

Accounting Domain – Business Details for AI Test Case Generation

This document provides business context to help an AI Agent (RAG-enabled) generate high-quality test cases for the svc-accounting microservice.

1. Business Overview

The Accounting Service manages financial accounts and applies business rules when processing credits and debits. It ensures the integrity of balances and enforces standard accounting constraints.

2. Core Business Entities

Account

- Represents a customer or system-owned ledger.
- Attributes:
 - Account Number (unique)
 - Name
 - Balance (BigDecimal)

Transaction

- Represents money movement.
- Types:
 - CREDIT → increases balance
 - DEBIT → decreases balance
- Business rules enforced:
 - No negative balances allowed.
 - Amount must be positive.
 - Debit cannot exceed current balance.

3. Key Business Rules for Test Generation

1. Every account starts with a zero balance.

2. Account numbers must be unique.

3. CREDIT transactions:

- Must accept positive amounts.
- Must correctly update balance.

4. DEBIT transactions:

- Must reject amounts larger than current balance.
- Must handle exact balance debit gracefully.

5. Error handling:

- "Account not found"
- "Invalid amount"
- "Insufficient balance"

4. Example Business Scenarios

Scenario 1 — Basic Credit Flow

- Create Account ACC1001
- Post CREDIT of 500
- Expected balance → 500

Scenario 2 — Debit With Sufficient Balance

- Debit 200 from balance 500
- Expected balance → 300

Scenario 3 — Debit With Insufficient Balance

- Attempt to debit 600 from balance 300
- Expected → Error: Insufficient balance

Scenario 4 — Multiple Transactions

- CREDIT 1000

- DEBIT 300
- CREDIT 200
- Expected final balance → 900

5. Purpose for AI Test Generation

This domain knowledge allows the AI agent to:

- Understand business constraints
- Generate meaningful unit & integration tests
- Validate error scenarios
- Cover edge cases and boundary conditions
- Produce reliable JUnit tests that match accounting rules

6. Notes

This document is intended for RAG ingestion so the agent can produce smarter, context-rich test cases for svc-accounting.