AI 時代でも押さえたい！
TAM 視点で見る「決済でつまずく前の超基本」

ストライプジャパン株式会社
テクニカルアカウントマネージャー
Mong Xiang Theng
藤原 蓮

2026/8/1

自己紹介

stripe

Mong Xiang Theng (モン)

テクニカルアカウントマネージャー

藤原 蓮

テクニカルアカウントマネージャー

TAM（Technical Account Manager）:

- 有償サポートの一部プランで提供される専任担当者
- 運用フェーズにおける決済パフォーマンスの改善などを中心に技術的な支援を提供

本日の目的

AI 時代でも知っておきたいこと

AI開発と運用のギャップ

「運用の視点」してから見えてくること

- ✕「コードが書ける」≠「安定して運用できる」
- ✕ 表面上は動いても、運用後に初めて見えてくる課題
- ✕「こう動くはず」という想定と実際の挙動のズレ

TAM 視点で2点をピックアップ

ギャップ・ズレが典型的に出るポイント

01 エラーハンドリング

API のレスポンスの意味を正しく理解

02 決済パフォーマンス

3DS とオーソリ

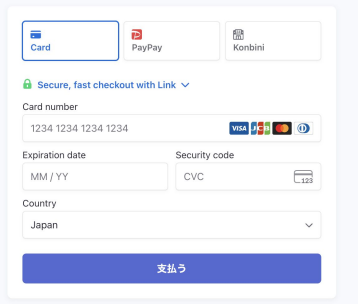
stripe

01 エラーハンドリング
API のレスポンスの意味を正しく理解

stripe

ケース A：決済処理を開始で 200 OK レスポンス

Payment Element：カードや各種決済手段を一つの埋め込み UIで提供する Stripe の決済フォーム




```
...
// 入力された決済情報で支払いを確定し、完了後は指定　URLへリダイレクトする
 const { error } = await stripe.confirmPayment({
   elements,
   confirmParams: {
     return_url: 'http://example.com/complete.html',
   },
 });
...
```

API

```
POST
/v1/payment_intents/:
id/confirm
```

→

HTTP Status Code :
200 OK

→

❌「決済成功！後段の処理を継続」

stripe

オブジェクトの状態を確認

- Stripe ではオブジェクトの作成・更新・状態遷移を追う設計
- 200 OK が意味するのは「 API リクエストが正常に受理された」こと
- 決済が成功したかは　payment_intent オブジェクトのステータス（ status）を検証する必要あり

決済ステータスの遷移フロー　(3DS発生 + Auto Capture)

`POST /v1/payment_intents/:id/confirm` ➜ **HTTP 200**
APIリクエストの送信（決済の確認・確定要求 ）

↓

`requires_action` ⏳ **アクション待ち**
3Dセキュアユーザーによる追加認証ステップが必要

↓

`succeeded` ✅ **支払い成功（完了）**
決済が正常に完了、このステータスをもって初めて支払い完了

payment_intent オブジェクト

未完了

```
{
 "id": "pi_XXXXXXX",
 "object": "payment_intent",
 . . . .
 "status": "requires_action"
}
```

成功

```
{
 "id": "pi_XXXXXXX",
 "object": "payment_intent",
 . . . .
 "status": "succeeded"
}
```

stripe

ケース B：子アカウントへ送金で 500 Error エラーレスポンス

Connect で支払いと送金を別々に作成する方式 (Separate charges and transfers)


```
const transfer = await
stripe.transfers.create({
  amount: 2000,
  currency: 'usd',
  destination: 'acct_xxxxx',
  transfer_group: 'ORDER100',
});
```

API

POST
/v1/transfers

→

HTTP Status Code :
500
Internal Server Error

→

❌「別の冪等キーでリトライしよう」

⚠「同じ冪等キーでリトライすればいつか成功する」

stripe

500 Internal Server Error の意味

Stripe の 500は「失敗」ではなく「結果不確定」

- **成功か失敗か判断できない状態**
  リクエスト処理中に接続が切れた場合などで発生しうる
- **自動調整機能（ロールフォワード）**
  内部で状態を調整し、後から処理を確定させることがある
- **キャッシュの動作**
  /v1/transfers API などでは、同一冪等キーを利用した場合でも500ステータスコードがキャッシュされ結果は同じになる可能性がある

二重送金が発生するシナリオ例

1. POST /v1/transfers をコール
2. Internal Server Error が発生し 500をレスポンス
3. 実際には送金が完了していた
4. 「失敗」と誤認し、別のリクエストを再送信する

⚠️ **二重返金が成立**

stripe

状態(ステータス)を正しくハンドルするために

Webhook イベントで非同期に受け取るのが重要

一時的な同期レスポンス（HTTP status code）だけに依存せず、信頼性の高いイベント駆動設計を構築

イベントサンプル

支払いが成功

payment_intent.succeeded

```
{
 "id": "evt 3NqABCDEFGHIJKLM",
 "object": "event",
 "type":
"payment intent.succeeded",
 . . . . . .
}
```

非同期でステータス変化への対応

【ケースA：200】

非同期プロセスとして Webhook にて通知を受け取り、オブジェクトのステータスを確認

【ケースB：500 Error】

同期リトライのみでは判断できない。 Webhook で処理の進行状況を把握。

注意点

Webhook は重複する可能性あり

→ event.id で重複排除

イベントの順序は保証していない

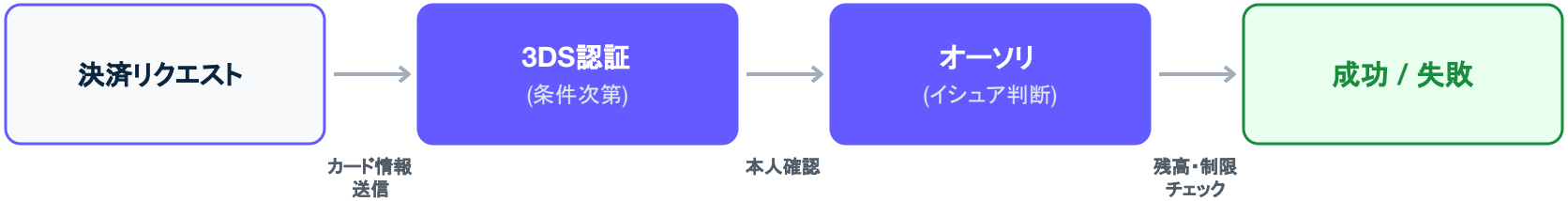
→ Webhook 受信後にオブジェクトを GET して最新ステータスを取得

stripe

02 決済パフォーマンス
3DSとオーソリ

stripe

Authentication (3DS) と Authorization (オーソリ)



⚠ 注意点と重要仕様

- **3DS認証は条件次第で発動**：Radar rule、イシュア要求、または規制要件（SCAなど）に基づいて実行されます。
- **3DS成功 ≠ 決済成功**：本人確認（3DS）が通っても、その後のオーソリ段階で残高不足等により落ちることがあります。
- **ソフトディクラインの存在**：オーソリ拒否時にイシュアから3DSを要求され、認証後に再試行して成功するパターンもあります。

stripe

Authentication：3DSとライアビリティシフト

3DSとは？

カードの持ち主本人かどうかをイシュア（カード発行会社）が確認するステップ

例: SMS認証、銀行アプリ承認

❌ よくある誤解

「3DSやれば不正チャージバックは全部イシュア負担」

✅ 実態：ライアビリティシフト ≠ 100%保護

① シフト適用には条件がある

- 認証成功 → **シフトあり**
- 認証試行済み → **シフトあり** (イシュア未対応でも OK)
- 認証なし → **シフトなし**

②「不正利用」のチャージバックのみ対象

商品未達・説明と違うなどの理由は対象外

③ SetupIntentの3DS ≠ 課金時のシフト

カード保存時に認証しても、後続の PaymentIntentには適用なし (定期課金は要注意)

④ 3DSは取引単位

一度3DSを通過しても、別の決済には引き継がれない

stripe

Authorization

Raw (未加工)

リアルタイム分析用途

すべてのオーソリ試行をそのままカウント。1回の決済で3回試行した場合、Rawデータでは3件としてカウントされます。

Deduplicated (重複排除)

正確なデータ分析用途

「同一決済の複数回試行」をグループ化し、最後の結果のみで集計。
1回決済で3回試行し最後が成功なら、成功率は100%（Rawでは33%）。

stripe

ダッシュボード使い倒そう

支払いの分析レポート

SQL・コード不要で今すぐ使える

ダッシュボードの「支払いの分析」から、すぐに各種データを可視化できます。

- **支払い成功の分析**
  成功率の全体像 + decline理由の特定
- **認証の分析**
  3DS通過率・離脱率の可視化
- **不審請求の申し立て分析**
  チャージバック状況のモニタリング
- **決済手段の分析**
  決済手段ごとのパフォーマンス比較
- **最適化分析**
  Stripeの自動最適化の効果検証

データ活用を深掘り

高度なクエリと自動連携

Sigma
SQLを使って自由にクエリを作成。スケジュール配信にも対応。

Data Pipeline
自社のSnowflakeやBigQuery環境へデータを自動でシームレスに連携。

Console AI アシスタント
自然言語でStripeデータに質問。Dashboard上で直接AIに質問可能です。「先月の拒否理由の内訳は？」と聞くだけで回答。

Console待機リスト登録：docs.stripe.com/dashboard/console

Stripe
Technical Account Manager

Software is only half the story

Like any high-performance instrument, your Stripe solution requires ongoing, specialized care and expertise

stripe

TAM：ユーザーのビジネスを知る専任パートナー

決済最適化ワークショップ

Auth rate・不正対策・コスト改善のセッション

パフォーマンス監視

決済異常を早期検知・カスタムレポート

テクニカルパートナーシップ

定期的な技術相談・最新アップデートの共有

イベント対応

セール・キャンペーン時の事前準備と支援

※ 提供メニューはサポートプランによります

User Story

有償サポート導入事例

DeNA: Stripe プレミアムサポートの活用により決済承認率を 10% 向上

詳しく見る →

LG公式オンラインショップの決済基盤にStripeを採用しダッシュボードで不正決済を可視化

詳しく見る →

stripe

Thank you !