Operations Manual

Running events on OpsCore Eventz

Version 1.0 · Effective 29 August 2026

How the platform is organized

Everything belongs to an organization. Users, events, registrations, attendees, invoices and audit records are all scoped to one, and nothing crosses that boundary. A request for a record belonging to another organization is answered as though it does not exist, so a wrong link or a guessed identifier reveals nothing.

Inside an organization, work is divided by department and by event. A user can be given a role across the whole organization, or scoped to a single department or event — which is how you give a campus coordinator control of their own event without giving them the finance ledger.

Roles and permissions

The platform ships twelve built-in roles carrying seventy-one permissions across fourteen categories: Events, Registrations, Attendees, Schedule, Partners, Check-In, Certificates, Communications, Surveys, Finance, Reports, Integrations, Administration and Security.

Permissions are checked on the server for every action. The interface hides controls a user cannot use, but hiding is a courtesy, not the security boundary — a user who constructs the request by hand is still refused.

You can create custom roles when the built-in ones do not fit. Built-in roles cannot be edited, so a custom role is the supported way to vary access.

- Organization Administrator — full control including users, roles, finance configuration and integrations.
- Event Administrator — creates and runs events end to end, including schedule, partners, certificates and communications.
- Registration Manager — registrations, approvals, waitlists, transfers, imports and exports.
- Finance Administrator — invoices, payments, refunds, reconciliation and financial exports.
- Check-In / Event Staff — the check-in station and attendance, without access to finance or settings.
- Report Viewer / Auditor — read-only across reporting and the audit log.
- Speaker, Sponsor, Exhibitor, Vendor, Attendee — portal roles, each seeing only their own records.

Creating and publishing an event

The event wizard takes five steps: basics, date and time, location, registration window and capacity, then ownership. An event is created as a draft and nothing is public until you publish it.

Before an event can be published the platform checks it is actually ready — that it has at least one active registration type, that each type has a registration form attached, and that the dates make sense. Anything missing is listed on the overview tab as a publish blocker, with a link to the screen that fixes it.

Publishing opens registration and makes the public event page live at its own address. You can change every detail afterwards; changes to a live event are recorded in the event's history with who made them.

Registration forms and conditional logic

Each registration type carries its own form. The form builder offers the full field set — short and long text, email, telephone, number, date, time, dropdown, multi-select, radio, checkbox, yes/no, address, organization, job title, file upload, signature, consent, and headings.

Conditional rules run as IF/THEN: when an answer matches, the rule can show or hide a field, make it required or optional, add a fee, or assign a registration type. The same rule engine runs in the browser as the registrant types and again on the server when they submit, so a rule cannot be bypassed by editing the page.

A field that has already been answered by a registration is deactivated rather than deleted when you remove it, so historic answers keep their labels.

Taking registrations

Registrants complete the public form without needing an account, if the organization allows it. Capacity and waitlist policy are applied at the moment of submission, pricing and discounts are calculated on the server, and the confirmation number is issued once the record is committed.

Registrations requiring approval sit in Pending Review until an administrator approves or denies them, with the reason recorded either way. Waitlisted registrations hold a position in the queue and can be promoted individually or in bulk.

Every modification — approval, denial, transfer, cancellation, promotion — is written to the registration's change history with the actor and the reason.

Check-in and attendance

The check-in station is built for a phone or tablet at a door. Staff can search by name, scan the QR code on a confirmation, or type a confirmation number.

Check-in is recorded at the event level and, where sessions require it, per session. Attendance records carry the status, minutes attended and credit hours, which is what certificate eligibility is later assessed against.

Attendance can be corrected afterwards by staff with the right permission; corrections are audited.

Certificates

Certificate templates are designed once and reused. Eligibility can be set on attendance, on specific sessions, on total credit hours, on survey completion, or granted manually.

Issued certificates carry a unique certificate number that anyone can check on the public verification page without signing in. Verification confirms the certificate and shows whether it is still valid.

A certificate with an error is corrected by reissuing: a new certificate is issued with a new number and the original is marked superseded. Both stay on file, so the correction history is intact. A certificate issued in error is revoked, and the reason is shown to anyone who verifies its number.

Finance

Registration fees, add-ons, discount codes and tax settings are configured per event. Discount codes are validated and the registration repriced on the server at submission, so a code cannot be forced by editing the page.

Invoices and receipts are generated as PDFs carrying your organization's branding. Payments can be recorded manually, against an invoice, or marked as no-charge — which means the full financial workflow can be demonstrated and operated without a payment processor connected.

Card data is never stored. Where a processor is connected, the platform keeps the card brand, the last four digits and the processor's reference, and nothing else.

Financial exports require the finance export permission and every export is written to the audit log with who ran it and what it covered.

Communications

Messages are built from templates with merge fields, previewed and test-sent before they go out, and targeted at an audience defined by event, registration status or registration type.

Automations run as WHEN/IF/THEN — when a registration is confirmed, if it is of a given type, then send a given template.

Every message is recorded in the notification log whether or not it was delivered. When no email provider is configured the platform records the message and marks it as skipped because no provider is connected. It does not report a message as sent when nothing was sent.

Reporting and exports

The standard report library covers registration, attendance, financial, event performance, engagement and partner reporting. The custom report builder works from a dataset, then fields, filters, grouping and sorting, and a saved report can be re-run or exported.

Exports are available as CSV, Excel-compatible CSV, JSON and PDF, and are audited.

Report schedules can be saved and show their next run time, but they are not executed by a background worker in this build. Scheduled reports must be run manually until that worker exists; the screen says so rather than implying delivery.

Integrations

Payment processing, finance system synchronisation, email, SMS and single sign-on are all adapter-based and each requires credentials you supply.

Until credentials are entered an integration shows as requiring credentials and does not operate. Once saved, credentials cannot be read back — an administrator sees a masked hint, and can replace, rotate or revoke them. Each of those actions is audited.

The finance system integration also supports a mapped manual export, so an organization can meet its finance obligations before any live connection is approved.

The API

An organization can issue API keys for its own systems. A key is shown once at creation and only its hash is stored; it carries an explicit set of scopes and an optional expiry, and can be revoked at any time.

The API is scoped to the issuing organization, rate limited per key, and refuses any call outside the key's granted scopes. Registrations created through the API run the same validation, capacity, pricing and automation path as the public form.

Security and audit

Sign-in attempts are recorded. Accounts lock for fifteen minutes after five consecutive failures. Multi-factor authentication is available and can be required, and an administrator can end every active session for a user at once.

Privileged actions are written to an append-only audit log with the actor, the entity, a summary and the before and after values. The audit log is searchable and exportable by users with the audit permission.

Security incidents are recorded with detection time, severity, affected data, containment and resolution, so notification obligations can be assessed against a real record.

Retention and privacy requests

Retention periods are configured per record type and shown to administrators as your organization's stated policy.

They are applied by administrators rather than swept automatically: this build does not delete records on a schedule of its own. Plan for that when you write your own retention procedure, and record each deletion through the platform so it lands in the audit log.

Data subject requests — access, correction, deletion — are logged with their status and outcome so the handling is auditable.