, workflow, view or page is refused, so you can reload and retry.
- **Clearer refusals** — many errors that used to look like server failures now come back as a client error with a stable code.
- **Files are temporary until a record saves them**, and sign-in can start with a named identity provider.

Details: For integrators

If you operate the platform

> ⚠ **Read before upgrading**
>
> In 1.10 the **combined-services pod is the only deployment shape**: the alternative of one deployment per service is gone. Read Infrastructure & operations and the 1.10 upgrade checklist before you upgrade.

- **Multi-tenant cells** — tenant routing at the gateway, fair per-tenant queues, dormant tenants and tenant moves between cells.
- **Tenant backup, restore and export**, and a maintenance mode that holds a tenant's requests while it is restored or moved.
- **Field encryption** needs a master key, or an external key vault.

Details: Infrastructure & operations

How to read this page

Features are recorded in **milestone buckets** rather than exact patch numbers, because a capability is rarely useful on its own: it lands with the screens, permissions and documentation around it. The buckets are **1.2**, **1.4**, **1.6**, **1.8** and **1.10**, plus **Long-standing** for anything that was already there before 1.2. A feature is listed under the milestone it **first became available in**, even if it was later also made available on an older line.

Throughout the documentation you will see the same vocabulary as a badge — a heading or list item marked `` `Since 1.10` `` became available in that milestone. A feature with no badge has usually been there a long time.

`Since dev` marks work on the development line, which is heading toward the next release (since 1.10 was cut, that is the 1.11 line). It is not in 1.10 or any earlier release, it may still change before it ships, and you will not find it on your installation unless you run that line. When the next release is tagged, each marker is replaced by its version number once that item is confirmed to be in it.

What changed in the documentation itself, rather than the product, is in the change log.

1.10

The largest infrastructure change recorded here, alongside a full pass on data protection. Five themes dominate: encryption and step-up verification, totals and grouping reaching general availability, account-security additions,

builder-facing governance and automation detail, and — the one operators must not miss — the combined-services pod becoming the **only** deployment shape.

Releases. 1.10.0 (2026-11-17) and **1.10.1** (2026-11-18), for both the server and the web portal. Further fixes will ship in **1.10.2**, which is not released yet. Everything on the line carries the `` `Since 1.10` `` badge; where it matters, an entry says which release it arrives in ("from 1.10.1", "from 1.10.2"). Work that is in the 1.10 code but is not a released feature — a proof of concept awaiting a product decision, or a screen whose app is only on the development line — stays `` `Since dev` `` and is not listed here.

Technical details

The 1.10.0 and 1.10.1 tags exist in both the backend and the web portal repositories. The Helm chart line that deploys 1.10 has reached chart `1.11.0`, still with app version `1.10.0`.

Everyday use

Working with records

- **A warning when someone else saves the record you are editing** — an optional check that refuses a save which would overwrite another person's work, plus a live notice while you are still editing. **Off by default**; an operator turns it on. See Entries [Using OSOS › Create and Edit Records].
- **Kanban boards by a reference field** — columns set to All, Used or Specific, your own column choice where the administrator allows it, and checks before a card moves. See Working With Data [Using OSOS › Create and Edit Records].
- **Category options from the entry form** — add or edit an option while filling in a record, where the administrator allows it; a category field can also hold several values. See Working With Data [Using OSOS › Create and Edit Records].
- **Files are kept only when you save** — an attached file waits in temporary storage until the record is saved, with a clear message when it has expired ("upload it again") or storage is briefly unavailable. See Working With Data [Using OSOS › Create and Edit Records].
- **Each row of a repeated block is recognised by a hidden id**, from release 1.10.1, and the upgrade gives one to every row saved before it. When a view lets you edit only some of a block's columns, deleting or reordering rows no longer moves the other columns' values between rows; a row you add starts empty in the columns you cannot edit. See Working With Data — saving through a view [Using OSOS › Create and Edit Records].
- **Pasted markdown is formatted** in an editor that works in markdown. See Working With Data [Using OSOS › Create and Edit Records].
- **One link style** for relation, URL and email values. See Working With Data [Using OSOS › Create and Edit Records].
- **Entry QR codes are generated in the browser** — no wait, no retry button, and the code always points at the entry's canonical link.

Finding and exporting

- **Quick export to Excel or CSV** — where the view allows export, an **Export Entries** button on any display exports every record that matches what you see. Up to 10,000 records download straight away; a larger export reaches a signed-in user as a "file ready" notification. You can also export just the rows you selected, up to 150, and a background export that fails no longer leaves a broken file or a "file ready" notification. It is web only, and CSV exists nowhere else. See Exporting Data [Using OSOS › Export a List].
- **A time range no longer breaks an export** — a ranged Time field is written as "from - to" instead of failing the whole file.

- **Quick filters with names, placement, order, groups and several defaults**, on the web and in the mobile app. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **Saved and quick queries that ask for values** — a parameterized query opens a *Query parameters* dialog, and each applied quick filter keeps its own values. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **An Expression as a filter value**, with a hint saying what the formula must return and a warning when it returns the wrong kind of value. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **One pagination bar on every paged list** — the item range, go to a page, and 20, 50 or 100 items per page. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **Clearing a sort stays cleared** instead of snapping back to the view's default.

Workflows and requests

- **Step-up verification on a workflow action, including in bulk.** A designer can require a second verification before a particular step action runs. Selecting several entries and running a gated action **now asks for one verification covering the whole selection**, rather than being unavailable in bulk. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **Acting on a group's step without taking ownership first**, where the page allows it, and an *update conflict* instead of a silent double action when someone else moved the workflow first. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **Reassign and Change Owner can carry a comment**, which now also appears in the entry's Discussion tab alongside ordinary comments and workflow-step notes. See Comments & Collaboration [Using OSOS].

Signing in and your account

- **TOTP recovery codes** — enrolling an authenticator app now also issues 8 one-time recovery codes for when the device is unavailable. See Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **Sessions & Sign-in Attempts** — review and end your own signed-in sessions device by device, and see recent sign-in attempts against your account. Ending another session really revokes it. See Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **Verification codes over WhatsApp**, where the organization turns it on and your phone number is verified. See Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **Sign-in** — a link can name the provider to start with, the login screen is skipped when one external provider is the only way in (on by default), and a page explains why the portal cannot start. See Signing In & Choosing a Space [Using OSOS].
- **A clear "try again shortly" while your organization is being restored or moved**. See Signing In & Choosing a Space [Using OSOS].
- **A side index on the User Preferences page**, a drawer on narrow screens. See Portal Navigation [Using OSOS › Portal Layout & Navigation].

Notifications and comments

- **Notifications** — copies forwarded from someone who delegated to you, notifications that open a page when clicked, and push sent once per device. See Receiving Notifications [Using OSOS].
- **Live notifications stay in your current space** — one that belongs to another space no longer appears while you work. See Receiving Notifications [Using OSOS].
- **Comments from visitors who are not signed in** show as *Anonymous* instead of breaking the public record page. See Comments & Collaboration [Using OSOS].

- **An unresolvable mention in a comment** shows as “Unknown” rather than breaking the entry page.

On mobile

- Grouping and totals at parity with the web, decrypt by tapping the masked value (with the same step-up check), a column-header quick-options menu, a working Reset on Style classification filters, an ad hoc step’s description as a tooltip, and a Duplicate command that always shows a label. See Web vs Mobile [Using OSOS].
- Recovery codes saved through the phone’s own save dialog, passkey rows showing when each was added and last used, Security as a tab of the User Center, and a dedicated Edit Basic Information form. See Account Security [Using OSOS › Account Security & Revealing Protected Values] and Web vs Mobile [Using OSOS].

Smaller changes — mention notifications now work on an installation that hosts several organizations, archiving a record someone else just archived says so clearly, and dashboards load their cards in one request.

Protecting data

- **Encrypted fields** — a field can be stored **encrypted at rest**, as a separate option from Sensitive. The two are **mutually exclusive**: a field is either masked-but-readable-with-a-grant, or encrypted. An encrypted field cannot be searched, filtered, aggregated, or used as the entity’s main title, and it is excluded from reports, templates and notification tokens. See Sensitive Fields [Build Apps › Protect a field].
- **Revealing an encrypted value needs its own step-up verification.** A user who holds the field’s grant reveals an encrypted value after a **single-use** identity check — the same authenticator, SMS, email, passkey or password methods used elsewhere, but proving who you are once, for this one reveal, with no standing session and no duration to pick. The value re-masks as soon as the field leaves the screen, and every reveal is recorded in the audit trail. See Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values].
- **A sensitive-data access policy on reactions, templates and summaries** — choose whether an automation, a generated document or a summary may see protected values at all, always, or only according to the person who ran it. A summary the policy denies is hidden entirely rather than shown blank. See Sensitive Fields [Build Apps › Protect a field].
- **Naming a protected field in the Administration query builder** no longer requires permission to read it. Nothing about which values you can see changes. See Permissions [Administration › Permissions & access].
- **Where the encryption key comes from** — besides the installation-wide master key, an operator can keep the key in **HashiCorp Vault** or **Azure Key Vault**. See Sensitive Fields — Encryption providers [Build Apps › Protect a field] and Configuration Reference.

Technical details

The installation-wide master key is set with `FIELD_ENCRYPTION_MASTER_KEY`. The Vault and Azure Key Vault providers work in `wrap` or `direct` mode.

Totals and grouping

- **Entity aggregates** — define Count, Sum, Min, Max or Average over a field once on the entity, then enable them per view. Totals are computed on the server over everything the list matches, not over the page you happen to be looking at.
- **Entry grouping** — group a list by one or more fields, with collapsible group headers, a subtotal row per group and a grand-total row. You can group by a reference field, and a localized text field groups on the language you are reading in. A protected field collapses into a single masked group rather than being refused.

- Group blocks follow the sort you applied, ordered by what you can actually see after filtering. See Entity Aggregates & Entry Grouping [Build Apps › Totals & grouping] for the design side and Entries [Using OSOS › Create and Edit Records] for everyday use.

For builders

Entities, fields and views

- **Moving and deleting** — a system administrator can move an entity, or up to 50 at once, to another module, and delete an entity or a module permanently once nothing depends on it. See Data Modeling: Entities [Build Apps › Create and design an entity] and Modules [Build Apps].
- **An Indexes tab** for custom database indexes on an entity, with their status and a retry; **tag groups** per entity; and an administrator's **Export Entries** (XLSX or CSV). See Data Modeling: Entities [Build Apps › Create and design an entity].
- **Permanently deleting entries also removes their virtual-entity copies**. See Data Modeling: Entities [Build Apps › Create and design an entity].
- **Multi-value category fields** (turning Multiple on is a **breaking** change for the entity) and **options users may manage** from the entry form. See Category [Build Apps › Category field].
- **Stable row ids in repeatable blocks**, and from release 1.10.1, saving rows through a view pairs each row with its stored row, so the columns the view does not let people edit stay with the right row. See Block fields [Build Apps › Block field] and Block: saving rows through a view [Build Apps › Block field].
- **Proxy (virtual entity) view governance** — a Proxy entity is now bound to a governing Proxy view that sets its capability ceiling (which fields, screens and create/edit/archive operations it exposes) and, where the view allows it, a write on the proxy reaches the source entity. See Virtual Entities — Proxy view governance [Build Apps › Virtual entities].
- **View configuration** — Kanban column modes and per-card move checks, quick filter names, placement, groups and defaults, and the gallery Quick export. See Data Views [Build Apps › Data Views & Faces].
- **Saving a view or page** refuses to overwrite someone else's newer change, and a view or page installed from a package can be renamed and given its own icon. See Data Views [Build Apps › Data Views & Faces] and Installed views [Build Apps › Data Views & Faces].
- **Layout references on the end-user screens** are resolved; the builder screens that use them stay `` `Since dev` ``. See Data Views [Build Apps › Data Views & Faces].
- **Admin Bar** — from a public page, jump straight to its definition in Administration, permissions allowing. See Portal Navigation [Using OSOS › Portal Layout & Navigation].

Queries, summaries and templates

- **Queries** — parameterized quick queries with per-query values, quick filter groups and a Queries tab toolbar. See Queries [Build Apps › Queries & filters].
- **The Expression value source** in criteria, with its result shape. See Criteria dialects [Integrate & Extend › Criteria and field-path dialects].
- **Duplicate a summary** — the same duplicate option reactions and views already offer, now on a Summary too, allowed to whoever may edit it. See Dashboards & Summaries [Build Apps › Summaries, dashboards & counters].
- **Copy and paste summary colours**. See Dashboards & Summaries [Build Apps › Summaries, dashboards & counters].
- **Conditional export templates**. See Templates [Build Apps › Export templates].

Reactions

- **A localized description, execution and failure counts on the list**, and a redesigned Reactions tab (search, filter, sort, grouping). See Reactions & Automation [Build Apps › Reactions & automation].
- **Set and Remove Permission actions**, capped by the access of whoever saved the reaction, and a Custom Function's result usable by chained reactions. See Reactions & Automation [Build Apps › Reactions & automation].
- **A Send WhatsApp action** that uses the space's approved message templates, with its options and template picker described in full. See the action options reference [Build Apps › Reactions & automation].
- **Files in a reaction's options are kept on save**, and a Call API action that tries to set one of the platform's own reserved headers is refused when you save it. See Files in reaction options [Build Apps › Reactions & automation].
- **Longer-running reaction actions pause and resume** instead of blocking while they wait for a reply, and runs interrupted by a restart are picked up again. See Configuration Reference — reaction saga configuration.

Technical details

- A view or page save based on an out-of-date copy answers `409 UPDATE_CONFLICT`.
- Saving block rows through a view: the columns the view does not grant are restored only for a row sent back with its `__id`; a row without one beside rows that have one is a new row; a migration gives every stored row an id.
- Reaction files on save: `400 FILE_EXPIRED`, `503 STORAGE_SERVICE_UNAVAILABLE`. A Call API action naming a reserved platform header: `400 RESERVED_HEADER_NAME`.
- Trigger context is stored only when a listener needs it.
- Long-running reaction actions use a durable bus workflow (the reaction saga mechanism) with a background sweep for runs a pod restart interrupted. It is in the 1.10 release.

For integrators

- **A per-view API document** — turn on **Swagger document** for a view and it publishes its own OpenAPI description, browsable in the product and usable to generate a client. See Calling the HTTP API [Integrate & Extend].
- **Optimistic concurrency on entry writes** — send back the entry's update identifier with your change and a conflicting write is refused instead of silently winning. See Data Modeling: Entities [Build Apps › Create and design an entity].
- **A concurrency guard on workflow writes** — a write against a workflow that changed in the meantime is refused. See Workflow model and API reference [Integrate & Extend].
- **Typed refusals instead of server errors** — a set of refusals of the caller's input or entitlement now answer with a client error and a stable error code (some of them also shipped in 1.8 updates), with new write refusals for plan limits. See Calling the HTTP API [Integrate & Extend] and its plan limits [Integrate & Extend › Calling the HTTP API].
- **Version 3 reads answer with the raw body again**, and tenancy and space refusals have their own codes. See Calling the HTTP API [Integrate & Extend].
- **Files are uploaded as temporary, then kept on save** — the platform records which record owns each file, a file that was never saved expires, and an orphan-file sweeper can clear what no record references. See Files [Integrate & Extend › Working with Files] and File Storage & Attachments [Integrate & Extend › Files: storage rules].
- **The gateway drops headers a client could use to pass itself off** as another service, or as a request made through a view. See Calling the HTTP API [Integrate & Extend].
- **Sign-in for integrations** — start sign-in with a named provider, bootstrap a browser client without a client secret, and per-tenant OpenID providers. See Authentication & OAuth [Integrate & Extend].

- **Functions** — per-tenant provisioning, and caller authentication on the provisioning webhook. See Functions [Integrate & Extend › Functions: run your own code].
- **Reactions** — a run-history endpoint per reaction and a system-wide Reaction Logs API. See Reactions & Automation [Build Apps › Reactions & automation].
- **Several statistic queries by id in one call.** See Queries [Build Apps › Queries & filters].
- **WhatsApp** — sending through the space's approved templates. See the WhatsApp API [Integrate & Extend › Calling the HTTP API].
- **Events while a tenant is under maintenance** are parked rather than delivered. See Data Model Events [Integrate & Extend › Data Model & Events].

Technical details

- Entry writes: send the entry's `updateIdentifier`; a conflicting write answers `409 UPDATE_CONFLICT`. Workflow writes answer `409 UPDATE_CONFLICT` the same way.
- `GET /api/v3/…` answers with the raw body.
- Typed refusals answer a `4xx` with a stable `error.code` instead of `500`.
- A file that was never saved expires with `400 FILE_EXPIRED`; the record of which entry owns each file is the file reference ledger.
- The gateway removes a client's `x-service` and clears its `x-view-id`.
- Sign-in with a named provider uses `idp_hint`.

Packages and the in-app Store

Publishing

- **Packages, published versions and a review inbox** — a package is a permanent publishing identity that owns modules; its versions are immutable builds; and nothing reaches the marketplace without a second person approving it. See Packages, Marketplace & the In-App Store [Integrate & Extend].
- **Connecting and setting up publishing are two separate steps**, and publishing and reviewing follow the marketplace groups — an administrator who is not in them cannot publish or review. The publisher terms are asked at approval, not when sending. See Connecting and setting up publishing [Integrate & Extend › Packages, Marketplace & the In-App Store] and The three marketplace groups [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Publishing revoked by the marketplace**, from release 1.10.2 — the package page shows the date and the marketplace's reason and stops offering to send; installing and updating keep working. See Publishing revoked by the marketplace [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **A README before the first version**, from release 1.10.2 — the package page says that a version keeps a copy of the README, and warns about a missing README only once it is known to be missing. See In the admin area [Integrate & Extend › Packages, Marketplace & the In-App Store].

Installing and updating

- **Installing from the in-app Store** — browse the marketplace catalog, install, fork or uninstall. What an installation created is read-only until you fork it. The marketplace connection is **off until an operator enables it**. The store can be browsed without a connection; only installing needs one, and installing connects first. See Packages, Marketplace & the In-App Store [Integrate & Extend].

- **Updating an installed app in place** to a newer listed version, with a preview of the release notes in between — no uninstall and reinstall. See Updating an installed app in place [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Recovering from a failed install, and uninstalling a fork** — a failed install keeps what it created protected and can be uninstalled; a forked installation can be uninstalled too. See Fork and uninstall [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Reporting a failed install or update to the app's publisher**, from release 1.10.1 — the installation page previews exactly what is sent, then sends the error and the failed resources, with an optional note and your contact. See Reporting a failed install or update [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Store and Installed polish** — Installed is filtered by *Status* and *Source* from the toolbar instead of tabs, store apps share one icon tile, and the set-up-publishing and connect steps explain themselves. See Store and Installed polish [Integrate & Extend › Packages, Marketplace & the In-App Store].

Exporting and importing

- **The packaging screens** — a package page with its history, export records (deleting one removes the record, not the file), an Installed list filtered by status and source, an installation page, and installed views and pages you can rename and give an icon. See The packaging screens [Integrate & Extend › Packaging & distribution] and Export records [Integrate & Extend › Packaging & distribution].
- **Who may export and import** — export needs Developer on the exported items; import is for space administrators only. See Packaging [Integrate & Extend › Packaging & distribution].
- **Mapped-field decisions** — when a mapped field's source cannot be matched by name, the installer decides what it copies from, and the import does not start until every such field is decided. See Mapped fields: the installer decides [Integrate & Extend › Packaging & distribution].
- **Package format 1.2** — a package can carry mapping hints so an imported mapped field still resolves when its parent entity is not in the package, and the import screen offers to **repoint a broken mapped field** instead of dropping it silently. See Schema versioning [Integrate & Extend].
- **Package formats 1.3 and 1.4**. See Schema versioning [Integrate & Extend].
- **Aggregates are packageable** and appear in the dependency graph, though an existing graph will not show them until it is rebuilt. See Resource schemas [Integrate & Extend › Package Resource Schemas].

Administration

People, delegation and sign-in

- **Delegation** — a delegation can forward the delegator's notifications and emails to the delegate, and an Allow All delegation no longer carries System Admin unless *Allow administration* is set. See Delegation [Administration] and Permissions [Administration › Permissions & access].
- **Users** — an application user can carry an expiry date and an issuer, and deactivating a user ends all of that user's sessions. A Space's Users tab can be searched and filtered. See Users [Administration › Managing users] and Spaces & Domains [Administration › Spaces & domains].
- **Identity providers** — a sign-in link can name the provider to start with, the login page is skipped when one external provider is the only way in, a default accent colour, and a separate identity setup for each organization on a shared installation. See Identity Providers [Administration › Sign-in methods & single sign-on].

Plans and licenses

- **The plan's entries limit is enforced** — warnings to system administrators at 80% and 100%, new entries refused only past 110%, and installing a package that would pass it refused up front. There is no web screen

showing entries usage. See Plan limits [Administration › Your license & plan limits].

Notifications, monitoring and branding

- **A Reaction Logs page** for system administrators, and permanent entity deletion recorded in the audit trail. See Monitoring, Logs & Audit [Administration › Monitoring: logs & audit].
- **Notifications** can carry an in-app link, delegates receive copies, and stale push registrations are cleaned up. See Notification Configuration [Administration › Notifications & email templates].
- **WhatsApp** — a notification channel with its own log tab, a sign-in and multi-factor code channel, and settings held in the new **System Settings** entity. There is no dedicated administration screen for them: the entity is edited through the generic entry editor or the API. See Organization Settings — System Settings [Administration › Organization settings], Notification Configuration — WhatsApp [Administration › Notifications & email templates], Monitoring — WhatsApp logs [Administration › Monitoring: logs & audit] and Identity Providers — WhatsApp [Administration › Sign-in methods & single sign-on].
- **A per-tenant brand** — a shared installation can serve organizations of several brands, each with its own product name, logo, theme and email sender. See Organization Settings [Administration › Organization settings].
- **An authorization shadow check** — optional, it compares access decisions in the background and **never changes one**. See Permissions [Administration › Permissions & access].

Technical details

- Deactivating a user removes every session record.
- The provider-naming sign-in link uses `idp_hint`; identity providers also gained public bootstrap endpoints.
- The marketplace entitlement is `marketplacePublish`.
- The authorization shadow check uses OpenFGA.

Infrastructure & operations

Before you upgrade

- **BREAKING — the combined-services pod is the only deployment shape.** The twelve service applications no longer have a standalone deployment to fall back to, and a chart value that tries to turn the combined pod off is refused. Addressing and routing are unchanged; only the pod boundary and the transport between co-located services move. See Service Topology and Deployment — the combined-services pod: rollout and rollback before upgrading.
- **Read the chart upgrade notes and the 1.10 upgrade checklist** (nine items) before upgrading across the 1.10 chart line. See Helm Chart — upgrade notes, Chart additions, 1.10.73 to 1.10.91, Helm Chart — chart additions, 1.10.92 to 1.11.0 and Deployment — 1.10 upgrade checklist.
- **The field-encryption master key** is required wherever encrypted fields are used, and cannot be changed once an environment has stored data with it. See Configuration Reference.
- **The scheduler key repair** is in the 1.10 release, and the chart turns it **on by default** from a later chart version (turn it off while upgrading from a release before 1.10). See Tenant Lifecycle — rollout flags.
- **A failed migration now fails the job** instead of reporting success over a half-migrated database, and the full-text index repair no longer lets one bad collection block the whole upgrade. See Helm Chart.

Multi-tenancy

- **Multi-tenancy runtime** — every service pod now receives its tenancy context from shared configuration. See Service Topology — Multi-tenancy runtime.
- **Multi-tenant cells** — guarded cell values, dormant tenants, fair per-tenant queue chunks, a migration job that lists the tenants it could not migrate, and shared-cell placement. See Helm Chart — multi-tenant cell values, Tenant Lifecycle and Tenant Provisioning.
- **The gateway** resolves the tenant from the host, rate-limits each tenant, strips forged space and credential headers, and answers an unknown host with "not found". See Gateway Operations.
- **Gateway header hardening** — a client's service and view headers are removed or cleared at the gateway. See Gateway Operations.
- **Space-scope enforcement** — enforced in the code, configurable per deployment. See Runbooks.
- **Tenant backup, restore and export** in a multi-tenant installation — API only, for the tenant's system administrator, with no screen in 1.10, and unavailable until the operator provides the backup tooling and the settings the chart does not render. See Tenant backup, restore and export.
- **Tenant maintenance** — while a tenant is restored, or held for a move to another cell, its requests are answered with "try again later" and its queued work is parked. See Maintenance & Operations, Tenant Lifecycle and Tenant Provisioning — moving a tenant.

New and optional pieces

- **New optional pieces** — a jobs pod, queue-depth autoscaling, signed storage links, the marketplace connection, and the web server configuration source. See Helm Chart and Service Topology — the jobs pod.
- **The on-prem/LDAP directory importer can now be deployed by the chart**, the same way the Azure AD importer already was, instead of requiring its own hand-run scheduled job. See Directory Sync — Chart-deployed on-prem importer [Administration › Users from your company directory].
- **Orphan file sweeper** — on in the chart, in dry-run, so it logs what it would delete and deletes nothing until you turn dry-run off. See Helm Chart and Runbooks.
- **Optional authorization shadow check**, the WhatsApp settings, a brands catalogue, and identity and storage files kept in Secrets. See Helm Chart — OpenFGA.
- **V2 web routing** — opt-in and off by default, at `/v2` of every main host; it replaces the earlier preview, whose setting now fails the render. The V2 web app itself stays `` `Since dev` ``. See Helm Chart — the V2 web app.
- **Backend images** are built from two new build files. See Deployment.

Resilience and monitoring

- **Redis failover** — clients follow the new master after a failover without a restart. See Maintenance & Operations.
- **Per-tenant and per-handler metrics**, and a tenant selector on the combined-services dashboard. See Observability.
- **Feature metrics and runbooks** for the shadow check, the sweeper, Redis failover, retention, backups and restores. See Observability and Runbooks.
- **New environment variables** and **five new runbooks**. See Configuration Reference and Runbooks.

Technical details

- **Combined-services pod.** The twelve service applications are `data`, `workflow`, `reaction`, `timeline`, `logging`, `dependency`, `packaging`, `audit`, `comments`, `functions`, `integrations` and `notification_api`. Chart `1.10.4` deleted their 24 per-application `deployment.yaml` / `hpa.yaml` templates and made `core.enabled: true` the default; `core.enabled: false` is refused at render time. Addressing (the

Kubernetes Services, Kong routing) is unchanged. A chart upgrade also makes `core.mongoUrlTemplate` mandatory.

- **Chart line.** Upgrade notes cover `1.10.12` to `1.10.72` (twelve notes), then `1.10.73` to `1.10.91`, then `1.10.92` to `1.11.0` (app version still `1.10.0`). The scheduler key repair is on by default from chart `1.10.93`; set it to `false` while upgrading from a release before 1.10. OpenFGA keeps up to 20 idle Postgres connections per pod from chart `1.10.94`.
- **Configuration.** Tenancy context is `TENANCY_MODE` / `APP_ID` from a shared ConfigMap. The field-encryption master key is `FIELD_ENCRYPTION_MASTER_KEY`. The jobs pod is selected with `CORE_ROLE`; the web image takes an nginx config source.
- **Gateway 2.6.x.** Per-tenant rate limiting answers `429` / `503`; an unknown host answers `404`. A client's `x-service` is stripped by the chart's gateway configuration and its `x-view-id` is cleared by the gateway plugin, which needs a gateway image built after `2.6.11`.
- **Tenant backup and maintenance.** Backup, restore and export are for the tenant's `SystemAdmin` and answer `503` until the operator provides Percona Backup for MongoDB. A tenant in maintenance answers `503 TENANT_IN_MAINTENANCE` with `Retry-After`.
- **V2 web routing** is `endUser.enabled`; it replaces the `webV2` preview key.
- **Images.** Backend images are built from `Dockerfile.workspace` and `Dockerfile.service`.
- **Dashboard.** The tenant selector is the `$tenant` variable on the merged-core dashboard.

1.8

Four themes dominate: sensitive data, workflow ownership, the mobile app catching up with the web portal, and a reworked custom-action model for builders.

Sensitive data

- **Sensitive fields and step-up unmask** — a field can be marked sensitive, so its value shows as `****` in lists, on the entry, in exports and in history for anyone not allowed to read it, and it cannot be filtered on either, so a value can't be inferred through search. Seeing a value takes a **step-up verification** — an authenticator code, or a one-time code by email or SMS — for a period you choose. Editing still works blind. See Entries [Using OSOS › Create and Edit Records].
- **Sensitive Permissions administration** — grants are managed from a dedicated page, given **per user or per group**, and can be set to **expire**. See Permissions [Administration › Permissions & access].
- **Sensitive fields on mobile** — masked values, in-place editing of masked fields, and the grant-management screens are all now in the app. Removing a grant now **deletes** it outright, on mobile and on the web portal alike. See Web vs Mobile [Using OSOS].
- **Sensitive data reveal is now per session, not per account** — unmasking a value on one device no longer leaves it visible to the same account signed in elsewhere, and the reveal ends the moment that sign-in session does. See Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values].
- **Administration needs a license that includes it** — without it, the Administration menu is not shown; everyday entry work is unaffected. See Your license [Administration › Your license & plan limits].
- **Opening a holder's full access record from the inherited-permissions view** — a user or group row in the sensitive-field inherited-permissions dialog is now clickable, opening that holder's complete grant record instead of only a summary. See Sensitive Permissions [Administration].

- **A granted sensitive field on mobile unmasks immediately on reveal**, instead of needing a full screen reopen. See Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values].

Workflows

- **Unowned group steps and Take Ownership** — a step handed to a group can sit unowned until someone claims it, and a group-owned step lists every member with their own state, so people who have not acted yet stay visible. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **External Step** — a step can hand execution off to an external system and resume when that system calls back, displayed full screen, inline, in a side panel, a dialog or a popup. See Web vs Mobile [Using OSOS].
- **Automations on a step's owner change** — react to the owner of a step changing, to notify the new owner or record the hand-over.
- **Cleaner history after a reset** — the execution history shows only the current run, so a step can no longer appear as both executed and current.
- **A Data View's workflow tab can default to Path or History** — previously it always opened on Path. See Data Views & Faces [Build Apps].
- **List, card, badge, kanban and tree reads now hide a workflow-step-hidden field's value**, not just its label — the cell renders blank and can't be inline-edited there. See Working With Data [Using OSOS › Create and Edit Records].
- **An ad-hoc workflow step can carry a short description** when inserted, shown afterward as a tooltip. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].

Everyday use

- **Human verification when signing in** — sign-in, multi-factor and password reset can require a short verification check. See Signing In & Choosing a Space [Using OSOS].
- **More accurate Empty and Filled conditions** — these now also check that the field is present on the entry, which fixes repeatable content reporting as Filled when it had no rows. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **Reordering repeatable values** — repetitions of a repeatable field or block can be dragged into order while adding or editing, on web and mobile. See Entries [Using OSOS › Create and Edit Records].

For builders

- **Custom Action** replaces the Custom Event trigger, configured with a custom action level — UI, Local or External (renamed from Module/Internal partway through 1.8) — plus a *Contextual* switch, in place of the old UI Event toggle.
- **Execute Custom Action** — a new automation action that runs a Local- or External-level custom action on the current entry, one reached through a reference field, or (later in 1.8) through a proxy entity, with parameter binding from a constant, a field value, an expression, or a pointer. See Reactions & Automation [Build Apps › Reactions & automation].
- **Entity designer deep links** — the open tab is kept in the address bar, so it can be bookmarked, shared, and reached with browser Back and Forward.
- **Quick edit enforces business rules** instead of silently skipping them.
- **Per-language entry history** — a multilingual field's change is shown language by language, so editing only the Arabic text is no longer invisible when reading the history in English.

- **Live mobile preview** in the page builder now covers pages as well as data views, and re-renders in place while you edit. See Web vs Mobile [Using OSOS].
- **A view-scoped template export reaches every non-sensitive entity field**, not only the triggering view's displayed columns. See Templates [Build Apps › Export templates].
- **The field-config panel gained a searchable jump-to-section index** — a navigation aid for a field with many configuration blocks, not a new field option. See Data Modeling: Entities [Build Apps › Create and design an entity].

For integrators

- **Application-user API tokens can be regenerated from the admin area**, with a grace period so the old token keeps working while the new one rolls out. See Regenerating a token [Integrate & Extend › Use an API token].
- **A signed-in user can discover their own modules and views** through the API, without needing admin-scoped access to enumerate them. See Modules [Build Apps].
- **Two more routes on every view's execution plan**, added to existing views by an upgrade step. See Workflow model and API reference [Integrate & Extend].

Infrastructure & operations

- **The Grafana datasource plugin changed how panels are authored** — the query cascade now starts at the Module instead of the Entity, and an ordinary (non-admin) editor's own access now suffices to build a panel. **Breaking:** a panel option for the system time field is removed, replaced by four always-present audit-stamp columns; a saved panel using the old option needs re-authoring. See Grafana Dashboards [Integrate & Extend].
- **The human-verification and link-shortener services no longer build from backend source** — both now ship as images published by their own repositories, deployed either per cell or as one shared cluster-wide instance (that choice is unchanged). See Helm Chart.
- **The message-queue setup step is self-healing** on a fresh install or a retried upgrade, instead of occasionally requiring an operator to intervene. See Helm Chart.

Technical details

- The Grafana datasource plugin version is v1.2.0; the removed panel option is `systemTimeField`.
- The two services are `altcha` and `shortener`; the self-healing step is the RabbitMQ vhost creation.

Delivered to 1.8 as updates

Shipped in 1.8 updates and carried into 1.10:

- **A permission grant cannot exceed your own level**, and a malformed grant record no longer hides a user's system grants. See Permissions [Administration › Permissions & access].
- **Authoring a reaction needs Developer access or higher on its entity**, and a Call API action refuses private and reserved network destinations unless an operator allow-lists them. See Calling the HTTP API [Integrate & Extend] and Configuration Reference.
- **The Call API limits in full** — besides private destinations, the whole exchange must finish within 30 seconds, bodies are capped at 10 MiB, and a token's value can never add a header. See Call API: what the platform allows [Build Apps › Reactions & automation].
- **Refusals that used to look like server errors** — executing a hidden workflow action or taking ownership of a step that already has an owner is now refused as invalid, and another service's refusal is passed on with its own status. See Calling the HTTP API [Integrate & Extend].

- **Edit is shown on an entry after its workflow completes**, to the owner of the completed step, and a button that runs a job waits for it, with the values it saved appearing straight away on public pages. See Working With Data [Using OSOS › Create and Edit Records].
- **A view that hides its page header still shows Save, Edit and the workflow buttons** — only the title part is hidden now; on the 1.10 line from release 1.10.2. See Working With Data — saving through a view [Using OSOS › Create and Edit Records].
- **Saved CSS classes show again in a view's custom-style settings**, instead of appearing blank after a reload; on the 1.10 line from release 1.10.2. See Data Views [Build Apps › Data Views & Faces].
- **A duplicated reaction can change its event**, and a source the new event rules out falls back to *Self*. See Defining a reaction [Build Apps › Reactions & automation].

Technical details

- Grant above your own level: `PERMISSION_GRANT_EXCEEDS_CALLER_ACCESS`.
- Reaction authoring without Developer access: `REACTION_AUTHORING_FORBIDDEN`.
- The Call API allow-list for private destinations: `CALL_API_ALLOWED_PRIVATE_DESTINATIONS`.
- The hidden-action and already-owned-step refusals answer `422` instead of `500`; another service's `4xx` is passed on with its own status.

1.6

- **Sentinel & TLS for high availability** and a dedicated **health check** — see Operations.
- **Update audit diff** — see Queries [Build Apps › Queries & filters].
- **Extended export** — see Templates [Build Apps › Export templates].
- **Signed, cacheable file links** — a browser can load a file in a plain image tag without a token, and a file whose content is missing is reported as not found. Delivered to the 1.6 line as an update (and to 1.8). See File Storage & Attachments [Integrate & Extend › Files: storage rules].
- **Secrets redacted in reaction history**, as an update to the 1.6 and 1.8 lines. See Reactions & Automation [Build Apps › Reactions & automation].
- **Templates on virtual entities**, as an update to the 1.6 and 1.8 lines. See Templates [Build Apps › Export templates].
- **Workflow corrections** — a step's field rules still apply to its participants on an admin page (an update to the 1.6 and 1.8 lines), and an ownerless step can be reassigned again, because the owner picker lists every active user (an update to the 1.6 line). See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps] and Workflow model and API reference [Integrate & Extend].
- **Delegation fixes** — only the delegate can act under a delegation, and signing out ends the delegation you were acting under; updates to the 1.6 and later lines. See Delegation [Administration].
- **A sign-in deep link**, added to the 1.6 line. See Signing In & Choosing a Space [Using OSOS].
- **Pickers search every language** — a reference or user picker finds a record by a title entered in any of the organization's languages; an update to the 1.6 and 1.8 lines. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **Grants no longer hide each other** — Edit and Developer on the same module are both honoured, and an Admin grant on a module or an entity opens nothing; an update to the 1.6 and 1.8 lines. See Permissions [Administration › Permissions & access].

Technical details

- A signed file link whose bytes are missing answers `404`.
- The sign-in deep link is `/signin`.

1.4

- **Sub-workflows** and **conditional target steps** — see Workflows [Build Apps › Workflows & approvals].
- **Entity-scoped relations** — see Queries [Build Apps › Queries & filters].
- **Template parameters** — see Templates [Build Apps › Export templates].
- **A Cancel link on the sign-in page**, as an update to the 1.4, 1.6 and 1.8 lines. See Signing In & Choosing a Space [Using OSOS].
- **Large Excel template exports streamed page by page**, as an update to the 1.4, 1.6 and 1.8 lines. See Templates [Build Apps › Export templates].
- **Request and summary corrections** — completed requests found under Inbox → Completed after a sub-workflow, request lists repaired, and a summary's "N Entries" caption that counts with the view's query; updates to the 1.4, 1.6 and 1.8 lines. See Services & Requests [Using OSOS › Services & Submitting Requests] and Dashboards & Summaries [Build Apps › Summaries, dashboards & counters].
- **Reaction runtime corrections**, such as the loop guard scoped to the workflow step. See Reactions & Automation [Build Apps › Reactions & automation].
- **Security hardening** — the sign-in rate limiter stays on unless an operator explicitly turns it off, and the functions provisioner is called with its own token instead of a shared key. See Configuration Reference and Functions [Integrate & Extend › Functions: run your own code].
- **Password-reset links** — the token is stored hashed, asking again replaces the earlier link so only the newest one works, and reset and API tokens are kept out of the email log; an update to the 1.4, 1.6 and 1.8 lines. See Account Security [Using OSOS › Account Security & Revealing Protected Values] and Notification Configuration [Administration › Notifications & email templates].
- **A stale copy of an entry no longer undoes a workflow step move** made by a Move Workflow Step reaction; an update to the 1.4, 1.6 and 1.8 lines.
- **Saving through a view keeps the repeated-block columns the view does not let you edit**, instead of wiping them from every row; an update to the 1.4, 1.6 and 1.8 lines, and in 1.10 from release 1.10.1. See Working With Data — saving through a view [Using OSOS › Create and Edit Records].
- **A Yes/No (Boolean) field whose default is True opens ticked** on the web create form; an update to the 1.4 and 1.8 lines, reaching the 1.6 and 1.10 lines in their next update. See Boolean fields [Build Apps › Boolean field].

Technical details

- The identity rate limiter is on unless `APPLY_RATE_LIMITER` is exactly `false`.
- The functions provisioner is called with a bearer token instead of the shared key.

1.2

- **The `Under` operator** for querying hierarchies — see Queries [Build Apps › Queries & filters].
- **Custom variables** in templates — see Templates [Build Apps › Export templates].

- **Hiding the API description hides all of it**, not only the explorer page; an update to the 1.2 line and every later one. See Configuration Reference.

Technical details

`ENABLE_SPEC=false` hides the whole API spec, not only the explorer page.

Before 1.2 — Long-standing

Everything the platform was already built on: entities and entries, views and pages, services and requests, workflows and reactions, the permission model, spaces and multi-tenancy. These carry the **Long-standing** marker rather than a version, and are described throughout Using OSOS [Using OSOS › Start Here] and Build apps [Build Apps › Build apps — start here].

Not yet attributed to a milestone

These capabilities are documented, but the release that introduced them has not been established, so they are deliberately listed without a version rather than guessed at:

- **Packaging** — export built resources from one environment and import them into another. See Packaging [Integrate & Extend › Packaging & distribution].
- **Virtual entities** — joiner, proxy and aggregator entities materialized from other entities. See Entities [Build Apps › Create and design an entity].
- **Expanded field types** — Map (Geo), localized text, Sequence, Icon, Switch, Block, Password, rich text. See Field Types [Build Apps › Choose a field type].
- **Virtual (computed) fields and advanced queries** — filtering across relations, on computed fields and on workflow metadata, plus aggregate operators. See Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List].
- **Statistic queries, dashboards and charts** — see Summaries and Dashboards [Using OSOS › Dashboards, Counters & Reports].
- **Functions (custom code)** — standalone and value validators, and reusable logic called from reactions and templates. See Reactions [Build Apps › Reactions & automation].
- **Bulk actions** — bulk edit, version upgrade, archive/restore and permanent delete. See Entries [Using OSOS › Create and Edit Records].
- **Templates and export** — XLSX export and DOCX/HTML/XLSX rendering with PDF conversion. See Templates [Build Apps › Export templates].
- **Preview mode** — build views, dashboards and statistic queries against deterministic sample data. See Preview mode [Build Apps › Preview Mode].
- **Human verification on public entry points** — see Human Verification [Administration › Human verification (bot protection)].

Change log

What changed in **this documentation**, and when. For what changed in the product itself, see Versions.

Keeping the two apart matters: a page can change because the product changed, or because the page was wrong, incomplete or badly organized. This log records the second kind as plainly as the first, so you can tell whether something reads differently because it *is* different.

How this log works

The documentation is brought up to date on a **fixed cadence**, once per two-week sprint, rather than continuously. Each run compares the documentation against the current source code, updates what drifted, and adds one entry here. Entries between those runs cover structural or editorial work on the site itself.

The newest entry is first. Dates are the date the change landed.

2026-11-20 — reader-first rollout

The reader-first model from the pilot below now covers the whole site.

- **Development is split in two.** Build apps [Build Apps › Build apps — start here] is for citizen developers who build applications in the portal without code. Integrate & extend [Integrate & Extend › Integrate & extend — start here] is for developers who connect other systems through the API. The old addresses of the moved pages redirect to their new place.
- **Every Using, Administration and Build-apps page was rewritten** for its reader: the goal first, steps with the labels you see on screen, then good-to-know notes and troubleshooting. Source paths, endpoints and configuration keys were moved out of these pages.
- **About sixty new companion pages** in Integrate & extend, Setup & maintenance and Core hold the technical detail that was moved. Reader pages link to them through **Go deeper** boxes.
- **One PDF per section and language**, linked from each section's header and from the home page, instead of a single PDF for the whole site.
- **Real credentials removed** from the examples. Every example secret is now a placeholder.
- **One brand parameter set for deployment builds**: OSOS, Apptomator or a client's own name, logos, favicon and colours, for the site and the PDFs.

2026-11-20 — reader-first pilot

A **pilot of a reader-first structure**, not a sync. The team reported that pages for end users, app builders and administrators read like engineering notes, and a review confirmed it. This entry rewrites a representative slice to the new model so it can be judged before the rest of the site follows:

- **Front door.** The home page now asks what you want to do — use apps, build apps, run your organization, integrate through the API, or deploy and operate — and the Overview names five audiences instead of four.
- **Rewritten for their readers:** the Using, Administration and Build-apps start pages; Services & requests [Using OSOS › Services & Submitting Requests]; Permissions & access [Administration] and Roles; Sign-in methods & single sign-on [Administration], with a Microsoft Entra ID set-up; Workflows & approvals [Build Apps].

- **New task pages:** Let a group see only its requests [Administration › Let the HR group see only HR requests] and Publish a service with an approval workflow [Build Apps].
- **Technical depth moved, not dropped**, to new companion pages for integrators and operators, reached through **Go deeper** boxes and collapsed **Technical details** blocks. Links into deeper sections now degrade to plain text in a narrower build.
- **Corrections found on the way**, each checked against the code: saving a request does not send it on; the real request statuses and sidebar labels; opening a single record needs Edit since 1.8; choosing None removes a grant rather than denying; the workflow **Return** action goes to the first step.

2026-11-20

A sync covering the backend, the web portal, the Helm chart (up to chart `1.11.0`, still deploying app version `1.10.0`) and the function provisioner. The mobile app, the API gateway and the other services did not change in this period. **The 1.10 releases are now tagged**: `1.10.0` on 2026-11-17 and `1.10.1` on 2026-11-18. The 1.10 branches already carry fixes for `1.10.2`, which is not tagged yet, so a few items say which release they arrive in ("from 1.10.1", "from 1.10.2"). A fix that also reached an older line carries the earliest line it reached; most of the new work is on the development line and is marked `` `Since dev` ``. See Versions § 1.10 for the themed list.

Corrections that change what a page tells you:

- **A chained reaction is held to its triggering space only on the development line.** On 1.10 a reaction chained onto another reaction's success or failure, given a wider space scope, runs once per space, each time with the original space's entry; give it the Triggering Space only. The development line runs it once and refuses to save a wider scope. See Reactions & Automation — chaining reactions [Build Apps › Reactions & automation].
- **Filtering Installed by status and source from the toolbar is in 1.10.0.** The 2026-11-17 entry below, the Versions page and the Development FAQ said it was on the development line only and that 1.10 shows tabs. The `1.10.0` tag carries it, with the other Store and Installed refinements. See Store and Installed polish [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Whether the scheduler key repair runs depends on the line and the chart.** The 2026-11-17 entry said it stays off by default. A 1.10 backend leaves it off unless it is switched on, but from chart `1.10.93` the chart switches it on for the whole deployment; the development line turns it on by default and removes its rollout flag. When upgrading from a release before 1.10, switch it off until every scheduler pod runs the new release. See Helm Chart — scheduler key repair on by default.
- **A reaction's condition is no longer checked in a temporary database collection.** The sequence diagram of an entry create said so; since 1.10 the condition runs as a query over the entry held in memory, and nothing is written. See Key Flow Sequences.

Highlights, by who will notice them:

- **Using the portal.** When a view lets you edit only some columns of a repeated block, saving keeps the other columns (an update reaching 1.4 and later lines), and from 1.10.1 each row is recognised by a hidden id, so deleting or reordering rows is safe. A Yes/No field whose default is True opens ticked, and a view that hides its page header keeps its Save, Edit and workflow buttons. See Working With Data — saving through a view [Using OSOS › Create and Edit Records].
- **Administering.** On the development line, regenerating a function's token is recorded in the audit trail as a function token rotation, and the token is not e-mailed to you. See Monitoring, Logs & Audit [Administration › Monitoring: logs & audit].

- **Building.** An administrator can report a failed store install or a stopped update to the app's publisher (from 1.10.1); a package page shows when the marketplace has revoked publishing, and asks for a README before the first version (from 1.10.2). The reactions page now covers who may author a reaction, the Call API limits, the Send WhatsApp options and files held in reaction options in full. See Packages, Marketplace & the In-App Store [Integrate & Extend] and Reactions & Automation [Build Apps › Reactions & automation].
- **Building, on the development line.** Functions gain encrypted secrets, delivered as environment variables and never exported in a package, and an application user of their own. Stale package draft and README saves are refused, a page widget that shows a view outside the package can be mapped at install, and pasted web images are copied into a README. See Function secrets and per-function identity [Integrate & Extend › Functions: run your own code] and Page widgets pointing outside the package [Integrate & Extend › Packaging & distribution].
- **Operating.** Chart `1.10.92` to `1.11.0`: scheduler key repair on by default, OpenFGA keeping its Postgres connections open, and two settings for development-line backends. The 1.10 upgrade checklist has a ninth item. A development-line backend needs function provisioner `1.6.0` or later **before** it is upgraded. See Helm Chart — chart additions, 1.10.92 to 1.11.0 and Functions in a Multi-Tenant Deployment.
- **The FAQs** gained fourteen answers on these changes, and the answers this run made wrong were rewritten: the Installed filters, the chart version and upgrade notes, and the scheduler key repair.

Now `` `Since 1.10` ``, after being `` `Since dev` ``: the Store and Installed refinements (the Installed toolbar filters, the store icon tile, the set-up-publishing and connect explanations, and the smaller fixes). Each was checked against the `1.10.0` tag.

Marked `` `Since dev` ``, and why: these are on the development line only.

- **Function secrets and each function's own identity.** Their data migrations are numbered `1.10.0-a53` and `1.10.0-a54`, but they are not on the 1.10 branch.
- **Holding a chained reaction to its triggering space**, and refusing chained reactions that could never fire.
- **Removing the scheduler key repair's rollout flag.**
- **Stale-save refusal for package drafts and READMEs**, page-widget mappings at install, and pasted web images in a README.
- **Speech input, the shell builder screens and the V2 web app**, which stay as described in the 2026-11-17 entry.

One fix ships as a patch on every line rather than in a milestone: access to an entity named in a request's address is now decided on the entity that name actually resolves to. It has been delivered to the 1.4, 1.6, 1.8 and 1.10 maintenance branches. *(Updated 2026-11-23: it ships in 1.4.120, 1.6.70 and 1.8.29; on the 1.10 line it ships in the next patch after 1.10.1.)* It is described in the internal engineering pages.

2026-11-17

A sync at the **1.10 general-availability cut**, covering the backend, web portal, API gateway, Helm chart, and the tenant-management and space-provisioner services, up to chart `1.10.91`. The backend and the web portal each now have a 1.10 release branch carrying version `1.10.0`. The `1.10.0` tag is not yet published, so the line keeps `` `Since 1.10` `` under the convention this site ratified for it. The mobile app did not change in this period. A fix that also reached a published line carries the earliest line it reached, and this run moved several items from `` `Since 1.10` `` back to `` `Since 1.4` ``, `` `Since 1.6` `` or `` `Since 1.8` `` for that reason. See Versions § 1.10 for the themed list, and Delivered to 1.8 as updates for the fixes that went back to older lines.

Corrections that change what a page tells you:

- **Human verification on public forms is not automatic, and it does not let a form through when it fails.** The pages said every anonymous form got the check and that an outage let the submission through unverified. In fact an administrator turns it on view by view, it is **off by default**, signed-in users are never challenged, and a form that requires it is refused while the verification service is down. See Services & Submitting Requests [Using OSOS].
- **An installed app can now be updated in place.** The 2026-11-05 entry below, and the pages it described, said there was no update flow. See Updating an installed app in place [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **The in-app Store can be browsed without a connection**, connecting and setting up publishing are two separate steps, and publishing and reviewing need the marketplace groups **even for administrators**. See Connecting and setting up publishing [Integrate & Extend › Packages, Marketplace & the In-App Store].
- **Deleting a package version or export removes the record, not the file**: the ZIP stays in storage. See Export records [Integrate & Extend › Packaging & distribution].
- **Many refusals that answered `500` now answer `4xx` with a stable code.** Some of them shipped in 1.8 patches and are tagged that way. See Calling the HTTP API — typed refusals [Integrate & Extend › Calling the HTTP API].
- **With the default `local` field-encryption provider, the wrapped data keys are stored in the platform's own database.** A page that implied they never were has been corrected. See the Setup FAQ.
- **Tenant maintenance is now reachable**: restoring a tenant, or holding it for a move to another cell, answers `503 TENANT_IN_MAINTENANCE`. Pages that called the maintenance state not yet reachable have been rewritten. See Maintenance & Operations.

Highlights, by who will notice them:

- **Using the portal.** Files you attach are kept only when you save; pasted markdown is formatted; you can export only the rows you selected; verification codes can arrive over WhatsApp; signing in while your organization is restored or moved says to try again shortly; and live notifications stay in your current space. See What's New — Latest in 1.10 [Using OSOS › What's New].
- **Administering.** The plan's entries limit is enforced, with warnings at 80% and 100% of it and refusal only past 110%, and no web screen for entries usage. WhatsApp arrives as a notification and sign-in code channel, configured in the new System Settings entity, which has **no dedicated screen**. An OpenFGA authorization shadow check can be turned on; it never changes a decision. See Plan limits [Administration › Your license & plan limits], Organization Settings — System Settings [Administration › Organization settings] and Permissions [Administration › Permissions & access].
- **Building.** Files are uploaded as temporary and kept on save, with a file reference ledger, `FILE_EXPIRED` and an orphan-file sweeper. Views and pages installed from a package can be renamed and given an icon, and the packaging screens gained a package history, export records and installation pages. See File Storage & Attachments [Integrate & Extend › Files: storage rules] and The packaging screens [Integrate & Extend › Packaging & distribution].
- **A new page: Tenant backup, restore and export.** It is for operators of a multi-tenant deployment. These are API calls only, with no screen in 1.10, and they answer `503` until the operator provides Percona Backup for MongoDB and the settings the chart does not render.
- **Operating.** Read the upgrade notes up to chart `1.10.91` and the new 1.10 upgrade checklist. New on this line: V2 web routing at `/v2` (opt-in, through `endUser.enabled`, replacing `webV2`), Redis clients that follow a Sentinel failover, the orphan-file sweeper (on in the chart, in dry-run), optional OpenFGA, a brands catalogue, WhatsApp settings, and five more runbooks. See Helm Chart — upgrade notes, Deployment — 1.10 upgrade checklist and Runbooks.

- **The FAQs** in every section gained answers for these changes, about eighty in all. Answers this run made wrong were rewritten: human verification on public forms, “no update flow”, the scheduler key repair, the chart version, the gateway headers and the encryption keys.

Now `Since 1.10`, after being `Since dev`: each was checked against the 1.10 release branches, item by item.

- the reaction saga mechanism, and the message bus’s transactional outbox that made it necessary;
- the tenant maintenance state, now that restores and moves use it;
- the scheduler key repair, which **stays off by default**;
- layout references on the end-user route (the backend half only);
- live notifications filtered to your current space;
- formatting of markdown pasted into a markdown editor, which was added to the 1.10 branch after the cut.

Still marked `Since dev`, and why: these are in no released version.

- **Speech input** — the code is in the 1.10 branches, but it is a proof of concept that awaits a product decision, off by default.
- **The shell builder screens**, and the Layout builder template that uses layout references — the app exists only on the development line. The backend shell entity and the `shell_builder` entitlement are `Since 1.10`.
- **The V2 web app itself** — only its routing at `/v2` is part of 1.10.
- **Filtering Installed by status and source from the toolbar**, and the other packaging-screen refinements made after the branch point. In 1.10 the Installed page uses tabs. *(Corrected on 2026-11-20: these refinements are in the `1.10.0` release, and Installed is filtered from the toolbar there too. See the entry above.)*

See Helm Chart — the V2 web app and Store and Installed polish [Integrate & Extend › Packages, Marketplace & the In-App Store].

2026-11-05

A sync covering the backend, web portal, mobile app, API gateway and Helm chart: about six weeks of work on the 1.10 line, up to chart `1.10.72`. **1.10 is still not tagged as a general-availability release.** The backend is at `1.10.0-alpha.72` and the web portal at `1.10.0-alpha.59`; the mobile app ships as `1.10.2`. Work on that line carries `Since 1.10`, by the convention this site ratified for the 1.10 development stream. A fix that was also delivered to a published line carries the earliest line it reached (`Since 1.4`, `Since 1.6` and so on). See Versions § 1.10 for the themed list.

This run also **corrected several statements that were wrong**, some because the code changed and some because the page had been wrong all along. They come first, because a page you relied on may now say the opposite of what it said before.

Corrections that change what a page tells you:

- **CSV export now exists, but in only two places.** The pages said there was no CSV export anywhere. A list’s new **Quick export** (on the web, where the view allows export) and an administrator’s **Export Entries** on an entity page offer **Excel or CSV**. The older Table export, templates and reports stay XLSX/PDF, and the mobile app has no Quick export. See Exporting Data [Using OSOS › Export a List] and Data Views — Gallery quick export [Build Apps › Data Views & Faces].
- `GET /api/v3/…` **answers with the raw body again** on the 1.10 line. The HTTP API page had described the `{success, result}` envelope as universal. `/v3` writes, `v1` and `v2` keep the envelope, and a 1.8 deployment still envelopes `/v3` reads. See Calling the HTTP API [Integrate & Extend].

- **An unknown host is now `404 SPACE_ERROR`** at the gateway (it was `400`). A service reached directly without a space answers `400 SPACE_CONTEXT_MISSING` (it was a `500`). See Calling the HTTP API — tenancy refusals [Integrate & Extend › Calling the HTTP API].
- **An Allow All delegation no longer passes on System Admin** unless the delegation has *Allow administration* set. See Permissions [Administration › Permissions & access].
- **Reactions.** The Create and Update Data Entity actions are retired: they are refused with `422` on save and skipped at run time. A reaction's run history is for system administrators only. See Reactions & Automation [Build Apps › Reactions & automation]. The aggregate copy endpoint the pages listed never reached a release and has been removed; see Entity Aggregates & Entry Grouping [Build Apps › Totals & grouping].
- **Operators.** Backend images are built from `Dockerfile.workspace` and `Dockerfile.service`; the shared backend `Dockerfile` is gone. The Sentry trace sample rate defaults to 1% in code, while the chart renders `0.1`. Pages that called multi-tenant ("multi") mode not operator-ready now describe it as opt-in, with the chart refusing an unsafe configuration. See Deployment, Observability and Helm Chart — multi-tenant cell values.
- **Two security fixes reached older lines.** `ENABLE_SPEC=false` now hides the whole API spec (`` `Since 1.2` ``), and the identity rate limiter is on unless `APPLY_RATE_LIMITER` is exactly `false` (`` `Since 1.4` ``). See Configuration Reference and its identity additions.
- **One anchor was renamed.** Links to *Upload validation & security* on the file storage page now use `#upload-validation-security`.

Highlights, by who will notice them:

- **Using the portal.**
    - Kanban boards can be split by a reference field, and you may be able to choose the columns yourself.
    - Quick filters have their own names, placement and groups.
    - Saved and quick queries ask for their values.
    - A formula (Expression) can be a filter value.
    - Every list has one pagination bar, with go-to-page and 20/50/100 rows per page.
    - You can add a category option from the entry form.
    - A delegator's notifications can be forwarded to you.
    - You can act on a group's step without pressing Take ownership first, and you see an *update conflict* message when someone acted first.
    - Sign-in goes straight to your only provider, and a page explains why the portal cannot start.
    - On mobile, recovery codes are saved through the phone's save dialog, and Edit Basic Information is its own form.

    See Working With Data [Using OSOS › Create and Edit Records], Filtering, Sorting & Search [Using OSOS › Search, Filter, Sort and Group a List], Receiving Notifications [Using OSOS], Executing Workflow Actions [Using OSOS › Acting on Workflow Steps] and Signing In [Using OSOS › Signing In & Choosing a Space].

- **Administering.** Delegations can forward the delegator's notifications and emails. There is a Reaction Logs page, sign-in links can start a named provider, and a Space's Users tab can be searched. See Delegation [Administration], Monitoring, Logs & Audit [Administration › Monitoring: logs & audit] and Identity Providers [Administration › Sign-in methods & single sign-on].

- **Building.** You can move entities between modules, delete an entity or a module permanently, add custom indexes and define tag groups. Category fields can hold several values; turning that on is a **breaking** change for

the entity. Reactions gain Set and Remove Permission actions, export templates can be conditional, and packages move to schema `1.3` / `1.4`. See Data Modeling: Entities [Build Apps › Create and design an entity], Category fields [Build Apps › Category field] and Reactions & Automation [Build Apps › Reactions & automation].

- **A new page: Packages, Marketplace & the In-App Store [Integrate & Extend].** It covers package identity, published versions, the review inbox, installing from the in-app Store, and why an installed resource is read-only. The marketplace connection is **off until an operator enables it**, and there is no update flow yet.

- **Operating.** Read the twelve upgrade notes for chart `1.10.12` to `1.10.72` before you upgrade. New on this line: the optional jobs pod, queue-depth autoscaling, dormant tenants, gateway 2.6.x with per-tenant resolution and rate limiting, per-tenant metrics, and five new runbooks. See Helm Chart — upgrade notes, Gateway Operations and Runbooks.

- **The FAQs** in every section gained answers for these changes. Answers this run made wrong (CSV, the combined-services pod, `SPACE_ERROR`, the shared Dockerfile) were rewritten.

Still marked `` `Since dev` ``, and why: some code on the 1.10 line is not released even under the convention above. It is a proof of concept, off by default with nothing that uses it yet, or not reachable at all. That covers:

- speech input and the V2 web preview;
- the shell builder app and layout references;
- the tenant maintenance state;
- the scheduler key repair, which is so far the only rollout flag (the rollout-flag mechanism itself is `` `Since 1.10` ``);
- the reaction saga mechanism.

See Helm Chart — speech input and Tenant Lifecycle — rollout flags.

2026-09-22

A sync covering the backend, web portal, mobile app and Helm chart, and the release that carries last cycle's development-line work into general availability as **1.10**. Two kinds of change land in this entry together: genuinely new capabilities from this cycle, and the reclassification of everything the 2026-09-05 entry below described — it was accurate as `` `Since dev` `` the day it was written, and ships as `` `Since 1.10` `` as of this release. See Versions § 1.10 for the full, themed list with a link to every page it touches.

Highlights:

- **Decrypting an encrypted field now requires its own step-up identity check** — a single-use proof, distinct from the longer Sensitive-field unmask session. It used to be one click with no check at all; that changed before 1.10 shipped, so pages that still described the one-click behavior have been corrected. See Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **That same kind of step-up check now covers a bulk workflow action in one prompt**, instead of being unavailable once you select more than one entry. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **Multi-tenancy is now a first-class runtime concern** for operators — every service pod carries tenancy context from a shared ConfigMap, part of the same infrastructure work that reshaped deployment (see the breaking change below). See Service Topology.

- **Field encryption can now point at HashiCorp Vault or Azure Key Vault**, not only the deployment-wide master key. See Sensitive Fields [Build Apps › Protect a field].
- **Your account security screen gained two capabilities**: TOTP recovery codes for when your authenticator device is unavailable, and **Sessions & Sign-in Attempts**, where ending a session really ends it. See Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **A Proxy (virtual entity) view now governs what its proxy may do** — a capability ceiling on which fields, screens and operations it exposes, and, where the view allows it, a write that reaches the source entity. See Virtual Entities [Build Apps › Virtual entities].
- **BREAKING for operators: the combined-services pod is no longer optional.** Chart `1.10.4` deleted the alternative, one-Deployment-per-service shape entirely — every 1.10 deployment now runs the combined pod, and `core.mongoUrlTemplate` becomes mandatory. Read Service Topology before you upgrade.
- **Everything marked `` `Since dev` `` in the entry below is now `` `Since 1.10` `` ** — the development line it documented has shipped as a release. Nothing in the product changed between the two entries; only its official availability did.
- Smaller, and all user-visible: Reassign and Change Owner can carry a comment, visible in the Discussion tab; a Reaction gains a localized description, execution/failure counts and a run-history endpoint; a Summary can be duplicated; the mobile column-header quick-options menu; and an **Admin Bar** that jumps from a public page straight to its Administration definition.

Corrections made in the process:

- Three claims in the 2026-09-05 entry below had quietly stopped being true since it was written: decrypt's "no verification step," bulk workflow step-up being unavailable, and the combined-services pod being optional. That entry is left as written, with the change noted there; see it and Versions § 1.10 for the current, correct facts.

2026-09-05

A sync covering the backend, web portal, mobile app and Helm chart. Almost everything below is **on the development line only** — it carries the `` `Since dev` `` marker, it is in no released version, and it will not be present on your deployment unless you are running that line. The marker is a filter: pick a released milestone in the version control on any page and the development-only material disappears. See Versions for the list of what is in it.

Highlights:

- **A field can now be stored encrypted, and that is a different thing from marking it sensitive.** The two are mutually exclusive — a field is either masked but readable by anyone you grant access to, or encrypted. Revealing an encrypted value takes one click and **no verification step**, where revealing a sensitive one still asks you to verify. In exchange, an encrypted field cannot be searched, filtered, totalled, or used as an entity's main title. See Sensitive Fields [Build Apps › Protect a field] and, for what it feels like as a user, Account Security [Using OSOS › Account Security & Revealing Protected Values].
- **Lists can now group and total themselves.** Define Count, Sum, Min, Max or Average on an entity, enable them on a view, and a list gains collapsible groups, a subtotal per group and a grand total. Two things a reader should not confuse: `-` means there was nothing to total, `****` means it was totalled and withheld from you. This is a new page — Entity Aggregates & Entry Grouping [Build Apps › Totals & grouping] — with the everyday side on Entries [Using OSOS › Create and Edit Records].
- **An optional check refuses a save that would overwrite someone else's work**, and warns you live while you are still editing. It is **off unless an operator turns it on**, which is why the page says so before it says anything else. See Entries [Using OSOS › Create and Edit Records].

- **A workflow action can require a second verification before it runs** — and deliberately cannot be run that way on several entries at once. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **A view can publish its own API document.** Turn on *Swagger document* and that view gets an OpenAPI description you can browse in the product or point a client generator at. See Calling the HTTP API [Integrate & Extend].
- **The documentation now describes a deployment shape it had never mentioned.** The twelve service applications can be hosted in one or two processes instead of twelve, with calls between co-located services made in-process. It is **disabled in the shipped chart**, and nothing about addressing or the gateway changes while it is off. Correcting the pages around it also fixed a number of statements that had quietly stopped being true — that each service is always its own process, which health-check paths and metrics endpoints to use, and the chart version stamps. See Service Topology.
- **Package format 1.2**, which lets a package carry the hints an imported mapped field needs, and lets you repoint a broken one during import instead of having it dropped silently. The operator rule that comes with it: upgrade the target environment before the source. See Schema versioning [Integrate & Extend].
- **A failed database migration now fails the job.** It used to log the error and exit successfully, so Kubernetes reported the job complete and Helm reported the upgrade succeeded over a half-migrated database. See Helm Chart.
- Smaller, and all user-visible: entry QR codes are generated in your browser, so there is no wait and no retry button; a time *range* no longer fails an entire export; clearing a sort stays cleared; a comment mention that cannot be resolved shows as "Unknown" instead of breaking the entry page; and real-time notifications from another space no longer arrive in the one you are in.

Correction, added 2026-09-22. Three of the facts above have since changed — each was accurate the day this entry was written, and each stopped being true before any release shipped it, so the entry is left as written and the change is recorded here instead of by quietly editing history:

- Decrypting an encrypted field is **no longer** one click with no verification step. It now requires its own single-use identity check on every reveal.
- Step-up verification on a workflow action is **no longer** unavailable in bulk. Selecting several entries and running a gated action now asks for one verification covering the whole selection.
- The combined-services pod described above **is no longer optional**. Chart `1.10.4` deleted the alternative (one-Deployment-per-service) shape entirely, so every 1.10 deployment runs the combined pod.

See Versions § 1.10 for each capability's current, correct description.

Corrections made in the process, none of them caused by this cycle's code:

- The local-setup page told you to edit a `docker-compose.override.yml` that has never existed in the backend repository, and quoted configuration values the tracked file does not contain. Both are fixed, and the seeded super-admin address now matches what the code actually uses rather than what the example file says. See Running Locally.
- The packaging page still said a `1.2` package would be refused outright. That is true of a released deployment and no longer true of the development line; the page now says both.
- Around fifteen internal reference pages still pointed at frontend source paths that a library restructure had moved. They were resolved one symbol at a time, because two of the old libraries were split rather than renamed — a blind rename in an earlier pass had already attributed one layer to the wrong library.
- The last three ASCII-art diagrams on the site are now real diagrams.

2026-08-27

A sync covering the backend, web portal, mobile app, Helm chart and the Grafana datasource plugin. Highlights:

- **List and card views now hide a workflow-step-hidden field's value, not just its label** — a field a step marks hidden no longer leaks its value into list, card, badge, kanban or tree reads; the cell renders blank and can't be edited inline there either. See Working With Data [Using OSOS › Create and Edit Records] and, for the design side, Workflows [Build Apps › Workflows & approvals].
- **An ad-hoc workflow step can carry a short description** when you insert one, shown later as a tooltip. See Executing Workflow Actions [Using OSOS › Acting on Workflow Steps].
- **A signed-in user can now discover their own modules and views** without an admin's help — a new endpoint replaces the old admin-gated lookup chain. See Modules [Build Apps].
- **The Grafana datasource plugin (v1.2.0) changed how panels are built** — the query cascade now starts at the Module instead of the Entity, and an ordinary (non-admin) editor's own access now suffices to author a panel. **Breaking:** the `systemTimeField` panel option is gone, replaced by four always-present audit-stamp columns — a saved panel that used it needs re-authoring. See Grafana Dashboards [Integrate & Extend].
- **`altcha` and `shortener` no longer build from backend source** — both now ship as images published by their own repositories, consumed either as a per-cell deployment or a shared cluster-wide instance (that choice itself is unchanged). See Helm Chart.
- **The RabbitMQ vhost-creator step is now self-healing** on a fresh install or a retried upgrade, instead of occasionally getting stuck and needing an operator to intervene. See Helm Chart.
- **Opening a user's or group's full access record is one click away** from the sensitive-field inherited-permissions view, instead of only showing a summary. See Sensitive Permissions [Administration].
- **A granted sensitive field on mobile now unmasks immediately when revealed**, without needing to fully reopen the screen. See Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values].
- Smaller fixes folded in along the way: the field-config panel gained a searchable jump-to-section index (a navigation aid, not a new field option); a view-scoped template export now reaches every non-sensitive field instead of only the triggering view's displayed columns; a tree-view relation sidebar, filter-first panel, and report-preview loading sequence were each fixed; a single resource can again be unchecked alone in the packaging export tree; and contextual help gained its first real write-up on Data Views & Faces [Build Apps].

Two backend changes — a package-schema update for cross-environment map-field remapping, and a migration job that now fails loudly instead of silently — sit past the `1.8.0` release cut with no later milestone published yet, so they are held out of this documentation until a `1.9` line exists to tag them against.

SOT-A (curated site): 26 page pairs (52 files) extended across `admin`, `core`, `dev`, `setup`, and `using`, both locales — no new page, no IA-manifest change. **SOT-B (`ai-generated-docs`):** 54 files across both locales, verified against source with cited paths. **API reference:** `data-service` and `workflow-service` snapshots refreshed against the new backend commit; the generated endpoint reference grew from 375 to 376 operations, and its "hand-maintained" list dropped `shortener-service` (extracted to its own repository, alongside `altcha-service`) in favour of a new "extracted from the backend" note.

2026-08-21

New: OSOS as a Backend, and the package contract — two guide clusters for developers who work against the platform without its UI.

- Development → OSOS as a Backend [Integrate & Extend › OSOS as a Backend] — building an external application on the HTTP API: a 15-minute quickstart, credentials and their lifecycle, tenancy, reading and writing at client scale, files, live subscriptions, long-running work, an error catalogue grouped by originating exception, client generation from spec snapshots, environments, and the security boundary. Alongside it, a **generated endpoint reference** — 375 operations across seven services, regenerated from committed OpenAPI snapshots rather than maintained by hand, with the four spec-less Express services documented manually and declared as such.
- Development → Packaging [Integrate & Extend › Packaging & distribution] — the package artifact as a contract: the ZIP layout and manifests, per-resource payload schemas (Tier 1), the dependency graph, the import API and its state machine, schema versioning, and the three supported ways to produce a package — including the honest statement that a single standalone JSON carrying a whole working system is not something the platform produces.
- The existing Calling the HTTP API [Integrate & Extend] page was corrected in step: the response envelope is **universal** — the old version-conditional split it described was removed from the product, and the generated specs now document the envelope themselves.

A large sync covering the backend, web portal, mobile app, Helm chart and API gateway. Highlights:

- **Sensitive data reveal now works per sign-in session, not per account** — unmasking a value on one device no longer leaves it visible to the same account signed in elsewhere, and the reveal ends the moment that session does. A new guide, Account Security & Sensitive Data Reveal [Using OSOS › Account Security & Revealing Protected Values], covers verification methods and the reveal wizard end to end.
- **Administration now needs a license that includes it** — without it, the Administration menu is not shown; everyday entry work is unaffected. See Your license [Administration › Your license & plan limits].
- **Execute Custom Action was reworked**, including parameter binding and new scope types. Builder-facing level names changed from `Module` / `Internal` to `Local` / `External` — a one-way change on the platform side. See Reactions & Automation [Build Apps › Reactions & automation].
- **Application-user API tokens can now be regenerated from the admin area**, with a grace period so the old token keeps working while the new one rolls out. See Regenerating a token [Integrate & Extend › Use an API token].
- **A Data View's workflow tab can default to Path or History**, and several sub-workflow ownership and timing issues are fixed. See Data Views & Faces [Build Apps].
- **Human verification (Altcha) now also covers the identity pages** — sign-in, MFA and password reset all require a solved challenge. See Human Verification [Administration › Human verification (bot protection)].
- **The Grafana integration gained dashboard-authoring role configuration, chart-rendered routing, and a documented upgrade caveat** — and a correction: the Helm chart page previously said Grafana could not be installed through the chart at all. It now can, with one sharp edge kept prominent: the plugin itself still is not installed automatically. See Grafana Dashboards [Integrate & Extend] and Grafana Datasource Plugin.
- **A response-shape inconsistency in the HTTP API reference is corrected** — the page previously described older API versions as unwrapped where every version is now enveloped the same way. See HTTP API [Integrate & Extend › Calling the HTTP API].

2026-08-19

New: MCP Server for AI Agents — a guide cluster under Development → MCP Server [Integrate & Extend › MCP Server for AI Agents] covering the `osos-mcp` module, which lets an AI agent work with the platform through its HTTP API.

Three things the pages state plainly, because each one changes how you should use it:

- The module is **early access** at version 0.2.0 and is **not suitable for reliable standalone work**. No tool has yet been verified against a live tenant with a real credential.
- Every page states **which `osos-mcp` version it describes**. When the module is updated, that is what tells you whether a page still applies.
- The module **requires a valid license**; it refuses requests otherwise.

2026-08-06

- Added this **General** section as the site's front door, bringing the platform overview, core concepts, version history, this change log and the certification paths together in one place ahead of the audience sections.
- Split the certification program out into its own repository, and introduced a token so links to it resolve correctly wherever it is published.
- Replaced the header's long site title with the brand lockup for the current language — Latin wordmark in English, Arabic wordmark in Arabic — with a small *Docs / التوثيق* label beneath it.

2026-08-05

- Documentation sync against the 1.8 line: backend `alpha.63` – `alpha.67`, frontend `alpha.79` – `alpha.84`, mobile app 1.8.40, chart 1.8.25.
- Documented the reworked **Custom Action** trigger model and the new **Execute Custom Action** automation, entity-designer deep links, quick-edit validation, per-language entry history, and the mobile app's sensitive-field grants, masked inline editing and repeatable reordering.
- Added an **Introduction** section and made the audience map build-aware.

2026-07-30

- Documentation sync against the 1.8 line: backend `alpha.51` – `alpha.62`, frontend `alpha.63` – `alpha.78`, mobile app 1.8.38, chart 1.8.24.
- Documented sensitive fields and step-up unmask, the Sensitive Permissions administration page, External Step, unowned group steps and Take Ownership, the sign-in human-verification step, and repeatable reordering.

2026-07-24

- Published the site to its hosted home, with a build that verifies internal links and assets before deploying.
- Added a flag that marks pages and whole sections as internal, so a public build physically excludes them.
- Merged the remaining loose material into the curated site and added the local-development appendix.

2026-07-23

- Restructured the site end to end: restored the section landing pages, removed duplicate stubs, and rebuilt the sidebar so groups are explicit, ordered and bilingual.
- Replaced ASCII diagrams with real rendered diagrams, and ported the diagram set from the previous documentation into both languages.
- Improved the Arabic throughout against a glossary built from human translations, rather than leaving machine phrasing in place.
- Reviewed the previous documentation page by page and folded what was missing into the current site.

- Rewrote the frequently-asked questions as classified, searchable entries.

Certification paths

This page is for anyone who would rather learn OSOS as a course than look things up. It shows which certification fits your role and where to start.

Alongside this documentation there is a full **training and certification program** for OSOS: seven certifications across three career paths and four levels — 47 training days, 188 contact hours, in English and Arabic.

The documentation is a reference: you come to it with a question. The certification program is a curriculum: it takes you from nothing to competent in a defined order, with labs and an assessment at the end.

The three paths

Path	For	Ends at
Analyst	Turning a business need into a buildable specification.	Certified Expert
Developer	Building the data, screens and automation.	Certified Expert
Technician	Administering an organization on the platform, then deploying, operating and scaling the platform itself.	Certified DevOps Engineer

All three converge at the top, in the Master certification.

The seven certifications

Code	Certification	Path	Level	Hours
ACP [OSOS Training and Certification › ACP — Certified Practitioner]	Certified Practitioner	Foundation — everyone	1 · Practitioner	8
ACA [OSOS Training and Certification › ACA — Certified Analyst]	Certified Analyst	Analyst	2 · Associate	16
ACD [OSOS Training and Certification › ACD — Certified Developer]	Certified Developer	Developer	2 · Associate	16
ACSA [OSOS Training and Certification › ACSA — Certified System Admin]	Certified System Admin	Technician	2 · Associate	16
ACE [OSOS Training and Certification › ACE — Certified Expert]	Certified Expert	Analyst + Developer	3 · Expert	28
ACDE [OSOS Training and Certification › ACDE — Certified DevOps Engineer]	Certified DevOps Engineer	Technician	3 · Expert	20
ACM [OSOS Training and Certification › ACM — Certified Master]	Certified Master	All converge	4 · Master	84

Which certification fits your role

The documentation is organised by what you do. This is how those roles map to the courses:

If you…	Documentation	Start with	Then
Use apps in the portal	Using OSOS [Using OSOS › Start Here]	**ACP**	—
Turn business needs into specifications for others to build	Build apps [Build Apps › Build apps — start here]	**ACP**	**ACA**, then **ACE**

If you…	Documentation	Start with	Then
Build applications without code	Build apps [Build Apps › Build apps — start here]	**ACP**	**ACD**, then **ACE**
Connect other systems through the API, webhooks or functions	Integrate & extend [Integrate & Extend › Integrate & extend — start here]	**ACP**, **ACD**	**ACE**, which covers the API, integrations and functions
Administer your organization: users, access, sign-in, look and feel	Administration [Administration › Start here: Administration]	**ACP**	**ACSA**
Install, upgrade and operate the platform	Setup & Maintenance	**ACP**, **ACSA**	**ACDE**

ACSA — Certified System Admin is the course for tenant administrators: it covers the **Administration** area of a running organization, not servers. It sits on the Technician path because it is the prerequisite for **ACDE**, the course that deploys and operates the platform.

How progression works

ACP is the universal gate — everyone starts there, whichever path they are headed down. From it the three Level-2 certifications branch by path. **ACE** requires ACA or ACD; **ACDE** requires ACSA. **ACM** requires both ACE and ACDE, because a master is expected to hold the solution path and the technical path at once.

Each level goes deeper rather than wider: a concept met at Level 1 returns at Level 3 with the internals attached. No course re-teaches what an earlier one already covered at the same depth.

Where to start

- Never used the platform → **ACP — Certified Practitioner** [OSOS Training and Certification], two days, no prerequisites.
- Already using it and want the whole picture → the program blueprint [OSOS Training and Certification › OSOS Training & Certification Program].
- Choosing between paths → the program overview [OSOS Training and Certification] lays the three paths side by side.

Every course is grounded in this documentation: nothing is taught that the product does not do.