

# AI agents (Preview)

From the SourceLace manual: An agent is a job you set up once and SourceLace runs for you: every weekday morning, every Monday, two days before each QBR in your calendar, or as soon as something happens, such as a new priority 1 case. It reads your systems with your access, then writes a summary, dr

SourceLace version 0.1.0 · Made October 7, 2026

The latest version is at <https://sourcelace.onrender.com/docs>

# AI agents (Preview)

An **agent** is a job you set up once and SourceLace runs for you: every weekday morning, every Monday, two days before each QBR in your calendar, or as soon as something happens, such as a new priority 1 case. It reads your systems with **your** access, then writes a summary, drafts emails or proposes changes to records for you to approve. Agents are in **Preview**: new, offered for pilots and provided as is, and we are still shaping them with early customers.

Agents are in the SourceLace app under **Agents**. They are included on the Team plan (drafts and summaries only), Business and Enterprise; see [Plans and limits](#).

Agents run on SourceLace by default. If your organization has a **runner** (agents inside your own network, your cloud account or data center), you can choose it for an agent under **Where it runs**: its runs, results and approvals then stay in your network. See **Runners: agents inside your own network** in this manual.

New to agents? Follow [Build your first agent](#), step by step. To share an agent with your team, see [Make templates for your team](#).

## Starting from a template

The quickest way to start is **Agents → Templates**. Each template is a ready-made job, written for the systems you have:

| Area             | Templates                                                                            |
|------------------|--------------------------------------------------------------------------------------|
| Sales            | Weekly pipeline review, Stale opportunity nudges, Meeting prep brief, Call follow-up |
| Customer success | QBR prep, Renewal reminder, Account health score update                              |
| Support and IT   | Sev 1 escalation, Daily ticket backlog digest                                        |
| Finance          | Overdue invoice reminders, Month-end close status                                    |
| Supply chain     | Late purchase orders, Low stock alert                                                |
| HR               | New hire onboarding                                                                  |
| RevOps           | CRM data hygiene, Closed-won handoff                                                 |

Pick one and press **Use template**. The wizard asks:

1. **Systems.** Which of your sources to use for each need, such as your CRM (Salesforce, HubSpot or Dynamics 365) and your mailbox (Gmail or Outlook). Only sources you can use are offered.
2. **Questions.** A few settings, such as "Days without activity" or "Minimum amount".
3. **When and how.** The schedule in your time zone, and for record changes whether each one waits for your approval.
4. **Name and results.** Where results go: a new project named after the agent, or one of yours.

The agent is created as a **draft**. Check its instructions, press **Dry run**, then **Turn on**.

## Changing your answers later

On an agent made from a template, **Change answers** asks the template's questions again with your current answers filled in. The instructions and actions are made again from the template, as a new version; the name, systems, schedule, limits and approval choices stay, and your own edits to the instructions are replaced.

Admins can hide templates people should not use. Templates that change records show "Changes records, which agents do available on the Business plan" on a plan where agents make drafts only. Anyone who may create agents can write templates for the organization in the [template editor](#), or turn one of their agents into a template with **Save as template**.

## Template editor

Your organization can make its own templates: **Agents → Templates → New template**, or **Save as template** on any agent (the editor opens filled in from the agent; nothing is saved until you save). The person who made a template and your admins can **Edit** it later.

The template editor is split into tabs, one for each part below: **About**, **Systems**, **Questions**, **Instructions**, **When it runs**, **Actions** and **Limits**. Every field has an  with what it does and an example. **Save**, **Check** and **Try it** stay at the bottom of the screen and work on every tab at once; a tab with something to fix shows a red count, and **To fix before saving** above the tabs links to the tab of each problem.

### About it

- **Name:** what people see in the template list. Name the job, such as "Overdue invoice reminders".
- **Category:** the team it is for (Sales, Customer success, Support and IT, Finance, Supply chain, HR, RevOps, or Other). People filter the list by it.
- **What it does:** one or two sentences on what an agent made from it does and what it produces.
- **Who it is for:** such as "Sales managers". Shown on the template's card.

### System slots

The kinds of system the agent needs, such as a CRM and a mailbox. Each person picks one of their own sources for each slot when they use the template.

- **Label:** shown in the wizard, such as "CRM" or "Mail (for drafts)".
- **Short name:** used in the instructions as {source.crm} (the chosen source's id) or {system.crm} (its product name).
- **Systems that fit:** which systems fit the slot. **Any CRM** adds Salesforce, HubSpot and Dynamics 365 at once; likewise Any ERP, Any HR system, Any ticket system, Any mailbox, Any chat and Any data warehouse.
- **Required:** when off, people may leave the slot empty, and instruction lines starting [if crm] are left out.

## Questions

What people answer when they use the template: **text**, **number** (with a smallest and largest value), **choice** (from a list), **yes or no**, or an **email address**. Each has a **question** as people read it, a **short name** (lowercase letters, digits and \_) to use as {name} in the instructions, optional **help text** shown under it, and an optional **default answer**. Turn off **People must answer it** to allow an empty answer.

## Instructions and placeholders

What the agent should do, as you would brief a new colleague. Use:

- {stale\_days} for the answer to a question,
- {source.crm} and {system.crm} for the system chosen for a slot,
- [if update\_crm] at the start of a line to keep the line only when a yes/no question is answered yes, or an optional slot is filled.

The editor checks every placeholder as you type: one that names no question and no slot is shown under the instructions with its line number, and the template cannot be saved until it is fixed. **Check** runs the full check on the server.

## Default trigger, actions and limits

- **When it runs:** when people press Run, on a schedule, or before calendar events. People can change it in the wizard and the editor. Templates saved from an agent that watches for something keep that query under **Advanced: the queries it watches**.
- **What it may do:** drafts and documents run on their own; record changes and chat posts wait for the owner's approval unless you choose otherwise and an admin allows it. **Only when** adds an action only when a yes/no question is answered yes. For record changes, give for each system that fits the slot the object and the fields the agent may set; leave a system empty if it has no good place for it.
- **Suggested limits:** steps, records, credits per run and runs per day. People can change them when they edit their agent.

## Versions

Each save makes a new version. Agents remember the template and version they were made from (shown on the agent's page); they do not change when the template does.

## Try it

**Try it** fills the template in with your own systems and answers and shows the agent it would make: when it runs, what it reads, what it may do and the full instructions, plus anything that would stop it being created. Nothing is saved or run.

## SourceLace library

The templates under **Agents → Templates** marked as SourceLace's come from the SourceLace library. The people who run SourceLace add, edit, retire and target them in the app (**Agents → Templates → SourceLace library**), with no software release:

- **Offered to:** every customer, or only chosen customers.
- **Status:** *published* templates are offered; *retired* ones no longer are, and agents made from them keep working.
- **Versions:** each save is a new version, as for your own templates.

Your admins can still hide any library template from your organization.

In the library editor, the **Who sees it** settings are on the template's **About** tab.

## Describe it

The quickest way to a new agent is **Agents → New agent**, then **What should this agent do?** Write a few sentences, such as "Every Monday, find deals with no activity for 3 weeks and draft a nudge to each owner", and press **Draft it**.

- If a template fits, SourceLace offers it first ("This looks like the Stale opportunity nudges template"). **Use it** opens the template's wizard; **No, draft it for me** carries on.
- Otherwise your organization's AI model drafts the agent: its instructions, the systems from your own sources, what it may do, when it runs and its limits. You see a short summary of what it will do and notes on anything left out (for example a system you do not have, or record changes on a plan that makes drafts only).
- The draft opens in the editor **unsaved**. Nothing runs until you save it and turn it on; try a **Dry run** first.

Safe defaults: email drafts and documents run on their own (they change nothing until you send or use them); record changes and chat posts always wait for your approval; limits start from your plan's. To react to something happening, it picks one of the ready-made triggers below rather than writing a query.

The AI model sees only what you wrote and the names and kinds of your systems, never any data from them. Drafting counts as one chat question against your organization's AI use.

## Writing your own agent

The editor is split into tabs, so you never scroll through one long form: **The job, Systems and actions, When it runs, Results** and **Limits and alerts**. Each tab starts with one line on what it is for and a **Learn more** link here, and every field has an ⓘ next to its name: click it (or press Enter on it) to see what the field does, what happens if you leave it empty, and an example. The tab you were on is kept in the page's address, so a reload or a shared link opens the same tab. **Save** (or **Create draft**) stays at the bottom of the screen and saves every tab at once; if something is missing, the tab with the problem shows a red count and opens, with the field to fix selected.

- **The job** tab: the **Name**, an optional one-line **Description** shown in the agents list, **What it does** (what to look at, what to compare and what to produce, as you would tell a new colleague), and an optional **Skill to follow**: one of your organization's skills (how-tos) that the agent follows step by step.
- **Systems and actions** tab, **Systems it may read**: pick only what the job needs. The agent cannot read any other source.

- **Systems and actions** tab, **What it may do**: each action is listed explicitly. *Change records* names the system, the object (such as Opportunity), the fields and whether it may create, update or both. *Draft emails* creates drafts in your Gmail or Outlook; nothing is ever sent. *Make a document* adds a file to the results. *Post in Slack or Teams* posts in the channels you list for the action, and nowhere else; each message waits for your approval unless an admin allows posting without approval. Anything not listed is refused.
- **When it runs** tab: only when you press Run, on a schedule (hourly, daily, weekdays, weekly or monthly, at a time in a time zone), a set number of hours before calendar events whose title contains some words, when something happens, or when another system calls it.
- **Results** tab: the **Results project** where each run's summary and files become a chat, **Where it runs** (when your organization has runners) and the outside agents it may ask.
- **Limits and alerts** tab: the safe defaults from your plan, marked recommended (see below).

## Advanced

Settings most agents never need, with safe defaults from your plan:

- **Limits** (on the **Limits and alerts** tab): the most steps (tool calls), records changed and credits per run, and the most runs per day (from midnight UTC).
- **When a run fails** (on the **Limits and alerts** tab): a notice in the app, and optionally an email draft to you in your own mailbox.
- **The query it watches** (on the **When it runs** tab, under **Advanced: the query it watches**): for agents that run when something happens, the query itself (see below).
- For each action, under its own **Advanced**: its **short name** (the id approvals and the audit trail use) and its **description**.

## The agent's page

An agent's own page shows its status, trigger, next run and owner at the top, then tabs: **Setup** (what it reads, may do and produces, its versions, **Change answers**, **Save as template** and **Delete agent**), **Runs**, and, when they apply, **Shadow**, **Connections** (its start address and whether agents on other platforms may call it) and **Sharing** (team agent).

Each save makes a new **version**; every run records which version it used.

Schedules follow your time zone through daylight-saving changes: 08:00 stays 08:00. On the night clocks go forward, a time inside the skipped hour runs at the same minute after the jump (02:30 becomes 03:30).

## When something happens

An agent can watch a system and run once for each new record, such as each new priority 1 case or each opportunity closed won. Choose the system under **In which system** and a ready-made choice under **What starts a run**: *New priority 1 cases* (Salesforce, ServiceNow, Zendesk, Jira) or *Opportunities closed won* (Salesforce, HubSpot, Dynamics 365). For anything else, choose **My own query** and write it under **Advanced: the query it watches**, on the same **When it runs** tab: a query in the system's own language with {since} where the time to look back to goes, for example:

```
SELECT Id, Subject FROM Case WHERE Priority = 'High' AND CreatedDate > {since}
```

SourceLace runs the query as you every few minutes (5 minutes at the most often). Each record it finds starts one run, and the run is told which record it is for. The same record never starts a second run, unless you choose a field such as LastModifiedDate to run again each time it changes. When you turn the agent on, records that already exist are noted and skipped. If many records appear at once, at most a set number of runs start per check and the rest follow at the next check.

The Sev 1 escalation and Closed-won handoff templates use this, with the right query for each system.

## Start from another system (Preview)

Instead of SourceLace checking a system every few minutes, the system itself can start an agent the moment something happens: your CRM or service desk's automation, an event broker, or a script calls the agent's **start address** (a webhook). This is in **Preview**: new, offered for pilots and provided as is.

- The agent runs **as its owner**, with the owner's access, exactly as on any other trigger. A team agent runs once for each subscriber, as them.
- What the other system sends reaches the run as **data about what happened** (such as a case number and its fields), never as instructions: it cannot change what the agent may read or do.
- Shadow mode, approvals and every limit apply as usual. An agent in shadow only does dry runs, whoever starts it.

### Set up its start address

1. Open the agent and choose **Edit**. On the **When it runs** tab, choose **When another system calls it**, and save.
2. On the agent's page, on the **Connections** tab, under **Start address (webhook)**, choose how the other system signs in:
  - **Bearer secret** (the usual choice): the system sends Authorization: Bearer <secret>.
  - **HTTP Basic**: the user name is the hook id shown on the page, and the password is the secret. Choose this for systems that only offer Basic, such as SAP Event Mesh.
3. Optionally turn on **Require a signature on every call**: the system must also sign each call with a second, signing secret (an HMAC-SHA256 of the body, in the X-SourceLace-Signature header).
4. Press **Make its start address**. Copy the **secret** (and signing secret) now: it is shown only once. SourceLace keeps only a fingerprint of it. Store it in the other system's credential store, never in a flow's text.
5. Copy the **address** and use the example call on the page to try it. Turn the agent on (or try it in shadow first).

The page also shows the agent's **name** (such as sev1-escalation), unique in your organization. You can rename it at any time; the start address never changes.

**Rotate secret** makes a new secret at once; the old one stops working. **Turn off** refuses every call until you turn it on again. The agent's owner and your admins can do both at any time.

## What the other system sends

- JSON, at most 256 KB: Content-Type: application/json. CloudEvents 1.0 are accepted too, in structured mode (application/cloudevents+json) or binary mode (ce- headers).
- To avoid starting the same run twice when a system sends again, add an Idempotency-Key header (such as the case number): a second call with the same key within 24 hours starts nothing and gets the first answer again. A CloudEvent's id, or an event\_id or replayId field, works the same way without the header.
- For a team agent, a field "sourcelace\_subscriber": "<email>" runs it for that one subscriber only.
- SourceLace answers 202 with the run's id once the run is queued. The run's result appears on its page, in the agent's project, and as usual in **Approvals** for changes.

Each agent's start address takes at most 60 calls a minute unless your admin chose fewer (**Webhook calls per agent per minute** on the [Limits](#) page); more are answered "try again later".

## Only some events

Under **Only start for some events**, add conditions on the event's fields, such as data.Priority **equals** 1, or Status **is one of** New, Escalated. All of them must match. Other events are answered as filtered and start nothing. Use dots for fields inside fields (data.Account.Name).

## Example: a priority 1 case starts an agent

Your support team wants a briefing as soon as a priority 1 case is created, without waiting for SourceLace's next check.

1. Make an agent such as "Sev 1 briefing": it reads the CRM and the service desk, and drafts an email to the account team. Under **When it runs**, choose **When another system calls it**.
2. Make its start address and copy the secret.
3. In your CRM or service desk's automation, add a step that runs when a case is created with priority 1 and sends an HTTP POST to the start address, with the secret in the Authorization header and the case as JSON:

```
POST <start address>
Authorization: Bearer <secret>
Content-Type: application/json
Idempotency-Key: case-00123
```

```
{"case": "00123", "priority": "1", "account": "Corvanta", "subject": "Checkout down"}
```

4. Turn the agent on (or try it in shadow for a few days). Each priority 1 case now starts one run within seconds, with the case's details.

## Start an agent from a Salesforce Flow

1. In SourceLace, make the agent's start address with **Bearer secret** and copy the address and secret.

2. In Salesforce Setup, create an **External Credential** (authentication protocol **Custom**) with a principal that holds the secret as a parameter, and a **Named Credential** whose URL is the start address. Add a custom header `Authorization` with the value `Bearer` followed by the secret parameter. This keeps the secret in Salesforce's credential store, out of your flows.
3. Give the people or the integration user that run the flow access to the external credential's principal (a permission set).
4. Create a **record-triggered flow** (for example on Case, when it is created and Priority equals 1) and add an **HTTP callout** (or Apex action) that sends a POST to the Named Credential with the record's fields as JSON. Add an `Idempotency-Key` header with the record id.
5. Activate the flow and create a test record. The run appears on the agent's page.

To start the agent from Salesforce platform events instead, send the event's `replayId` in the body: SourceLace uses it to start each event only once.

## Start an agent from SAP business events through SAP Event Mesh

SAP S/4HANA and other SAP systems publish business events (such as a sales order created) to SAP Event Mesh, which delivers them to webhooks as CloudEvents.

1. In SourceLace, make the agent's start address with **HTTP Basic** and copy the address, the hook id (the user name) and the secret (the password).
2. In SAP Event Mesh, create a **queue** subscribed to the topic of the events you want.
3. Create a **webhook subscription** for that queue: the webhook URL is the start address, the content type is `application/json` (or CloudEvents), and authentication is **Basic** with the hook id and the secret.
4. Event Mesh checks the address first with a handshake; SourceLace answers it for a start address that is on. Then start (resume) the subscription.
5. Add conditions in SourceLace if only some events should start the agent, such as type **equals** the event type, or a field under data.

Each event's id makes sure an event delivered twice starts the agent once. SAP Integration Suite, advanced event mesh can deliver the same way, through a REST delivery point to the start address.

## Start from another agent

Another AI agent, such as an assistant in your CRM, can start your SourceLace agents when it is connected to SourceLace and signed in as you. It sees three tools: `list_agents` (the agents you may start), `run_agent` (start one now by its name or id, with optional details as input) and `get_agent_run` (how one of your runs went).

- It can start only what you could start yourself: your own agents, any agent if you are an admin, and team agents you subscribe to. A team agent then runs only your own run, as you.
- Your agent runs with its own instructions and actions. The other agent's input reaches it as data, never as instructions.
- Shadow mode, approvals and every limit apply as usual, and every call is in the audit log.
- It can read only your own runs.

## Work with agents from other platforms (A2A)

Many agent platforms and integration platforms support **A2A** (Agent2Agent), an open standard for agents on different platforms to work together. SourceLace speaks it in both directions: agents elsewhere can ask your SourceLace agents to do work, and your SourceLace agents can ask agents elsewhere for help. This is in **beta**: your SourceLace provider turns it on for your organization, and it is off until then.

The same rules hold as everywhere else in SourceLace:

- **The calling agent signs in as a person.** Before another platform's agent can call yours, someone in your organization signs in to SourceLace from that platform, and every task runs as them, with their access. Nobody gets more than they already have.
- **Approvals stay in SourceLace.** Changes an agent proposes wait on the [Approvals](#) page, as always. An agent on another platform can never approve anything.
- **What arrives from outside is data, never instructions.** It cannot change what your agent may read or do.
- Shadow mode, the trust ramp, limits and your data protection rules apply unchanged, and every call is in the audit log.

## Let other platforms call your agent

1. Open the agent. Under **Callable by other agents**, turn it on. It is off for every agent until its owner, a co-owner or an admin turns it on.
2. Copy the **agent card** address shown there and add it on the other platform. The card tells the other platform the agent's name, its one-line description, and how to sign in. It never includes the agent's instructions.
3. On the other platform, sign in to SourceLace when it asks. From then on, it can send your agent tasks.

Agents that are not callable have no card at all: to anyone outside, they do not exist. Your organization's callable agents are also listed together in one catalog, so a platform can find them all at once.

## What the calling agent sees

The other platform sends a task, such as "Summarize open escalations for Corvanta", and follows it until it is done:

- **Working**, then **Completed** with the run's summary and a link to the run in SourceLace.
- **Needs sign-in** when the agent reads a source the person has not connected yet. The task comes with a link to the **Sources** page; after the person connects, the task carries on.
- **Needs input** when the run proposed changes that wait for approval. The task comes with a link to the **Approvals** page, where the person (or whoever your rules name) decides. Approving from the other platform is not possible.
- **Rejected** when the person may not run the agent, or a limit stops it; **Failed** when the run fails; **Canceled** when the other platform cancels it.

Who may start an agent this way is the same as for **Run now**: its owner, a co-owner or an admin; for a team agent, each subscriber starts only their own run. Each person can make at most 120 of these calls a minute unless your admin chose fewer (**A2A calls per person per minute** on the [Limits](#) page).

## Ask agents on other platforms

Your agents can also ask agents on other platforms, such as a partner's pricing agent or an agent your IT team built elsewhere.

1. An admin opens **Agents**, then **Outside agents**, and adds the other agent by its **agent card** address. SourceLace reads the card and shows the agent's name, what it can do, and how it signs in. Only addresses SourceLace may reach are accepted, with the same rules as REST sources.
2. In an agent's setup, under **Agents on other platforms it may ask**, check the ones this agent may ask. An agent can ask only the outside agents its setup names.
3. While it runs, the agent can send one of them a question and use the answer. The answer reaches it as data, never as instructions.

What your agent sends goes through your organization's data protection rules first, so protected values leave SourceLace as placeholders. **Require approval before sending** is on for every outside agent you add: each message then waits on the **Approvals** page with exactly what would be sent and to whom, and goes out only when someone approves it. Turn it off only for outside agents you trust with your data. An agent in shadow, or doing a dry run, never sends anything.

Example: Corvanta's renewal agent prepares each week's renewal list. Its admin adds the reseller's quoting agent as an outside agent, and the renewal agent asks it for a price on each renewal. Each question waits in Approvals until the renewals manager approves it, so nothing leaves Corvanta unseen.

## How SourceLace signs in to them

For each outside agent, the admin chooses one of:

- **Shared account**: one token or API key for the whole organization. It is saved encrypted and never shown again, to anyone.
- **Personal sign-in**: each person signs in to the other platform once, on the **Outside agents** page, and agents then ask it as the person they run as. The other platform sees only what that person may see there.

Removing an outside agent stops every agent from asking it at once. People who leave your organization are signed out of every outside agent.

## Limits

Each agent has limits that stop a run, or a busy agent, before it does more than you meant. You set them in the agent's settings:

| Setting                   | What it limits                         | Default | Allowed  |
|---------------------------|----------------------------------------|---------|----------|
| <b>Most steps per run</b> | Tool calls in one run, such as queries | 30      | 1 to 100 |

| Setting                             | What it limits                                                       | Default | Allowed            |
|-------------------------------------|----------------------------------------------------------------------|---------|--------------------|
| <b>Most records changed per run</b> | Records one run may propose to change or change                      | 10      | 0 to 1,000         |
| <b>Most credits per run</b>         | AI use and steps of one run together                                 | 400     | 10 to 100,000      |
| <b>Most runs per day</b>            | Runs of this agent in one day                                        | 24      | 1 to 500           |
| <b>Hours before the event</b>       | For calendar agents: how long before each event it runs              | 48      | 1 to 336 (14 days) |
| <b>Check every (minutes)</b>        | For agents that run when something happens: how often the query runs | 15      | 5 to 1,440         |
| <b>Most runs per check</b>          | Runs started by one check; the rest follow at the next               | 5       | 1 to 20            |

A run that reaches a limit stops and says which one; see [When a run fails](#).

To finish sooner, a run does lookups that do not depend on each other (such as reading several objects' fields, or querying two systems) at the same time; changes, drafts and messages wait for them and then happen one at a time, in order. Each lookup still counts as one step. How many run at once is the organization's **Agent lookups at the same time** limit ([Limits](#)).

## Pause or stop

- **Pause** skips the agent's schedule and triggers until it is turned on again. It can still be run by hand (Run now, Dry run).
- **Stop** is for when something is wrong: nothing runs, not even by hand, until its owner turns it on again.

## Runs and results

Each run's page shows why it ran, what it found (the summary), each step (the tool, the source and object, how many rows, and how long it took), the changes it proposed and the credits it used. The steps never include the data the agent read: that stays in your systems.

When a run has something to say, its summary and files also become a chat in the agent's project, so you can ask follow-up questions there.

The **Agents** page lists 50 agents at a time, with a search box (name, description, owner or when it runs) and a **Status** filter. An agent's page lists its newest runs; **Older runs** loads more, and **Result** shows only runs that ended one way, such as failed.

A **dry run** reads and plans like a real run, but changes nothing and drafts nothing: its page shows what it would have done. An agent that is still a draft can only be dry-run.

## Try it in shadow (Preview)

Before an agent works for real, you can let it **try in shadow** for a few days. Shadow mode is in **Preview**: new, offered for pilots and provided as is.

In shadow, the agent runs exactly when it would for real (on its schedule, before your meetings, or when something happens), with your access, but it **never changes, drafts or posts anything**. Everything it would have done is kept for you to look at, on the agent's **Shadow** tab.

Meanwhile you and your team keep doing the work as usual.

Then compare:

- **Check now** (for a change to an existing record): SourceLace looks at the record as it is now and tells you whether it already holds what the agent proposed (**Matches**: someone did the same thing), something else (**Differs**), or no longer exists (**Gone**). It only looks: nothing is written, and only that one word is kept.
- **Agree** or **Disagree** with each thing it proposed, with a note if you like.
- Mark each run **Good** or **Needs work**.

When enough runs are reviewed (3 unless your admin chose more; the tab says "2 of 3 runs reviewed") and the latest one is good, press **Go live**. From then on the agent works for real, with approvals as usual. If you edit the agent while it is in shadow, review a few runs of the new version before it goes live. An admin can let an agent go live sooner, with a reason.

## Days in shadow and approved changes

Your admin can also ask agents to earn trust over time, as many teams do: read-only for a while, then a person approves what it does for a while, before anything happens on its own.

- **Days in shadow.** An agent stays in shadow for at least this many days before it can go live, even once its runs are reviewed. The Shadow tab shows both, for example "2 of 3 runs reviewed · Day 12 of 30", and when it can go live. An admin can still let it go live sooner, with a reason, and that is recorded.
- **Approved changes after going live.** For this many days after an agent goes live, every change and message it makes waits for a person's approval in **Approvals**, even the ones your admin allowed without approval. Email drafts are not affected. The agent's page says "Writes need approval until" a date; after that, changes allowed without approval happen on their own again.

Your admin can also require every new agent to try in shadow first. Then an agent that is already working cannot get a new kind of change (for example, a new field, a new mailbox or a new channel) until it has tried that in shadow too.

Agents in shadow count toward your plan's number of agents that are on.

## Team agents (Preview)

Instead of everyone building their own copy of the same agent, one person can **share** an agent as a **team agent**, and others **subscribe** to it. Team agents are in **Preview**: new, offered for pilots and provided as is.

- **One agent, run for each person.** The agent has one owner and one set of instructions, systems, actions and trigger. Each time it is due, it runs once for each subscriber, **as that subscriber**: with their own sign-ins and their own access, never the owner's. Their results go to their own project, and any drafts and changes waiting for approval are theirs.
- **Your own answers.** The owner can ask each subscriber a few questions, such as "Your accounts" or "Your region". Your answers shape your runs only. You can change them at any time.

- **For the owner.** A team agent's page lists its subscribers and their runs a page at a time; search the subscribers by email or status.
- **Subscribing.** Find shared agents under **Agents → Team agents**, check what they read and may write, and press **Subscribe**. **My subscriptions** lists the ones you have. If an agent needs a system you have not connected, your runs are skipped with a note saying what to connect. **Unsubscribe** stops your runs.
- **For groups.** An admin can subscribe a whole group: the agent then runs for each member. The owner can also keep an agent for some groups only, when your organization uses access by group.
- **Changes.** When the owner edits the agent, everyone gets the change. If an edit lets the agent read a new system or write something new, your runs pause until you accept the new version.
- **Shadow mode** applies to the whole team agent: while it is in shadow, nobody's run changes or drafts anything.
- **For the owner. Share as team agent** is on the agent's page, on the **Sharing** tab. Its team page lists the subscribers, where each one stands (for example "must connect a source") and every subscriber's runs.
- **Co-owners.** The owner can name a few co-owners (up to 5), who can change the agent just like the owner. Co-owners must be allowed to use the systems the agent reads.
- **Handing over.** The owner, or an admin, can hand a team agent to someone else in the organization. That person becomes the owner once they accept, and the agent starts running for them too. Until then nothing changes, and they can also decline.
- **If the owner leaves.** The agent keeps running for its subscribers, as before. If it has a co-owner, the longest-standing one becomes the owner. If not, the agent **needs a new owner**: it keeps running, but nobody can change it until someone accepts it, and your admins are told so they can pick a new owner.
- **Plans and credits.** A team agent counts as one agent toward your plan's number of agents that are on, however many people subscribe. Each run for each person uses credits like any agent run. Your organization sets how many people one team agent may run for (200 unless your admin chose fewer), and the **Usage** page shows each team agent's runs.

## Approvals

Changes to records wait in **Agents → Approvals**. The page shows 50 at a time, with **Show more** for the rest and a search box (agent, source, object, record ID or owner). Each shows the record, every field's current and new value, and why the agent proposed it. **Approve and apply** checks the record again and applies the change with your access at that moment:

- If the record changed since the proposal, the approval is marked *stale* and nothing is written; the next run proposes again.
- If your access no longer allows the change, it fails and nothing is written.
- Approvals expire after 7 days.

An admin can allow some objects to be changed **without approval** (Enterprise plan). Even then, the agent's owner must choose that for the action, the change uses the owner's access, only the

listed fields can be set, and every change is in the audit log.

## Safety

- **An agent never has more access than its owner.** It reads and writes as you, through the same access by group, write switches and audit as when you ask yourself. If you lose access to a source, or leave the organization, your agents pause and say why (a team agent keeps running for its subscribers instead, see Team agents).
- **Data is data, not instructions.** If a record or email the agent reads asks it to do something else, such as "delete all accounts", it is refused: an agent can only use its listed sources and actions.
- **No email is sent.** Email actions create drafts in your mailbox for you to review and send. Slack and Teams messages go only to the channels listed for the action, after your approval unless an admin allowed otherwise.
- **Data protection applies.** Fields your admin hid never reach the AI, and masked values reach it only as placeholders; the agent's drafts and changes get the real values. See [Data protection](#).
- **Limits stop runaway runs.** A run stops at its step, record or credit limit, when your organization's AI use reaches its maximum, or when an admin presses the stop switch.
- **What is kept:** the agent's settings (encrypted with your organization's key), each run's times, counts and steps, and, encrypted, its summary and proposed changes. The rows an agent reads are not stored, but its summaries can quote values and each proposed change holds the record's current and proposed values. Runs and approvals are deleted with the agent and after your organization's chat retention period.

## When a run fails

If a run fails or stops at a limit, you see a notice across the app and under **Needs your attention** on the Agents page, with the reason and a link to the run. You also see a notice when SourceLace paused one of your agents, or when half or more of its runs in the last 7 days did not finish.

In the agent's settings, under **When a run fails**, you can also get an email draft in your own Gmail or Outlook, at most one a day per agent.

## Credits

Agents use AI like a question in the app, plus a small amount per step (each tool call). The **Usage** page counts agent runs with the rest of your organization's AI use. Each agent's page and the Agents list show the credits it used this month.

## For admins

**Agents → Settings** (admins only):

- **Stop all agents.** One switch stops every agent at once; runs in progress stop at their next step.
- **Who can create agents:** everyone (the default), people in some groups, or admins only.

- **Changes without approval:** the objects agents may change without asking (Enterprise plan). For Slack or Teams posts, allow the object messages of that source.
- **Every agent runs on one of our runners** (when runners are on): agents are saved and turned on only when they name one of your runners. See [Runners](#).
- **New agents try in shadow first** (Preview, when shadow mode is on for your organization): no agent is turned on until people reviewed its shadow runs, and a working agent cannot get a new kind of change without trying it in shadow. See [Try it in shadow](#). Under **Limits** you can ask for more reviewed runs than SourceLace's 3, never fewer.
- **Days in shadow before going live** and **Days of approved changes after going live** (Preview, with shadow mode): how long agents stay in shadow before they go live, and how long a person approves every change and message of an agent that has just gone live. Both are 0 (off) unless you set them; 30 and 30 is a common choice. See [Days in shadow and approved changes](#).

**Agents → Outside agents** (admins, when A2A is on): the agents on other platforms your agents may ask, how SourceLace signs in to each, and whether each message waits for approval. See [Ask agents on other platforms](#).

Admins see every agent in the organization, and **Agents → Overview** shows them all with their owner, runs and failures in the last 7 days, credits and waiting approvals, the most failing first. Admins can pause, stop or delete any agent, and dry-run or run it; a run still acts as the agent's owner, with the owner's access. Only the owner can edit an agent, and the co-owners of a team agent. Creating, editing, turning on and off, each run and each approval are in the [audit log](#).

Self-hosted servers run agents in the same process as the app. A Super Admin can turn this off under **Platform settings → Switches (Run AI agents on this server)**; it takes effect at the next restart, and the saved value applies to every server). To run them on some servers only, set `SOURCELACE_AGENT_SCHEDULER=false` in the environment of the others, which wins over the app's value; two servers never run the same run.