

# Multi-Seller Payment Routing – Three Approaches Explained

This document explains three different approaches to handle multi-seller payment routing when Razorpay Route is not yet enabled. The explanations are beginner-friendly and focus on understanding the logic clearly.

## The Core Problem

In a multi-seller e-commerce platform, a single customer payment must be split between multiple sellers and the platform (commission). Razorpay Route normally handles this. However, when Route is not enabled (especially in test mode), we need alternate logic to simulate or prepare for this behavior.

### Option 1: Simulated Split (Database-Only Logic)

In this approach, no real routing or transfer happens. After payment success, the backend calculates how much each seller should receive and stores this information in the database.

- Payment is collected normally (test mode).
- Seller share and platform commission are calculated in backend.
- Split amounts are saved in database tables (orders, seller payouts).
- Nothing appears in Razorpay dashboard regarding splitting.

**Analogy:** Writing in a notebook that you owe someone money, but not paying yet.

### Option 2: Abstracted Routing Logic (Switch-Based)

This approach is an improved version of Option 1. The routing logic is isolated into a single function or service, making it easy to replace simulated logic with real Razorpay Route later.

- Payment success triggers a single 'distribute money' function.
- If Route is disabled → simulated DB split is used.
- If Route is enabled later → Razorpay Route API is called.
- No UI or major backend rewrite required in future.

**Analogy:** Same electrical wiring, switching power source from inverter to main supply later.

### Option 3: Mock Razorpay Route API

In this advanced approach, a fake or mock Razorpay Route API is created in the backend. The system behaves as if real routing is happening, but all responses are simulated.

- Fake transfer API returns realistic Razorpay-like responses.
- Used for testing failures, retries, refunds, and payout states.
- No real Razorpay interaction or money movement.
- More complex and not necessary for beginners.

**Analogy:** Acting in a movie scene where money is transferred, but it is only acting.

## Recommended Choice for Learning Projects

For personal or learning projects, Option 2 (Abstracted Routing Logic) is the best choice. It is simple, clean, interview■friendly, and future■proof. When Razorpay Route is enabled, only one module needs to be replaced.