

Stripe Payment Integration for Salesforce – Business Case Study

1. Executive Summary

A company needed to streamline their payment collection process for Opportunity records inside Salesforce. The existing manual method required sending payment links through external tools, manually tracking payment status, and updating Opportunity records by hand. This resulted in delays, errors, and poor customer experience.

To solve this, a fully automated Stripe–Salesforce integration was implemented. The solution allows agents to generate payment links directly inside Salesforce, send them to customers, automatically update payment status, and redirect customers to branded success/failure pages.

2. Business Problem

Challenges Faced:

- Payment links were being created manually outside Salesforce.
- No automatic connection between payment and Opportunity records.
- Agents had to manually update payment status (Paid/Failed).
- Poor visibility for management — no real-time tracking.
- No branded success/cancel pages after payment.
- Manual emailing of payment links.
- No unified experience inside Salesforce console.

These issues increased effort, caused delays, and reduced accuracy.

3. Business Requirements

The business needed:

1. Ability to generate a Stripe payment link directly from an Opportunity.
 2. Store the link in a custom Payment object.
 3. Show the link instantly in an Agent chat window.
 4. Allow the agent to send the link through email automatically.
 5. Redirect customers to custom-branded Success/Failure LWC pages.
 6. Update payment status in Salesforce after successful payment.
 7. Maintain audit logs and reporting on payments.
-

4. Implemented Solution

A three-part architecture was designed:

A. Apex Integration Layer

A custom Apex class (`StripeDirectService`) communicates with Stripe Checkout API.

Key Responsibilities:

- Create payment session in Stripe.
- Generate payment link for Opportunity.
- Insert `Payment__c` record.
- Set Opportunity `Payment_Status__c` to *Pending*.

B. Agentforce Action (Invocable Apex)

The invocable class (`Agent_CreatePayment`) is used by the Salesforce Agent.

It returns only the payment link, which:

- Appears inside the Agent chat window.
- Can be sent by the Agent through email.
- Is fully secure and controlled via Salesforce Trusted URLs.

C. Stripe Webhook + Success/Failure LWC Pages

To capture customer payment status:

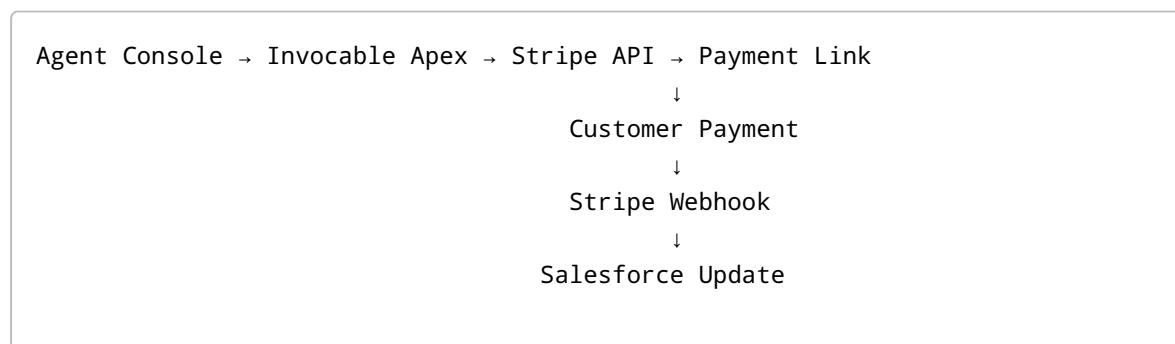
- A Stripe webhook sends a POST request back to Salesforce when payment is successful.
- Salesforce updates Opportunity and `Payment__c` status to **Paid**.

Customers visiting the link are redirected to:

- A custom **Success Page (LWC + Experience Cloud Site)**
- A custom **Failure/Cancel Page (LWC + Experience Cloud Site)**

Both pages display confirmation details to the customer.

5. Technical Architecture Diagram



↓
Success/Failure Experience Pages

6. Key Salesforce Components Created

Custom Objects:

- Payment_c
- Payment_Link_Full_c
- Amount_c
- Payment_Status_c
- Opportunity_c (Lookup)

Apex Classes:

- StripeDirectService.cls
- Agent_CreatePayment.cls

LWC Pages:

- stripePaymentResult

Experience Cloud Sites:

- /success
- /cancel

7. Integration Flow – End to End

Step 1 – Agent Requests Payment Link

Agent clicks "Create Payment Link" → triggers Invocable Apex.

Step 2 – Stripe API Generates Link

Apex sends Opportunity info (Amount, Name, Email) → Stripe.

Step 3 – Payment_c Record is Created

Payment link is saved.

Step 4 – Link is Shown in Agent Chat

Agent receives: "**Here is your payment link:**" and the URL.

Step 5 – Agent Emails the Link

Agent clicks "Send Email" → link is sent automatically to the customer.

Step 6 – Customer Completes Payment

Stripe checkout → redirects to Success/Failure Experience Cloud site.

Step 7 – Stripe Webhook Updates Salesforce

Opportunity is updated to:

- **Payment_Status_c = Paid** (if success)
 - **Payment_Status_c = Failed** (if canceled)
-

8. Benefits Delivered

Business Benefits:

- **90% reduction in agent effort** (no more manual payment tracking).
- **Real-time payment updates** on Opportunity.
- **Better customer experience** with branded success/failure pages.
- **Less human error** due to automation.
- **Complete audit tracking** of payments.

Technical Benefits:

- Modular Apex (clean separation of concerns).
 - Easy to extend for multi-currency or subscription billing.
 - Secure integration using Salesforce Named Credentials.
 - Reusable payment architecture across other Salesforce objects.
-

9. Future Enhancements

- Support for recurring/subscription billing.
 - Ability for customers to download a payment receipt (LWC PDF).
 - Payment reminder automation using Flows.
 - Refund handling via Stripe API.
 - Dashboard for finance users.
-

10. Conclusion

This Stripe–Salesforce integration solves the core business challenge of streamlining payments inside Salesforce. It enables agents to generate and share payment links in real time, track customer payments automatically, and improve operational efficiency.

This architecture is production-ready, scalable, and can be extended for any industry such as travel, education, retail, SaaS, etc.