Purchase Scenarios Catalog

> Teligant Platform - Comprehensive Purchase & Payment Reference

Table of Contents

Overview

Architecture Summary

```
┌─────────────────────────────────────────────────────────────────┐
│                        TELIGANT PLATFORM                         │
├─────────────────────────────────────────────────────────────────┤
│  PURCHASE TYPES                                                  │
│                                                                 │
│  ┌──────────┐  ┌──────────────┐  ┌────────────┐  ┌────────────┐  │
│  │   OTP    │  │ SUBSCRIPTION │  │ MEMBERSHIP │  │CONSULTATION│  │
│  │(One-Time)│  │ (Recurring)  │  │  (Access)  │  │ (Service)  │  │
│  └──────────┘  └──────────────┘  └────────────┘  └────────────┘  │
│                                                                 │
├─────────────────────────────────────────────────────────────────┤
│  PAYMENT PROCESSOR: PayTheory (GraphQL API + Webhooks)          │
├─────────────────────────────────────────────────────────────────┤
│  DATABASE: PostgreSQL + Prisma ORM                              │
└─────────────────────────────────────────────────────────────────┘
```

Key Entities

Entity	Description	Database Model
Order	Main purchase record	`Order`
LineItem	Individual product in order	`OrderLineItem`
Refill	Scheduled subscription refill	`OrderRefill`
Transaction	Payment record	`PayTheoryTransaction`
Subscription	Recurring payment schedule	`PayTheorySubscription`
PaymentMethod	Saved card token	`PayTheoryPaymentMethod`

Purchase Types

1. One-Time Purchase (OTP)

Single purchase of physical products or lab tests without recurring billing.

Characteristics: - Single charge at approval - No automatic refills - Requires doctor approval for prescription items - Shipped once fulfilled

Product Types: - Physical products (non-prescription) - Prescription medications (single supply) - Lab tests

2. Subscription

Recurring purchase of medications with automatic refills and billing.

Characteristics: - Automatic recurring charges - Scheduled refills based on billing cycle - -7 day offset on first refill - Can be paused/resumed/canceled - Requires doctor approval

Billing Cycles: - 30-day (monthly) - 60-day (bi-monthly) - 90-day (quarterly) - most common - 120-day (4-month) - 180-day (6-month)

3. Membership

Access-based subscription providing benefits and discounts.

Characteristics: - Monthly or annual billing - Unlocks member benefits (discounts, auto-orders) - Health assessment access - Priority support

Billing Options: - Monthly recurring - Annual (upfront or monthly)

4. Initial Consultation

Service purchase for video/chat consultation with a doctor.

Characteristics: - Charged upfront before consultation - Time-limited availability - Can result in prescription orders - Refundable if not completed

---

Scenario Catalog

SC-001: One-Time Purchase - Physical Product (Non-Prescription)

```
scenario_id: SC-001
name: "OTP - Physical Product (Non-Prescription)"
type: ONE_TIME_PURCHASE
requires_approval: false
payment_timing: IMMEDIATE

flow:
  1. Patient adds non-prescription product to cart
  2. Patient completes checkout with payment method
  3. Payment charged immediately
  4. Order status: APPROVED
  5. Order sent to fulfillment
  6. Order shipped to patient

status_progression:
  - PENDING → APPROVED → SENT_TO_PHARMACY → ORDER_SHIPPED

payment:
  charge_timing: "At checkout"
  refund_eligible: true
  refund_window: "Before shipping"
```

SC-002: One-Time Purchase - Prescription Medication

```
scenario_id: SC-002
name: "OTP - Prescription Medication"
type: ONE_TIME_PURCHASE
requires_approval: true
payment_timing: ON_APPROVAL

flow:
  1. Patient completes health questionnaire
  2. Patient adds prescription product to cart
  3. Patient completes checkout (payment method saved, not charged)
  4. Order status: AWAITING_REVIEW
  5. Doctor reviews prescription request
  6. IF APPROVED:
     - Payment charged
     - Order sent to pharmacy
     - Prescription fulfilled and shipped
  7. IF DENIED:
     - Patient notified
     - No charge made
     - May suggest alternatives

status_progression:
  approved: AWAITING_REVIEW → APPROVED → SENT_TO_PHARMACY → ORDER_SHIPPED
  denied: AWAITING_REVIEW → DENIED

payment:
  charge_timing: "Upon doctor approval"
  refund_eligible: true
  refund_window: "Before pharmacy processing"
```

SC-003: One-Time Purchase - Lab Test

```
scenario_id: SC-003
name: "OTP - Lab Test"
type: ONE_TIME_PURCHASE
requires_approval: false
payment_timing: IMMEDIATE
product_type: LAB_TEST

flow:
  1. Patient selects lab test
  2. Patient completes checkout
  3. Payment charged immediately
  4. Lab requisition generated
  5. Patient visits lab location
  6. Results delivered to platform
  7. Doctor reviews results (if applicable)

status_progression:
  - PENDING → APPROVED → IN_PROGRESS → PROCESSED

payment:
  charge_timing: "At checkout"
  refund_eligible: true
  refund_window: "Before lab visit"

integration:
  provider: "ChooseHealth"
```

SC-004: Subscription - 30-Day Cycle (New)

```
scenario_id: SC-004
name: "Subscription - 30-Day Cycle (New)"
type: SUBSCRIPTION
billing_cycle: EVERY_DAY_30
supply: DAY_30
requires_approval: true

flow:
  1. Patient completes health questionnaire
  2. Patient selects 30-day subscription option
  3. Patient completes checkout
  4. Order status: AWAITING_REVIEW
  5. Doctor reviews and approves
  6. First payment charged
  7. Refill schedule created with -7 day offset
  8. First refill: Day 23 (30 - 7 = 23)
  9. Subsequent refills: Every 30 days from first refill

refill_schedule:
  subscription_start: "Day 0"
  first_refill: "Day 23"  # -7 day offset applied
  second_refill: "Day 53"
  third_refill: "Day 83"
  pattern: "First refill shifted, then regular 30-day intervals"

status_progression:
  - AWAITING_REVIEW → ACTIVE

payment:
  initial_charge: "Upon approval"
  recurring_charges: "Every 30 days from first refill"

business_rule: SUBSCRIPTION_PAYMENT_SHIFT_DAYS = 7
```

SC-005: Subscription - 90-Day Cycle (New)

```
scenario_id: SC-005
name: "Subscription - 90-Day Cycle (New)"
type: SUBSCRIPTION
billing_cycle: EVERY_DAY_90
supply: DAY_90
requires_approval: true

flow:
  1. Patient completes health questionnaire
  2. Patient selects 90-day subscription option
  3. Patient completes checkout
  4. Order status: AWAITING_REVIEW
  5. Doctor reviews and approves
  6. First payment charged
  7. Refill schedule created with -7 day offset
  8. First refill: Day 83 (90 - 7 = 83)
  9. Subsequent refills: Every 90 days from first refill

refill_schedule:
  subscription_start: "Day 0"
  first_refill: "Day 83"  # -7 day offset applied
  second_refill: "Day 173"
  third_refill: "Day 263"
  pattern: "First refill shifted, then regular 90-day intervals"

status_progression:
  - AWAITING_REVIEW → ACTIVE

payment:
  initial_charge: "Upon approval"
  recurring_charges: "Every 90 days from first refill"
```

SC-006: Subscription - Pause

```
scenario_id: SC-006
name: "Subscription - Pause"
type: SUBSCRIPTION_ACTION
action: PAUSE
requires_approval: false

preconditions:
  - Subscription status is ACTIVE
  - No pending refill in processing

flow:
  1. Patient/Admin requests pause
  2. System updates Order status to PAUSED
  3. PayTheory subscription CANCELED (workaround - no native pause)
  4. canceledAt timestamp recorded
  5. Upcoming refills marked as inactive
  6. User action logged: SUBSCRIPTION_PAUSED
  7. ActiveCampaign event: DOCTOR_REVIEW_PAUSED

status_progression:
  - ACTIVE → PAUSED

side_effects:
  - PayTheory subscription canceled (not paused)
  - Original refill schedule preserved in OrderRefill records

known_issues:
  - "Uses cancel+recreate workaround (PayTheory has no native pause)"
```

SC-007: Subscription - Resume

```
scenario_id: SC-007
name: "Subscription - Resume"
type: SUBSCRIPTION_ACTION
action: RESUME
requires_approval: false

preconditions:
  - Subscription status is PAUSED

flow:
  1. Patient/Admin requests resume
  2. System updates Order status to NEW (then ACTIVE after processing)
  3. Existing OrderRefill statuses reset to null
  4. NEW PayTheory subscription created from resume date
  5. New refill dates calculated from resume date
  6. -7 day offset recalculated from resume date (BUG: loses original offset)
  7. User action logged: SUBSCRIPTION_RESUMED
  8. ActiveCampaign event: DOCTOR_REVIEW_RESUMED

status_progression:
  - PAUSED → NEW → ACTIVE

side_effects:
  - New PayTheory subscription created
  - Refill schedule recalculated from resume date

known_issues:
  - "BUG: Original -7 day offset lost on resume"
  - "Refill dates shift unpredictably after resume"
  - "Potential duplicate refills if overlapping dates"
```

SC-008: Subscription - Cancel

```
scenario_id: SC-008
name: "Subscription - Cancel"
type: SUBSCRIPTION_ACTION
action: CANCEL
requires_approval: false

preconditions:
  - Subscription status is ACTIVE or PAUSED

flow:
  1. Patient/Admin requests cancellation
  2. System updates Order status to CANCELED
  3. PayTheory subscription canceled
  4. All future refills marked as CANCELED
  5. User action logged: SUBSCRIPTION_CANCELED
  6. No refund for current period (already fulfilled)

status_progression:
  - ACTIVE → CANCELED
  - PAUSED → CANCELED

payment:
  refund: "No automatic refund for current period"
  future_charges: "Stopped immediately"
```

SC-009: Subscription - Denial

```
scenario_id: SC-009
name: "Subscription - Doctor Denial"
type: SUBSCRIPTION_ACTION
action: DENY
requires_approval: true

preconditions:
  - Subscription status is AWAITING_REVIEW
  - Doctor has reviewed request

flow:
  1. Doctor reviews subscription request
  2. Doctor denies (with reason)
  3. Order status updated to DENIED
  4. If payment was taken: Refund issued
  5. Patient notified with denial reason
  6. May suggest alternative products/consultations

status_progression:
  - AWAITING_REVIEW → DENIED

payment:
  refund: "Full refund if already charged"
```

SC-010: Membership - Monthly

```
scenario_id: SC-010
name: "Membership - Monthly Billing"
type: MEMBERSHIP
billing_cycle: MONTHLY
requires_approval: false

flow:
  1. Patient selects membership plan
  2. Patient completes checkout
  3. First payment charged immediately
  4. Membership activated
  5. Benefits unlocked (discounts, auto-orders, assessments)
  6. Recurring monthly charges

benefits:
  - Product discounts (percentage off)
  - Auto-order eligible items
  - Health assessment access
  - Priority support

status_progression:
  - PENDING → ACTIVE

payment:
  initial_charge: "At checkout"
  recurring_charges: "Monthly"
```

SC-011: Membership - Annual

```
scenario_id: SC-011
name: "Membership - Annual Billing"
type: MEMBERSHIP
billing_cycle: ANNUAL
requires_approval: false

flow:
  1. Patient selects annual membership
  2. Patient completes checkout
  3. Full annual payment charged
  4. Membership activated for 12 months
  5. Benefits unlocked
  6. Renewal reminder before expiration

benefits:
  - Same as monthly + annual discount
  - Price lock for membership year

status_progression:
  - PENDING → ACTIVE

payment:
  initial_charge: "Full annual amount at checkout"
  renewal: "Annual (with reminder)"
```

SC-012: Initial Consultation - Video

```
scenario_id: SC-012
name: "Initial Consultation - Video"
type: CONSULTATION
service_type: VIDEO
requires_approval: false
payment_timing: IMMEDIATE

flow:
  1. Patient selects consultation type
  2. Patient selects available time slot
  3. Patient completes checkout
  4. Payment charged immediately
  5. Appointment token created
  6. Consultation conducted at scheduled time
  7. Doctor may prescribe medications
  8. If prescription: New order created

status_progression:
  completed: PENDING → IN_PROGRESS → PROCESSED
  refunded: PENDING → REFUNDED
  expired: PENDING → EXPIRED

payment:
  charge_timing: "At booking"
  refund_eligible: true
  refund_conditions:
    - "Canceled 24+ hours before appointment"
    - "Doctor unavailable"
    - "Technical issues preventing consultation"
```

SC-013: Subscription - Refill Payment Success

```
scenario_id: SC-013
name: "Subscription Refill - Payment Success"
type: SUBSCRIPTION_REFILL
trigger: SCHEDULED_PAYMENT
payment_result: SUCCESS

flow:
  1. Scheduled refill date reached
  2. PayTheory processes recurring payment
  3. Webhook: payment.success received
  4. Transaction recorded: CAPTURED
  5. OrderRefill status updated: PAID
  6. Child order created for this refill
  7. Order sent to pharmacy
  8. Next refill date confirmed

webhook_event: "payment.success"
transaction_status: CAPTURED
refill_status: PAID
```

SC-014: Subscription - Refill Payment Failure

```
scenario_id: SC-014
name: "Subscription Refill - Payment Failure"
type: SUBSCRIPTION_REFILL
trigger: SCHEDULED_PAYMENT
payment_result: FAILURE

flow:
  1. Scheduled refill date reached
  2. PayTheory attempts payment
  3. Payment fails (insufficient funds, card expired, etc.)
  4. Webhook: payment.failed received
  5. Transaction recorded: FAILED with failureReasons
  6. Retry logic (if configured)
  7. Patient notified to update payment method
  8. Subscription may be paused after X failures

webhook_event: "payment.failed"
transaction_status: FAILED
retry_policy: "3 attempts over 7 days"
escalation: "Pause subscription after final failure"
```

SC-015: Payment Method - Add New Card

```
scenario_id: SC-015
name: "Payment Method - Add New Card"
type: PAYMENT_METHOD
action: CREATE

flow:
  1. Patient navigates to payment settings
  2. PayTheory hosted form displayed
  3. Patient enters card details
  4. Card tokenized by PayTheory
  5. Token stored in PayTheoryPaymentMethod
  6. Card available for future purchases

data_stored:
  - paymentMethodId (PayTheory token)
  - payload (card metadata: last4, brand, expiry)
  - NOT stored: Full card number, CVV
```

SC-016: Payment Method - Update Default

```
scenario_id: SC-016
name: "Payment Method - Update for Subscription"
type: PAYMENT_METHOD
action: UPDATE

flow:
  1. Patient selects subscription to update
  2. Patient selects new payment method
  3. PayTheory subscription updated with new payment method
  4. Future charges use new card
  5. Confirmation displayed

preconditions:
  - New payment method already saved
  - Subscription is ACTIVE
```

Status Workflows

One-Time Purchase Status Flow

```
stateDiagram-v2
    [*] --> PENDING: Order Created
    PENDING --> AWAITING_REVIEW: Requires Prescription
    PENDING --> APPROVED: No Prescription Required
    AWAITING_REVIEW --> APPROVED: Doctor Approves
    AWAITING_REVIEW --> DENIED: Doctor Denies
    APPROVED --> SENT_TO_PHARMACY: Payment Captured
    SENT_TO_PHARMACY --> ORDER_SHIPPED: Pharmacy Ships
    ORDER_SHIPPED --> [*]: Complete
    DENIED --> [*]: Closed
```

Subscription Status Flow

```
stateDiagram-v2
    [*] --> AWAITING_REVIEW: Order Created
    AWAITING_REVIEW --> ACTIVE: Doctor Approves
    AWAITING_REVIEW --> DENIED: Doctor Denies
    ACTIVE --> PAUSED: User/Admin Pauses
    PAUSED --> ACTIVE: User/Admin Resumes
    ACTIVE --> CANCELED: User/Admin Cancels
    PAUSED --> CANCELED: User/Admin Cancels
    CANCELED --> [*]: Closed
    DENIED --> [*]: Closed
```

Consultation Status Flow

```
stateDiagram-v2
    [*] --> PENDING: Booked
    PENDING --> IN_PROGRESS: Consultation Starts
    IN_PROGRESS --> PROCESSED: Completed
    IN_PROGRESS --> SENT_TO_PHARMACY: Prescription Issued
    PENDING --> REFUNDED: Canceled
    PENDING --> EXPIRED: Time Passed
    PROCESSED --> [*]: Complete
    SENT_TO_PHARMACY --> [*]: Complete
    REFUNDED --> [*]: Closed
    EXPIRED --> [*]: Closed
```

Billing Cycles

Supported Cycles

Cycle	Enum Value	Days	Use Case
30-day	`EVERY_DAY_30`	30	Standard monthly
60-day	`EVERY_DAY_60`	60	Bi-monthly
90-day	`EVERY_DAY_90`	90	Quarterly (most common)
120-day	`EVERY_DAY_120`	120	4-month supply
180-day	`EVERY_DAY_180`	180	6-month supply
Annual	`ANNUAL`	365	Membership
One-time	`ONE_TIME_PAYMENT`	N/A	Single purchase
Upfront	`CHARGE_UPFRONT`	N/A	Pre-payment

First Refill Offset Rule

Business Rule: First subscription refill is shifted 7 days earlier to improve engagement.

```
SUBSCRIPTION_PAYMENT_SHIFT_DAYS = 7

Formula:
  first_refill_date = subscription_start + billing_cycle_days - 7
  subsequent_refills = previous_refill + billing_cycle_days
```

Example (90-day subscription starting Jan 1): - Subscription Start: Jan 1 - First Refill: Jan 1 + 90 - 7 = **Mar 25** (Day 83) - Second Refill: Mar 25 + 90 = Jun 23 (Day 173) - Third Refill: Jun 23 + 90 = Sep 21 (Day 263)

Business Rules

BR-001: Prescription Approval Required

```
rule_id: BR-001
name: "Prescription Approval Required"
applies_to: [ONE_TIME_PURCHASE, SUBSCRIPTION]
condition: "Product.requirePrescription == true"

behavior:
  - Order created with status AWAITING_REVIEW
  - Payment NOT charged until approval
  - Doctor must review within SLA
  - Denial requires reason
```

BR-002: First Refill Shift

```
rule_id: BR-002
name: "First Refill 7-Day Shift"
applies_to: [SUBSCRIPTION]
constant: SUBSCRIPTION_PAYMENT_SHIFT_DAYS = 7

behavior:
  - First refill date shifted 7 days earlier
  - Subsequent refills use normal intervals
  - Purpose: Improve patient engagement
```

BR-003: Pause Uses Cancel+Recreate

```
rule_id: BR-003
name: "Pause Workaround"
applies_to: [SUBSCRIPTION]
reason: "PayTheory has no native pause API"

behavior:
  - Pause: Cancel PayTheory subscription
  - Resume: Create NEW PayTheory subscription
  - Side effect: Refill schedule recalculated from resume date
  - Known issue: Original -7 day offset lost
```

BR-004: Membership Benefits Application

```
rule_id: BR-004
name: "Membership Benefits"
applies_to: [MEMBERSHIP]

benefits:
  - Percentage discount on eligible products
  - Auto-order for specific items
  - Health assessment access
  - Member-only products

application:
  - Discount applied at checkout
  - Benefits tracked in StoreProductMembershipBenefitApplication
```

BR-005: Payment Method Security

```
rule_id: BR-005
name: "Payment Tokenization"
applies_to: [ALL]

behavior:
  - Card numbers NEVER stored in database
  - PayTheory handles PCI compliance
  - Only token and metadata stored
  - Soft delete (deletedAt) for removed cards
```

PayTheory Integration

API Configuration

```
// Environment Variables
PAY_THEORY_API_URL=https://api.start.paytheory.com/graphql
PAY_THEORY_SECRET_KEY=<secret>
PAY_THEORY_MERCHANT_UID=<merchant-id>
PAY_THEORY_WEBHOOK_URL=<webhook-endpoint>
```

Account Codes

Used for categorizing transactions in PayTheory:

Code	Purpose
`Initial Consultation`	Video/chat consultations
`Membership`	Membership purchases
`Subscription`	Recurring medication orders
`Lab Test`	Laboratory testing
`Physical Product`	One-time product purchases

Webhook Events

Event	Trigger	Action
`payment.success`	Payment captured	Update transaction, create refill order
`payment.failed`	Payment declined	Log failure, notify patient, retry
`subscription.created`	New subscription	Link to order
`subscription.canceled`	Subscription ended	Update status
`dispute.created`	Chargeback initiated	Flag for review

Transaction Statuses

Status	Description
`PENDING`	Payment initiated
`AUTHORIZED`	Payment authorized, not captured
`CAPTURED`	Payment completed
`FAILED`	Payment declined
`REFUNDED`	Payment refunded
`DISPUTED`	Chargeback in progress

Edge Cases & Known Issues

ISSUE-001: Pause/Resume Loses Refill Offset

Severity: High **Status:** Known Bug

Problem: When a subscription is paused and resumed, the -7 day first refill offset is recalculated from the resume date, not preserved from the original subscription.

Example: - Original: Start Jan 1, First refill Jan 24 (30-7=23 days) - Pause Feb 15 - Resume Mar 10 - Actual: New first refill Apr 2 (Mar 10 + 30 - 7) - Expected: Preserve relative schedule

Impact: Patient refill dates shift unpredictably

ISSUE-002: Potential Refill Duplication

Severity: Medium **Status:** Under Investigation

Problem: On resume, new OrderRefill records are created from the resume date, but existing refills are kept with status reset to null. This may cause duplicate charges for overlapping dates.

ISSUE-003: No Timezone Handling

Severity: Low **Status:** Known Limitation

Problem: Refill dates are calculated using server timezone (UTC). Patients in different timezones may see different dates in UI.

ISSUE-004: Merchant-of-Record Architecture

Severity: Critical (Business) **Status:** Decision Needed

Problem: Teligant is merchant-of-record for all tenants. All payments flow to Teligant's account first.

Implications: - Manual fund distribution to tenants - Teligant liable for all chargebacks - Complex accounting - Security risk: Single API key compromise affects all tenants

Recommendation: Evaluate migration to Stripe Connect or PayFac model

Testing Reference

Test Scenarios by Category

One-Time Purchase Tests

Test ID	Scenario	Expected Result
OTP-001	Non-prescription checkout	Immediate charge, APPROVED status
OTP-002	Prescription checkout	AWAITING_REVIEW, no charge
OTP-003	Doctor approval	Payment charged, SENT*TO*PHARMACY
OTP-004	Doctor denial	DENIED status, no charge
OTP-005	Lab test purchase	Immediate charge, requisition generated

Subscription Tests

Test ID	Scenario	Expected Result
SUB-001	New 30-day subscription	First refill at Day 23
SUB-002	New 90-day subscription	First refill at Day 83
SUB-003	Subscription approval	ACTIVE status, payment charged
SUB-004	Subscription pause	PAUSED status, PayTheory canceled
SUB-005	Subscription resume	New PayTheory subscription created
SUB-006	Subscription cancel	CANCELED, future refills stopped
SUB-007	Refill payment success	PAID refill, child order created
SUB-008	Refill payment failure	FAILED transaction, retry initiated

Membership Tests

Test ID	Scenario	Expected Result
MEM-001	Monthly membership purchase	Immediate charge, benefits activated
MEM-002	Annual membership purchase	Full amount charged, 12-month access
MEM-003	Membership discount application	Correct discount on eligible products

Consultation Tests

Test ID	Scenario	Expected Result
CON-001	Book consultation	Payment charged, appointment created
CON-002	Cancel consultation (24h+)	Full refund
CON-003	Consultation completed	PROCESSED status
CON-004	Prescription from consultation	New order created

File References

Backend (NestJS)

Component	Path
Order Core Service	`src/shared/modules/order/modules/order/order-core.service.ts`
Subscription Service	`src/shared/modules/purchase-scope/purchase-subscription/purchase-subscription.service.ts`
Refill Service	`src/shared/modules/purchase-scope/purchase-subscription-refill/purchase-subscription-refill.service.ts`
PayTheory Client	`src/shared/providers/pay-theory/pay-theory-api.client.ts`
Status Enums	`src/shared/enums/purchase-*.enum.ts`
Checkout Controller	`apps/patient/core/modules/order-scope/order/layers/order-bl/controllers/checkout.controller.ts`

Frontend (Patient Portal)

Component	Path
Subscriptions API	`src/providers/store/api/subscriptions/subscriptions.ts`
Orders API	`src/providers/store/api/orders/`
Checkout Page	`src/app/(guest)/checkout/page.tsx`
Subscription List	`src/app/panel/(orders)/subscriptions/page.tsx`

Database

Model	Description
`Order`	Main purchase record
`OrderLineItem`	Products in order
`OrderRefill`	Scheduled refills
`PayTheoryTransaction`	Payment records
`PayTheorySubscription`	Recurring schedules
`PayTheoryPaymentMethod`	Saved cards

Version History

Version	Date	Changes
1.0.0	2024-12-02	Initial catalog creation

Generated from codebase analysis. Last updated: December 2, 2024