# FLUIDCLOUD

## User Manual

The website · buying online · videos & manuals
the service desk · partners & suppliers · projects · the Big 5 · going live

SceptreTechScience Pty Ltd
FLUIDERP 77.1 · go-live edition · 16 October 2026
Companion to the film FLUIDCLOUD_Demo.mp4

# Contents


**1 About this manual** 2

Conventions 2

A note on the examples 2

**2 How FLUIDCLOUD works** 3

Capture once, post everywhere 3

**3 Getting started** 4

The address and the three sides 4

Your first sign-in 4

The launcher 5

Anatomy of a screen 5

**4 Registering from the website** 7

**5 Watching the videos and reading the manuals** 9

**6 Buying a service online** 10

**7 Online orders and invoicing (staff)** 13

**8 Requests and the service desk** 16

**9 Partners and suppliers** 20

**10 Projects** 22

**11 Messages between the sides** 25

**12 Downloads and the app** 26

**13 Publishing videos and manuals (staff)** 28

**14 Reports and the numbers** 29

**15 Going live (administrator)** 30

Before you start 30

Steps 30

Website links 31

Security notes 31

**16 Messages, refusals and what to do** 32

**17 Data reference and glossary** 33

**Appendix A Walk it once on the demonstration package** 34

**Appendix B Administrator notes** 35

# 1 About this manual

FLUIDCLOUD is SceptreTechScience's platform for selling, delivering and supporting FLUIDERP — and it runs **on** FLUIDERP. Prospects and customers reach it from the website; partners and suppliers have their own door; SceptreTechScience's staff run the Big 5, projects and the service desk behind it. This manual describes the go-live edition (FLUIDERP 77.1, 16 October 2026) screen by screen, in the browser client (`erp/erp-web`) that the website opens. The Android and Windows apps show the same screens, from the same definitions.

| If you are… | Read |
|---|---|
| a prospect or a client | 3 Getting started · 4 Registering · 5 Videos & manuals · 6 Buying online · 8 Requests · 10 Projects (milestones) · 11 Messages · 12 Downloads |
| a partner or a supplier | 3 · 9 Partners & suppliers · 8 Requests · 11 Messages |
| support / implementation staff | 3 · 4 · 8 · 9 · 10 · 11 · 13 Publishing the library |
| finance staff | 3 · 7 Online orders & invoicing · 14 Reports · 16 Refusals |
| the platform administrator | everything, especially 2 · 15 Going live · 17 Data reference · Appendix B |

## Conventions

- **Bold** names a button, a menu tile or a field as it appears on screen; `code` names a table (collection) or a definition.
- “Staff side”, “Clients side” and “Partners & Suppliers side” are the three sign-in sides — three clusters of FLUIDCLOUD (chapter 2).
- Every procedure gives **Who** may do it, **Where** it is, the **Table** it writes, the steps, **what the button writes**, how to **check** it and what to do **if the system refuses**.

## A note on the examples

The screenshots and examples use the demonstration package shipped with FLUIDCLOUD: Jacaranda Lodge (Maria Chen), St Brigid's College (David Okonkwo), Coastal ERP (Sam Whitaker) and SceptreTechScience's staff. Names, amounts and dates are illustrative. A few “after the button” screens (the invoice an online order becomes, a new registration) were drawn from the posting definitions to show exactly what the system writes; Appendix A walks the same story on a demo install. A live install (chapter 15) contains none of this data.

# 2 How FLUIDCLOUD works

FLUIDCLOUD is three FLUIDERP clusters on one gateway. Each cluster has its own logins, its own data and its own colours. Nothing is shared: records cross from one cluster to another only through **bridges** — configuration rows (`erpcfg.tblbridge`) that say which records, which fields and in which direction. A client never reads the staff database; the staff side receives exactly the fields a client sent.

**Staff**
`cloudapi`
SceptreTechScience
Big 5 · systems · releases · service desk · projects · shop back-office · library

**Clients**
`cloudclient`
prospects & customers
registration · price list · basket & card · videos & manuals · requests · downloads · projects

**Partners & Suppliers**
`cloudpartner`
logins issued by staff
referrals · work orders · requests · library · messages

## Capture once, post everywhere

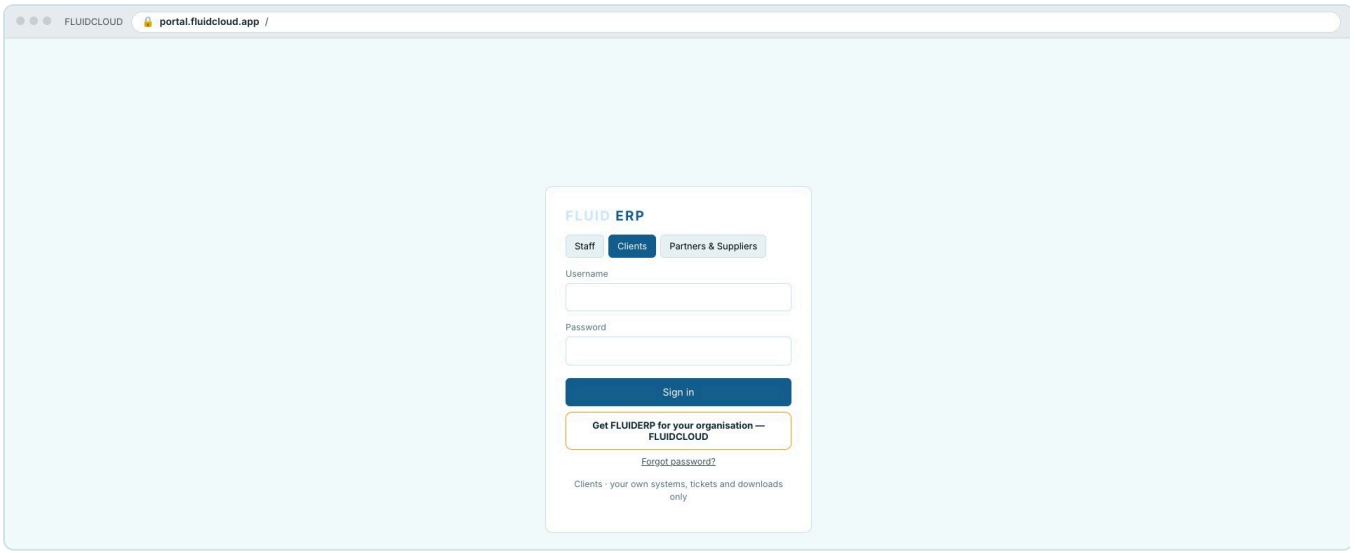
Every piece of information is typed once, where it starts — a registration by the prospect, an order by the buyer, a request by whoever asks — and travels by bridge to the side that acts on it. Buttons on the staff side are **posting definitions**: one press writes every row the business event needs (an order becomes the core sales invoice, its lines and the card receipt). Nothing is re-keyed, and every number in a report is read from the posted rows.

| Rule | What it means for you |
|---|---|
| 1 One record of truth | The staff cluster owns the customers, invoices, projects and releases. The other sides see mirrors that update within the minute. |
| 2 Only what you may see travels | A client sees only rows addressed to their own login; a partner only rows for their own login; staff-only fields never leave. |
| 3 The answer goes back where the question came from | A request's status and response are written back onto the requester's own row on their side. |
| 4 Nothing is published before it is clean | Uploads and posts pass the clean gate (AI or a reviewer) before anyone else can see them. |
| 5 The ledger moves only by posting | Invoices, receipts and journals are written by Post buttons from the core Big 5 definitions, never typed into the ledger. |

# 3 Getting started

## The address and the three sides

Everyone uses one address — the FLUIDCLOUD portal, linked from sceptretechscience.com. The sign-in page shows the three sides as a toggle; choose yours before typing your username. The website's "Get FLUIDERP" button can open the registration form directly: `https://<portal>/apply/register?side=cloudclient`.



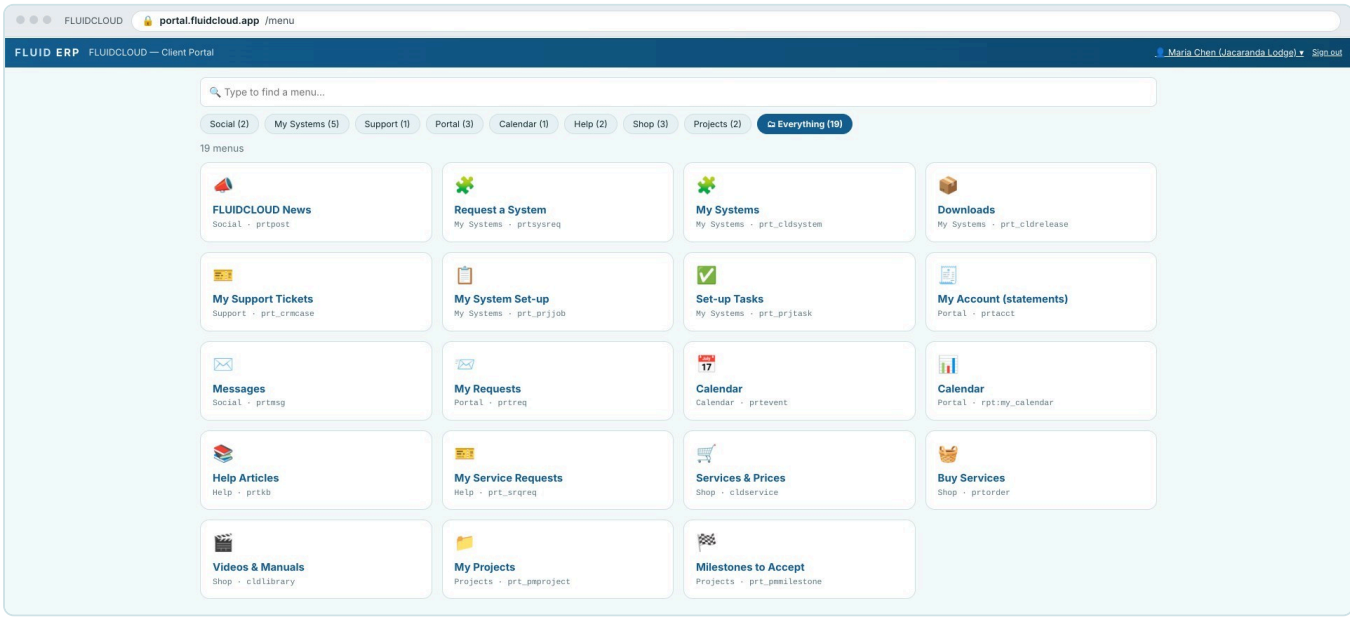

*The sign-in page on the Clients side. Register sits under Sign in; Forgot password? e-mails a code to the address on your login.*

| Side | Who signs in | Username |
|---|---|---|
| Staff | SceptreTechScience staff | your staff login (e.g. jade.support) |
| Clients | registrants (prospects) and clients | your e-mail address |
| Partners & Suppliers | partners and suppliers | the e-mail address SceptreTechScience registered |

## Your first sign-in

On a live install every login starts with a random password that the administrator hands you once. Sign in, open your name (top right) → 🔑 **Change Password**, and choose your own. A registrant enters the 6-digit code sent to their e-mail the first time they sign in.

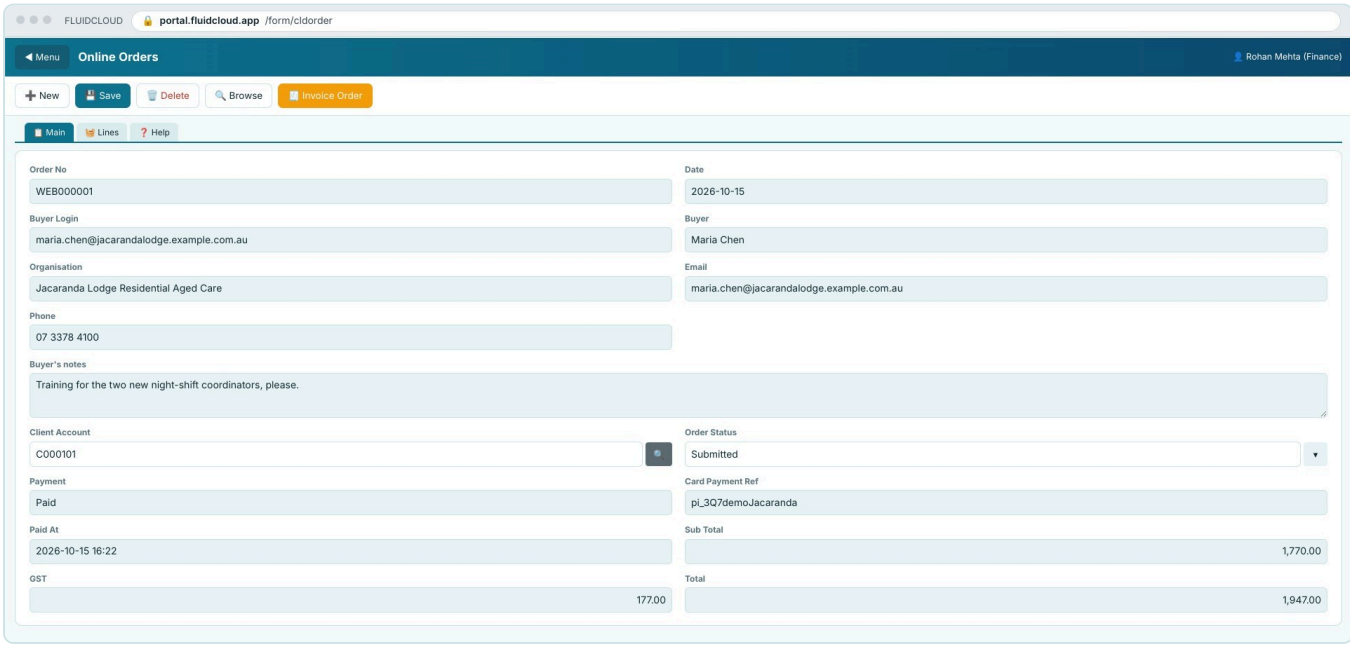
## The launcher




*A client's launcher. Chips filter by module; the search box finds a menu by name; each tile shows its module and id.*

The launcher shows only the menus your group may open. Type in 🔍 **Type to find a menu...** to filter; the chips (Shop, Projects, Help...) show one module. Reports open as reports (📊), forms as forms.

## Anatomy of a screen




*A form on the staff side: the bar (◀ Menu, title, you), the toolbar, the tabs (📋 Main, line tabs, ❓ Help) and the fields.*

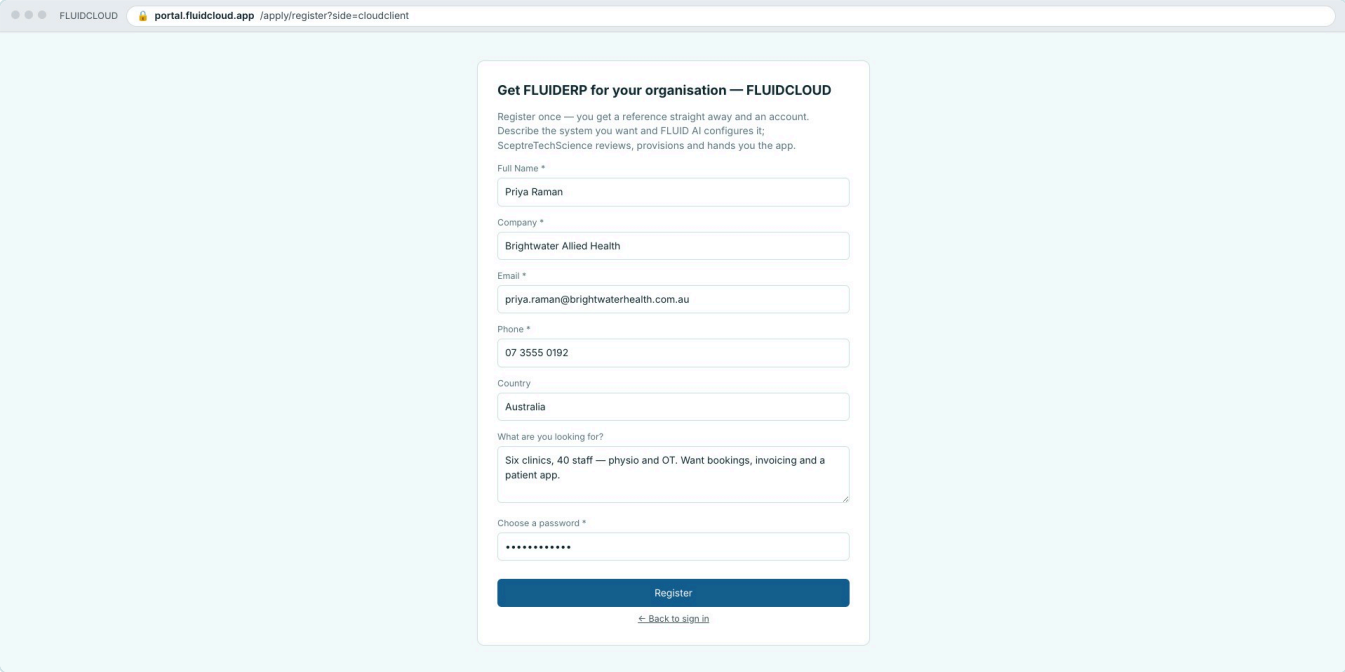
| Button | What it does |
|---|---|
| ➕ New | a blank record |
| 💾 Save | saves the record and all its line tabs together |
| 🗑️ Delete | deletes the record (when your group may) |
| 🔍 Browse | the list of records — search, open a row, 📄 Export CSV |
| coloured buttons (🧾, 🎫, Post…) | posting definitions: they write the rows of a business event (each is described in this manual) |
| 💳 Pay by Card | only on the basket — the card payment for that order |
| ❓ Help tab | the form's own help: every field, every button |

A read-only **enquiry** (🔎 List) is a mirror of staff data: search it and open a row; you cannot change it. Grey fields are calculated or written by the system.

# 4 Registering from the website

## Register as a prospect

Who anyone  Where sign-in page → Clients → **Get FLUIDERP for your organisation**  Table `prtreg → (bridge) portalreg`



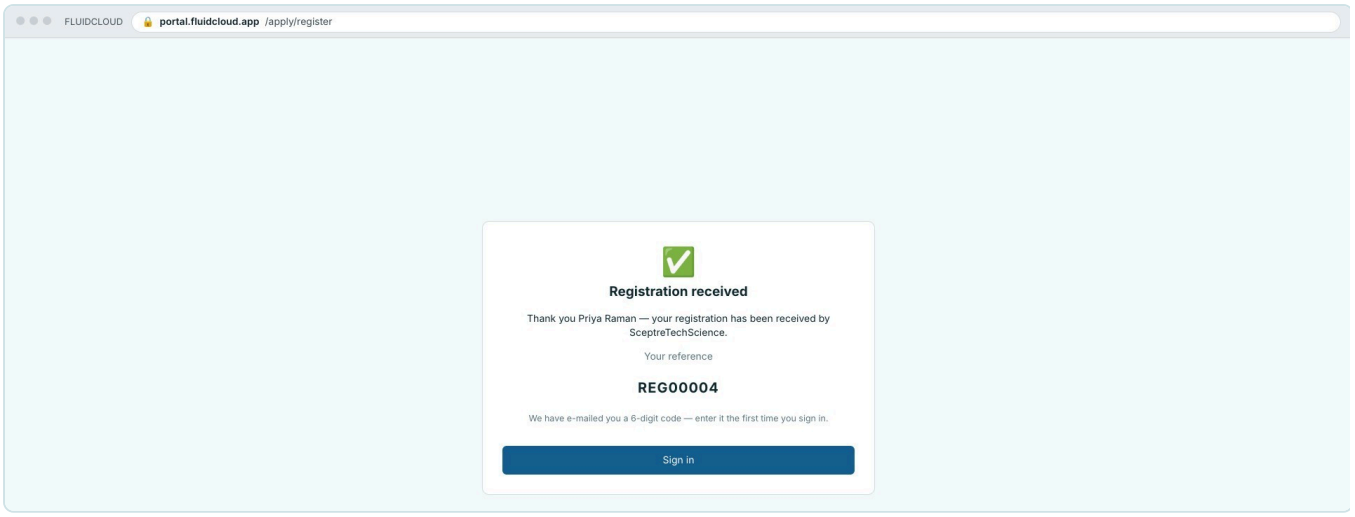

*The registration form. Its fields, required fields and limits are configuration (erpcfg.tblpublic).*

1. Open the portal from the website; the Clients side is selected.
2. Press **Get FLUIDERP for your organisation — FLUIDCLOUD**.
3. Type your name, organisation, e-mail, phone, country, a few words about what you need, and a password. Fields marked * are required.
4. Press **Register**. Note the reference (REG#####) shown at once.
5. Open the e-mail from FLUIDCLOUD and, the first time you sign in, enter the 6-digit code.

| **What Register writes** posting `tblpublic register` | |
|---|---|
| `prtreg` | REG##### · status Submitted · channel Client Portal |
| `tblusr` | a login = your e-mail, group **registrant**, password hashed |
| `portalreg (staff)` | the same registration — never the password or the code |

**Check it.** The success page shows your reference; 🔎 Check my status (on the form) shows it at any time with your e-mail.

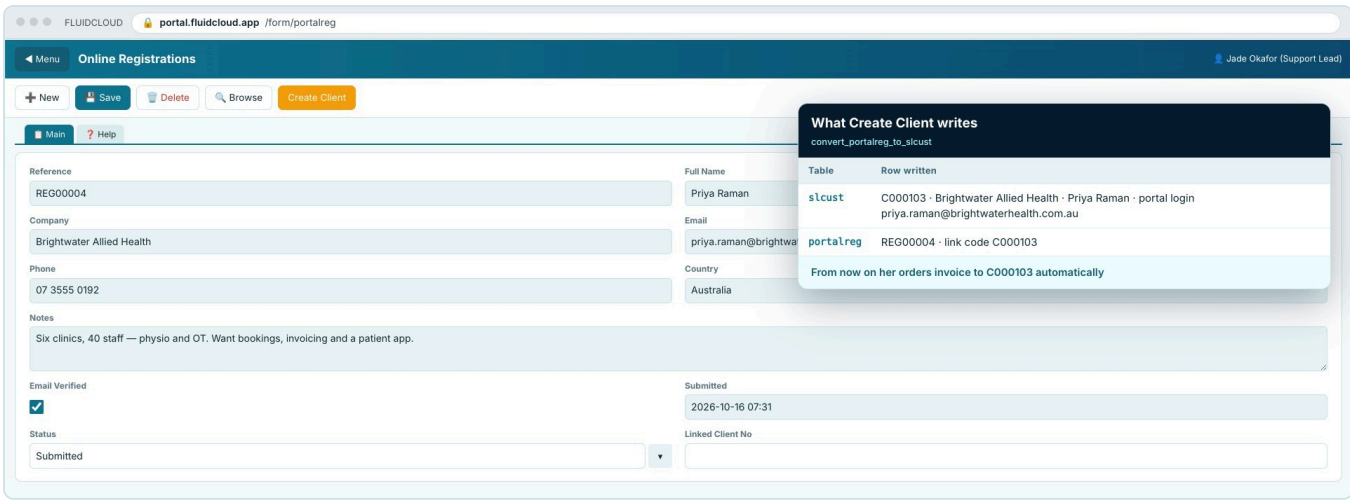
**If the system refuses.** “Too many requests” — the form accepts 30 registrations an hour from the internet; try again later. A missing required field is named on the form.




*The success page: the reference at once, and the code by e-mail.*

## Turn a registration into a client account

| Who | Where | Table |
|---|---|---|
| support, implementation, cloud_admin | Staff → 🪪 **Online Registrations** | `portalreg` → `slcust` |




*Create Client: the registration becomes the core customer record, linked to the person's login.*

1. Open the registration (REG#####). Check the e-mail is verified and the organisation is real.
2. Press **Create Client**.
3. Open 👤 **Customers** (slcust) to complete the address, terms and credit limit.

| **What Create Client writes** posting `convert_portalreg_to_slcust` | |
|---|---|
| `slcust` | C###### · the organisation · contact · e-mail · **portal login = the registrant's e-mail** |
| `portalreg` | link code = the new client number |

**Check it.** The registration's link code shows the client number; the person's online orders now invoice to that client automatically.

# 5 Watching the videos and reading the manuals

## Find and play a demo video or manual

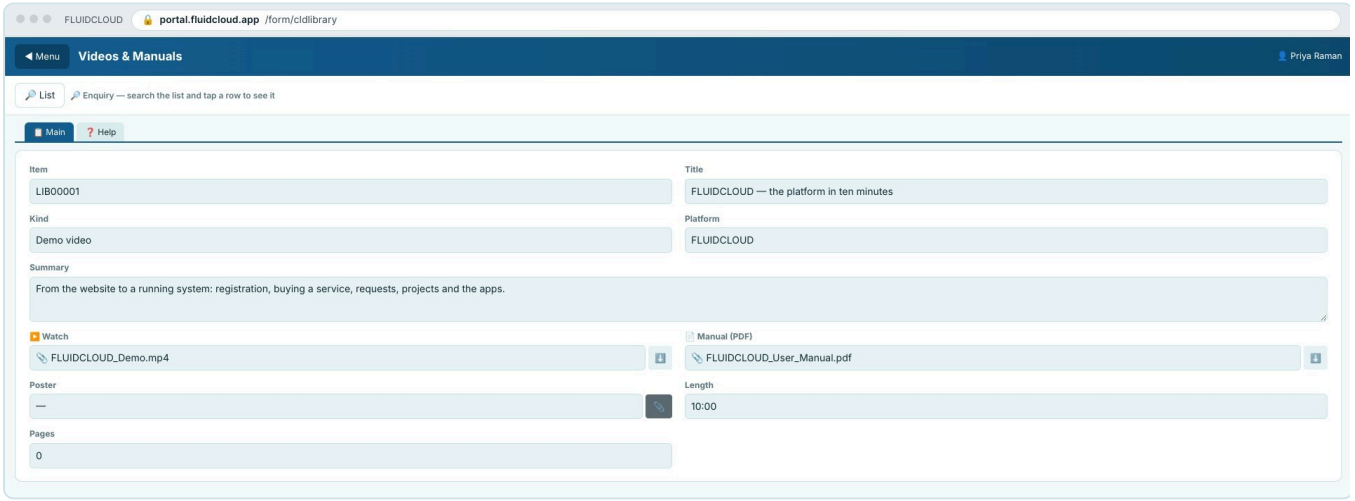
Who registrants, clients, partners, suppliers  Where 🎬 **Videos & Manuals**  Table `cldlibrary (mirror)`




*A prospect's library: the platform film, its manual and the vertical demos — not the partners' implementation guide.*

1. Open 🎬 **Videos & Manuals** — you see only the items published for your group.
2. Open an item.
3. Press ▶️ **Watch** to play the video; it streams at the best quality your connection allows and remembers where you stopped.
4. Press 📄 **Manual (PDF)** to open the manual; print it or save it as PDF from the document viewer.

**Check it.** Prospects see items for “Prospects & clients” and “Everyone”; clients also see “Clients only”; partners and suppliers see “Partners & suppliers” and “Everyone”.

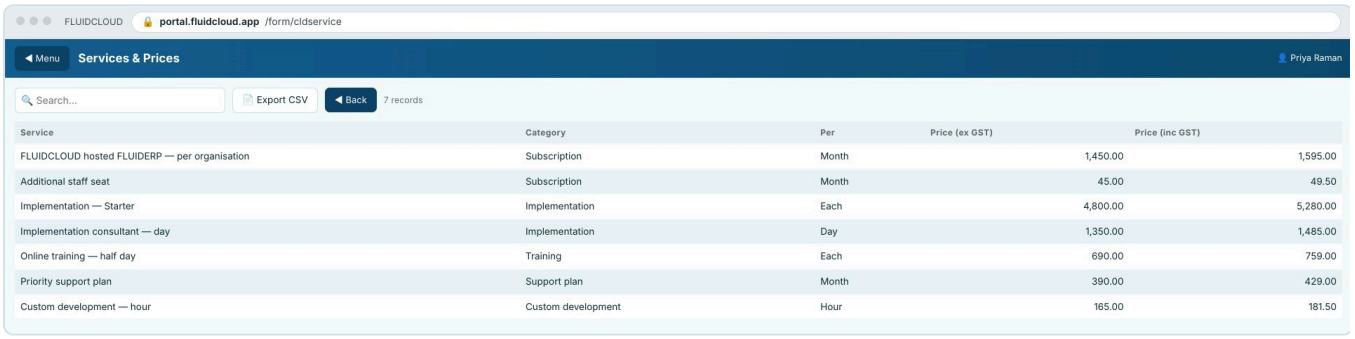



*An item: the video, the PDF and the poster, with its length and pages.*

# 6 Buying a service online

## See what can be bought, at today's prices

Who registrants and clients  Where 🛒 **Services & Prices**  Table `cldservice (mirror)`




*The price list, read from the staff catalogue. A service hidden or retired by staff disappears here at once.*

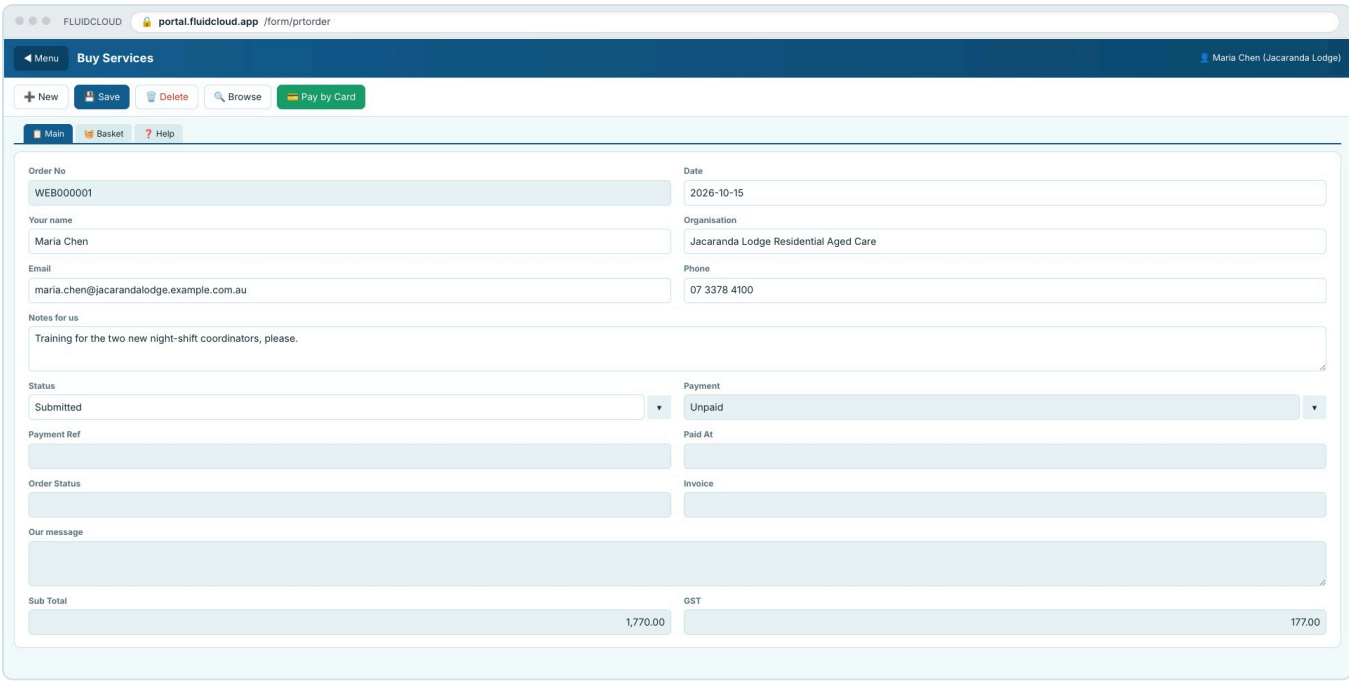
1. Open 🛒 **Services & Prices**.
2. Prices are shown ex GST and inc GST, per unit (each, hour, day, month...).
3. A linked demo video or manual is named on the service — find it in 🎬 Videos & Manuals.

| Service code | Service | Per | Ex GST | Inc GST |
|---|---|---|---|---|
| SUB-CLOUD | FLUIDCLOUD hosted FLUIDERP — per organisation | Month | 1,450.00 | 1,595.00 |
| SUB-SEAT | Additional staff seat | Month | 45.00 | 49.50 |
| IMP-STARTER | Implementation — Starter | Each | 4,800.00 | 5,280.00 |
| IMP-DAY | Implementation consultant — day | Day | 1,350.00 | 1,485.00 |
| TRN-ONLINE | Online training — half day | Each | 690.00 | 759.00 |
| SUP-PRIORITY | Priority support plan | Month | 390.00 | 429.00 |
| DEV-HOUR | Custom development — hour | Hour | 165.00 | 181.50 |

The starter catalogue shipped with FLUIDCLOUD (illustrative prices — the administrator sets the real ones before the shop opens, chapter 15).

## Order and pay by card

Who registrants and clients  Where 🧺 **Buy Services**  Table `prtorder + prtorderlines → (bridge) cldorder`



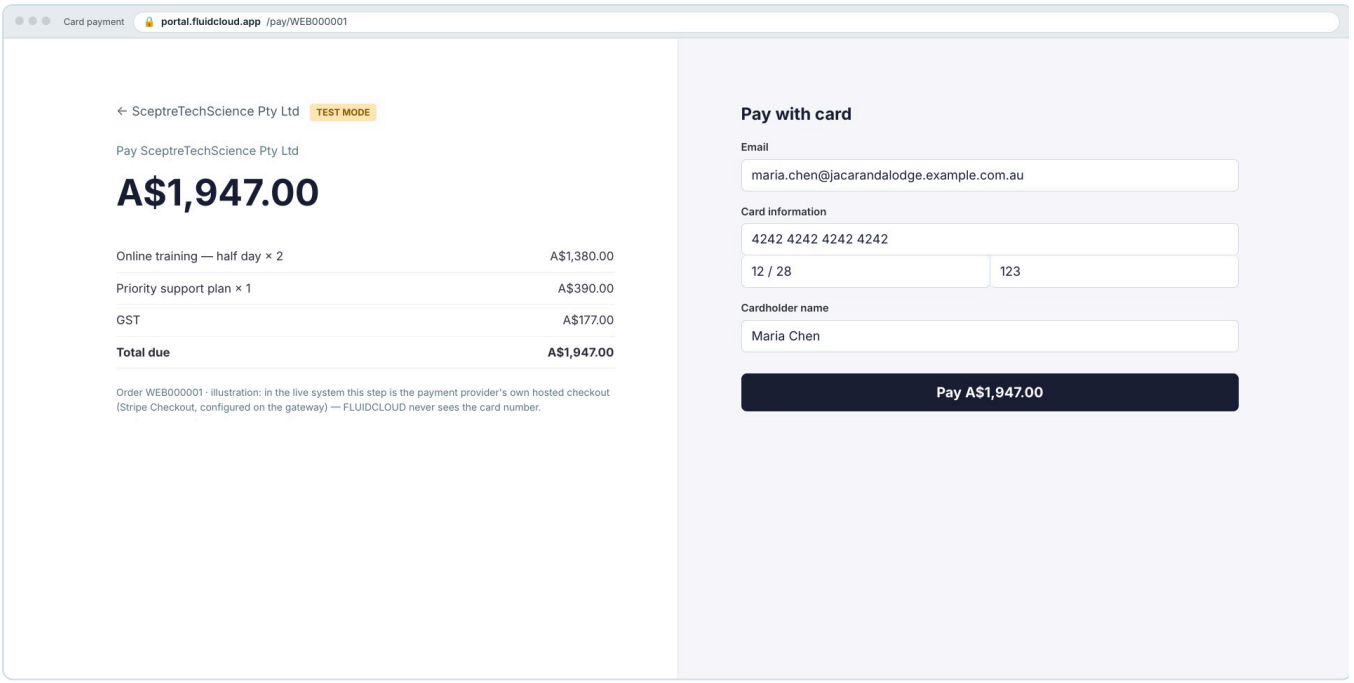

*The order: two training sessions and the priority support plan — A$1,770.00 + GST A$177.00 = A$1,947.00.*

1. Open 🧺 **Buy Services** and press ➕ **New**.
2. On the 🧺 **Basket** tab press ➕ **Add New**, pick a service with 🔍 and type the quantity. Description, price and GST are filled from the price list.
3. Back on 📋 **Main**, check your name and organisation, add notes, set **Status = Submitted** and press 💾 **Save**.
4. Press 💳 **Pay by Card**. The provider's secure checkout opens with the order's total; complete the card payment.
5. Return to the portal and reopen the order: **Payment = Paid** with the payment reference.
6. Prefer an invoice? Submit without paying — SceptreTechScience invoices you with 14 days to pay.

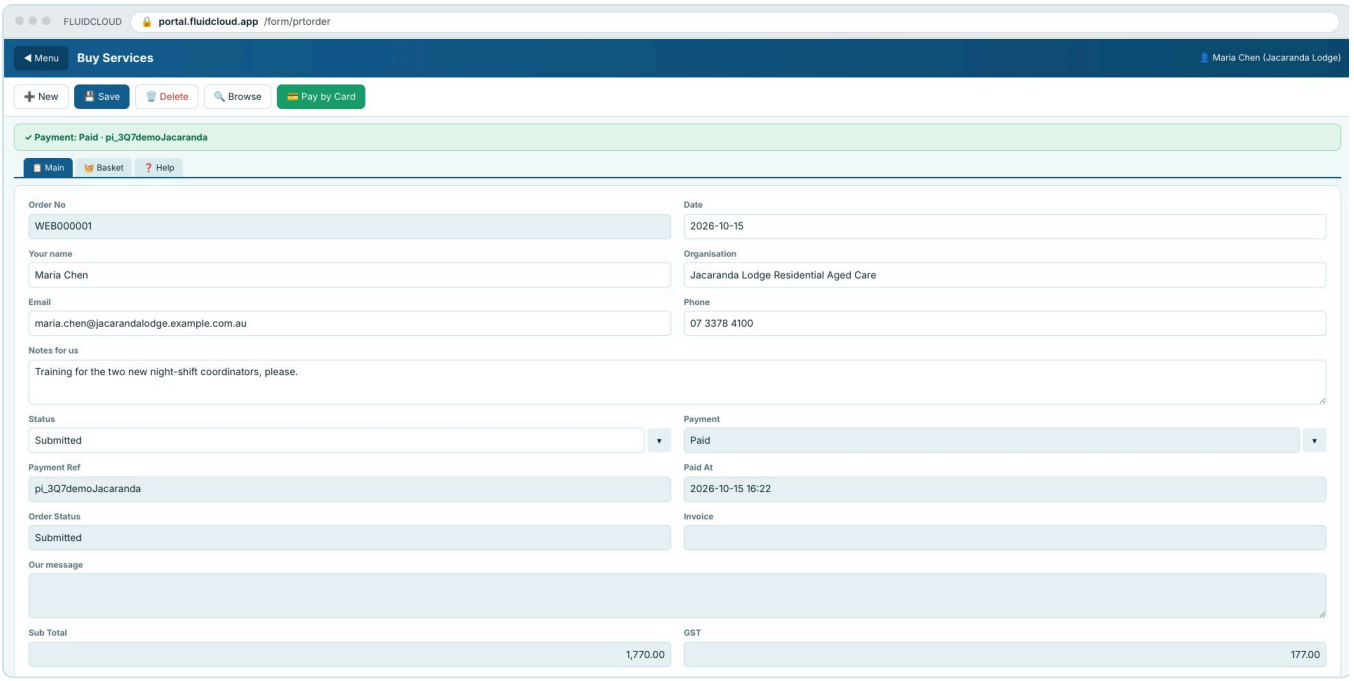
| **What Submitted (Save) writes** | posting `bridge prtorder_to_staff` |
|---|---|
| `cldorder (staff)` | the order with its WEB number, your details, totals; Status Submitted |
| `cldorderlines (staff)` | the lines, prices frozen as you saw them |

**Check it.** Order Status, Invoice and Our message on the order are written by SceptreTechScience as they deliver.

**If the system refuses.** 💳 Pay by Card refuses while the order is not Submitted. A card declined at the provider leaves Payment = Unpaid; try again from the order.




*The card step. In the live system this is the payment provider's own hosted checkout (Stripe Checkout); FLUIDCLOUD never sees the card number. (Test mode shown.)*

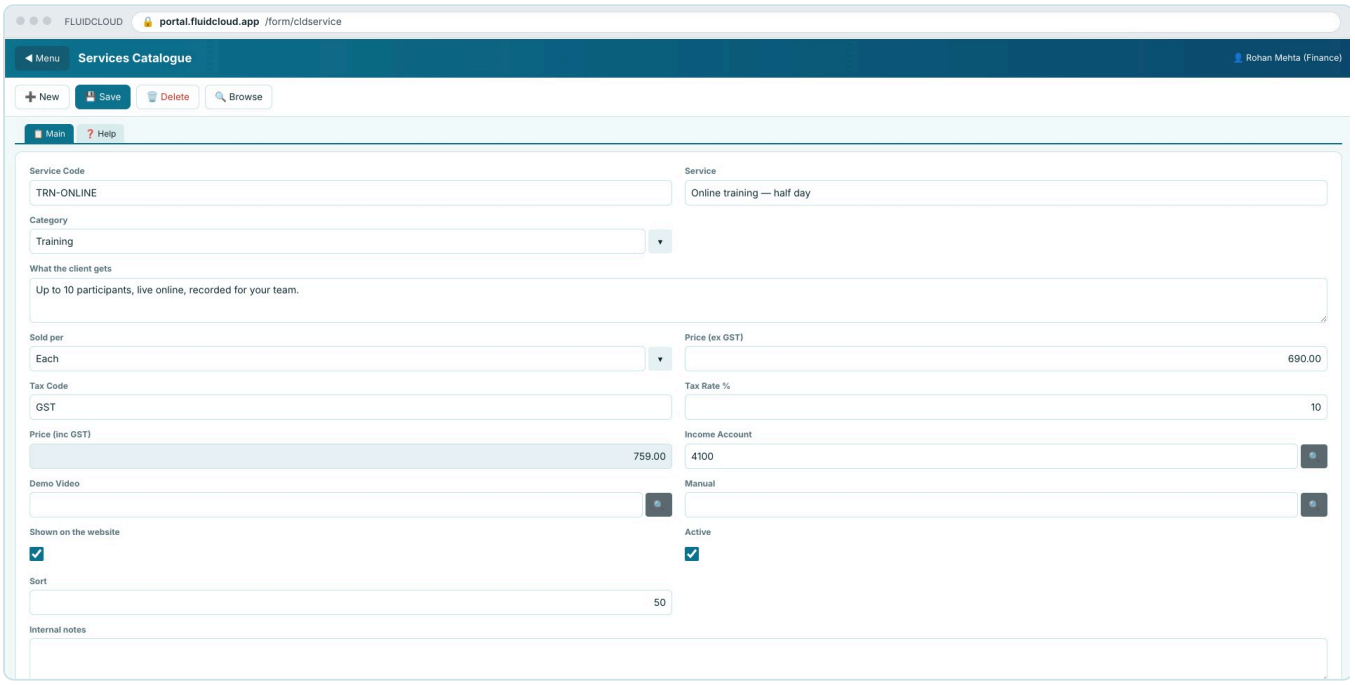



*Back on the order: Payment = Paid, with the provider's reference.*

# 7 Online orders and invoicing (staff)

## Keep the services catalogue

Who finance, implementation, cloud_admin  Where Staff → 🛒 **Services Catalogue**  Table `cldservice`



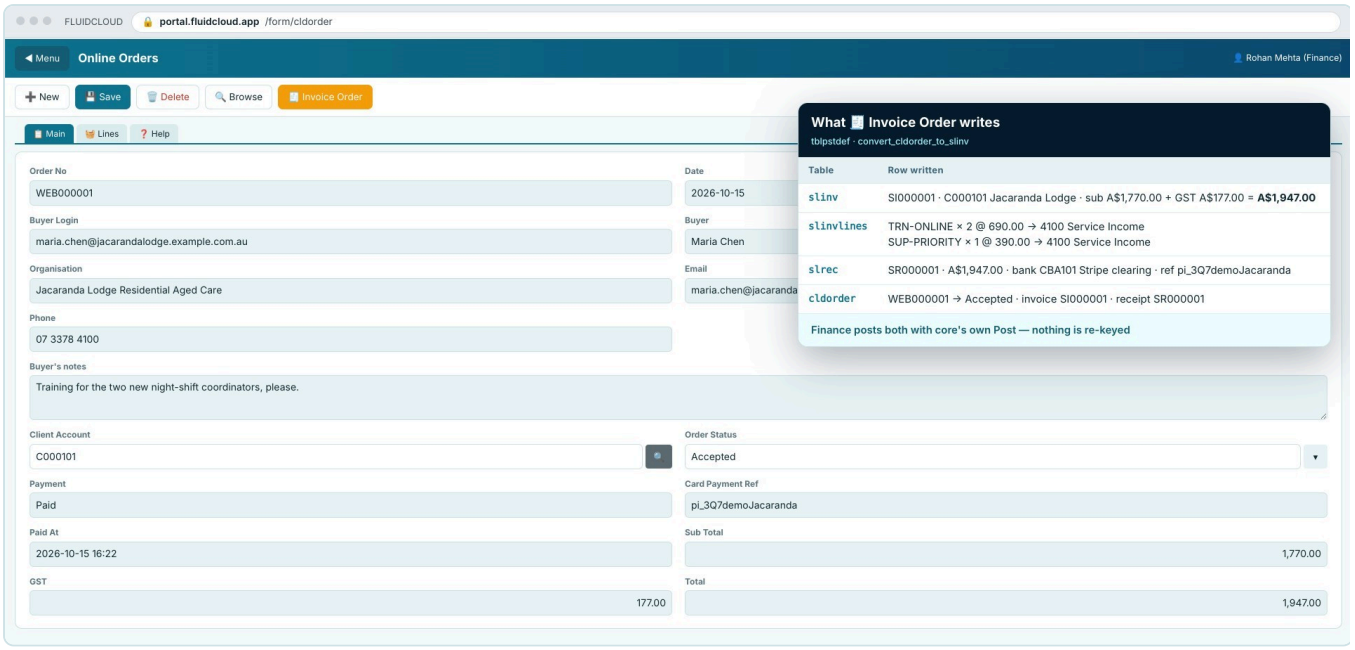

*A service in the catalogue.*

1. Open a service or press ➕ New: code, name, category, what the client gets, sold per, **price ex GST**, tax code and rate, income account (4100 by default).
2. Link a demo video and a manual from 🎬 Videos & Manuals.
3. **Active** and **Shown on the website** both on = listed on the portal. Turn either off to take it off sale.
4. 💾 Save — the portal's price list changes within the minute.

**Check it.** Report 📊 **Services Price List** shows the catalogue as the website shows it.

## Invoice an online order

Who finance, support, cloud_admin  Where Staff → 🛒 **Online Orders**  Table `cldorder → slinv · slinvlines · slrec`



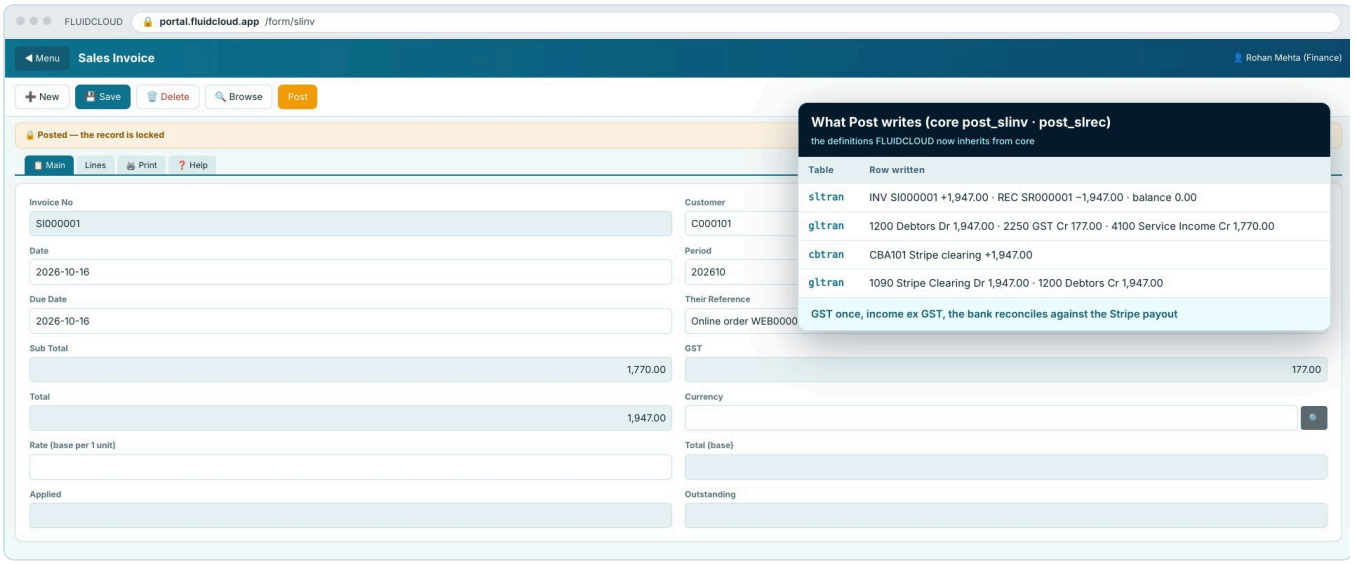

*Invoice Order and what it writes.*

1. Open the order (WEB######). Payment shows Paid with the provider's reference when the card was paid.
2. Check **Client Account** — it is filled from the buyer's portal login. Empty means the buyer is still a registrant: press **Create Client** on their Online Registration first.
3. Press **Invoice Order**.
4. Open the new invoice in **Sales Invoice**, review it and press **Post**; open the receipt in **Customer Receipt** and press **Post**.
5. Write a message to the buyer in **Message to the buyer** and set Order Status (In delivery, Completed) as you deliver; Save.

| What Invoice Order writes | posting `convert_cldorder_to_slinv` |
|---|---|
| `slinv` | SI###### · the client · today · reference “Online order WEB######” · sub-total, GST, total |
| `slinvlines` | one line per order line: service, quantity, price, the service's income account, GST |
| `slrec` | only when Paid: SR###### · the total · bank **CBA101 Stripe clearing** · the card reference |
| `cldorder` | Status Accepted · the invoice and receipt numbers (the button disappears) |

**Check it.** The order shows Sales Invoice and Card Receipt; the buyer's order shows Accepted and the invoice number.

**If the system refuses.** “Link the order to a client account first …” — Create Client on the registration, reopen the order. “The order is cancelled.” — nothing to invoice.




*Posting the invoice and the receipt moves the sales ledger, the cashbook and the general ledger (core post_slinv / post_slrec).*

| Posted | Ledger rows (WEB000001) |
|---|---|
| Sales invoice SI000001 | sltran INV +1,947.00 · gltran 1200 Dr 1,947.00 · 2250 GST Cr 177.00 · 4100 Cr 1,770.00 |
| Receipt SR000001 | sltran REC −1,947.00 · cbtran CBA101 +1,947.00 · gltran 1090 Dr 1,947.00 · 1200 Cr 1,947.00 |

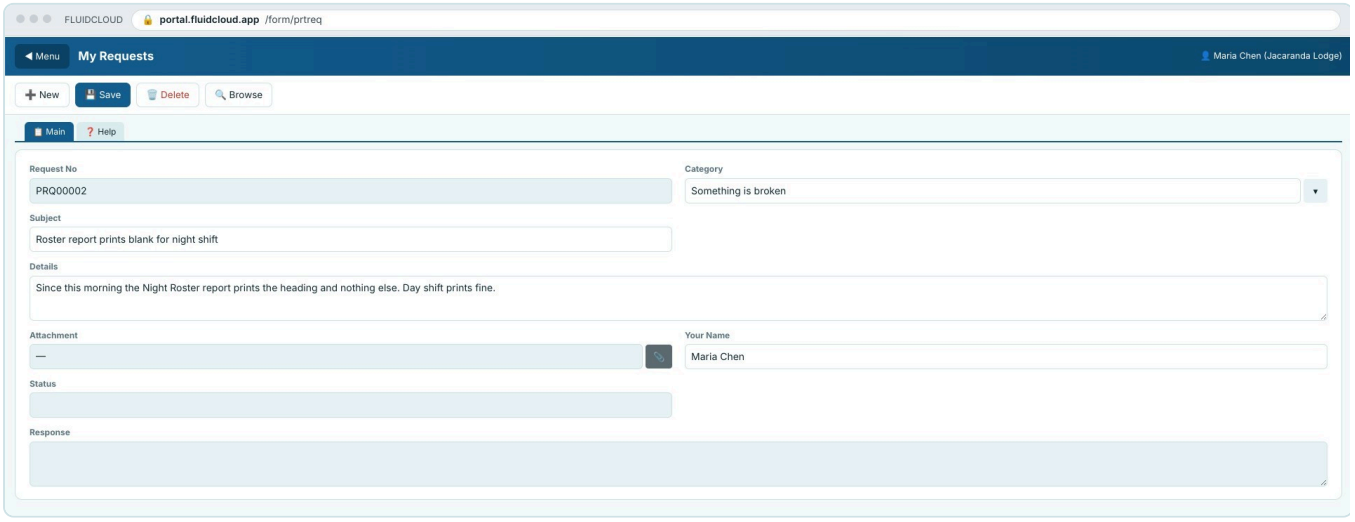
**A card payment after you invoiced.** If the buyer pays by card after the order was invoiced on account, Payment turns Paid but Card Receipt stays blank: enter the receipt in Customer Receipts against that invoice — bank CBA101 Stripe clearing, reference the Card Payment Ref. **Stripe payouts** land in your operating bank account; reconcile CBA101 against the payout with a transfer.

# 8 Requests and the service desk

Every request — from a client, a partner, a supplier or a member of staff — ends in one service desk. Requests from the portals arrive in 📮 **Portal Requests**; staff raise 🎫 **Service Requests** directly. The catalogue decides the team and the SLA, the impact × urgency matrix the priority, and the status and response go back to the requester.

## Send a request (client, partner or supplier)

**Who** clients, partners, suppliers **Where** 📨 **My Requests** **Table** `prtreq → (bridge) portalreq`




*A client's request: category Something is broken.*

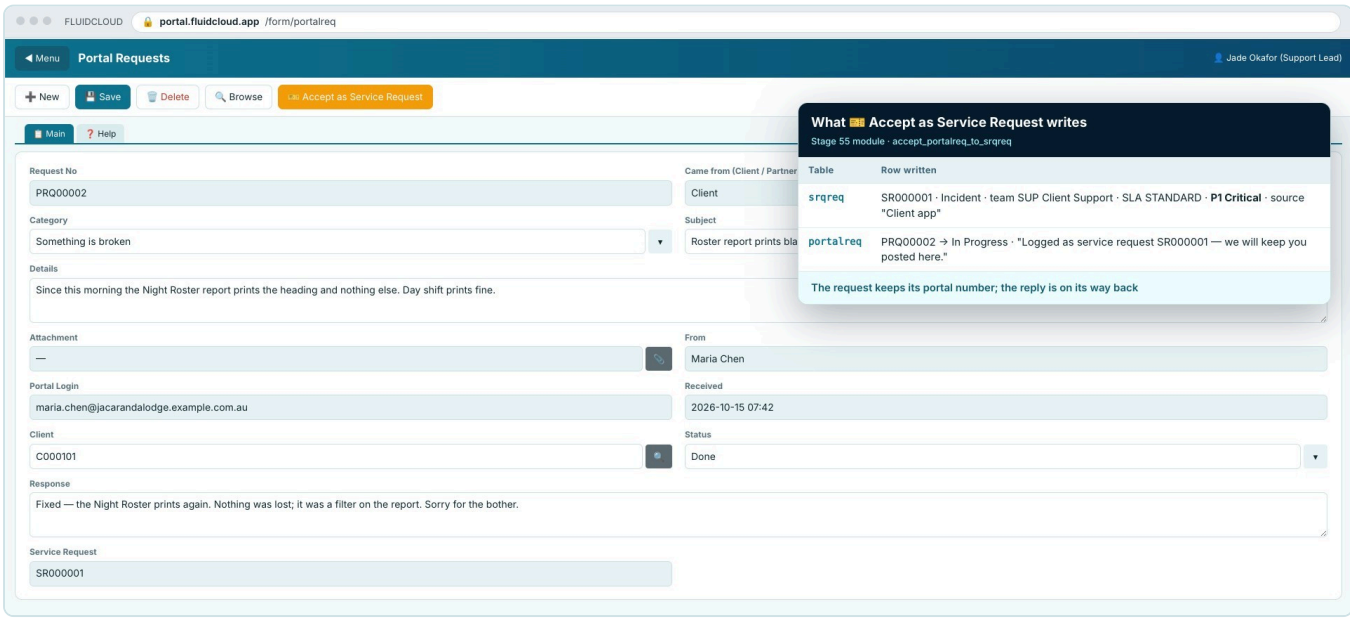
1. Open **My Requests**, press ➕ New.
2. Choose the category (table below), type a subject and the details; attach a screenshot if it helps.
3. 💾 Save. The request is on its way.
4. Come back to the same request: **Status** and **Response** are written by the service desk.

**Check it.** Client requests are numbered PRQ#####, partner and supplier requests PPR##### — they never collide in the staff in-tray.

| Category on the portal | Desk category | Kind | Team | SLA |
|---|---|---|---|---|
| Something is broken | SUP-INCIDENT Something is broken | Incident | SUP | STANDARD |
| Account query | SUP-ACCOUNT Account / sign-in | Service request | SUP | STANDARD |
| Billing query | FIN-BILLING Invoice / payment | Service request | FIN | STANDARD |
| Change request | IMP-CHANGE Change to my system | Change | IMP | STANDARD |
| Training or how-to | SUP-HOWTO Training / how-to | Question | SUP | STANDARD |
| Feedback | SUP-FEEDBACK Feedback | Question | SUP | STANDARD |
| Complaint | SUP-COMPLAINT Complaint | Incident | SUP | STANDARD |
| Question | SUP-QUESTION Question | Question | SUP | STANDARD |
| — (staff only) | OPS-RELEASE Release / upgrade | Change | OPS | INTERNAL |
| — (staff only) | STAFF-IT Staff IT request | Service request | OPS | INTERNAL |
| — (staff only) | STAFF-ADMIN Staff admin / facilities | Service request | FIN | INTERNAL |
| — (staff only) | PROJECT Project request | Change | IMP | INTERNAL |
| — (staff only) | OTHER Other | Question | SUP | STANDARD |

## Accept a portal request as a service request

**Who** support, implementation, ops, finance **Where** Staff → 🚨 **Portal Requests** **Table** `portalreq → srqreq`

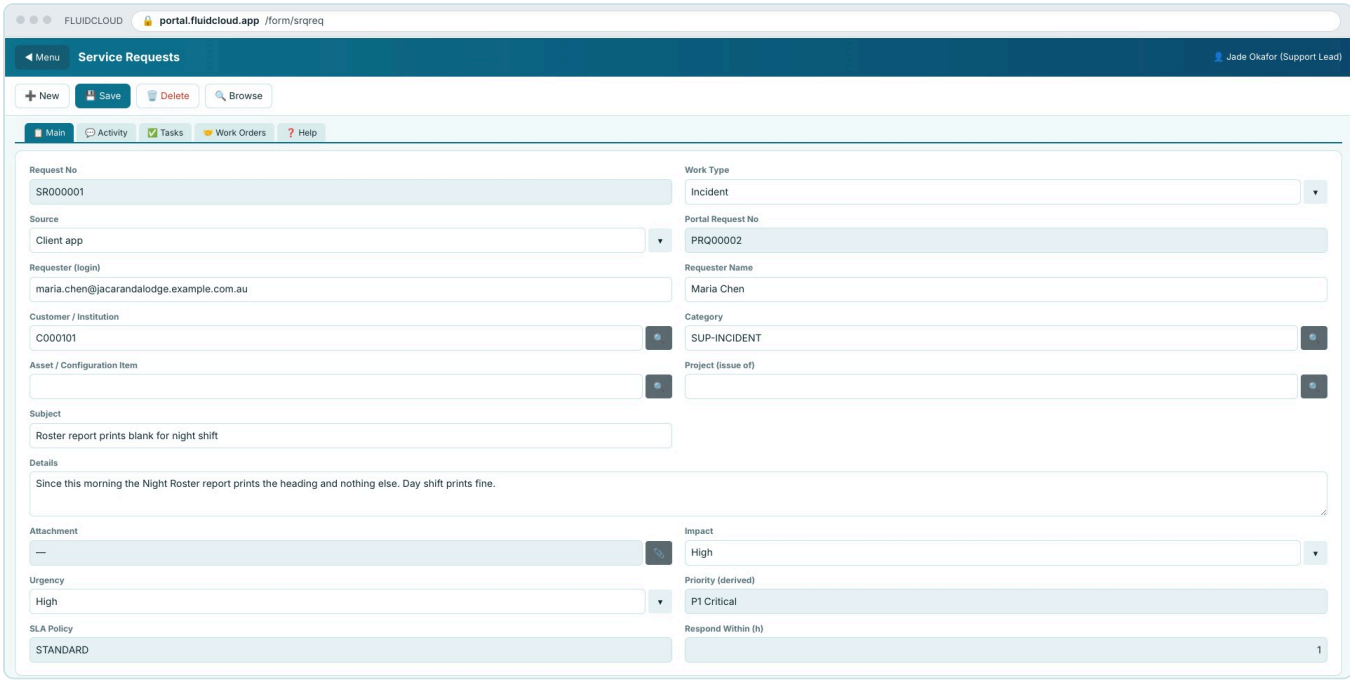



*Accept as Service Request and what it writes.*

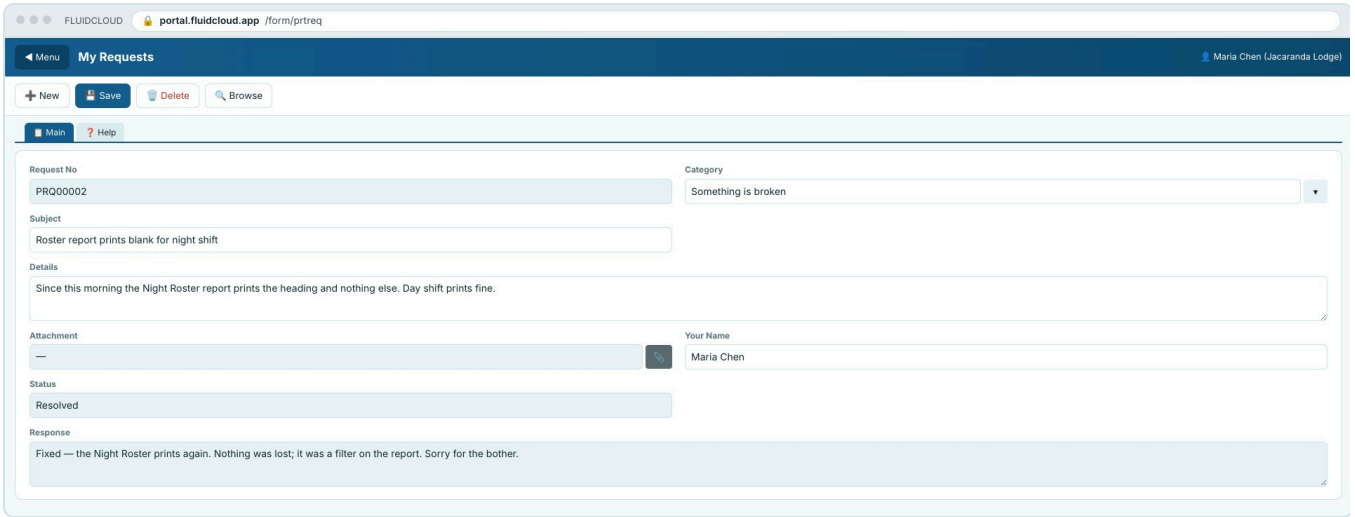
1. Open the request. **Came from** shows Client or Partner.
2. Press 🎫 **Accept as Service Request**.
3. Open the new SR in 🎫 **Service Requests**: assign it, log work on 💬 **Activity** (public replies reach the requester, internal notes do not).
4. Write the **Response**, set Status Resolved; or press ✅ Accept as Task, 📁 Create Project or 🤝 Assign to Partner when it is bigger.

| What 🎫 Accept as Service Request writes | posting `accept_portalreq_to_srqreq` |
|---|---|
| `srqreq` | SR###### · kind, team, SLA from the catalogue · priority from the matrix · source “Client app” / “Partner app” · the portal number kept |
| `portalreq` | In Progress · “Logged as service request SR###### — we will keep you posted here.” |

**Check it.** The requester's My Requests row shows the SR number at once and every status change after it.



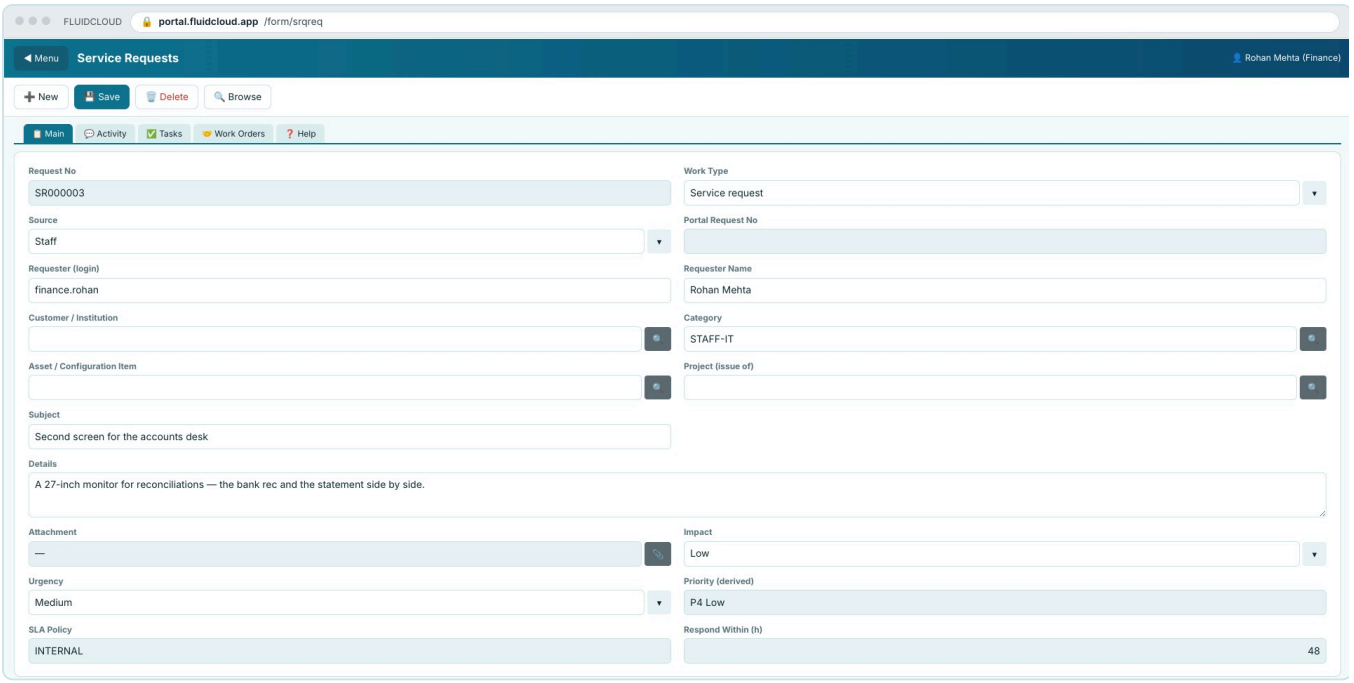

*SR000001: P1 Critical, resolve within 8 hours — the clocks run from opening.*




*Back on the client's side: Resolved, with the answer.*

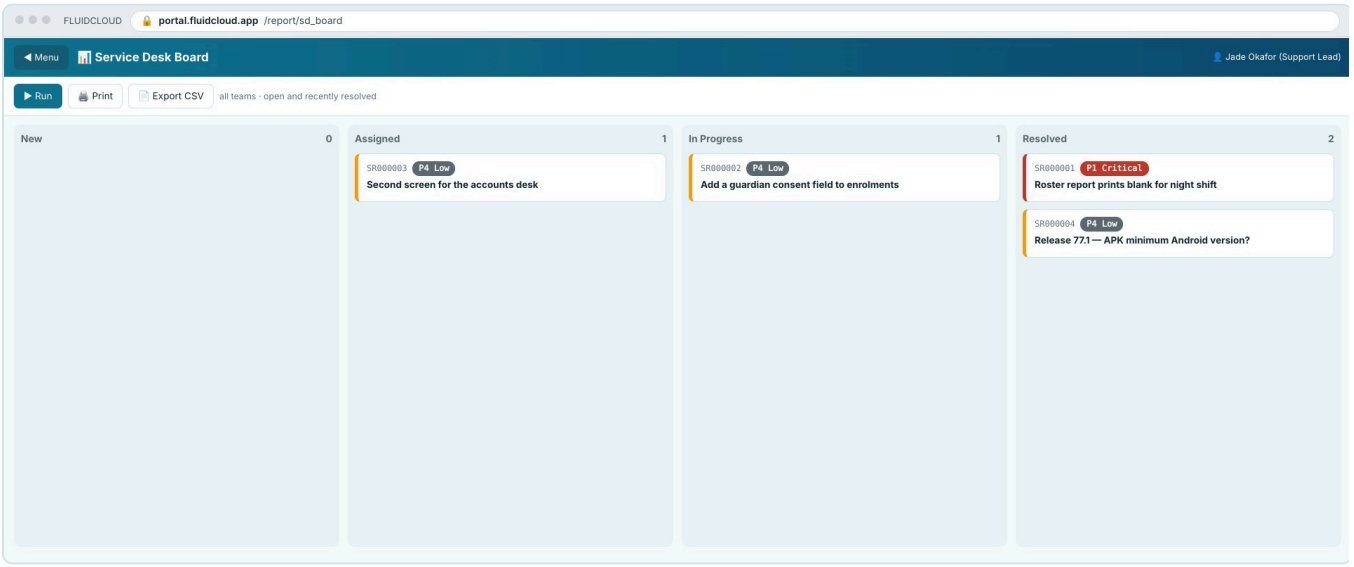
## Raise a staff request

Who all staff Where Staff → 🎫 **Service Requests** Table `srqreq`




*Rohan's staff IT request, routed to Platform Operations.*

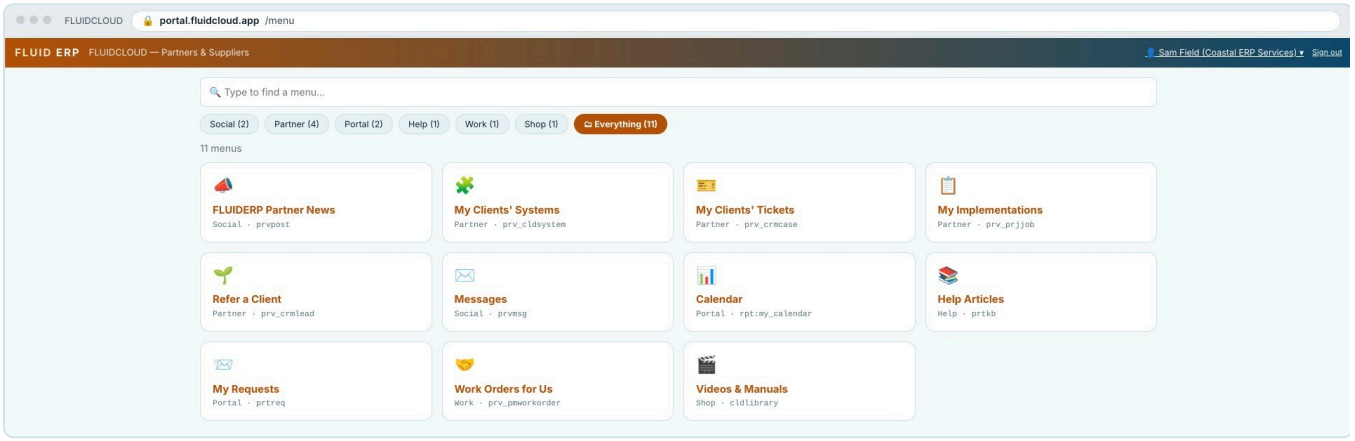
1. ➕ New; source Staff; category Staff IT request, Staff admin / facilities, Release / upgrade or Project request.
2. Describe it, 💾 Save. The catalogue routes it.




*🗂 Service Desk Board — every request, every source, by status.*

# 9 Partners and suppliers

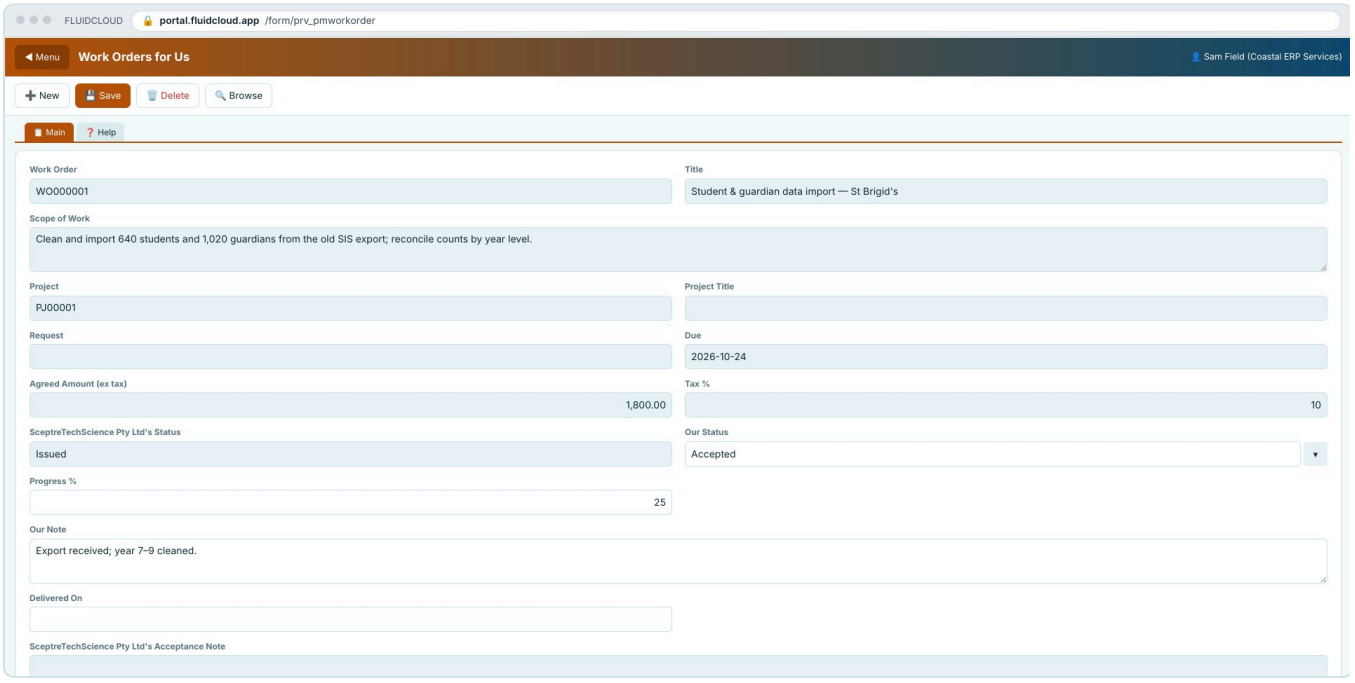
The Partners & Suppliers side has two groups. **partner** — implementation partners: referrals, the systems they implement, tickets, work orders, requests, the library and messages. **supplier** — anyone SceptreTechScience buys work from: work orders, requests, help articles, the library and messages. Logins are issued by SceptreTechScience (Users on that side, by the administrator).




*A partner's launcher.*

## Work a work order

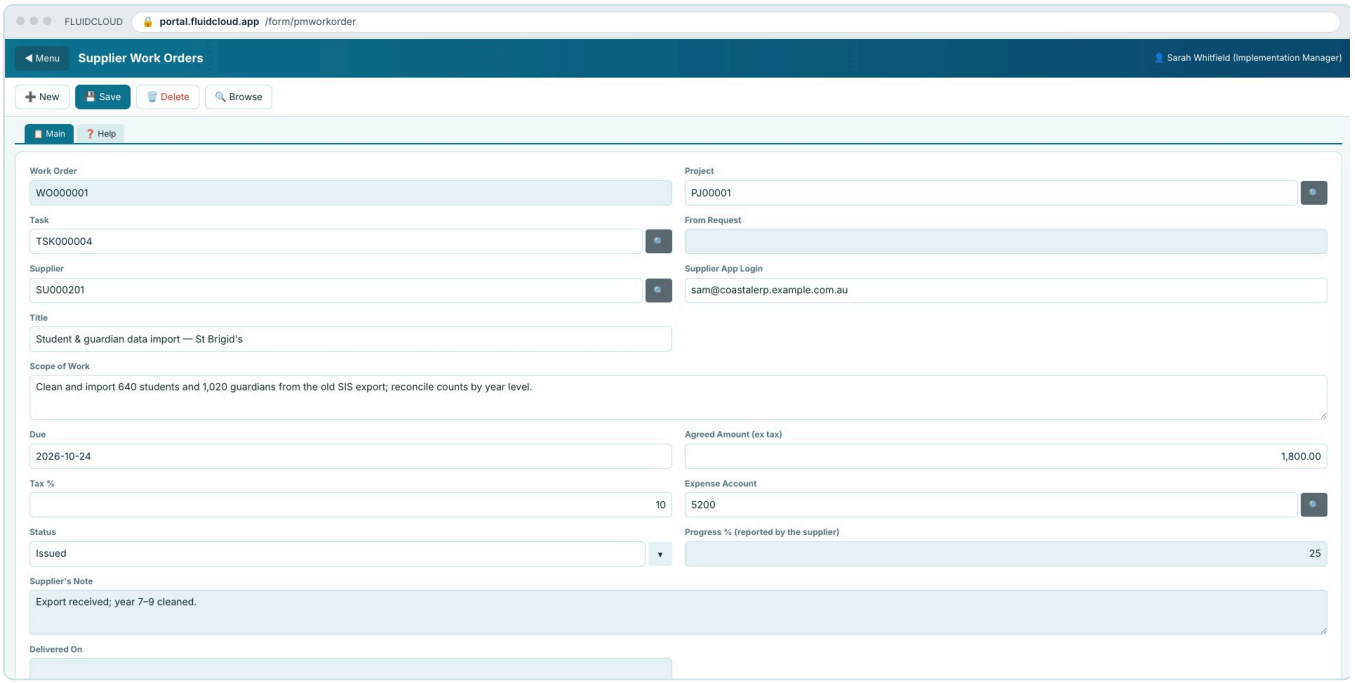
Who partners, suppliers Where 🤝 **Work Orders for Us** Table `prv_pmworkorder ↔ pmworkorder`




*WO000001 on the supplier's side.*

1. Open the work order: title, scope, due date, amount, the project it belongs to.
2. Set **My status** (Accepted, In Progress, Delivered), **Progress %** and a **note**; 💾 Save.
3. When SceptreTechScience accepts the delivery, the order shows Accepted by us and, later, Invoiced.

**Check it.** Your status, progress and note appear on the staff work order (they travel back by bridge).




*The same work order on the staff side; 🧾 Raise Supplier Invoice writes the core purchase invoice (5200 Subcontracted Services).*

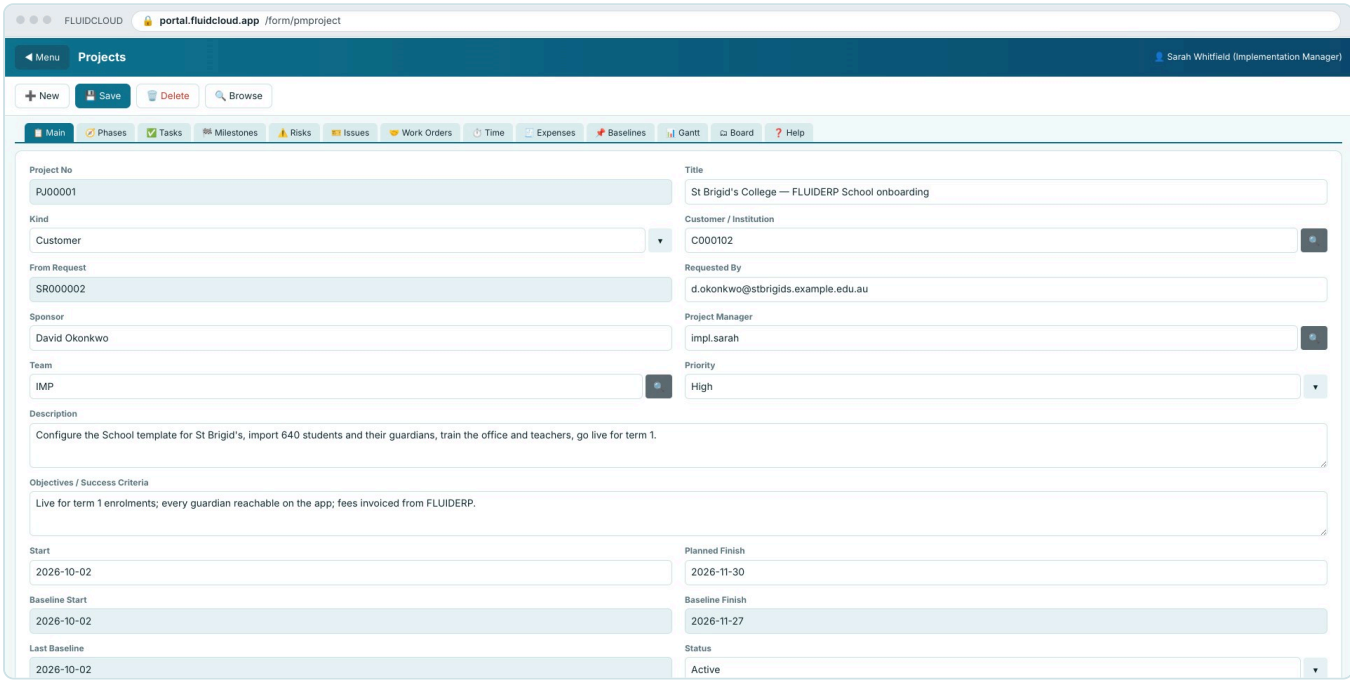
# 10 Projects

## Run a project

Who implementation (managers), finance

Where Staff → 📁 **Projects** · 🧭 **Service & Projects Station**

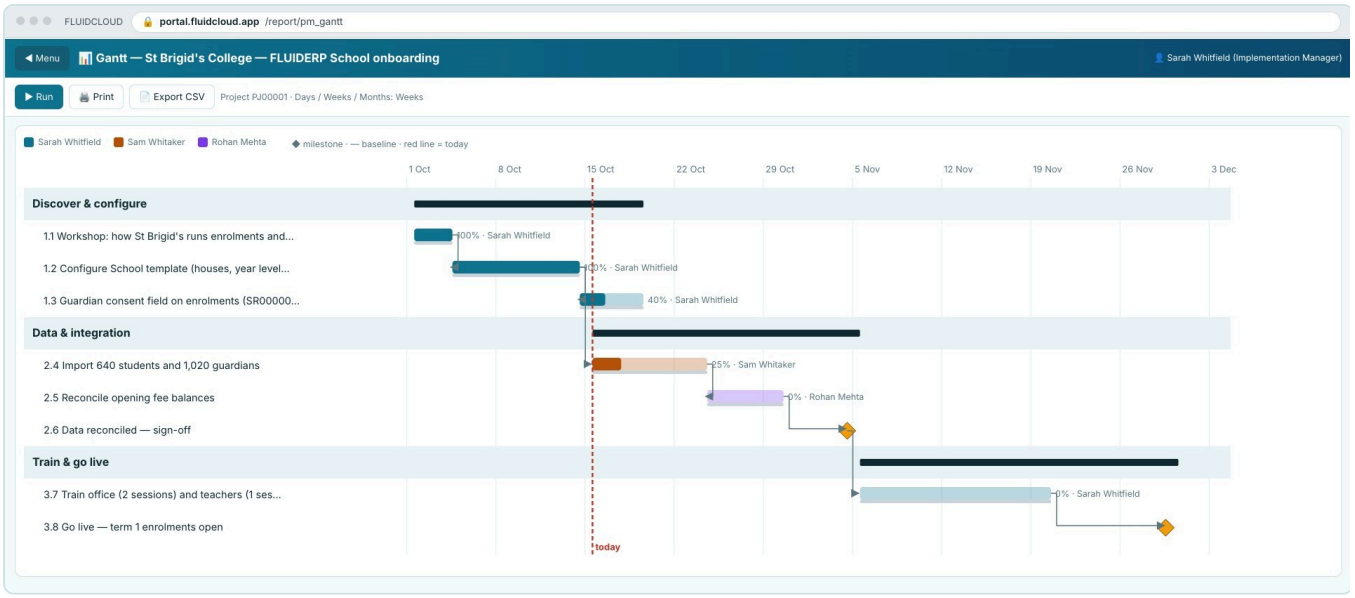
Table `pmproject` · `pmphase` · `pmtask` · `pmmilestone`



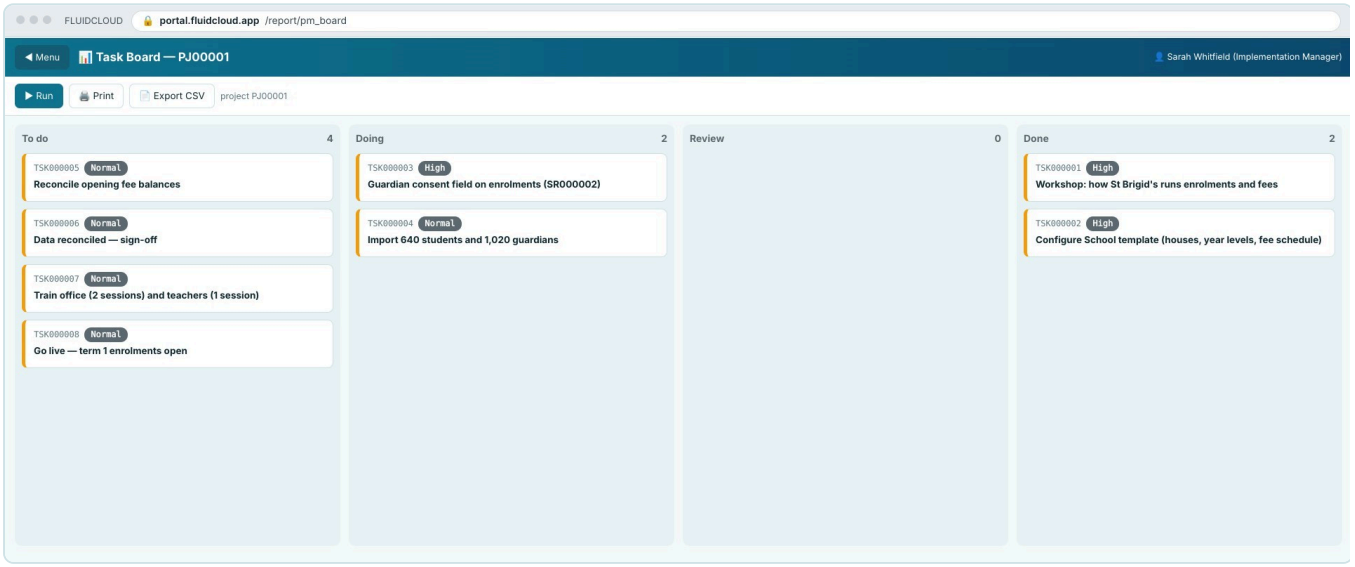

*PJ00001 — St Brigid's onboarding.*

1. Create the project (or 📁 Create Project from a service request): customer, manager, billing (fixed price, time & materials, milestones), budget, start and finish.
2. Add phases, tasks (owner, start, finish, hours, predecessor) and milestones (value, due). 📌 **Set Baseline** when the plan is agreed.
3. Keep task status and % complete current; log time on tasks.
4. Tick **Released** on the project and its milestones to show them to the customer.
5. 🧾 **Invoice Milestones** bills achieved milestones through the core sales invoice; 🧾 Bill Unbilled Time bills time & materials.

**Check it.** The project shows % complete, SPI / CPI, health (Green / Amber / Red), the next milestone and unbilled value — all calculated.



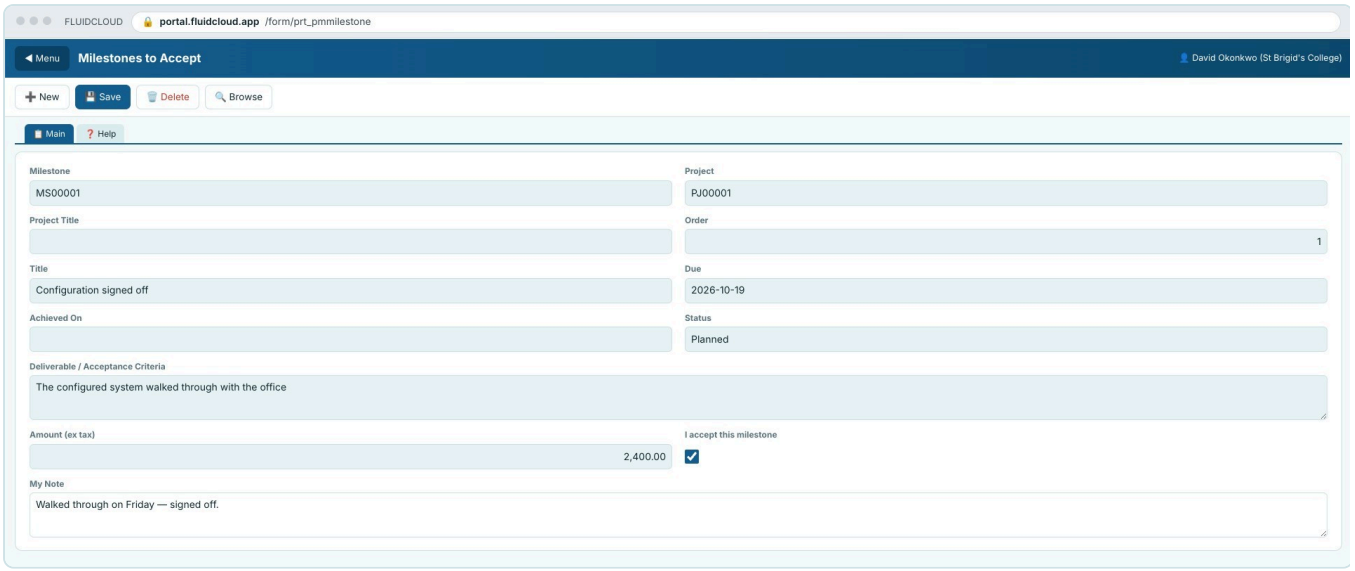

📊 *Gantt: bars filled to % complete, colour by owner, ◆ milestones, links, the baseline and today.*




🗂 *Task board: the same tasks by status.*

## Accept a milestone (customer)

Who clients Where 🏁 **Milestones to Accept** · 📁 **My Projects** Table `prt_pmmilestone`




*David accepts Configuration signed off.*

1. Open the milestone, read the deliverable.
2. Tick **Accepted**, add a note, 💾 Save.

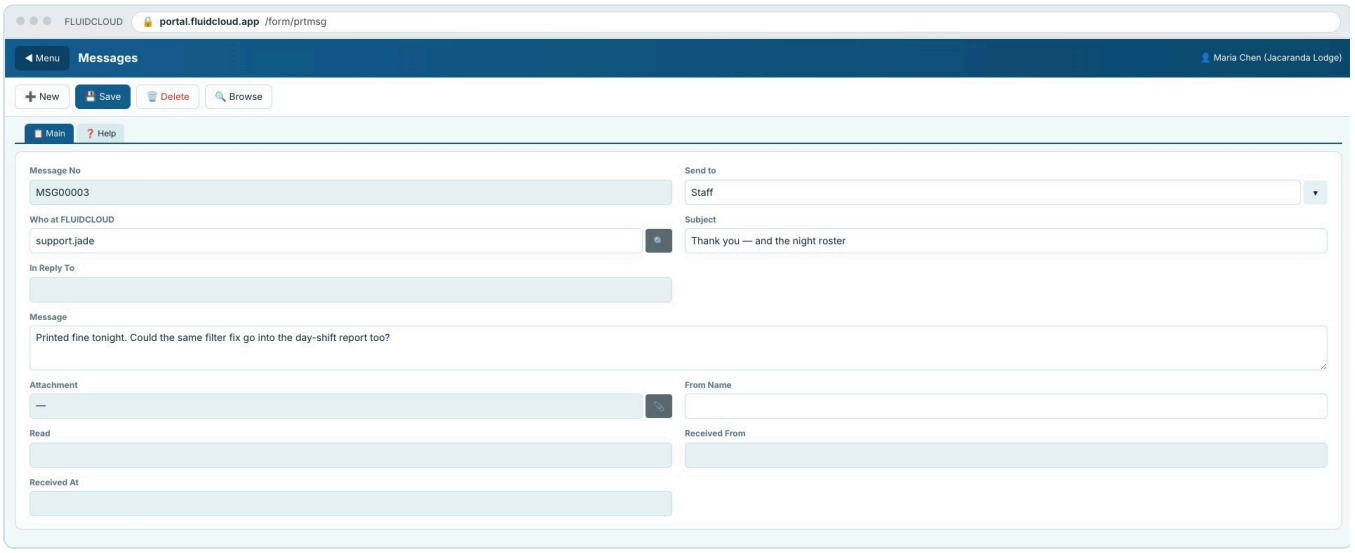
> **Check it.** The project manager sees your acceptance on the milestone.

**System Set-up Checklists** (staff) / **My System Set-up** (clients) are the short checklists FLUIDCLOUD creates when a system request is approved; **Projects** are the full project management above. Module projects are numbered PJ#####.

# 11 Messages between the sides

## Write to SceptreTechScience

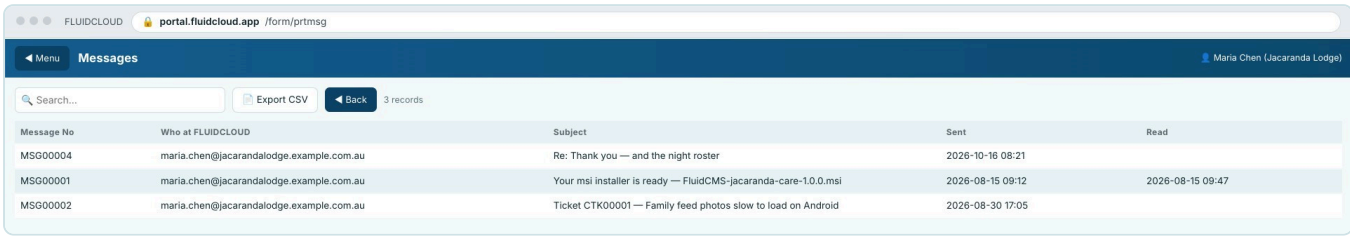
Who clients, partners, suppliers  Where ✉️ **Messages**  Table `prtmsg / prvmsg → msgmsg`




*Maria writes to Jade.*

1. ➕ New; To side **Staff**; To = the person (e.g. support.jade) or leave it to the team.
2. Subject, message, attachment; 💾 Save.

**Check it.** The reply arrives in the same Messages screen. Staff write to a client with To side Client and the client's login.

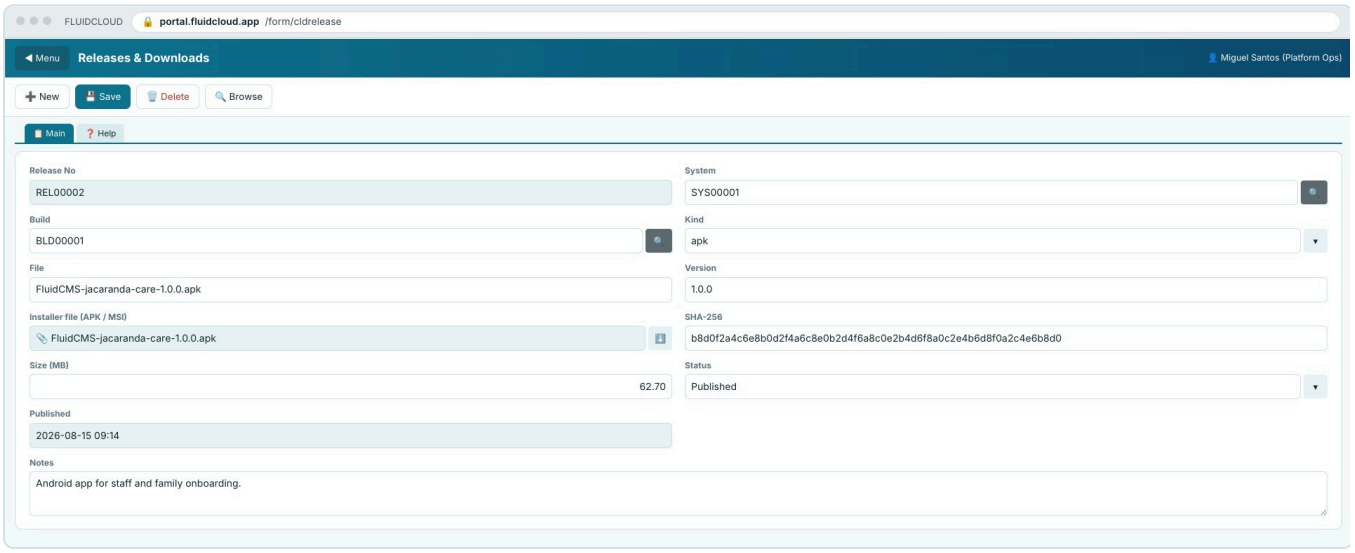



*The reply in Maria's inbox. Every message is validated by the receiving side's clean gate.*

# 12 Downloads and the app

## Publish an installer (staff)

Who ops, cloud_admin  Where Staff → 📦 **Releases & Downloads**  Table `cldrelease → prt_cldrelease`

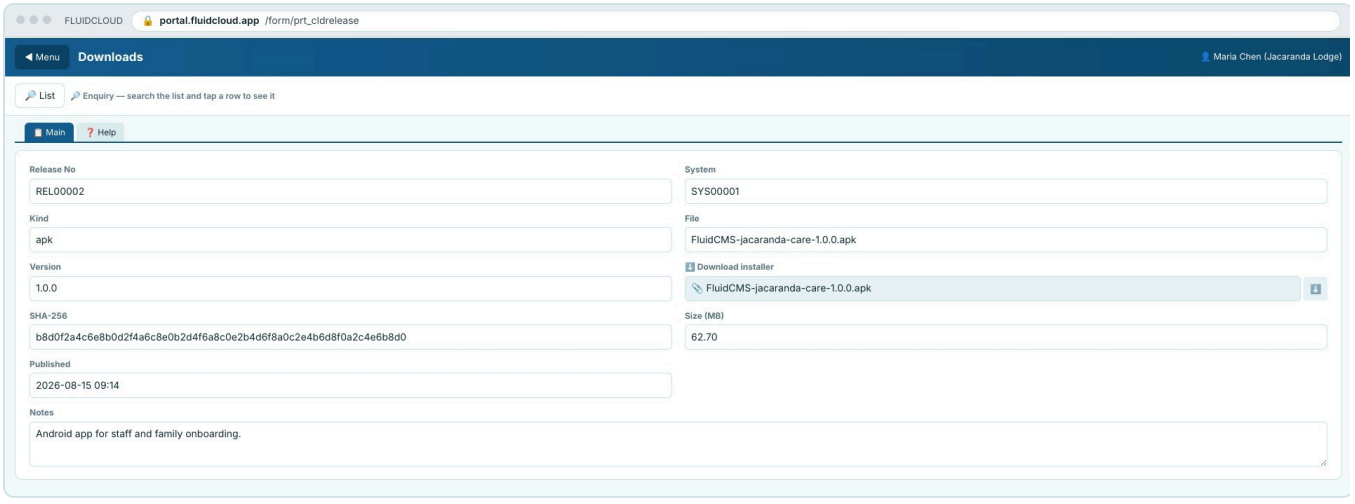



*A release with its APK.*

1. Open the release; attach the APK or MSI in **Installer file**; check version and SHA-256.
2. Press 📣 **Publish**. Only Published releases reach the client.

## Download and install the app (client)

Who clients  Where 📦 **Downloads**  Table `prt_cldrelease`




*Downloads on the client's side.*

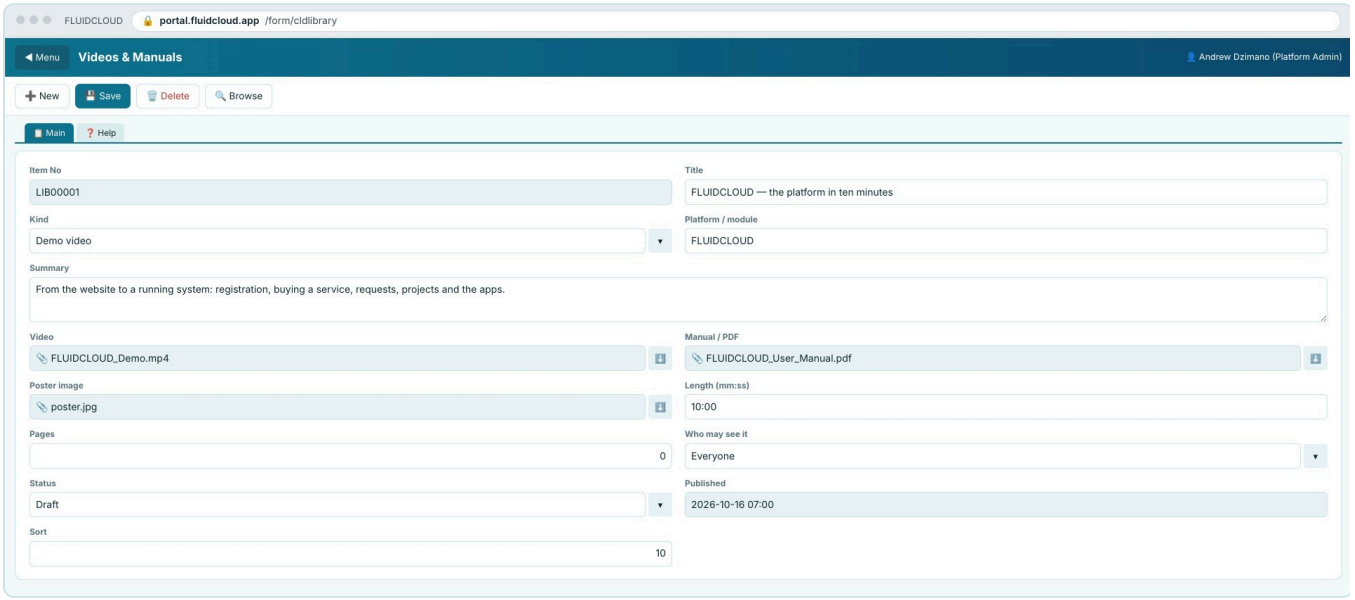
1. Open the release; press ⬇️ beside **Download installer**.
2. Compare the SHA-256 on the screen with your device's.
3. Android: allow the install when asked (Android 8.0 or newer). Windows: run the MSI.
4. Open the app, choose your side on the sign-in toggle, sign in.

**Check it.** The app is optional — every screen in this manual works in the browser.

# 13 Publishing videos and manuals (staff)

## Publish a video or manual

Who cloud_admin, ops, implementation  Where Staff → 🎬 **Videos & Manuals**  Table `cldlibrary → (bridge) client and partner libraries`




*A new library item, Draft, files uploaded.*

1. ➕ New: title, kind, platform, summary; upload the video (MP4), the PDF and a poster; choose **Who may see it**; Status **Draft**; 💾 Save.
2. The clean gate validates the text and every file. With an AI provider configured this takes a minute; without one the files wait as **Flagged** for a reviewer.
3. A reviewer (cloud_admin, ops, support) opens 🛡️ **Content Validation Queue**, checks each Flagged item, sets **Passed** with a reason and saves.
4. When the item shows Passed, set Status **Published**, 💾 Save. It appears on the portals within the minute.
5. To take it down, set Status **Withdrawn** — it disappears everywhere without being deleted.

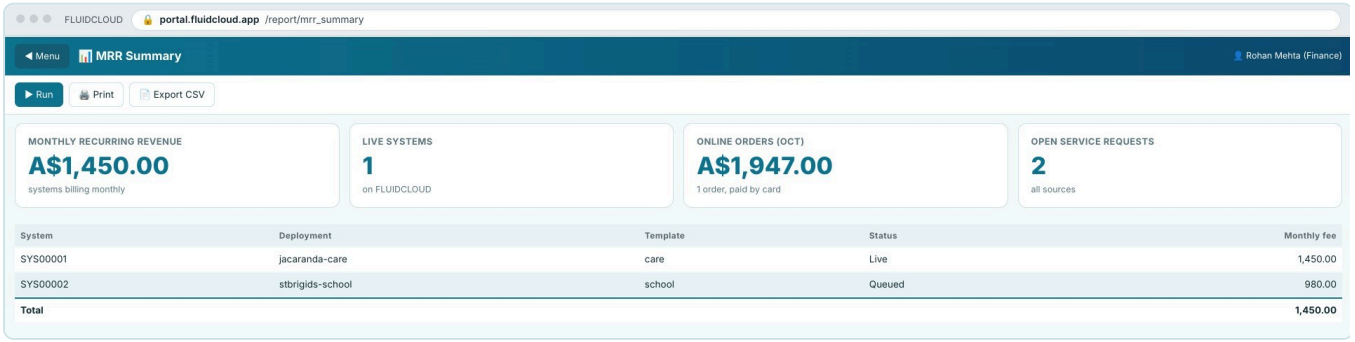
| **What Published (Save) writes** | posting `bridges cldlibrary_to_client / _to_partner` |
|---|---|
| `cldlibrary (client)` | the item and its files (copied with their verdict), visible to registrants and/or clients |
| `cldlibrary (partner)` | the item for partners and/or suppliers |

**If the system refuses.** “The AI content validation has not passed on this record …” — Published is refused until the item and its files have passed. Clear the Flagged files in the queue first.



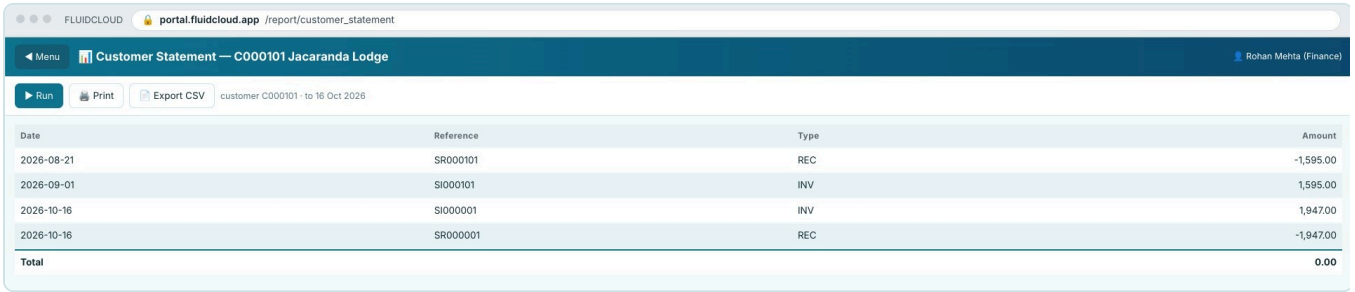

*The Content Validation Queue: files held Flagged for a reviewer.*

# 14 Reports and the numbers




*📊 MRR Summary: recurring revenue, live systems, online orders, open requests.*

| Report | Shows |
|---|---|
| 📊 MRR Summary | monthly recurring revenue by system |
| 📊 Online Orders | every portal order, its payment and invoice |
| 📊 Services Price List | the catalogue as the website shows it |
| 📊 Videos & Manuals Register | what is online, for whom, and its validation |
| 🗂 Service Desk Board · SLA Watch · Ageing · Resolved | the desk |
| 📊 Gantt · Portfolio Gantt · Board · Health · Workload | projects |
| Unbilled Time · Unbilled Milestones · Work Orders · Project Statement | project money |
| Trial Balance · GL Detail · Aged Debtors · Customer Statement · GST Summary · Income Statement · Balance Sheet | the Big 5 |




*A customer statement: the subscription and its payment, the online order and its card receipt — balance 0.00.*

Every report has ▶ Run, 🖨 Print (the document viewer — print or save as PDF) and 📄 Export CSV.

# 15 Going live (administrator)

A live install contains FLUIDCLOUD's configuration and none of the demonstration: no sample clients, invoices, tickets, requests, partners or logins. It is built by `setup/golive.py`, run by the installer when you pass a live file.

## Before you start

| You need | Notes |
|---|---|
| A server (Ubuntu or Windows) reachable from the internet | the gateway listens on HTTPS; the browser client on 8444 |
| A domain name for the portal, pointing at the server | e.g. portal.sceptretechscience.com — Let's Encrypt issues the certificate |
| Your company details | legal name, address, phone, e-mail, website, ABN, bank lines for invoices |
| Your staff logins | name, group (cloud_admin, ops, support, implementation, finance) and a real e-mail each |
| Stripe account | secret key and webhook secret (test or live) |
| An outbound mail relay | for codes, password resets and notifications |
| Optional: an AI moderation endpoint | any endpoint answering the OpenAI moderation shape; otherwise reviewers clear uploads |

## Steps

1. Copy `setup/live/fluidcloud.live.example.json` to `setup/live/fluidcloud.live.json` and replace every CHANGE-ME: company, bank lines, staff logins. Review the starter prices in the services catalogue (or remove `cldservice` from keepData to start empty).
2. Check it: `python3 setup/golive.py --check --setup <spec>` lists anything that still reads like the demonstration. The install refuses while the list is not empty.
3. Install: `sudo ./install.sh --institution fluidcloud --live setup/live/fluidcloud.live.json`. Choose Let's Encrypt, your domain, the mail relay, and Stripe (secret key and webhook secret are asked hidden and stored only in the protected environment `/etc/fluid-erp.env`).
4. In Stripe, add the webhook endpoint `https://<domain>:<port>/cloudclient/api/payments/webhook` for checkout.session.completed / expired / async_payment_*.
5. Install the browser client: `sudo bash scripts/install-web-client.sh` — the website links to it.
6. Hand out the first passwords from `setup/initial-passwords.txt` (0600), have everyone change theirs, then delete the file.
7. Give the **compliance** / reviewer duties to people: add logins to the client and partner sides' `compliance` group if those sides need a reviewer (no login ships in it).
8. Upload this film and manual to 🎬 Videos & Manuals (chapter 13) and publish them for Everyone.

## Website links

| Button on sceptretechscience.com | Link |
|---|---|
| Get FLUIDERP / Register | `https://<portal>/apply/register?side=cloudclient` |
| Client & partner sign-in | `https://<portal>/` |
| Download the app | Downloads, after sign-in (installers are per client system) |

## Security notes

- Every login is hashed (PBKDF2); a login with no password can never sign in; there is no demo login on a live install.
- Gateway keys, token secrets, the Stripe and mail secrets live only in the protected environment; a leaked appsettings.json leaks nothing.
- Only reviewers see the content queue, the lexicon, e-signatures, records retention and legal holds — never customers.
- Keep `setup/setup.lock.json`, `setup/generated-secrets.env`, `setup/initial-passwords.txt`, `setup/live/*.live.json`, `setup/out/` and `dist/` out of source control.

# 16 Messages, refusals and what to do

| The system says | Why | Do this |
|---|---|---|
| Link the order to a client account first … | the buyer is still a registrant | Create Client on the Online Registration, reopen the order |
| The order is cancelled. / The order has no value. | nothing to invoice | — |
| The AI content validation has not passed on this record … | text or files not yet Passed | clear Flagged items in 🛡️ Content Validation Queue (reviewers) |
| only the uploader or a reviewer | a file still under validation | wait for the verdict, or ask a reviewer |
| The customer is on credit hold … | Post on an invoice for a held client | release the hold on the client |
| 💳 Pay by Card does nothing | order not Submitted | set Status Submitted, 💾 Save |
| payments return 503 | no Stripe secret key in the environment | administrator: re-run the installer's payments step |
| This form is not available on this side. | a public form opened on the wrong side | use ?side=cloudclient in the link |
| live install REFUSED — the spec still carries demonstration values | placeholders in the live file | fix each item listed, run --check again |
| Not delivered — … | a message to someone who does not exist / blocked you | check the address |

# 17 Data reference and glossary

| Table | Side | What it is |
|---|---|---|
| cldservice | staff (mirror on clients) | the services catalogue / price list |
| prtorder · prtorderlines | clients | the buyer's basket and order |
| cldorder · cldorderlines | staff | online orders as received |
| cldlibrary | staff (mirrors on clients, partners) | videos and manuals |
| portalreg / prtreg | staff / clients | registrations |
| portalreq / prtreq | staff / clients, partners | requests from the portals |
| srqreq · srqact · srqkb · srqcat · srqteam · srqsla · srqpri | staff | service requests, activity, help articles, catalogue, teams, SLA, priority matrix |
| pmproject · pmphase · pmtask · pmmilestone · pmworkorder · pmtime · pmexpense | staff | projects |
| prt_pmproject · prt_pmmilestone · prt_srqreq | clients | the customer's view of projects and requests |
| prv_pmworkorder | partners & suppliers | work orders for the supplier |
| cldsystem · cldrequest · cldrelease · cldbuild · cldserver · cldlicence · cldkey · cldtoken | staff | the platform: systems, requests, releases, builds, servers, licences, vault, tokens |
| slcust · slinv · slinvlines · slrec · sltran · gltran · cbtran · glacct · cbacct | staff | the Big 5 sales ledger, GL and cashbook |
| msgmsg / prtmsg / prvmsg | each side | messages |
| sysmodq · sysmodlex | each side (reviewers) | content validation queue and lexicon |

| Term | Meaning |
|---|---|
| Bridge | a configuration row that copies chosen records and fields from one cluster to another |
| Cluster / side | one FLUIDERP database with its own logins: Staff, Clients, Partners & Suppliers |
| Posting definition | a button that writes the rows of a business event |
| Clean gate | validation of text and files (AI or a reviewer) before anyone else can see them |
| Registrant | a prospect who has registered but is not yet a client |
| SLA | the response and resolve times a request's category promises |
| Release | a version of a client's system: its installer and checksum |
| Stripe clearing (CBA101 / 1090) | the account card receipts post to until Stripe pays out |

# Appendix A Walk it once on the demonstration package

Install the demonstration (`sudo ./install.sh --institution fluidcloud`, no --live) on a test server and sign in with the logins below. **These passwords exist only in the demonstration — never on a live install.**

| Side | Login | Password | Group |
|---|---|---|---|
| Staff | cloudadmin | cloudadmin | cloud_admin |
| Staff | support.jade | support.jade | support |
| Staff | impl.sarah | impl.sarah | implementation |
| Staff | finance.rohan | finance.rohan | finance |
| Staff | ops.miguel | ops.miguel | ops |
| Clients | maria.chen@jacarandalodge.example.com.au | client123 | client |
| Clients | d.okonkwo@stbrigids.example.edu.au | client123 | client |
| Clients | tom.reilly@reillymfg.example.com.au | client123 | registrant |
| Partners | sam@coastalerp.example.com.au | partner123 | partner |

1. Clients side, Tom Reilly (registrant): 🎬 Videos & Manuals, 🛒 Services & Prices.
2. Maria: 🧺 Buy Services → WEB000001 is already Submitted and Paid. Staff, finance.rohan: 🛒 Online Orders → WEB000001 → 🧾 Invoice Order → Post the invoice and the receipt.
3. Maria: My Requests → PRQ00002 (Resolved). Staff, support.jade: 🎫 Service Requests → SR000001 → 💬 Activity.
4. Partners, Sam: 🤝 Work Orders for Us → WO000001; My Requests → PPR00001.
5. Staff, impl.sarah: 📁 Projects → PJ00001 → 📊 Gantt.
6. David: 🏁 Milestones to Accept → MS00001.
7. Staff, cloudadmin: 📊 MRR Summary (A$1,450.00).

# Appendix B Administrator notes

The bridges between the three sides (erpcfg.tblbridge):

| Direction | What travels |
|---|---|
| Staff → Clients | news, events, messages, systems, releases (Published only), tickets, set-up checklists, statements, request status, help articles, released projects and milestones, price list, library, order status |
| Staff → Partners & Suppliers | news, messages, systems, tickets, set-up checklists, request status, help articles, work orders (Issued / Accepted by us / Invoiced / Cancelled), library |
| Clients → Staff | messages, requests, registrations, system requests, tickets, posts, request ratings, orders and their lines |
| Partners & Suppliers → Staff | messages, referrals, posts, requests, work-order progress (Accepted / In Progress / Delivered) |

Generated by `tools_gen_tune_cloud.py` on top of `tools_gen_fluidcloud.py`; regenerate the chain with `scripts/regen-fluidcloud-golive.sh`. Checks: `institutions/fluidcloud/validate.py`, `verify_all.py` (gates 26–29 cover the go-live review). Known follow-ups: FluidCloud's messages use the side-to-side bridges, not yet the 75.0.9 addressed messages (partners cannot write to clients directly); on a multi-server deployment the order status bridge needs a loop guard; the browser sign-in page shows “FLUID ERP” and opens on the Staff side unless the link names ?side=.