

Swap your Stripe secret key for a scoped one

Both Sylus Stripe plugins now issue Restricted API Keys, and both are dropping support for secret keys. Around half an hour of developer time* closes a standing security exposure — and it is work you will have to do anyway.

THE RISK YOU ARE CARRYING TODAY

Your integration authenticates with a secret key (`sk_live_...`), which has **unrestricted access to your Stripe account**. It only has to leak once — a stale `.env`, a commit, a CI log, a departing contractor's laptop — and whoever holds it can:

read your **customer records and full payment history** (a reportable personal-data breach under GDPR), issue **refunds**, and **trigger payouts** from your balance. A Restricted API Key from the Sylus app can do **five things only**: create and confirm payments, open Checkout Sessions, capture and refund, and read events to reconcile state.

	SECRET KEY <code>SK_</code>	RESTRICTED KEY <code>RK_</code>
Scope of access	Every API resource	Only what is granted
If it leaks	Anything, including payouts	Only what the key allows
Stripe's position	Use sparingly, protect aggressively	Preferred wherever possible

"Stripe recommends always using RAKs instead of unrestricted secret keys." — [Stripe docs](#)

WHAT WE'RE ASKING YOU TO DO

1

Install the Sylus app for Stripe

On your own Stripe account, from the Stripe App Marketplace. Free, and installed in a few clicks.

[Install now →](#)

2

Copy the Restricted API Key

The app's Settings page shows your publishable key and a generated `rk_live_...` key, with copy buttons.

3

Paste it over your `sk_` key

In your Sylus payment method configuration — or in your custom integration's secret-key setting. Save.

No replatforming. No downtime. No code changes. No new contract.

WHY THIS IS ON THE CLOCK

Since June 2026

The Sylus 2.x plugin (v2.0.0, 15 June) accepts **only** `rk_`. The Sylus 1.x plugin rejects `sk_` on save from v2.1.0.

December 2026

Sylus 1.14 LTS reaches end of support. On 1.x you have roughly five months before security fixes stop entirely.

Every day until you swap

A secret key has **no expiry**. Yours has been valid every day since it was created and stays valid until someone rotates it.

UPGRADE GUIDE — FIND YOUR TRACK

TRACK 1 · SYLIUS 2.X

On the Stripe plugin, older release

Repo moved: `FLUX-SE/SylusStripePlugin` → `Sylus/StripePlugin`

1. `composer require flux-se/sylus-stripe-plugin:^2.0.1` — needs Sylus 2.0.4+
2. Paste the `rk_` key into *Restricted API key*. Save.

Shortest track — already the right plugin.

TRACK 2 · SYLIUS 1.X

On the Payum Stripe plugin

Repo moved: `FLUX-SE/SylusPayumStripePlugin` → `Sylus/PayumStripePlugin`

1. `composer require flux-se/sylus-payum-stripe-plugin:^2.0.18`
2. Paste the `rk_` key over the `sk_` value. Save.

Saved `sk_` values keep working after the update, so no outage. Then plan the move to 2.x before December.

TRACK 3 · ANY VERSION

Custom Stripe integration

Nothing to upgrade — the app is **not tied to any plugin**. Use the key wherever your code reads a Stripe secret key.

Restricted keys are **drop-in replacements** for secret keys, so this usually needs no back-end changes.

Check scope coverage first. If you also call **Customers, Subscriptions, Connect, Payouts or Radar**, those requests will fail authorisation — talk to us before switching.

What it actually costs your team. The admin change is two minutes; the honest number is a `composer update`, a staging test order and your normal release — budget around **half an hour of developer time per store***, no downtime and no customer-facing change. Rollback is one field, and Stripe lets you set a delayed expiry of up to seven days on the old key as a safety window.

Looking after several Sylus stores? The app installs per Stripe account, so each merchant needs their own install and their own key — there is no bulk switch. Send us the list and we will sequence it with you, flagging which stores need a Sylus upgrade first. Package names and namespaces are unchanged, so there is nothing to edit in `composer.json`.

INSTALL THE APP

[Stripe App Marketplace →](#)

DOCUMENTATION

[docs.sylus.com →](#)

QUESTIONS, OR HELP SCOPING AN UPGRADE



Magdalena Sadowska, Head of Product
magdalena.sadowska@sylus.com

* Happy-path estimate: the plugin as shipped, no customisation, on a supported Sylus minor. A customised checkout takes longer — ask us and we will size it. • The app's key also lets Stripe attribute your transactions to Sylus, and Sylus reinvests 100% of its payment-partnership revenue into Sylus Community Edition; your Stripe pricing and contract do not change. Telemetry is non-identifying (versions only), never customer or payment data. The GDPR note above is context, not legal advice. Stripe and Sylus are trademarks of their respective owners.