Healthcare claims: Stedi, Availity and Optum

From the SourceLace manual: How to connect a healthcare clearinghouse, so people and agents can ask a payer (a health plan) three questions: is this patient covered, where is this claim, and what was paid and why. A clearinghouse is a service that passes these standard questions and answers between

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

Healthcare claims: Stedi, Availity and Optum

How to connect a healthcare clearinghouse, so people and agents can ask a payer (a health plan) three questions: **is this patient covered**, **where is this claim**, and **what was paid and why**. A clearinghouse is a service that passes these standard questions and answers between providers and payers. SourceLace connects to three clearinghouses: **Stedi**, **Availity** and **Optum** (Optum's medical network APIs, formerly Change Healthcare). Connect the one your organization already uses.

Status:

- **Stedi: eligibility tested; claim status and remittance in testing.** Eligibility has been checked against a real Stedi test account. Claim status and remittance are covered by automated tests against a simulated version of Stedi's API, because Stedi answers them only for production accounts.
- **Availity: eligibility and claim status tested on Availity's Demo plan.** New. Built on Availity's published API documentation and covered by automated tests against a simulated version of it, then checked against a real Availity Demo plan account (Availity's ready-made test answers). Not yet tried against a Standard plan account.
- **Optum: eligibility, claim status and remittance tested in Optum's sandbox.** New. Built on Optum's published API documentation, covered by automated tests against a simulated version of it, and checked against Optum's real sandbox (ready-made test answers and sample remittance files). Not yet tried against an Optum production account.

Anything not yet tried against a real account is checked together with you on your first connection. Where a clearinghouse's own documentation does not say something, this page says so.

At a glance

Source	Kind	Reads	Changes	People sign in with	First test question
Stedi	`claims`	Eligibility and benefits, claim status, remittance (what was paid)	None (read-only)	One **shared** Stedi API key an admin makes	"Which questions does claims:stedi answer?"
Availity	`claims`	Eligibility and benefits, claim status	None (read-only)	One **shared** Availity app (client ID and secret)	"Which questions does claims:availity answer?"
Optum	`claims`	Eligibility and benefits, claim status, remittance (what was paid)	None (read-only)	One **shared** Optum client ID and secret	"Which questions does claims:optum answer?"

It is shared access. None of the three clearinghouses signs each person in separately, so everyone who may use the source acts as the one credential SourceLace was given. Narrow who can use the source with access by group. Keys and secrets are kept encrypted and never shown again.

SourceLace only reads. It asks eligibility, claim status and remittance questions. It never submits a claim and never changes anything at the clearinghouse or a payer. With Optum, it reads

remittance files from your report mailbox and never deletes them.

Before you connect: test data only

Everything a payer answers about a patient is **protected health information (PHI)**.

SourceLace has no HIPAA business associate agreement program yet, and does not claim HIPAA compliance. Until a business associate agreement (BAA) is signed between your organization and SourceLace, and SourceLace has matching agreements with the services it uses (hosting and the AI model provider), **use test data only**: a Stedi **test** API key with Stedi's listed test requests, an Availity app on Availity's **Demo** plan, or Optum's **sandbox**. Your agreement with your clearinghouse (Stedi publishes its own BAA; ask Availity or Optum about theirs) is a separate agreement between your organization and that clearinghouse. A qualified person at your organization decides whether and when real patient data may be used. See Security and privacy.

Patient details are masked by default

Patient names, member ids, dates of birth (all but the year) and patient account numbers are **masked for everyone** who uses the source. Payer and plan names, codes, dates of service and amounts stay readable, so questions such as "how much did this payer deny last month, and why" still work. An admin can turn masking off for the source with **Mask patient details**; your organization's data protection rules apply on top either way.

The audit trail keeps only which question was asked (eligibility, claim status or remittance) and the payer id, never the patient details that were typed in. Results are kept for at most 5 minutes.

What people can do

`search_schema` lists the three questions, and `describe_object` shows each one's columns. `query` (language `claims_request`) asks one question, written as a small JSON object:

Question	Example	Notes
Is this patient covered? (eligibility)	`{"transaction": "eligibility", "payer": "61101", "provider": {"npi": "1999999984", "organization_name": "Provider Name"}, "member": {"member_id": "HUMANA123", "first_name": "Jane", "last_name": "Doe", "date_of_birth": "1975-05-05"}, "service_types": ["30"]}`	One row per benefit: coverage status, plan, copay, coinsurance, deductible, out-of-pocket, network, time period. `service_types` are standard codes (30 is general medical care). Needs `member_id`, or `last_name` and `date_of_birth`.
Where is this claim? (claim status)	`{"transaction": "claim_status", "payer": "87726", "provider": {"npi": "1999999984", "organization_name": "Provider Name"}, "member": {"member_id": "...", "first_name": "...", "last_name": "..."}, "service_from": "2026-09-01", "service_to": "2026-09-30"}`	One row per claim the payer found: status category (such as F1 finalized/payment, F2 finalized/denial), status, billed and paid amounts, payment date and check number.

Question	Example	Notes
What was paid, and why? (remittance)	`{"transaction": "remittance", "received_from": "2026-10-01", "received_to": "2026-10-15"}`	Stedi and Optum. The remittance advice (ERA) received for you in those dates (at most 31 days at a time, and at most 50 advices per question): one row per claim, with the payment, billed, paid, patient responsibility and adjustment reasons such as `CO-45 120.00`. Or `{"transaction": "remittance", "era_id": "..."}` for one advice: with Optum, the `era_id` is the remittance file's name, such as `R5123456.AB`. Optum keeps remittance files in your report mailbox for up to 30 days.

An NPI (the provider's National Provider Identifier) is checked before anything is sent, so a mistyped one is caught at once.

What each clearinghouse answers:

	Eligibility	Claim status	Remittance
Stedi	Yes	Yes	Yes
Availity	Yes, one service type per question	Yes	Not yet: Availity's remittance API is still in beta
Optum	Yes	Yes, with the provider's `tax_id` and the patient's `date_of_birth`	Yes, from your Optum report mailbox

Optum needs a little more. For claim status, Optum needs the provider's tax ID (EIN) and the patient's date of birth: add `"tax_id": "12-3456789"` to `provider`, and `date_of_birth` and `gender` to `member` (or `date_of_birth` to `dependent`). SourceLace asks for them before sending anything. For eligibility, Optum needs a date of service: when the question gives none, SourceLace sends today's date.

Every clearinghouse's answers come back in the same columns, so a question, an agent or a report works the same whichever one you connect. Availity answers some questions after a short wait while it asks the payer; SourceLace waits up to about a minute, then tells you to ask again.

With a Stedi test key: eligibility answers come back for Stedi's listed test requests only (the eligibility example above is one of them). Stedi does not answer claim status with a test key, and remittance advice exists only for claims a production account has sent, so those two need a production key and a signed BAA.

On Availity's Demo plan: Availity answers with ready-made test answers, not real payers. The Demo plan has a small daily allowance (see Availity's plan details), and Availity counts every request against it, including each sign-in and each time SourceLace asks whether an answer is ready.

In Optum's sandbox: Optum answers only requests that use its own predefined test values (payer ids such as `00001`, NPIs such as `1760854442`, names such as `johnone doeone`), with ready-made answers.

Remittance in one question across systems

One remittance advice can be a table in one question across your systems, for example to put what a payer paid next to your own claim records. Name the table as `"claims:stedi"."remittance"` and give the advice's id in the question: `WHERE "era_id" = '<the advice id>'`. Three things to know:

- **Only a remittance, and only one advice at a time.** Eligibility and claim status answer one question about one patient, so they cannot be a table; ask them on their own. To find the ids first, ask for the remittance on its own with a date range, then use the result in the question across systems.
- **Patient details are protected** in a question across systems: ask for the payer, amounts, codes and dates by name instead of `*`.
- **Your clearinghouse must offer remittance** (the table above).

Every answer that read claims records carries the note "The text below is written by the assistant from health or claims records and is not checked; check the records." by design, so it never shows "all clear"; if the line cannot be built it says "Answer check: SourceLace could not check this answer" (see The line under an answer).

Ready-made agents

The **Healthcare claims and revenue cycle** pack (see Industry packs) sets up three read-only agents on this source: a weekly denial work list, a remittance reconciliation and a payer scorecard. They read remittance advice only, so they need a Stedi or Optum source, and a production account and a signed BAA before they have anything to read: with a test key or sandbox they report that no remittance arrived. Availity sources do not answer remittance yet. A qualified person on your team makes every billing, appeal and patient-balance decision.

What your admin does in Stedi

Once. About 10 minutes.

Who: someone with the **Admin** or **Developer** role in your organization's Stedi account. Stedi's **Operator** and **Read-only** roles cannot make API keys.

1. **Open the API keys page.** Sign in to the Stedi portal at portal.stedi.com, open your account settings, then **Developer settings**, then **API keys**.
2. **Make a test key.** Click **Generate new API Key**, name it `SourceLace (test)`, choose **Test**, and click **Generate**. Copy the key: Stedi shows it only once. Keep it for your SourceLace admin, and do not paste it into email or chat.
3. **Later, for real patients only after a BAA:** make a **Production** key the same way. A production key has the permissions of the person who made it, and keeps them even if that person's role changes later; Stedi does not offer a key limited to reading. Make it from an account whose role suits that.
4. **Enroll for remittance, if you will ask "what was paid".** Payers send remittance advice to one clearinghouse at a time, and many require enrollment first. In Stedi's payer list, a payer marked **Enrollment required** for remittance (ERA) needs that enrollment, which you request through Stedi. SourceLace cannot do this step for you.
5. **Give your SourceLace admin** the key. They add the source below.

Check that it worked: ask "Which questions does claims:stedi answer?" (you should see `eligibility`, `claim_status` and `remittance`), then, with a test key, ask the eligibility example above exactly as written. You should see rows with an active coverage status. If you see "Stedi refused the sign-in", look it up under When something goes wrong.

Stedi's own guides: Stedi healthcare documentation (see "Test mode", "Account settings" and "Supported payers").

What your admin does in Availity

Once. About 20 minutes, plus the time Availity takes to approve the request.

Who: someone at your organization who has, or creates, a user in Availity's developer portal (developer.availity.com) and belongs to your organization there. To be added to an existing organization in the portal, Availity asks for a support ticket naming the user, the organization and its Organization Admin. Availity does not publish other role rules: if a step below is missing for you, ask Availity Client Services.

1. **Sign in to Availity's developer portal.** Go to developer.availity.com and sign in, or click **Create account**, then **Sign Up**, enter the code Availity emails you, and set up multi-factor authentication.
2. **Create an app for SourceLace.** Open **My Apps**, click **Create a New App**, name it `SourceLace`, and click **Create App**.
3. **Subscribe the app to the HIPAA transaction APIs.** Open **API Products**, find the healthcare (HIPAA) transactions product that includes **Coverages** and **Claim Statuses**, click **More Info**, then **Access this Product**. For testing choose the **Demo** plan (free, approved automatically, ready-made test answers, no patient data). Click **Access with this Plan**, **Go to cart**, choose the `SourceLace` app, and click **Submit Request**.
4. **Later, for real patients only after a BAA:** request the **Standard** plan the same way. Availity asks you to contact its sales team (availity.com/contact-sales) and approves the plan once a contract is in place. SourceLace states no Availity prices or terms; ask Availity.
5. **Copy the client ID and secret.** Open **My Apps**, then **Approved Access**, open the product, and expand **Client info**. Copy the client ID (Availity calls it the API key) and the client secret. Availity may show the secret only once: if it is lost, open the app and click **Reset** under **Client Secret**, which stops the old one working.
6. **Note the scope.** In the product's details, look for the scope. Availity's guide uses `hipaa`; a new subscription may show two scopes. If you see anything other than `hipaa`, give it to your SourceLace admin exactly as shown.
7. **Give your SourceLace admin** the client ID, the secret and the scope (if not `hipaa`). Do not paste the secret into email or chat. They add the source below.

Payers decide which details they need for each question. If one is missing, the answer says so in the payer's words.

Check that it worked: ask "Which questions does claims:availity answer?" (you should see `eligibility` and `claim_status`), then the eligibility example above with your Demo plan. You should see rows with a coverage status. Demo answers are Availity's ready-made test answers, so they need not match the patient you asked about; that is expected. If you see "Availity refused SourceLace's sign-in", look it up under When something goes wrong.

Availity's own guides: Availity developer portal ("Getting started" and the "Availity API Guide").

What your admin does in Optum

Once. About 15 minutes, plus the time Optum takes to send credentials.

Who: someone who can request API access for your organization from Optum. For production, that is usually whoever holds your organization's contract with Optum.

1. **Request sandbox access.** Go to Optum's API sandbox access page (marketplace.optum.com, "API sandbox access"), choose the medical network (eligibility and claims) APIs, fill in your work email, name, company, role and headquarters state, and submit.
2. **Receive the sandbox client ID and secret.** Optum sends your organization a client ID and secret for its sandbox. Optum's pages do not say exactly how they arrive: if nothing comes, ask your Optum contact.
3. **Later, for real patients only after a BAA:** production needs a contract or subscription with Optum, through your Optum account, Optum's sales team or your Optum representative. Optum then gives a **separate** production client ID and secret. SourceLace states no Optum prices or terms; ask Optum.
4. **For remittance, ask about the report mailbox.** Optum keeps payers' remittance files in a report mailbox, for at most 30 days. Optum's guide says reading it needs a mailbox user name and password from your Optum implementation analyst. In Optum's sandbox, SourceLace read the sample files without them; for production, ask your Optum representative whether your mailbox needs them.
5. **Give your SourceLace admin** the client ID and secret, which environment they are for (sandbox or production), and the mailbox user name and password if you have them. Do not paste secrets into email or chat. They add the source below.

Check that it worked: ask "Which questions does claims:optum answer?" (you should see `eligibility`, `claim_status` and `remittance`), then, in the sandbox, ask with Optum's own test values, for example `{"transaction": "eligibility", "payer": "00001", "provider": {"npi": "1760854442", "organization_name": "happy doctors group"}, "member": {"member_id": "0000000001", "first_name": "johnone", "last_name": "doeone", "date_of_birth": "1880-01-02"}}`. For "what was paid", ask `{"transaction": "remittance", "era_id": "R5000000.WA"}`, one of the sandbox's sample files. Which combination of Optum's test values gives which ready-made answer is Optum's to say: if the sandbox refuses a question, try other values from Optum's list. If you see "Optum refused SourceLace's sign-in", look it up under When something goes wrong.

Optum's own guides: Optum developer portal (Eligibility and claims: "API environments", "Sandbox predefined fields and values" and "Claim responses and reports").

Add the source (SourceLace admin)

Kind: **Healthcare claims (clearinghouse)** (`claims`): pick **Stedi**, **Availity** or **Optum** on the **Choose the system** step of **Add a source**. There is no address to enter: each clearinghouse has fixed API addresses, which SourceLace already knows.

Option	Type	Default	Example	What it is
`vendor`	Choice	`stedi`	`availity`	**Clearinghouse**: Stedi, Availity or Optum. The picker sets it.

Option	Type	Default	Example	What it is
`secret`	Secret	(required)		Stedi: the **Stedi API key**. Availity and Optum: the **client secret**. Stored encrypted and never shown again. Leave it empty when editing to keep the saved one.
`client_id`	Text	(required for Availity and Optum)		**Client ID** of your Availity app or from Optum.
`scope`	Text	`hipaa`	`hipaa`	Availity only. **Scope**: leave empty unless your subscription shows another scope; two scopes are separated by a space.
`environment`	Choice	sandbox	`production`	Optum only. **Optum environment**: sandbox or production. Must match the client ID and secret.
`reports_username`	Text	(empty)		Optum only, optional. **Report mailbox user name**, for remittance.
`reports_password`	Secret	(empty)		Optum only, optional. **Report mailbox password**. Fill in both or neither.
`protect_patient_identifiers`	On or off	on	`off`	**Mask patient details.** On masks patient names, member ids, dates of birth (but the year) and account numbers for everyone.

When something goes wrong

What you see	What it means	What to do
"Stedi refused the sign-in. ..." (HTTP 401 or 403)	The key is wrong, was deleted, or belongs to another Stedi account.	An Admin or Developer checks the key in Stedi (step 2), then you enter the current key on the source.
"The payer id is not valid for this provider or this clearinghouse. ..." (payer code 79)	The payer id is wrong, or your clearinghouse does not reach that payer.	Look the payer up in your clearinghouse's payer list and use its payer id, keeping leading zeros.
"The payer's system is not answering right now. ..." (payer code 42 or 80)	The payer is down or slow.	Ask again in a few minutes.
"The payer does not recognize the member id. ..." or "... did not find this member. ..." (payer codes 72, 73, 75)	The member details do not match the payer's record.	Check the member id, name and date of birth against the member's card.
"The payer does not know this provider (NPI). ..." or "The provider is not authorized to ask this payer. ..." (payer codes 43, 41)	The NPI is wrong, or the provider is not enrolled with that payer for this question.	Check the NPI, and the provider's enrollment with the payer through your clearinghouse.
"The member's date of birth does not match the payer's record." (payer code 71)	The date of birth differs from the payer's.	Check it against the member's card.

What you see	What it means	What to do
"A remittance in ... is read one remittance advice at a time: add era_id = ..."	A question across systems named the remittance table without an advice id, or with several.	Add `WHERE "era_id" = '<the advice id>'`, or ask for the remittance on its own with a date range to find the ids.
"... cannot be read as a table in a query across sources: it answers one question about one patient ..."	A question across systems named eligibility or claim status.	Ask it on its own, then use its result in the question.
"... are protected patient columns, so they cannot be used in a query across sources. Name the other columns instead of using *."	A question across systems asked for a patient column (or *).	Name the other columns (payer, amounts, codes, dates) instead.
"The claims request is not valid: ..."	A key, value or date SourceLace does not accept, such as an NPI that fails its check digit.	Fix what the message names. The AI usually fixes the request itself.
"The payer answered with no coverage details and no reason."	The payer returned an empty answer.	Check the member id and date of birth, and try again later.
"SourceLace refused to make that call to the clearinghouse." or "SourceLace refused to send the clearinghouse credential to another address."	A link pointed outside the clearinghouse's address, and SourceLace never sends the credential anywhere else.	Try again. If it repeats, contact support.
"... answered with something other than data (HTTP ...)."	The clearinghouse had a problem, or something answered in its place.	Try again shortly. If it repeats, contact support.
"Stedi is limiting how fast SourceLace may call it. ..." (or Availity, or Optum; HTTP 429)	Too many questions for your clearinghouse account. On Availity's Demo plan, the daily allowance may be used up: it starts again at midnight UTC.	Wait a minute and ask again, or ask for less at once.
"The payer or Stedi did not answer in time. ..." (or Availity, or Optum; HTTP 500, 502, 503 or 504)	The payer took too long, or the clearinghouse had a problem.	Try again in a few minutes.
"Stedi did not send this check to the payer: CMS (Medicare) accepts requests from US addresses only." (HTTP 451)	Medicare's system refused the request's origin.	Tell support@sourcelace.com.
"Stedi does not have that remittance advice, or this account cannot see it." (HTTP 404)	The `era_id` is wrong or belongs to another Stedi account.	Ask for remittance by dates instead.
No rows for remittance	No remittance advice arrived in those dates, or the payer is not enrolled to send it to Stedi.	Check enrollment (step 4); a test key has no remittance advice.
"... needs the option secret ..."	The Stedi API key, or the Availity or Optum client secret, is missing.	Enter it.
"... needs the option client_id ..."	The Availity or Optum client ID is missing.	Enter it (Availity step 5; Optum step 2).

Availity:

What you see	What it means	What to do
"Availity refused SourceLace's sign-in with this source's client ID and secret (invalid_client). ..."	The client ID or secret is wrong or was reset, or the scope is not the one your subscription allows.	Check them under My Apps, Approved Access, Client info (step 5) and the scope (step 6), then enter them again.

What you see	What it means	What to do
"Availity refused the sign-in. ..." (HTTP 401)	The app's access changed or was revoked.	Check that the subscription is still approved (step 3), and the client ID and secret (step 5).
"Availity refused this question: your app's subscription does not include this API ..." (HTTP 403)	The app's subscription does not cover the API for that question.	Check the app's product and plan under My Apps, Approved Access (step 3); if the product lists the API, ask Availity Client Services.
"The payer could not use the request: change what it names. Availity said: ..."	The payer needs a detail the question did not give, or one is in the wrong form.	Add or fix what Availity names, such as the member's date of birth.
"The payer rejected the request." or "The payer did not answer." or "... down for maintenance ..."	Availity reached the payer, which refused, did not answer, or is down.	Try again later; if it keeps happening, check the provider's enrollment with the payer.
"Availity is still waiting for the payer's answer (still processing). ..."	The payer is slow; SourceLace stopped waiting after about a minute.	Ask again in a minute or two.
"Availity no longer has that answer (answers expire). ..." (HTTP 404)	Availity removed the answer before SourceLace read it.	Ask the question again.
"Availity checks one service type per eligibility question ..."	The question asked for several `service_types`.	Ask once per service type.
"Availity did not say where to find its answer. Try again in a few minutes."	Availity accepted the question but did not say where the answer will be.	Ask again in a few minutes.
"SourceLace does not read remittance through Availity yet ..."	Availity's remittance API is still in beta.	Ask about eligibility or claim status, or use Stedi or Optum for remittance.

Optum:

What you see	What it means	What to do
"Optum refused SourceLace's sign-in with this source's client ID and secret ..."	The client ID or secret is wrong, or they are for the other environment.	Check them, and that **Optum environment** matches them (sandbox and production have separate ones).
"Optum refused the sign-in, or this account may not use that API. ..." (HTTP 401 or 403)	Your Optum subscription does not include this API, or for remittance the mailbox user name or password is wrong.	Check with your Optum representative; enter the mailbox credentials again.
"Optum is still preparing a large remittance file. ..." (HTTP 413)	Optum prepares large files before they can be read.	Ask again in 5 minutes, not sooner.
"Optum does not have that remittance file, or it was deleted from the mailbox." (HTTP 404)	The file name is wrong, or the file is gone (files stay at most 30 days).	Ask for remittance by dates instead.
"Optum's claim status needs the provider's tax_id (its EIN) and the patient's date_of_birth ..."	Optum needs both for claim status.	Add them to the question.
"Optum refused the request: <field>: <what Optum said>. ..." (HTTP 400)	Optum found a detail missing or in the wrong form.	Change what it names and ask again.
"That is not an Optum remittance file name ..."	`era_id` is not in Optum's form, such as `R5123456.AB`.	Copy the name from an earlier answer's `era_id` column.
"Optum refused the remittance file name. Ask for remittance by dates instead."	Optum did not accept the file name.	Ask for the remittance by dates.

What you see	What it means	What to do
"The payer could not answer this question. (payer code ...)"	The payer refused the request with a code SourceLace has no words for yet.	Read the payer's own text after the code, and fix what it names.
No rows for remittance	No remittance file in your mailbox was made in those dates, or none has arrived yet.	Check the dates; in the sandbox, only Optum's sample files exist.