Checkout Session

Author: Josh S. Sakweli, Backend Lead Team
Last Updated: 2025-10-02
Version: v1.0

Base URL: `https://apinexgate.glueauth.com/api/v1/`

Short Description: The Checkout Session API manages the complete checkout process for e-commerce transactions. It handles creating checkout sessions from cart or direct purchase, managing payment intents, processing payments through multiple payment methods (Wallet, Cash on Delivery, Cards, Mobile Money), and supports advanced features like group purchasing and installment payments. Each session maintains state, inventory holds, and payment attempt tracking with automatic expiration handling.

Hints:

- All checkout sessions expire after 15 minutes by default
- Inventory is automatically held during active sessions and released upon expiration or cancellation
- Maximum 5 payment retry attempts allowed per session
- Group purchase sessions require WALLET payment method only
- Sessions in PAYMENT_PROCESSING status cannot be modified
- Installment payment type is planned for future release (placeholder only)
- All monetary values are in TZS (Tanzanian Shillings)
- Use ISO 8601 format for all datetime fields
- Sessions can only be updated when in PENDING_PAYMENT or PAYMENT_FAILED status

---

Standard Response Format

All API responses follow a consistent structure using our Globe Response Builder pattern:

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
```

```
  "action_time": "2025-10-02T10:30:45",
  "data": {
    // Actual response data goes here
  }
}
```

Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-10-02T10:30:45",
  "data": "Error description"
}
```

Standard Response Fields

Field	Type	Description
`success`	boolean	Always `true` for successful operations, `false` for errors
`httpStatus`	string	HTTP status name (OK, CREATED, BAD_REQUEST, NOT_FOUND, etc.)
`message`	string	Human-readable message describing the operation result
`action_time`	string	ISO 8601 timestamp of when the response was generated
`data`	object/string	Response payload for success, error details for failures

---

HTTP Method Badge Standards

For better visual clarity, all endpoints use colored badges for HTTP methods with the following standard colors:

- **GET** - GET - Green (Safe, read-only operations)
- **POST** - POST - Blue (Create new resources)
- **PATCH** - PATCH - Orange (Partial updates)

- **DELETE** - `DELETE` - Red (Remove resources)

Endpoints

1. Create Checkout Session

Purpose: Creates a new checkout session for processing a purchase. Supports multiple checkout types: direct product purchase, cart checkout, group purchasing, and installment payments (future).

Endpoint: `POST` `{base_url}/checkout-sessions`

Access Level: □□ Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user
Content-Type	string	Yes	Must be `application/json`

Request JSON Sample:

```
{
  "sessionType": "REGULAR_DIRECTLY",
  "items": [
    {
      "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
      "quantity": 2
    }
  ],
  "shippingAddressId": "f1e2d3c4-b5a6-7890-cdef-123456789abc",
  "shippingMethodId": "standard-shipping",
  "paymentMethodId": "p1a2y3m4-e5n6-7890-tdef-987654321abc",
  "metadata": {
    "couponCode": "SAVE20",
    "referralCode": "REF123",
```

```
    "notes": "Please handle with care"
  },
  "installmentPlanId": null,
  "groupInstanceId": null,
  "downPaymentPercent": 20,

}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
sessionType	string	Yes	Type of checkout session	enum: REGULAR_DIRECTLY, REGULAR_CART, GROUP_PURCHASE, INSTALLMENT
items	array	Conditional	Array of items to checkout. Required for REGULAR_DIRECTLY and GROUP_PURCHASE	For REGULAR_DIRECTLY: exactly 1 item. For GROUP_PURCHASE: exactly 1 item. Not used for REGULAR_CART
items[].productId	string (UUID)	Yes (if items provided)	Product identifier	Valid UUID format
items[].quantity	integer	Yes (if items provided)	Quantity to purchase	Min: 1, must not exceed product constraints
shippingAddressId	string (UUID)	Yes	User's shipping address identifier	Valid UUID format, must belong to authenticated user
shippingMethodId	string	Yes	Selected shipping method identifier	Must be valid shipping method
paymentMethodId	string (UUID)	No	Payment method identifier. If null, defaults to WALLET	Valid UUID format, must belong to authenticated user
metadata	object	No	Additional metadata for the session	Key-value pairs for coupons, referrals, notes, etc.
installmentPlanId	string (UUID)	No	Installment plan identifier (for INSTALLMENT type only)	Valid UUID format - PLACEHOLDER FOR FUTURE

Parameter	Type	Required	Description	Validation
groupInstanceId	string (UUID)	No	Group instance to join (for GROUP_PURCHASE type only)	Valid UUID format, group must be joinable

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Checkout session created successfully",
  "action_time": "2025-10-02T14:30:45",
  "data": {
    "sessionId": "c1d2e3f4-a5b6-7890-cdef-123456789abc",
    "sessionType": "REGULAR_DIRECTLY",
    "status": "PENDING_PAYMENT",
    "customerId": "u1s2e3r4-i5d6-7890-abcd-ef1234567890",
    "customerUserName": "john_doe",
    "items": [
      {
        "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
        "productName": "Premium Wireless Headphones",
        "productSlug": "premium-wireless-headphones",
        "productImage": "https://cdn.nextgate.com/products/headphones-001.jpg",
        "quantity": 2,
        "unitPrice": 150000.00,
        "discountAmount": 20000.00,
        "subtotal": 300000.00,
        "tax": 0.00,
        "total": 280000.00,
        "shopId": "s1h2o3p4-i5d6-7890-abcd-ef1234567890",
        "shopName": "TechWorld Electronics",
        "shopLogo": "https://cdn.nextgate.com/shops/techworld-logo.jpg",
        "availableForCheckout": true,
        "availableQuantity": 50
      }
    ],
    "pricing": {
      "subtotal": 300000.00,
      "discount": 20000.00,
```

```
  "shippingCost": 5000.00,
  "tax": 0.00,
  "total": 285000.00,
  "currency": "TZS"
},
"shippingAddress": {
  "fullName": "John Doe",
  "addressLine1": "123 Main Street",
  "addressLine2": "Apartment 4B",
  "city": "Dar es Salaam",
  "state": "Dar es Salaam Region",
  "postalCode": "12345",
  "country": "Tanzania",
  "phone": "+255123456789"
},
"billingAddress": {
  "sameAsShipping": false,
  "fullName": "John Doe",
  "addressLine1": "456 Business Ave",
  "city": "Dar es Salaam",
  "state": "Dar es Salaam Region",
  "postalCode": "12346",
  "country": "Tanzania"
},
"shippingMethod": {
  "id": "standard-shipping",
  "name": "Standard Shipping",
  "carrier": "DHL",
  "cost": 5000.00,
  "estimatedDays": "3-5 business days",
  "estimatedDelivery": "2025-10-07T14:30:45"
},
"paymentIntent": {
  "provider": "WALLET",
  "clientSecret": null,
  "paymentMethods": ["WALLET"],
  "status": "READY"
},
"paymentAttempts": [],
"inventoryHeld": true,
```

```
    "inventoryHoldExpiresAt": "2025-10-02T14:45:45",
    "metadata": {
      "couponCode": "SAVE20",
      "referralCode": "REF123",
      "notes": "Please handle with care"
    },
    "expiresAt": "2025-10-02T14:45:45",
    "createdAt": "2025-10-02T14:30:45",
    "updatedAt": "2025-10-02T14:30:45",
    "completedAt": null,
    "createdOrderId": null,
    "cartId": null
  }
}
```


Success Response Fields:

Field	Description
sessionId	Unique identifier for the checkout session
sessionType	Type of checkout (REGULAR_DIRECTLY, REGULAR_CART, GROUP_PURCHASE, INSTALLMENT)
status	Current status of the session (PENDING_PAYMENT, PAYMENT_PROCESSING, PAYMENT_COMPLETED, PAYMENT_FAILED, EXPIRED, CANCELLED, COMPLETED)
customerId	User ID who created the session
customerUserName	Username of the customer
items	Array of checkout items with product details, pricing, and availability
items[].productId	Product unique identifier
items[].productName	Product display name
items[].productSlug	URL-friendly product identifier
items[].productImage	Primary product image URL
items[].quantity	Quantity being purchased
items[].unitPrice	Price per unit in TZS
items[].discountAmount	Total discount applied to this item
items[].subtotal	Unit price × quantity
items[].tax	Tax amount for this item
items[].total	Final total for this item (subtotal + tax - discount)

Field	Description
items[].shopId	Shop identifier selling this product
items[].shopName	Shop display name
items[].shopLogo	Shop logo URL
items[].availableForCheckout	Whether product is currently available
items[].availableQuantity	Current stock quantity available
pricing	Summary of all pricing calculations
pricing.subtotal	Sum of all item subtotals
pricing.discount	Total discount applied
pricing.shippingCost	Shipping fee
pricing.tax	Total tax amount
pricing.total	Final amount to be paid (subtotal + shippingCost + tax - discount)
pricing.currency	Currency code (TZS)
shippingAddress	Complete shipping address details
billingAddress	Billing address (may differ from shipping)
billingAddress.sameAsShipping	Whether billing address matches shipping address
shippingMethod	Selected shipping method details
shippingMethod.id	Shipping method identifier
shippingMethod.name	Display name of shipping method
shippingMethod.carrier	Carrier company name
shippingMethod.cost	Shipping fee in TZS
shippingMethod.estimatedDays	Estimated delivery timeframe
shippingMethod.estimatedDelivery	Estimated delivery datetime
paymentIntent	Payment processing details
paymentIntent.provider	Payment provider (WALLET, CASH_ON_DELIVERY, CREDIT_CARD, etc.)
paymentIntent.clientSecret	Client secret for payment processing (null for WALLET/COD)
paymentIntent.paymentMethods	List of available payment methods
paymentIntent.status	Payment intent status (READY, PENDING, etc.)
paymentAttempts	Array of payment attempts made on this session
inventoryHeld	Whether inventory is currently held for this session
inventoryHoldExpiresAt	When the inventory hold will be released

Field	Description
metadata	Custom metadata provided during creation
expiresAt	When this checkout session expires
createdAt	Session creation timestamp
updatedAt	Last update timestamp
completedAt	Completion timestamp (null until completed)
createdOrderId	Order ID created after successful payment (null until completed)
cartId	Cart ID if session was created from cart (null for REGULAR_DIRECTLY)

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Validation failed",
  "action_time": "2025-10-02T14:30:45",
  "data": {
    "sessionType": "must not be null",
    "items": "Items are required for checkout",
    "shippingAddressId": "must not be null"
  }
}
```

Standard Error Types:

Application-Level Exceptions (400-499)

- `400 BAD_REQUEST`: Invalid request data, validation errors, business rule violations
- `401 UNAUTHORIZED`: Missing, invalid, or expired authentication token
- `403 FORBIDDEN`: Insufficient permissions or verification required
- `404 NOT_FOUND`: Product, payment method, shipping address, or other resource not found
- `422 UNPROCESSABLE_ENTITY`: Validation errors with detailed field information

Server-Level Exceptions (500+)

- `500 INTERNAL_SERVER_ERROR`: Unexpected server errors

Error Response Examples:

Bad Request - Invalid Session Type (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "REGULAR_DIRECTLY checkout supports only 1 item. Use REGULAR_CART for multiple items.",
  "action_time": "2025-10-02T14:30:45",
  "data": "REGULAR_DIRECTLY checkout supports only 1 item. Use REGULAR_CART for multiple items."
}
```


Bad Request - Insufficient Inventory (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient stock. Available: 3, Requested: 5",
  "action_time": "2025-10-02T14:30:45",
  "data": "Insufficient stock. Available: 3, Requested: 5"
}
```


Bad Request - Insufficient Wallet Balance (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS",
  "action_time": "2025-10-02T14:30:45",
  "data": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS"
}
```


Unauthorized - Token Missing (401):


```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:30:45",
  "data": "Authentication token is required"
}
```

Not Found - Product Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Product not found",
  "action_time": "2025-10-02T14:30:45",
  "data": "Product not found"
}
```


Not Found - Payment Method Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Payment method not found or does not belong to you",
  "action_time": "2025-10-02T14:30:45",
  "data": "Payment method not found or does not belong to you"
}
```


Validation Error (422):


```
{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-10-02T14:30:45",
  "data": {
    "sessionType": "must not be null",
    "items[0].quantity": "must be greater than or equal to 1",
    "shippingAddressId": "must not be null"
  }
}
```


---

2. Get Checkout Session by ID

Purpose: Retrieves detailed information about a specific checkout session by its ID. Only the session owner can access their session.

Endpoint: `GET` `{base_url}/checkout-sessions/{sessionId}`

Access Level: □□ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
sessionId	string (UUID)	Yes	Unique identifier of the checkout session	Valid UUID format

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Checkout session retrieved successfully",
  "action_time": "2025-10-02T14:35:45",
  "data": {
    "sessionId": "c1d2e3f4-a5b6-7890-cdef-123456789abc",
    "sessionType": "REGULAR_DIRECTLY",
    "status": "PENDING_PAYMENT",
    "customerId": "u1s2e3r4-i5d6-7890-abcd-ef1234567890",
    "customerUserName": "john_doe",
    "items": [
      {
        "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
        "productName": "Premium Wireless Headphones",
        "productSlug": "premium-wireless-headphones",
        "productImage": "https://cdn.nextgate.com/products/headphones-001.jpg",
        "quantity": 2,
        "unitPrice": 150000.00,
        "discountAmount": 20000.00,
        "subtotal": 300000.00,
        "tax": 0.00,
```

```
      "total": 280000.00,
      "shopId": "s1h2o3p4-i5d6-7890-abcd-ef1234567890",
      "shopName": "TechWorld Electronics",
      "shopLogo": "https://cdn.nextgate.com/shops/techworld-logo.jpg",
      "availableForCheckout": true,
      "availableQuantity": 50
    }
  ],
  "pricing": {
    "subtotal": 300000.00,
    "discount": 20000.00,
    "shippingCost": 5000.00,
    "tax": 0.00,
    "total": 285000.00,
    "currency": "TZS"
  },
  "shippingAddress": {
    "fullName": "John Doe",
    "addressLine1": "123 Main Street",
    "addressLine2": "Apartment 4B",
    "city": "Dar es Salaam",
    "state": "Dar es Salaam Region",
    "postalCode": "12345",
    "country": "Tanzania",
    "phone": "+255123456789"
  },
  "billingAddress": {
    "sameAsShipping": false,
    "fullName": "John Doe",
    "addressLine1": "456 Business Ave",
    "city": "Dar es Salaam",
    "state": "Dar es Salaam Region",
    "postalCode": "12346",
    "country": "Tanzania"
  },
  "shippingMethod": {
    "id": "standard-shipping",
    "name": "Standard Shipping",
    "carrier": "DHL",
    "cost": 5000.00,
```

```
      "estimatedDays": "3-5 business days",
      "estimatedDelivery": "2025-10-07T14:30:45"
    },
    "paymentIntent": {
      "provider": "WALLET",
      "clientSecret": null,
      "paymentMethods": ["WALLET"],
      "status": "READY"
    },
    "paymentAttempts": [],
    "inventoryHeld": true,
    "inventoryHoldExpiresAt": "2025-10-02T14:45:45",
    "metadata": {
      "couponCode": "SAVE20"
    },
    "expiresAt": "2025-10-02T14:45:45",
    "createdAt": "2025-10-02T14:30:45",
    "updatedAt": "2025-10-02T14:30:45",
    "completedAt": null,
    "createdOrderId": null,
    "cartId": null
  }
}
```


Success Response Fields:

Field	Description
sessionId	Unique identifier for the checkout session
sessionType	Type of checkout session
status	Current status of the session
customerId	User ID who owns the session
customerUserName	Username of the customer
items	Array of checkout items (see Create Checkout Session for details)
pricing	Pricing summary (see Create Checkout Session for details)
shippingAddress	Shipping address details
billingAddress	Billing address details
shippingMethod	Shipping method information

Field	Description
paymentIntent	Payment intent details
paymentAttempts	Array of all payment attempts made
inventoryHeld	Whether inventory is currently held
inventoryHoldExpiresAt	When inventory hold expires
metadata	Custom metadata
expiresAt	Session expiration datetime
createdAt	Session creation timestamp
updatedAt	Last update timestamp
completedAt	Completion timestamp (null if not completed)
createdOrderId	Order ID after successful payment (null if not completed)
cartId	Cart ID if created from cart

Error Response JSON Sample:


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T14:35:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```


Error Response Examples:

Not Found - Session Not Found or No Permission (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T14:35:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```


Unauthorized (401):


```
{
  "success": false,
```

```
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:35:45",
  "data": "Authentication token is required"
}
```


---

3. Get My Checkout Sessions

Purpose: Retrieves all checkout sessions belonging to the authenticated user, ordered by creation date (newest first).

Endpoint: `GET` `{base_url}/checkout-sessions`

Access Level: □□ Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Checkout sessions retrieved successfully",
  "action_time": "2025-10-02T14:40:45",
  "data": [
    {
      "sessionId": "c1d2e3f4-a5b6-7890-cdef-123456789abc",
      "sessionType": "REGULAR_DIRECTLY",
      "status": "PENDING_PAYMENT",
      "itemCount": 1,
      "totalAmount": 285000.00,
      "currency": "TZS",
      "expiresAt": "2025-10-02T14:45:45",
      "createdAt": "2025-10-02T14:30:45",
```

```
    "isExpired": false,
    "canRetryPayment": false,
    "itemPreviews": [
      {
        "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
        "productName": "Premium Wireless Headphones",
        "productImage": "https://cdn.nextgate.com/products/headphones-001.jpg",
        "quantity": 2,
        "unitPrice": 150000.00,
        "total": 280000.00,
        "shopName": "TechWorld Electronics"
      }
    ]
  },
  {
    "sessionId": "d2e3f4a5-b6c7-8901-def0-234567890abc",
    "sessionType": "REGULAR_CART",
    "status": "COMPLETED",
    "itemCount": 3,
    "totalAmount": 500000.00,
    "currency": "TZS",
    "expiresAt": "2025-10-01T10:15:30",
    "createdAt": "2025-10-01T10:00:30",
    "isExpired": false,
    "canRetryPayment": false,
    "itemPreviews": [
      {
        "productId": "p1r2o3d4-u5c6-7890-abcd-ef1234567890",
        "productName": "Smart Watch Series 5",
        "productImage": "https://cdn.nextgate.com/products/watch-005.jpg",
        "quantity": 1,
        "unitPrice": 350000.00,
        "total": 350000.00,
        "shopName": "Gadget Hub"
      },
      {
        "productId": "p2r3o4d5-u6c7-8901-bcde-f12345678901",
        "productName": "Wireless Mouse",
        "productImage": "https://cdn.nextgate.com/products/mouse-003.jpg",
        "quantity": 2,
```

```
          "unitPrice": 45000.00,
          "total": 90000.00,
          "shopName": "TechWorld Electronics"
        }
      ]
    },
    {
      "sessionId": "e3f4a5b6-c7d8-9012-ef01-345678901bcd",
      "sessionType": "PAYMENT_FAILED",
      "status": "PAYMENT_FAILED",
      "itemCount": 1,
      "totalAmount": 120000.00,
      "currency": "TZS",
      "expiresAt": "2025-10-02T15:00:00",
      "createdAt": "2025-10-02T14:45:00",
      "isExpired": false,
      "canRetryPayment": true,
      "itemPreviews": [
        {
          "productId": "p3r4o5d6-u7c8-9012-cdef-234567890123",
          "productName": "USB-C Cable 2m",
          "productImage": "https://cdn.nextgate.com/products/cable-002.jpg",
          "quantity": 5,
          "unitPrice": 15000.00,
          "total": 75000.00,
          "shopName": "Accessories World"
        }
      ]
    }
  ]
}
```


Success Response Fields:

Field	Description
sessionId	Unique identifier for the checkout session
sessionType	Type of checkout session
status	Current status of the session
itemCount	Number of items in the checkout

Field	Description
totalAmount	Total amount to be paid in TZS
currency	Currency code (TZS)
expiresAt	When this session expires
createdAt	When this session was created
isExpired	Whether the session has expired
canRetryPayment	Whether payment can be retried (true only if status is PAYMENT_FAILED and not expired)
itemPreviews	Array of preview information for items in checkout
itemPreviews[].productId	Product unique identifier
itemPreviews[].productName	Product name
itemPreviews[].productImage	Product image URL
itemPreviews[].quantity	Quantity being purchased
itemPreviews[].unitPrice	Price per unit
itemPreviews[].total	Total for this item
itemPreviews[].shopName	Shop selling this product

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:40:45",
  "data": "Authentication token is required"
}
```

Error Response Examples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:40:45",
  "data": "Authentication token is required"
}
```

4. Get My Active Checkout Sessions

Purpose: Retrieves only active checkout sessions (PENDING_PAYMENT or PAYMENT_FAILED status) that haven't expired yet, ordered by creation date (newest first).

Endpoint: `GET` `{base_url}/checkout-sessions/active`

Access Level: □□ Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Active checkout sessions retrieved successfully",
  "action_time": "2025-10-02T14:42:45",
  "data": [
    {
      "sessionId": "c1d2e3f4-a5b6-7890-cdef-123456789abc",
      "sessionType": "REGULAR_DIRECTLY",
      "status": "PENDING_PAYMENT",
      "itemCount": 1,
      "totalAmount": 285000.00,
      "currency": "TZS",
      "expiresAt": "2025-10-02T14:45:45",
      "createdAt": "2025-10-02T14:30:45",
      "isExpired": false,
      "canRetryPayment": false,
      "itemPreviews": [
        {
          "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
          "productName": "Premium Wireless Headphones",
```

```
          "productImage": "https://cdn.nextgate.com/products/headphones-001.jpg",
          "quantity": 2,
          "unitPrice": 150000.00,
          "total": 280000.00,
          "shopName": "TechWorld Electronics"
        }
      ]
    },
    {
      "sessionId": "e3f4a5b6-c7d8-9012-ef01-345678901bcd",
      "sessionType": "GROUP_PURCHASE",
      "status": "PAYMENT_FAILED",
      "itemCount": 1,
      "totalAmount": 120000.00,
      "currency": "TZS",
      "expiresAt": "2025-10-02T15:00:00",
      "createdAt": "2025-10-02T14:45:00",
      "isExpired": false,
      "canRetryPayment": true,
      "itemPreviews": [
        {
          "productId": "p3r4o5d6-u7c8-9012-cdef-234567890123",
          "productName": "USB-C Cable 2m",
          "productImage": "https://cdn.nextgate.com/products/cable-002.jpg",
          "quantity": 5,
          "unitPrice": 15000.00,
          "total": 75000.00,
          "shopName": "Accessories World"
        }
      ]
    }
  ]
}
```


Success Response Fields:

Field	Description
sessionId	Unique identifier for the checkout session
sessionType	Type of checkout session

Field	Description
status	Current status (PENDING_PAYMENT or PAYMENT_FAILED only)
itemCount	Number of items in the checkout
totalAmount	Total amount to be paid in TZS
currency	Currency code (TZS)
expiresAt	When this session expires
createdAt	When this session was created
isExpired	Always false for active sessions
canRetryPayment	Whether payment can be retried
itemPreviews	Array of preview information for items (see Get My Checkout Sessions for details)

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:42:45",
  "data": "Authentication token is required"
}
```

Error Response Examples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T14:42:45",
  "data": "Authentication token is required"
}
```

---

5. Update Checkout Session

Purpose: Updates an existing checkout session. Can modify shipping address, shipping method, payment method, or metadata. Only sessions in PENDING_PAYMENT or PAYMENT_FAILED status can be updated.

Endpoint: PATCH `{base_url}/checkout-sessions/{sessionId}`

Access Level: □□ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user
Content-Type	string	Yes	Must be `application/json`

Path Parameters:

Parameter	Type	Required	Description	Validation
sessionId	string (UUID)	Yes	Unique identifier of the checkout session	Valid UUID format

Request JSON Sample:

```
{
  "shippingAddressId": "f9e8d7c6-b5a4-3210-fedc-ba9876543210",
  "shippingMethodId": "express-shipping",
  "paymentMethodId": "p9a8y7m6-e5n4-3210-tdef-ba9876543210",
  "metadata": {
    "giftWrapping": true,
    "giftMessage": "Happy Birthday!"
  }
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
shippingAddressId	string (UUID)	No	New shipping address identifier	Valid UUID format, must belong to user
shippingMethodId	string	No	New shipping method identifier	Must be valid shipping method

Parameter	Type	Required	Description	Validation
paymentMethodId	string (UUID)	No	New payment method identifier	Valid UUID format, must belong to user
metadata	object	No	Additional or updated metadata	Key-value pairs, merged with existing metadata

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Checkout session updated successfully",
  "action_time": "2025-10-02T14:50:45",
  "data": {
    "sessionId": "c1d2e3f4-a5b6-7890-cdef-123456789abc",
    "sessionType": "REGULAR_DIRECTLY",
    "status": "PENDING_PAYMENT",
    "customerId": "u1s2e3r4-i5d6-7890-abcd-ef1234567890",
    "customerUserName": "john_doe",
    "items": [
      {
        "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
        "productName": "Premium Wireless Headphones",
        "productSlug": "premium-wireless-headphones",
        "productImage": "https://cdn.nextgate.com/products/headphones-001.jpg",
        "quantity": 2,
        "unitPrice": 150000.00,
        "discountAmount": 20000.00,
        "subtotal": 300000.00,
        "tax": 0.00,
        "total": 280000.00,
        "shopId": "s1h2o3p4-i5d6-7890-abcd-ef1234567890",
        "shopName": "TechWorld Electronics",
        "shopLogo": "https://cdn.nextgate.com/shops/techworld-logo.jpg",
        "availableForCheckout": true,
        "availableQuantity": 50
      }
    ],
    "pricing": {
      "subtotal": 300000.00,
```

```
    "discount": 20000.00,
    "shippingCost": 8000.00,
    "tax": 0.00,
    "total": 288000.00,
    "currency": "TZS"
  },
  "shippingAddress": {
    "fullName": "John Doe",
    "addressLine1": "789 New Address Street",
    "addressLine2": null,
    "city": "Dar es Salaam",
    "state": "Dar es Salaam Region",
    "postalCode": "12347",
    "country": "Tanzania",
    "phone": "+255987654321"
  },
  "billingAddress": {
    "sameAsShipping": false,
    "fullName": "John Doe",
    "addressLine1": "321 Payment Ave",
    "city": "Dar es Salaam",
    "state": "Dar es Salaam Region",
    "postalCode": "12348",
    "country": "Tanzania"
  },
  "shippingMethod": {
    "id": "express-shipping",
    "name": "Express Shipping",
    "carrier": "DHL",
    "cost": 8000.00,
    "estimatedDays": "1-2 business days",
    "estimatedDelivery": "2025-10-04T14:50:45"
  },
  "paymentIntent": {
    "provider": "CREDIT_CARD",
    "clientSecret": "pi_1234567890abcdef",
    "paymentMethods": ["CREDIT_CARD", "DEBIT_CARD"],
    "status": "READY"
  },
  "paymentAttempts": [],
```

```
    "inventoryHeld": true,
    "inventoryHoldExpiresAt": "2025-10-02T15:05:45",
    "metadata": {
      "couponCode": "SAVE20",
      "giftWrapping": true,
      "giftMessage": "Happy Birthday!"
    },
    "expiresAt": "2025-10-02T15:05:45",
    "createdAt": "2025-10-02T14:30:45",
    "updatedAt": "2025-10-02T14:50:45",
    "completedAt": null,
    "createdOrderId": null,
    "cartId": null
  }
}
```


Success Response Fields:

Field	Description
All fields	Same as Create Checkout Session response, with updated values
updatedAt	Timestamp showing when the session was last updated
pricing.shippingCost	May be recalculated if shipping method changed
pricing.total	May be recalculated if shipping method changed

Error Response JSON Sample:


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot update - payment has been completed",
  "action_time": "2025-10-02T14:50:45",
  "data": "Cannot update - payment has been completed"
}
```


Error Response Examples:

Bad Request - Cannot Update Completed Session (400):


```
{
  "success": false,
```

```
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot update a completed checkout session",
  "action_time": "2025-10-02T14:50:45",
  "data": "Cannot update a completed checkout session"
}
```


Bad Request - Cannot Update Cancelled Session (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot update a cancelled checkout session",
  "action_time": "2025-10-02T14:50:45",
  "data": "Cannot update a cancelled checkout session"
}
```


Bad Request - Cannot Update Expired Session (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot update an expired checkout session",
  "action_time": "2025-10-02T14:50:45",
  "data": "Cannot update an expired checkout session"
}
```


Not Found - Session Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T14:50:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```


Not Found - Payment Method Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
```

```
  "message": "Payment method not found or does not belong to you",
  "action_time": "2025-10-02T14:50:45",
  "data": "Payment method not found or does not belong to you"
}
```

---

6. Cancel Checkout Session

Purpose: Cancels an existing checkout session and releases held inventory. Cannot cancel sessions that are completed or have successful payment.

Endpoint: DELETE `{base_url}/checkout-sessions/{sessionId}/cancel`

Access Level: □□ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
sessionId	string (UUID)	Yes	Unique identifier of the checkout session to cancel	Valid UUID format

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Checkout session cancelled successfully",
  "action_time": "2025-10-02T14:55:45",
  "data": null
}
```

Success Response Fields:

Field	Description
data	Always null for cancel operations

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot cancel a completed checkout session",
  "action_time": "2025-10-02T14:55:45",
  "data": "Cannot cancel a completed checkout session"
}
```

Error Response Examples:

Bad Request - Cannot Cancel Completed Session (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot cancel a completed checkout session",
  "action_time": "2025-10-02T14:55:45",
  "data": "Cannot cancel a completed checkout session"
}
```

Bad Request - Cannot Cancel with Successful Payment (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot cancel - payment has been completed. Please contact support.",
  "action_time": "2025-10-02T14:55:45",
  "data": "Cannot cancel - payment has been completed. Please contact support."
}
```

Bad Request - Already Cancelled (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Checkout session is already cancelled",
```

```
  "action_time": "2025-10-02T14:55:45",
  "data": "Checkout session is already cancelled"
}
```

Not Found - Session Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T14:55:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```

---

7. Process Payment

Purpose: Initiates payment processing for a checkout session. Validates session status, inventory availability, and payment method before processing. Delegates to the appropriate payment provider based on payment method type.

Endpoint: POST `{base_url}/checkout-sessions/{sessionId}/process-payment`

Access Level: □□ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
sessionId	string (UUID)	Yes	Unique identifier of the checkout session	Valid UUID format

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Payment processed successfully",
  "action_time": "2025-10-02T15:00:45",
  "data": {
    "success": true,
    "paymentProvider": "WALLET",
    "transactionId": "txn_1234567890abcdef",
    "amount": 285000.00,
    "currency": "TZS",
    "status": "COMPLETED",
    "message": "Payment successful",
    "paymentMethod": "WALLET",
    "processedAt": "2025-10-02T15:00:45",
    "orderId": "ord_9876543210fedcba",
    "receiptUrl": null
  }
}
```


Success Response Fields:

Field	Description
success	Whether the payment was successful
paymentProvider	Payment provider used (WALLET, CASH_ON_DELIVERY, CREDIT_CARD, etc.)
transactionId	Unique transaction identifier
amount	Amount processed in TZS
currency	Currency code (TZS)
status	Payment status (COMPLETED, PENDING, PROCESSING, FAILED)
message	Human-readable payment result message
paymentMethod	Payment method type used
processedAt	Timestamp of payment processing
orderId	Order ID created after successful payment
receiptUrl	URL to payment receipt (may be null for some payment methods)

Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot process payment - session status: PAYMENT_COMPLETED",
  "action_time": "2025-10-02T15:00:45",
  "data": "Cannot process payment - session status: PAYMENT_COMPLETED"
}
```


Error Response Examples:

Bad Request - Invalid Session Status (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot process payment - session status: PAYMENT_COMPLETED",
  "action_time": "2025-10-02T15:00:45",
  "data": "Cannot process payment - session status: PAYMENT_COMPLETED"
}
```


Bad Request - Session Expired (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Checkout session has expired",
  "action_time": "2025-10-02T15:00:45",
  "data": "Checkout session has expired"
}
```


Bad Request - Insufficient Wallet Balance (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS",
  "action_time": "2025-10-02T15:00:45",
  "data": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS"
}
```


Not Found - Session Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T15:00:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```


---

8. Retry Payment

Purpose: Retries payment for a failed checkout session. Validates that the session is in PAYMENT_FAILED status, hasn't exceeded maximum retry attempts (5), and hasn't expired. Re-validates inventory availability and extends session expiration before retrying.

Endpoint: `POST` `{base_url}/checkout-sessions/{sessionId}/retry-payment`

Access Level: □□ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
sessionId	string (UUID)	Yes	Unique identifier of the checkout session	Valid UUID format

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Payment retry successful",
  "action_time": "2025-10-02T15:10:45",
  "data": {
    "success": true,
```

```
    "paymentProvider": "WALLET",
    "transactionId": "txn_retry_1234567890abcdef",
    "amount": 285000.00,
    "currency": "TZS",
    "status": "COMPLETED",
    "message": "Payment successful on retry",
    "paymentMethod": "WALLET",
    "processedAt": "2025-10-02T15:10:45",
    "orderId": "ord_retry_9876543210fedcba",
    "receiptUrl": null
  }
}
```


Success Response Fields:

Field	Description
success	Whether the payment retry was successful
paymentProvider	Payment provider used
transactionId	Unique transaction identifier for this retry
amount	Amount processed in TZS
currency	Currency code (TZS)
status	Payment status (COMPLETED, PENDING, PROCESSING, FAILED)
message	Human-readable payment result message
paymentMethod	Payment method type used
processedAt	Timestamp of payment processing
orderId	Order ID created after successful payment
receiptUrl	URL to payment receipt (may be null)

Error Response JSON Sample:


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot retry payment - session status: PENDING_PAYMENT. Expected:
PAYMENT_FAILED",
  "action_time": "2025-10-02T15:10:45",
  "data": "Cannot retry payment - session status: PENDING_PAYMENT. Expected: PAYMENT_FAILED"
```

```
}
```

Error Response Examples:

Bad Request - Invalid Status (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot retry payment - session status: PENDING_PAYMENT. Expected:
PAYMENT_FAILED",
  "action_time": "2025-10-02T15:10:45",
  "data": "Cannot retry payment - session status: PENDING_PAYMENT. Expected: PAYMENT_FAILED"
}
```

Bad Request - Session Expired (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Checkout session has expired. Please create a new checkout session.",
  "action_time": "2025-10-02T15:10:45",
  "data": "Checkout session has expired. Please create a new checkout session."
}
```

Bad Request - Max Retry Attempts Exceeded (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Maximum payment attempts (5) exceeded. Please create a new checkout session.",
  "action_time": "2025-10-02T15:10:45",
  "data": "Maximum payment attempts (5) exceeded. Please create a new checkout session."
}
```

Bad Request - Product No Longer Available (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Product 'Premium Wireless Headphones' is no longer available in requested
```

```
quantity. Please create a new checkout session.",
  "action_time": "2025-10-02T15:10:45",
  "data": "Product 'Premium Wireless Headphones' is no longer available in requested quantity.
Please create a new checkout session."
}
```


Bad Request - Insufficient Wallet Balance (400):


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS. Please
top up your wallet or update your payment method.",
  "action_time": "2025-10-02T15:10:45",
  "data": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS. Please
top up your wallet or update your payment method."
}
```


Not Found - Session Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Checkout session not found or you don't have permission to access it",
  "action_time": "2025-10-02T15:10:45",
  "data": "Checkout session not found or you don't have permission to access it"
}
```

Checkout Session Types

REGULAR_DIRECTLY

Direct product purchase without using a cart. Must include exactly 1 item in the request.

Characteristics:

- Single item purchase flow
- Items array must contain exactly 1 product
- Does not link to any cart

- cartId field will be null in response
- Best for "Buy Now" buttons or quick checkout flows

Example Use Case: User clicks "Buy Now" on a product detail page and proceeds directly to checkout.

REGULAR_CART

Checkout from existing shopping cart. Items are retrieved from the user's active cart.

Characteristics:

- Multi-item purchase flow
- Items array not required in request (fetched from cart automatically)
- Links to user's cart via cartId
- Cart is validated before checkout creation
- cartId field will be populated in response

Example Use Case: User adds multiple products to cart, reviews cart, and clicks "Proceed to Checkout".

GROUP_PURCHASE

Special checkout type for group buying where multiple users purchase the same product at a discounted group price.

Characteristics:

- Single item purchase at group price
- Items array must contain exactly 1 product
- Payment method MUST be WALLET (other methods not supported)
- Can join existing group via groupInstanceId or create new group
- Uses product's groupPrice instead of regular price
- Requires product to have groupBuyingEnabled = true
- Quantity cannot exceed product's groupMaxSize
- Group has time limit defined by product's groupTimeLimitHours

Example Use Case: User sees a product available for group buying at 80,000 TZS (regular price: 150,000 TZS). User can either start a new group or join an existing group with available seats.

Request Example (Create New Group):

```
{
  "sessionType": "GROUP_PURCHASE",
  "items": [{
```

```
    "productId": "prod-uuid",
    "quantity": 2
  }],
  "shippingAddressId": "addr-uuid",
  "shippingMethodId": "standard",
  "paymentMethodId": null,
  "groupInstanceId": null
}
```

Request Example (Join Existing Group):

```
{
  "sessionType": "GROUP_PURCHASE",
  "items": [{
    "productId": "prod-uuid",
    "quantity": 3
  }],
  "shippingAddressId": "addr-uuid",
  "shippingMethodId": "standard",
  "paymentMethodId": null,
  "groupInstanceId": "group-uuid-to-join"
}
```

INSTALLMENT

Status: PLACEHOLDER - Not Yet Implemented

Future checkout type for installment payment plans. Will allow users to split payments into multiple installments.

Planned Characteristics:

- Will require installmentPlanId in request
- Payment spread over multiple periods
- May require down payment
- Interest rates may apply based on plan
- Credit check may be required

Current Behavior: Returns error: "INSTALLMENT checkout not implemented yet"

Checkout Session Status Flow

Status Definitions

Status	Description	Can Update?	Can Cancel?	Can Pay?	Can Retry?
PENDING_PAYMENT	Session created, awaiting payment	Yes	Yes	Yes	No
PAYMENT_PROCESSING	Payment in progress	No	No	No	No
PAYMENT_FAILED	Payment failed, can retry	Yes	Yes	No	Yes
PAYMENT_COMPLETED	Payment successful	No	No	No	No
EXPIRED	Session expired	No	No	No	No
CANCELLED	User cancelled	No	No	No	No
COMPLETED	Order created successfully	No	No	No	No

Status Transition Flow

```
PENDING_PAYMENT
    ├─> PAYMENT_PROCESSING (when payment initiated)
    │       ├─> PAYMENT_COMPLETED (on success)
    │       │       └─> COMPLETED (after order creation)
    │       └─> PAYMENT_FAILED (on failure, can retry max 5 times)
    │               ├─> PENDING_PAYMENT (when retry initiated)
    │               └─> EXPIRED (after max retries or session timeout)
    ├─> CANCELLED (user cancels)
    └─> EXPIRED (15 minutes timeout)
```

Payment Methods Supported

WALLET

Internal wallet system for storing and using funds.

Requirements:

- User must have active wallet
- Sufficient balance required
- Balance checked before payment processing
- Instant payment processing
- No additional fees

Payment Flow:

1. Validate wallet is active
2. Check sufficient balance
3. Deduct amount from wallet
4. Create transaction record
5. Create order

Default Behavior: If no paymentMethodId provided in request, WALLET is used as default.

CASH_ON_DELIVERY

Pay in cash when order is delivered.

Requirements:

- No pre-payment required
- Payment collected by delivery agent
- Order created immediately
- May have geographic restrictions

Payment Flow:

1. Create payment intent with status "PENDING"
2. Create order immediately
3. Payment marked as "COD"
4. Delivery agent collects payment on delivery

CREDIT_CARD / DEBIT_CARD

Status: Planned - Not Yet Implemented

Future support for credit and debit card payments.

Planned Features:

- Card tokenization for security

- PCI compliance
- 3D Secure authentication
- Saved cards support
- International cards support

MNO_PAYMENT (Mobile Money)

Status: Planned - Not Yet Implemented

Future support for mobile money payments (M-Pesa, Tigo Pesa, Airtel Money, etc.).

Planned Features:

- USSD push notifications
- Callback confirmation
- Transaction reconciliation
- Multiple operators support

Other Payment Methods

Additional payment methods (PayPal, Bank Transfer, Cryptocurrency, Gift Card) are placeholders for future implementation.

Inventory Management

Inventory Hold Mechanism

When a checkout session is created, inventory is automatically held to prevent overselling.

Hold Behavior:

- Inventory held for 15 minutes (matches session expiration)
- Held quantity deducted from available stock
- Other users cannot purchase held inventory
- Hold automatically released on:
  - Session expiration
  - Session cancellation
  - Payment failure (after 5 retry attempts)
- Hold converted to permanent deduction on successful payment

Hold Fields:

- `inventoryHeld`: Boolean indicating if inventory is currently held
- `inventoryHoldExpiresAt`: Timestamp when hold will be released

Example:

```
{
  "inventoryHeld": true,
  "inventoryHoldExpiresAt": "2025-10-02T14:45:45"
}
```

Inventory Validation

Before processing payment or retry, system validates:

1. Product still exists and is active
2. Product still in stock
3. Requested quantity still available
4. Product hasn't been deleted

If validation fails, user must create new checkout session with current availability.

Session Expiration

Expiration Rules

Default Expiration: 15 minutes from creation

Expiration Extended When:

- Payment retry initiated (adds 15 minutes)
- Session updated successfully (adds 15 minutes)

What Happens on Expiration:

1. Session status changes to EXPIRED
2. Held inventory released back to available stock
3. Session cannot be updated or paid
4. Session cannot be cancelled (already effectively ended)
5. User must create new checkout session

Checking Expiration:

- `expiresAt` field shows expiration timestamp
- `isExpired` field (in summary responses) shows if session expired

Preventing Expiration

To prevent session expiration:

1. Complete payment before expiration
2. Update session if needed (extends timer)
3. If expired, create new checkout session

Best Practice: Display countdown timer to users showing time remaining before expiration.

Payment Attempts Tracking

Attempt Limits

Maximum Attempts: 5 per session

Attempt Tracking: Each payment attempt recorded with:

- Attempt number (1-5)
- Payment method used
- Status (SUCCESS, FAILED, RETRY_INITIATED)
- Error message (if failed)
- Timestamp
- Transaction ID (if available)

Example Payment Attempts:


```
{
  "paymentAttempts": [
    {
      "attemptNumber": 1,
      "paymentMethod": "WALLET",
      "status": "FAILED",
      "errorMessage": "Insufficient wallet balance",
      "attemptedAt": "2025-10-02T14:35:00",
      "transactionId": null
    },
```

```
    {
      "attemptNumber": 2,
      "paymentMethod": "WALLET",
      "status": "RETRY_INITIATED",
      "errorMessage": null,
      "attemptedAt": "2025-10-02T14:40:00",
      "transactionId": null
    },
    {
      "attemptNumber": 3,
      "paymentMethod": "WALLET",
      "status": "SUCCESS",
      "errorMessage": null,
      "attemptedAt": "2025-10-02T14:42:00",
      "transactionId": "txn_1234567890abcdef"
    }
  ]
}
```

After Max Attempts

When 5 attempts exceeded:

1. Session status set to EXPIRED
2. Inventory released
3. Cannot retry payment
4. User must create new checkout session

Metadata Usage

Purpose

Metadata allows storing custom key-value data with checkout sessions for business-specific needs.

Common Use Cases

Coupons & Discounts:

```
{
  "metadata": {
    "couponCode": "SAVE20",
    "couponDiscount": 20000,
    "couponAppliedAt": "2025-10-02T14:30:00"
  }
}
```

Referral Tracking:

```
{
  "metadata": {
    "referralCode": "REF123",
    "referredBy": "user-uuid",
    "referralBonus": 5000
  }
}
```

Gift Options:

```
{
  "metadata": {
    "isGift": true,
    "giftWrapping": true,
    "giftMessage": "Happy Birthday!",
    "giftRecipientName": "Jane Doe"
  }
}
```

Delivery Instructions:

```
{
  "metadata": {
    "deliveryInstructions": "Leave package at front desk",
    "deliveryTimePreference": "morning",
    "contactOnArrival": true
  }
}
```

Group Purchase:

```
{
  "metadata": {
    "groupInstanceId": "group-uuid",
    "isGroupLeader": true,
    "groupCreatedAt": "2025-10-02T14:30:00"
  }
}
```

Marketing Attribution:

```
{
  "metadata": {
    "source": "facebook_ad",
    "campaign": "summer_sale_2025",
    "medium": "social_media"
  }
}
```

Metadata Rules

- Stored as JSON object
- Key-value pairs with string keys
- Values can be strings, numbers, booleans, or nested objects
- No size limit enforced at API level (reasonable usage expected)
- Merged with existing metadata on update (not replaced)
- Persisted with session throughout lifecycle
- Accessible in order after completion

Error Handling Best Practices

Client-Side Handling

Before Creating Checkout:

1. Validate user has shipping address
2. Check product availability
3. Verify payment method exists (if not using wallet)
4. Display clear pricing breakdown

During Checkout:

1. Show expiration countdown timer
2. Handle session expiration gracefully
3. Provide clear error messages
4. Suggest actions (e.g., "Top up wallet" if insufficient balance)

Payment Failures:

1. Display specific error reason
2. Show remaining retry attempts
3. Offer to update payment method
4. Suggest alternative payment methods

Common Error Scenarios

Insufficient Inventory:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient stock. Available: 3, Requested: 5"
}
```

Action: Reduce quantity or create new session with available quantity.

Insufficient Wallet Balance:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 285000 TZS, Available: 150000 TZS"
}
```

Action: Top up wallet or change payment method.

Session Expired:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Checkout session has expired"
}
```

Action: Create new checkout session.

Product Unavailable:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Product is not available for purchase"
}
```

Action: Remove product from cart or select alternative product.

Max Retries Exceeded:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Maximum payment attempts (5) exceeded. Please create a new checkout session."
}
```

Action: Create new checkout session, possibly with different payment method.

Integration Examples

Example 1: Direct Product Purchase Flow

Step 1: Create Checkout Session

```
POST /api/v1/checkout-sessions
Authorization: Bearer {token}
Content-Type: application/json

{
  "sessionType": "REGULAR_DIRECTLY",
  "items": [{
    "productId": "prod-uuid",
    "quantity": 1
  }],
  "shippingAddressId": "addr-uuid",
```

```
  "shippingMethodId": "standard",
  "paymentMethodId": null
}
```

Step 2: Process Payment

```
POST /api/v1/checkout-sessions/{sessionId}/process-payment
Authorization: Bearer {token}
```

Step 3: Check Order Created Response will include `orderId` field after successful payment.

Example 2: Cart Checkout Flow

Step 1: Create Checkout from Cart

```
POST /api/v1/checkout-sessions
Authorization: Bearer {token}
Content-Type: application/json

{
  "sessionType": "REGULAR_CART",
  "shippingAddressId": "addr-uuid",
  "shippingMethodId": "express",
  "metadata": {
    "couponCode": "SAVE20"
  }
}
```

Step 2: Review Session

```
GET /api/v1/checkout-sessions/{sessionId}
Authorization: Bearer {token}
```

Step 3: Update if Needed

```
PATCH /api/v1/checkout-sessions/{sessionId}
Authorization: Bearer {token}
Content-Type: application/json

{
  "shippingMethodId": "standard",
```

```
  "metadata": {
    "giftWrapping": true
  }
}
```

Step 4: Process Payment

```
POST /api/v1/checkout-sessions/{sessionId}/process-payment
Authorization: Bearer {token}
```

Example 3: Group Purchase Flow

Step 1: Create Group Purchase Session

```
POST /api/v1/checkout-sessions
Authorization: Bearer {token}
Content-Type: application/json

{
  "sessionType": "GROUP_PURCHASE",
  "items": [{
    "productId": "prod-uuid",
    "quantity": 2
  }],
  "shippingAddressId": "addr-uuid",
  "shippingMethodId": "standard",
  "groupInstanceId": null
}
```

Step 2: Process Payment (WALLET only)

```
POST /api/v1/checkout-sessions/{sessionId}/process-payment
Authorization: Bearer {token}
```

Example 4: Payment Retry Flow

Step 1: Get Active Sessions

```
GET /api/v1/checkout-sessions/active
Authorization: Bearer {token}
```

Step 2: Identify Failed Session Look for session with `status: "PAYMENT_FAILED"` and `canRetryPayment: true`

Step 3: Update Payment Method (Optional)

```
PATCH /api/v1/checkout-sessions/{sessionId}
Authorization: Bearer {token}
Content-Type: application/json

{
  "paymentMethodId": "new-payment-method-uuid"
}
```

Step 4: Retry Payment

```
POST /api/v1/checkout-sessions/{sessionId}/retry-payment
Authorization: Bearer {token}
```

Rate Limiting

Rate Limits:

- Create Checkout: 20 requests per minute per user
- Get Sessions: 60 requests per minute per user
- Process/Retry Payment: 10 requests per minute per user
- Update/Cancel: 30 requests per minute per user

Rate Limit Headers:

```
X-RateLimit-Limit: 20
X-RateLimit-Remaining: 15
X-RateLimit-Reset: 1696258800
```

Rate Limit Exceeded:

```
{
  "success": false,
  "httpStatus": "TOO_MANY_REQUESTS",
  "message": "Rate limit exceeded. Please try again later.",
  "action_time": "2025-10-02T15:30:45",
```

```
  "data": "Rate limit exceeded. Please try again later."
}
```

Revision #2
Created 2 October 2025 06:00:25 by Admin Qbit
Updated 11 December 2025 08:26:43 by Admin Qbit