Sign Your People In From Your Own Portal

Last Modified on 09/04/2026 11:09 am EDT

Put a Pipeline icon on your intranet and one click takes an agent straight into their account, already signed in. Your developer builds the link; we issue the keys and the test user.

Introduction

If your brokerage runs its own portal (an intranet, an agent dashboard, a franchise platform), single sign-on is how Pipeline stops being a separate password. Your system vouches for who someone is, Pipeline trusts the vouching, and the login screen never appears.

Getting there is a small project rather than a setting, and it runs in a fixed order: confirm the fit, get a master admin to authorize the work, build and prove it in a development environment, then go to production. The whole thing costs nothing on our side. What it costs is developer time on yours, because writing the signed link is work only your developer can do.

When to Use This

- You run an agent portal or intranet and want Pipeline behind one click instead of a second password.
- Your IT vendor is asking for SAML metadata, an OAuth application, or an Okta connector.
- Someone told you to "get an API key" and you don't know who's allowed to ask for one.
- You're opening a second Pipeline account and want your existing sign-in to cover it.
- Your developer needs somewhere real to test before anything touches your live account.

Why This Isn't the SSO Your IT Team Is Picturing

Nearly everyone calls this SSO, and the word carries expectations we don't meet. Pipeline's is a signed-link API we built ourselves: your server constructs a URL carrying a user ID, a timestamp, and a hash, and we validate it. There's no SAML metadata to exchange, no OAuth consent screen, no identity-provider application to install, and no way for your platform to end a Pipeline session it started.

That matters most in the first conversation with an IT vendor, because a vendor expecting to point an identity provider at us will spend a week looking for something that doesn't exist. Send them our documentation early. We also don't build platform-specific integrations — there's no Moxi connector or Delta connector waiting to be switched on; there's one published method, and your developer implements it.

1. Check the Fit Before You Scope the Work

Read the mechanism first, because it decides whether your platform can do this at all. Your system needs to be able to generate a URL server-side containing the person's Pipeline User ID, a timestamp in ISO 8601 format, and an HMAC-SHA256 hash of those values signed with your account's secret key. If your portal is a page builder with no server-side code, that's the blocker to find now rather than after you've requested keys.

It's also worth setting expectations about scope. The API does single sign-on and nothing else — there's no

companion API for pulling transactions, creating users, or uploading documents. Sign your people in through this; move data through Zapier.

Sign your people into Pipeline straight from your own portal, with no second password to type. What Pipeline offers is a signed-link API we built ourselves — not SAML, and not OAuth.

See Single Sign-On

2. Have a Master Admin Authorize the Work in Writing

We release keys only after a master admin on the account asks us to, in writing — and that includes naming any outside developer or IT vendor who'll be working on it. Until that authorization exists we won't discuss keys or setup details with them, even if they already know your account name and can describe the project accurately.

The master admin sits at the top of your account — full access to every location, every transaction, and the settings no one else can touch.

See Master Admin

> Any master admin on the account can give that authorization. If your accounts are linked under an Enterprise, a designated Enterprise Admin is normally the authorized contact for key requests instead.

3. Get the Development Key and Test User

Setup runs against a development environment before it touches anything real. Our tech team issues a development secret plus a test user — name, email address, and Pipeline User ID — for dev.paperlesspipeline.com, so your developer has something to build against without an agent's live account in the loop. The production key follows once you tell us the development test worked.

Both keys are free. If there's a cost anywhere in this project, it's your developer's or your IT vendor's, not ours.

> The production secret is a password in every sense that matters — anything holding it can sign someone into your account. Keep it server-side, never in browser code, and treat a leak the way you'd treat a leaked admin password.

4. Build and Prove It in the Development Environment

Your developer's first working link is usually the second or third one they build, and the failure looks the same every time: the person lands on the ordinary login page instead of being signed in. That means Pipeline couldn't validate the link, and there are only a handful of reasons why — an expired or wrong test user ID, a timestamp that isn't UTC in ISO 8601 format, an `http` URL where it needs `https`, or a hash calculated over the wrong values.

An SSO link that drops someone on the login page means Pipeline couldn't validate it. These are the usual reasons, in the order worth checking.

See SSO Sign-In Lands on the Login Page

> One that catches people writing the link inside a templating system: the ampersands separating the URL parameters get HTML-encoded to `&amp;` on the way out, and the link arrives malformed. Check the URL as it's actually sent, not as it looks in your template.

5. Collect Your People's Pipeline User IDs

Every link is built for one person, from their Pipeline User ID — not their email address. Your portal needs a way to store that ID against each person, so gathering the list is real work worth scheduling rather than an afterthought. The IDs appear on user profiles once we turn SSO on for your account.

The Manage Users page lists everyone on your account — the active roster by default, with each person's record a click away.

See Users List

> Send that list to your developer yourself. We won't forward a roster of your people to an outside vendor, however thoroughly they've been authorized to work on the integration — the names and email addresses are yours to share.

6. Go to Production, Then Extend to Your Other Accounts

Tell us the development test succeeded and we'll issue the production key. Point your portal at the live URL, sign yourself in first, then roll out to a small group before the whole roster.

If your company runs more than one Pipeline account, you don't start over for each one. The same key can cover accounts in the same organization — accounts linked under an Enterprise carry SSO over automatically as they're added, and for separate accounts, write to us and we'll enable it and confirm which key to build against.

7. Leave the Password Login Alone

Turning SSO on doesn't close the ordinary door, and you shouldn't want it to. People can still go to the Pipeline login page and sign in with their email address and password, which is what keeps your rollout safe — if the portal is down, or someone's ID hasn't been mapped yet, they still get to work.

Log in to reach everything your role gives you access to — your transactions, documents, checklists, and tasks.

See Login