Which systems can send events to SourceLace

From the SourceLace manual: An event is a short message a system sends the moment something happens in it: a case is created, a payment arrives, a file is shared. An agent can start from an event instead of waiting for its next scheduled time, so it acts within seconds. This page lists which of your

SourceLace version 0.1.0 · Made November 22, 2026
The latest version is at https://sourcelace.ai/docs

Which systems can send events to SourceLace

An **event** is a short message a system sends the moment something happens in it: a case is created, a payment arrives, a file is shared. An agent can start from an event instead of waiting for its next scheduled time, so it acts within seconds. This page lists which of your systems can do that, how each one is connected, and what to use when a system cannot send events at all.

There are four ways to start an agent from a system:

Way	In plain words	Use it for
Live streams	SourceLace stays connected to the system and hears each event as it happens.	Salesforce, MongoDB, Gmail, message queues, your own connector
Webhook presets	The other system calls the agent's **start address** (a web address, also called a webhook) when something happens. SourceLace knows how that system proves who it is.	The systems in the list below
The general webhook	The same start address, for any system that can call a web address and send a secret.	Anything not in the list
Scheduled checks	SourceLace asks the system every few minutes what is new.	Systems that cannot send events

Whichever way starts an agent, the rest is the same: the agent runs as its owner, with the owner's access, and conditions, duplicate protection, limits, shadow mode, approvals and the audit trail all apply.

Live streams

SourceLace keeps one connection open for the agents that listen to the same source, and picks up exactly where it stopped after a restart, so nothing is lost while SourceLace updates. Nothing is checked on a timer: the connection waits for the system to speak.

- **Salesforce:** platform events, platform event channels and change data capture (a record created, changed or deleted). See Start when Salesforce sends an event.
- **MongoDB:** a document in a collection is added, changed or deleted. See Start when a MongoDB document changes.
- **Gmail:** a new email arrives in your inbox or a label. See Start when an email arrives in Gmail.
- **Message queues:** Amazon SQS, Google Pub/Sub, Apache Kafka, and brokers that speak AMQP 1.0 (Azure Service Bus, ActiveMQ and Artemis, Solace, RabbitMQ 4, IBM MQ and most JMS brokers). A queue inside your own network, such as IBM MQ, is read through the on-site SourceLace Agent, with no inbound port opened. See Start agents from your message queues.
- **Your own connector:** a connector your developers build with the Connector SDK can send events. See Start when your own connector sends an event.

Webhook presets

A **preset** is SourceLace knowing how one system proves it is really the sender: which signature or token to check, how to answer its first "is this address alive?" call, and how to avoid starting a run twice for an event sent twice. On the agent's **When it runs** tab, under **Who calls it**, choose the system. Under **Connect a system**, its guide gives numbered steps with your agent's start address filled in.

- **Microsoft 365, Teams and SharePoint** (through Microsoft Graph)
- **Zendesk**
- **Jira**
- **GitHub**
- **Stripe**
- **DocuSign Connect**
- **Okta event hooks**
- **Gong**
- **Slack** events, and **Slack commands** (a person types a command)
- **Microsoft Teams outgoing webhooks**
- **Salesforce Data 360** data actions
- **Marketing Cloud**
- **HubSpot workflows**
- **ServiceNow** flows and business rules
- **SAP Event Mesh**, which carries events from SAP S/4HANA, ECC and SuccessFactors
- **Amazon SNS**, which carries events from Amazon S3 and EventBridge
- **Azure Event Grid**
- **Google Cloud Pub/Sub push**, and Google Drive and Calendar push notifications

Details for each: Who calls it and Connect a system.

The general webhook

Any system that can call a web address and send a header can start an agent, even if it is not in the list. Leave **Who calls it** on **Any system**: SourceLace makes a start address and a secret, and the other system sends the secret as `Authorization: Bearer <secret>` (or as HTTP Basic, where it cannot send a bearer token). The guide **Any other system** has the steps. A system that sends the same event twice starts one run when it sends an `Idempotency-Key` header, a CloudEvent id, or an `event_id` field.

Systems that need scheduled checks

Some systems cannot send events: data warehouses such as Snowflake, BigQuery, Databricks, Redshift and Fabric, and some ERP and business systems. For these, give the agent a query and SourceLace runs it every few minutes (5 at the most often, 15 unless you change it, up to once a day): either "what is new since the last check" or "which records meet this condition now". Each record found starts one run, or one run covers the whole list. See When something happens and When a record meets a condition.

Snowflake and Databricks can also call a start address from their own alerts, so check **Connect a system** first if you want the faster way.

Why these checks are cheap

- **A check is not an agent run.** It is one read-only query, run as the agent's owner. No AI is used and no credits are spent unless the query finds something that starts a run.
- **It only looks at what is new.** The query uses the time of the last check, so the system returns the few records that changed since, not everything.
- **A quiet check ends quickly.** When nothing is found, the check does nothing and the next one is queued. A check that failed is retried later with growing pauses, so a system that is down is not hammered.
- **SourceLace keeps no copy.** It remembers only which records already started a run.
- **You choose how often.** A contract renewal list needs one check a day, not one every five minutes.

Your own system still answers each check, so ask it for only the records and fields the agent needs. For a data warehouse, ask your warehouse admin how its query use is billed.

Choosing between them

1. If the system is in the **live streams** list, use that: it is the fastest and needs no address on your side.
2. Else if it is in the **presets** list, use its preset and guide.
3. Else if it can call a web address, use the **general webhook**.
4. Else use a **scheduled check**.

When you only want to notify an agent that something happened, a webhook is enough. When one agent should hand another agent a task and wait for an answer, see Work with agents from other platforms (A2A).

When something goes wrong

What you see	What to do
The agent does not start when the system sends an event.	Open the agent's **Recent calls**. A refused call says why (wrong or missing secret or signature, a condition that did not match, the daily run limit). See Recent calls.
A system says the address did not answer its first check.	The agent must be on, and **Who calls it** must match that system. Some systems check the address before they use it.
Events stop arriving from a live stream.	The agent's page says why: usually a sign-in that needs renewing. Sign in to the source again.
A scheduled check finds nothing you expected.	Run the query in the system itself, and check that its time condition uses the time of the last check.