Group Purchase

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 Group Purchase API enables collaborative buying where multiple users join together to purchase products at discounted group prices. Users can create new groups, join existing groups, transfer between groups, and track their participations. The system automatically handles seat management, expiration, and order creation when groups are completed.

Hints:

- Groups automatically expire based on product's `groupTimeLimitHours` setting
- Groups complete automatically when all seats are filled
- Users can join the same group multiple times to buy more seats (Hybrid Approach)
- Transfer between groups only allowed for same product, shop, and price
- Empty groups are automatically soft-deleted after all participants transfer out
- Group codes are auto-generated with format: GP-XXXXXX (6 random characters)
- Only WALLET payment method supported for group purchases
- Purchase and transfer history tracked for each participant
- Seats are released when users transfer out of groups

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, 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

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

Endpoints

1. Get Available Groups for Product

Purpose: Retrieves all available (open, not expired, not full) group purchase instances for a specific product.

Endpoint: `GET` `{base_url}/group-purchases/product/{productId}/available`

Access Level: □□ Public (No Authentication Required)

Authentication: None

Request Headers:

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

Path Parameters:

Parameter	Type	Required	Description	Validation
productId	string (UUID)	Yes	Unique identifier of the product	Valid UUID format

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Available groups retrieved successfully",
  "action_time": "2025-10-02T14:30:45",
  "data": [
    {
      "groupInstanceId": "gp123456-7890-abcd-ef12-345678901234",
      "groupCode": "GP-A3X7K9",
      "productName": "Premium Wireless Headphones",
      "productPrimaryMedia": {
        "fileId": "9c1a2b3d-4e5f-6789-abcd-ef0123456789",
        "mediaType": "IMAGE",
        "status": "READY",
        "thumbUrl": "https://cdn.nextgate.com/products/headphones-001/thumb.webp",
        "mimeType": "image/webp"
      },
```

```
"shopName": "TechWorld Electronics",
"groupPrice": 80000.00,
"savingsPercentage": 46.67,
"currency": "TZS",
"totalSeats": 10,
"seatsOccupied": 4,
"seatsRemaining": 6,
"totalParticipants": 3,
"progressPercentage": 40.00,
"status": "OPEN",
"expiresAt": "2025-10-02T20:30:45",
"isExpired": false,
"isUserMember": false,
"participants": [
  {
    "userId": "user1234-5678-90ab-cdef-123456789012",
    "userName": "john_doe",
    "userProfileMedia": {
      "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "quantity": 2,
    "contributionPercentage": 50.00
  },
  {
    "userId": "user2345-6789-01bc-def1-234567890123",
    "userName": "jane_smith",
    "userProfileMedia": {
      "fileId": "4a5b6c7d-8e9f-0123-bcde-f34567890123",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/profiles/jane/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "quantity": 1,
    "contributionPercentage": 25.00
  },
  {
    "userId": "user3456-7890-12cd-ef12-345678901234",
```

```
      "userName": "bob_wilson",
      "userProfileMedia": null,
      "quantity": 1,
      "contributionPercentage": 25.00
    }
  ]
},
{
  "groupInstanceId": "gp234567-8901-bcde-f123-456789012345",
  "groupCode": "GP-B2Y8M5",
  "productName": "Premium Wireless Headphones",
  "productPrimaryMedia": {
    "fileId": "9c1a2b3d-4e5f-6789-abcd-ef0123456789",
    "mediaType": "IMAGE",
    "status": "READY",
    "thumbUrl": "https://cdn.nextgate.com/products/headphones-001/thumb.webp",
    "mimeType": "image/webp"
  },
  "shopName": "TechWorld Electronics",
  "groupPrice": 80000.00,
  "savingsPercentage": 46.67,
  "currency": "TZS",
  "totalSeats": 10,
  "seatsOccupied": 7,
  "seatsRemaining": 3,
  "totalParticipants": 5,
  "progressPercentage": 70.00,
  "status": "OPEN",
  "expiresAt": "2025-10-02T18:45:30",
  "isExpired": false,
  "isUserMember": true,
  "participants": [
    {
      "userId": "user4567-8901-23de-f234-567890123456",
      "userName": "alice_brown",
      "userProfileMedia": {
        "fileId": "5b6c7d8e-9f01-2345-cdef-456789012345",
        "status": "READY",
        "thumbUrl": "https://cdn.nextgate.com/profiles/alice/thumb.webp",
        "mimeType": "image/jpeg"
```

```
          },
          "quantity": 3,
          "contributionPercentage": 42.86
        }
      ]
    }
  ]
}
```

Success Response Fields:

Field	Description
groupInstanceId	Unique identifier for the group
groupCode	Human-readable group code (e.g., GP-A3X7K9)
productName	Name of the product in this group
productPrimaryMedia	Resolved primary media of the product — `{fileId, mediaType, status, thumbUrl, mimeType}`. `null` if the product has no media yet
shopName	Shop selling this product
groupPrice	Discounted price per unit in TZS
savingsPercentage	Percentage saved vs regular price
currency	Currency code (TZS)
totalSeats	Maximum number of seats in this group
seatsOccupied	Number of seats currently filled
seatsRemaining	Available seats remaining
totalParticipants	Number of unique participants
progressPercentage	Group completion percentage (0-100)
status	Group status (OPEN, COMPLETED, FAILED, DELETED)
expiresAt	When the group expires
isExpired	Whether the group has expired
isUserMember	Whether authenticated user is in this group
participants	Array of participant previews
participants[].userId	Participant's user ID
participants[].userName	Participant's username
participants[].userProfileMedia	Resolved profile media — `{fileId, status, thumbUrl, mimeType}`. `null` if no profile picture set

Field	Description
participants[].quantity	Number of seats this participant holds
participants[].contributionPercentage	Percentage of total seats this participant holds

Error Response Examples:

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"
}
```

---

2. Get Group by ID

Purpose: Retrieves detailed information about a specific group purchase instance including all participants and their histories.

Endpoint: `GET` `{base_url}/group-purchases/{groupId}`

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

Path Parameters:

Parameter	Type	Required	Description	Validation
groupId	string (UUID)	Yes	Unique identifier of the group	Valid UUID format

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Group retrieved successfully",
  "action_time": "2025-10-02T14:35:45",
  "data": {
    "groupInstanceId": "gp123456-7890-abcd-ef12-345678901234",
    "groupCode": "GP-A3X7K9",
    "productId": "prod1234-5678-90ab-cdef-123456789012",
    "productName": "Premium Wireless Headphones",
    "productPrimaryMedia": {
      "fileId": "9c1a2b3d-4e5f-6789-abcd-ef0123456789",
      "mediaType": "IMAGE",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/products/headphones-001/thumb.webp",
      "mimeType": "image/webp"
    },
    "shopId": "shop1234-5678-90ab-cdef-123456789012",
    "shopName": "TechWorld Electronics",
    "shopLogoMedia": {
      "fileId": "1d2e3f4a-5b6c-7890-defa-bc1234567890",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/shops/techworld/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "regularPrice": 150000.00,
    "groupPrice": 80000.00,
    "savingsAmount": 70000.00,
    "savingsPercentage": 46.67,
    "currency": "TZS",
    "totalSeats": 10,
    "seatsOccupied": 4,
    "seatsRemaining": 6,
    "totalParticipants": 3,
    "progressPercentage": 40.00,
    "status": "OPEN",
    "isExpired": false,
    "isFull": false,
    "initiatorId": "user1234-5678-90ab-cdef-123456789012",
    "initiatorName": "john_doe",
```

```
"durationHours": 24,
"createdAt": "2025-10-01T20:30:45",
"expiresAt": "2025-10-02T20:30:45",
"completedAt": null,
"maxPerCustomer": 5,
"isUserMember": true,
"myParticipantId": "part1234-5678-90ab-cdef-123456789012",
"myQuantity": 2,
"participants": [
  {
    "participantId": "part1234-5678-90ab-cdef-123456789012",
    "userId": "user1234-5678-90ab-cdef-123456789012",
    "userName": "john_doe",
    "userProfileMedia": {
      "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "quantity": 2,
    "totalPaid": 160000.00,
    "status": "ACTIVE",
    "joinedAt": "2025-10-01T20:30:45",
    "contributionPercentage": 50.00,
    "purchaseCount": 1,
    "hasTransferred": false,
    "purchaseHistory": [
      {
        "checkoutSessionId": "checkout1-2345-6789-0abc-def123456789",
        "quantity": 2,
        "amountPaid": 160000.00,
        "purchasedAt": "2025-10-01T20:30:45",
        "transactionId": "txn_1234567890abcdef"
      }
    ],
    "transferHistory": []
  },
  {
    "participantId": "part2345-6789-01bc-def1-234567890123",
    "userId": "user2345-6789-01bc-def1-234567890123",
```

```
        "userName": "jane_smith",
        "userProfileMedia": {
          "fileId": "4a5b6c7d-8e9f-0123-bcde-f34567890123",
          "status": "READY",
          "thumbUrl": "https://cdn.nextgate.com/profiles/jane/thumb.webp",
          "mimeType": "image/jpeg"
        },
        "quantity": 1,
        "totalPaid": 80000.00,
        "status": "ACTIVE",
        "joinedAt": "2025-10-01T21:15:20",
        "contributionPercentage": 25.00,
        "purchaseCount": 1,
        "hasTransferred": false
      },
      {
        "participantId": "part3456-7890-12cd-ef12-345678901234",
        "userId": "user3456-7890-12cd-ef12-345678901234",
        "userName": "bob_wilson",
        "userProfileMedia": null,
        "quantity": 1,
        "totalPaid": 80000.00,
        "status": "ACTIVE",
        "joinedAt": "2025-10-02T08:45:10",
        "contributionPercentage": 25.00,
        "purchaseCount": 1,
        "hasTransferred": false
      }
    ]
  }
}
```


Success Response Fields:

Field	Description
groupInstanceId	Unique identifier for the group
groupCode	Human-readable group code
productId	Product unique identifier
productName	Product name

Field	Description
productPrimaryMedia	Resolved primary media of the product — `{fileId, mediaType, status, thumbUrl, mimeType}`. `null` if the product has no media yet
shopId	Shop identifier
shopName	Shop name
shopLogoMedia	Resolved shop logo — `{fileId, status, thumbUrl, mimeType}`. `null` if the shop has no logo
regularPrice	Original product price in TZS
groupPrice	Discounted group price in TZS
savingsAmount	Amount saved per unit (regularPrice - groupPrice)
savingsPercentage	Percentage saved
currency	Currency code (TZS)
totalSeats	Maximum seats in group
seatsOccupied	Filled seats
seatsRemaining	Available seats
totalParticipants	Unique participants count
progressPercentage	Completion percentage
status	OPEN, COMPLETED, FAILED, or DELETED
isExpired	Whether group expired
isFull	Whether all seats filled
initiatorId	User who created the group
initiatorName	Initiator's username
durationHours	Group duration in hours
createdAt	Group creation timestamp
expiresAt	Expiration timestamp
completedAt	Completion timestamp (null if not completed)
maxPerCustomer	Maximum seats per customer
isUserMember	Whether authenticated user is member
myParticipantId	User's participant ID (if member)
myQuantity	User's seat quantity (if member)
participants	Array of detailed participant information
participants[].participantId	Participant unique identifier
participants[].userId	User ID

Field	Description
participants[].userName	Username
participants[].userProfileMedia	Resolved profile media — `{fileId, status, thumbUrl, mimeType}`. `null` if no profile picture set
participants[].quantity	Number of seats held
participants[].totalPaid	Total amount paid in TZS
participants[].status	ACTIVE, TRANSFERRED_OUT, or REFUNDED
participants[].joinedAt	Join timestamp
participants[].contributionPercentage	Percentage of total seats
participants[].purchaseCount	Number of purchases made
participants[].hasTransferred	Whether participated in transfers
participants[].purchaseHistory	Purchase records (only shown to participant owner)
participants[].transferHistory	Transfer records (only shown to participant owner)

Error Response Examples:

Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Group not found with ID: gp123456-7890-abcd-ef12-345678901234",
  "action_time": "2025-10-02T14:35:45",
  "data": "Group not found with ID: gp123456-7890-abcd-ef12-345678901234"
}
```

---

3. Get Group by Code

Purpose: Retrieves group information using the human-readable group code instead of UUID.

Endpoint: GET `{base_url}/group-purchases/code/{groupCode}`

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

Path Parameters:

Parameter	Type	Required	Description	Validation
groupCode	string	Yes	Group code (e.g., GP-A3X7K9)	Format: GP-XXXXXX (6 alphanumeric characters)

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Group retrieved successfully",
  "action_time": "2025-10-02T14:40:45",
  "data": {
    "groupInstanceId": "gp123456-7890-abcd-ef12-345678901234",
    "groupCode": "GP-A3X7K9",
    "productId": "prod1234-5678-90ab-cdef-123456789012",
    "productName": "Premium Wireless Headphones",
    "productPrimaryMedia": {
      "fileId": "9c1a2b3d-4e5f-6789-abcd-ef0123456789",
      "mediaType": "IMAGE",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/products/headphones-001/thumb.webp",
      "mimeType": "image/webp"
    },
    "shopId": "shop1234-5678-90ab-cdef-123456789012",
    "shopName": "TechWorld Electronics",
    "shopLogoMedia": {
      "fileId": "1d2e3f4a-5b6c-7890-defa-bc1234567890",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/shops/techworld/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "regularPrice": 150000.00,
    "groupPrice": 80000.00,
    "savingsAmount": 70000.00,
```

```
    "savingsPercentage": 46.67,
    "currency": "TZS",
    "totalSeats": 10,
    "seatsOccupied": 4,
    "seatsRemaining": 6,
    "totalParticipants": 3,
    "progressPercentage": 40.00,
    "status": "OPEN",
    "isExpired": false,
    "isFull": false,
    "initiatorId": "user1234-5678-90ab-cdef-123456789012",
    "initiatorName": "john_doe",
    "durationHours": 24,
    "createdAt": "2025-10-01T20:30:45",
    "expiresAt": "2025-10-02T20:30:45",
    "completedAt": null,
    "maxPerCustomer": 5,
    "isUserMember": false,
    "myParticipantId": null,
    "myQuantity": null,
    "participants": []
  }
}
```


Success Response Fields:

Field	Description
All fields	Same as "Get Group by ID" response

Error Response Examples:

Not Found (404):


```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Group not found with code: GP-INVALID",
  "action_time": "2025-10-02T14:40:45",
  "data": "Group not found with code: GP-INVALID"
}
```

4. Get My Groups

Purpose: Retrieves all groups that the authenticated user is a member of, optionally filtered by status.

Endpoint: `GET` `{base_url}/group-purchases/my-groups`

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

Query Parameters:

Parameter	Type	Required	Description	Validation	Default
status	string	No	Filter by group status	enum: OPEN, COMPLETED, FAILED, DELETED	null (all statuses)

Success Response JSON Sample:


```
{
  "success": true,
  "httpStatus": "OK",
  "message": "My groups retrieved successfully",
  "action_time": "2025-10-02T14:45:45",
  "data": [
    {
      "groupInstanceId": "gp123456-7890-abcd-ef12-345678901234",
      "groupCode": "GP-A3X7K9",
      "productName": "Premium Wireless Headphones",
      "productPrimaryMedia": {
        "fileId": "9c1a2b3d-4e5f-6789-abcd-ef0123456789",
        "mediaType": "IMAGE",
        "status": "READY",
        "thumbUrl": "https://cdn.nextgate.com/products/headphones-001/thumb.webp",
```

```
      "mimeType": "image/webp"
    },
    "shopName": "TechWorld Electronics",
    "groupPrice": 80000.00,
    "savingsPercentage": 46.67,
    "currency": "TZS",
    "totalSeats": 10,
    "seatsOccupied": 8,
    "seatsRemaining": 2,
    "totalParticipants": 5,
    "progressPercentage": 80.00,
    "status": "OPEN",
    "expiresAt": "2025-10-02T20:30:45",
    "isExpired": false,
    "isUserMember": true,
    "participants": [
      {
        "userId": "user1234-5678-90ab-cdef-123456789012",
        "userName": "john_doe",
        "userProfileMedia": {
          "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
          "status": "READY",
          "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
          "mimeType": "image/jpeg"
        },
        "quantity": 2,
        "contributionPercentage": 25.00
      }
    ]
  },
  {
    "groupInstanceId": "gp234567-8901-bcde-f123-456789012345",
    "groupCode": "GP-B2Y8M5",
    "productName": "Smart Watch Series 5",
    "productPrimaryMedia": {
      "fileId": "2a3b4c5d-6e7f-8901-bcde-f01234567890",
      "mediaType": "IMAGE",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/products/watch-005/thumb.webp",
      "mimeType": "image/webp"
```

```
      },
      "shopName": "Gadget Hub",
      "groupPrice": 250000.00,
      "savingsPercentage": 28.57,
      "currency": "TZS",
      "totalSeats": 15,
      "seatsOccupied": 15,
      "seatsRemaining": 0,
      "totalParticipants": 8,
      "progressPercentage": 100.00,
      "status": "COMPLETED",
      "expiresAt": "2025-10-01T18:20:30",
      "isExpired": false,
      "isUserMember": true,
      "participants": []
    }
  ]
}
```


Success Response Fields:

Field	Description
All fields	Same as "Get Available Groups for Product" response

---

5. Get My Participations

Purpose: Retrieves all active participations of the authenticated user across all groups with detailed participant information.

Endpoint: `GET` `{base_url}/group-purchases/my-participations`

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": "My participations retrieved successfully",
  "action_time": "2025-10-02T14:50:45",
  "data": [
    {
      "participantId": "part1234-5678-90ab-cdef-123456789012",
      "userId": "user1234-5678-90ab-cdef-123456789012",
      "userName": "john_doe",
      "userProfileMedia": {
        "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
        "status": "READY",
        "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
        "mimeType": "image/jpeg"
      },
      "quantity": 2,
      "totalPaid": 160000.00,
      "status": "ACTIVE",
      "joinedAt": "2025-10-01T20:30:45",
      "purchaseCount": 1,
      "hasTransferred": false,
      "checkoutSessionId": "checkout1-2345-6789-0abc-def123456789",
      "purchaseHistory": [
        {
          "checkoutSessionId": "checkout1-2345-6789-0abc-def123456789",
          "quantity": 2,
          "amountPaid": 160000.00,
          "purchasedAt": "2025-10-01T20:30:45",
          "transactionId": "txn_1234567890abcdef"
        }
      ],
      "transferHistory": []
    },
    {
      "participantId": "part2345-6789-01bc-def1-234567890123",
      "userId": "user1234-5678-90ab-cdef-123456789012",
      "userName": "john_doe",
```

```
    "userProfileMedia": {
      "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "quantity": 3,
    "totalPaid": 750000.00,
    "status": "ACTIVE",
    "joinedAt": "2025-10-02T10:15:20",
    "purchaseCount": 2,
    "hasTransferred": true,
    "checkoutSessionId": "checkout2-3456-7890-1bcd-ef2345678901",
    "purchaseHistory": [
      {
        "checkoutSessionId": "checkout2-3456-7890-1bcd-ef2345678901",
        "quantity": 1,
        "amountPaid": 250000.00,
        "purchasedAt": "2025-10-02T10:15:20",
        "transactionId": "txn_2345678901bcdef0"
      },
      {
        "checkoutSessionId": "checkout3-4567-8901-2cde-f34567890123",
        "quantity": 2,
        "amountPaid": 500000.00,
        "purchasedAt": "2025-10-02T12:30:15",
        "transactionId": "txn_3456789012cdef01"
      }
    ],
    "transferHistory": [
      {
        "fromGroupId": "gp345678-9012-cdef-1234-567890123456",
        "fromGroupCode": null,
        "toGroupId": "gp234567-8901-bcde-f123-456789012345",
        "toGroupCode": null,
        "transferredAt": "2025-10-02T11:45:30",
        "reason": "Transferred 1 seats from group GP-C3Z9N7"
      }
    ]
  }
```

```
    ]
}
```

Success Response Fields:

Field	Description
participantId	Participant unique identifier
userId	User ID of the participant
userName	Username
userProfileMedia	Resolved profile media — `{fileId, status, thumbUrl, mimeType}`. `null` if no profile picture set
quantity	Total seats held
totalPaid	Total amount paid in TZS
status	ACTIVE, TRANSFERRED_OUT, or REFUNDED
joinedAt	When user joined this group
purchaseCount	Number of purchases made in this group
hasTransferred	Whether user has transfer history
checkoutSessionId	Original checkout session ID
purchaseHistory	Array of all purchases in this group
purchaseHistory[].checkoutSessionId	Checkout session for this purchase
purchaseHistory[].quantity	Seats purchased
purchaseHistory[].amountPaid	Amount paid for this purchase
purchaseHistory[].purchasedAt	Purchase timestamp
purchaseHistory[].transactionId	Payment transaction ID
transferHistory	Array of all transfers involving this participation
transferHistory[].fromGroupId	Source group ID
transferHistory[].toGroupId	Target group ID
transferHistory[].transferredAt	Transfer timestamp
transferHistory[].reason	Transfer reason/description

---

6. Transfer Seats Between Groups

Purpose: Transfers seats from one group to another. Allows users to move their purchases between compatible groups (same product, shop, and price).

Endpoint: `POST` `{base_url}/group-purchases/transfer`

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:

```
{
  "sourceGroupId": "gp123456-7890-abcd-ef12-345678901234",
  "targetGroupId": "gp234567-8901-bcde-f123-456789012345",
  "quantity": 2
}
```

Request Body Parameters:

Parameter	Type	Required	Description	Validation
sourceGroupId	string (UUID)	Yes	Group to transfer from	Valid UUID, user must be active member
targetGroupId	string (UUID)	Yes	Group to transfer to	Valid UUID, must be different from source
quantity	integer	Yes	Number of seats to transfer	Min: 1, cannot exceed user's quantity in source group

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Seats transferred successfully",
  "action_time": "2025-10-02T15:00:45",
  "data": {
    "participantId": "part2345-6789-01bc-def1-234567890123",
    "userId": "user1234-5678-90ab-cdef-123456789012",
```

```
    "userName": "john_doe",
    "userProfileMedia": {
      "fileId": "3f4a5b6c-7d8e-9f01-abcd-ef2345678901",
      "status": "READY",
      "thumbUrl": "https://cdn.nextgate.com/profiles/john/thumb.webp",
      "mimeType": "image/jpeg"
    },
    "quantity": 3,
    "totalPaid": 0.00,
    "status": "ACTIVE",
    "joinedAt": "2025-10-02T15:00:45",
    "purchaseCount": 0,
    "hasTransferred": true,
    "checkoutSessionId": "checkout1-2345-6789-0abc-def123456789",
    "purchaseHistory": [],
    "transferHistory": [
      {
        "fromGroupId": "gp123456-7890-abcd-ef12-345678901234",
        "fromGroupCode": null,
        "toGroupId": "gp234567-8901-bcde-f123-456789012345",
        "toGroupCode": null,
        "transferredAt": "2025-10-02T15:00:45",
        "reason": "Transferred 2 seats from group GP-A3X7K9"
      }
    ]
  }
}
```


Success Response Fields:

Field	Description
participantId	Updated participant ID in target group
userId	User ID
userName	Username
userProfileMedia	Resolved profile media — `{fileId, status, thumbUrl, mimeType}`. `null` if no profile picture set
quantity	New total quantity in target group
totalPaid	Total paid (0 for transfers)
status	Participant status (ACTIVE)

Field	Description
joinedAt	Join timestamp (current time if new to target)
purchaseCount	Purchase count (0 for pure transfers)
hasTransferred	Always true for transferred participants
checkoutSessionId	Original checkout session ID
purchaseHistory	Purchase history (shown only to owner)
transferHistory	Transfer history including this transfer

Error Response Examples:

Bad Request - Same Source and Target (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Source and target groups must be different",
  "action_time": "2025-10-02T15:00:45",
  "data": "Source and target groups must be different"
}
```

Bad Request - Insufficient Seats (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Not enough seats to transfer. You have: 1, requested: 2",
  "action_time": "2025-10-02T15:00:45",
  "data": "Not enough seats to transfer. You have: 1, requested: 2"
}
```

Bad Request - Target Group Full (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Not enough seats available. Requested: 2, Available: 1",
  "action_time": "2025-10-02T15:00:45",
  "data": "Not enough seats available. Requested: 2, Available: 1"
}
```

Bad Request - Product Mismatch (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot transfer between groups with different products",
  "action_time": "2025-10-02T15:00:45",
  "data": "Cannot transfer between groups with different products"
}
```

Bad Request - Price Mismatch (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot transfer. Price mismatch: 80000.00 vs 75000.00",
  "action_time": "2025-10-02T15:00:45",
  "data": "Cannot transfer. Price mismatch: 80000.00 vs 75000.00"
}
```

Not Found - Not a Participant (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "You are not a participant in the source group",
  "action_time": "2025-10-02T15:00:45",
  "data": "You are not a participant in the source group"
}
```

---

7. Update Group Name

Purpose: Allow the group initiator to update the group name. Group names must be unique among active (OPEN) groups.

Endpoint: **PATCH** `{base_url}/group-purchases/{groupId}/name`

Access Level: □□ Protected (Requires Authentication - Group Initiator Only)

Authentication: Bearer Token

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication
Content-Type	string	Yes	application/json

Path Parameters:

Parameter	Type	Required	Description	Validation
groupId	UUID	Yes	Unique identifier of the group	Valid UUID format

Request Body:

Field	Type	Required	Description	Validation
groupName	string	Yes	New name for the group	3-100 characters, unique among active groups

Request JSON Sample:

```
{
  "groupName": "iPhone 15 Pro - Dar es Salaam Deal"
}
```

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Group name updated successfully",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```

Success Response Fields:

Field	Description
`data`	Always `null` — the endpoint confirms the rename via `message` only. Fetch Get Group by ID to see the updated `groupName`

Error Response JSON Samples:

Group not found:

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Group not found",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```

Not the initiator:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Only the group initiator can change the group name",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```

Group not active:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot rename group with status: COMPLETED",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```

Group expired:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot rename expired group",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```

Invalid name length:


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Group name must be between 3 and 100 characters",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```


Name already taken:


```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Group name already taken: iPhone 15 Pro - Dar es Salaam Deal",
  "action_time": "2025-12-30T10:00:00",
  "data": null
}
```


Standard Error Types:

Status	Message	Cause
`400 BAD_REQUEST`	Only the group initiator can change the group name	User is not the group initiator
`400 BAD_REQUEST`	Cannot rename group with status: {status}	Group is not OPEN
`400 BAD_REQUEST`	Cannot rename deleted group	Group has been deleted
`400 BAD_REQUEST`	Cannot rename expired group	Group has expired
`400 BAD_REQUEST`	Group name must be between 3 and 100 characters	Invalid name length
`400 BAD_REQUEST`	Group name already taken: {name}	Name exists on another active group
`401 UNAUTHORIZED`	User not authenticated	Missing or invalid token
`404 NOT_FOUND`	Group not found	Invalid group ID

Business Rules:

Rule	Description
Initiator only	Only the user who created the group can rename it
Active groups only	Group must have status `OPEN`

Rule	Description
Not expired	Group must not be past its expiration time
Not deleted	Group must not be soft-deleted
Unique name	Name must be unique among all active (OPEN) groups
Name length	Must be between 3 and 100 characters
Trimmed	Leading/trailing whitespace is automatically removed

Default Group Name: When a group is created, the name is auto-generated as:

```
{groupCode}-{productName}
```

Example: `GP-ABC123-iPhone 15 Pro`

Usage Example:

```
curl -X PATCH \
  'https://api.nexgate.com/api/v1/groups/550e8400-e29b-41d4-a716-446655440000/name' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIs...' \
  -H 'Content-Type: application/json' \
  -d '{
    "groupName": "iPhone 15 Pro - Dar es Salaam Deal"
  }'
```

Group Purchase Workflow

1. Creating a New Group

When a user makes a GROUP_PURCHASE checkout and no `groupInstanceId` is provided in metadata:

Flow:

1. User creates checkout session with `sessionType: GROUP_PURCHASE`
2. User processes payment (WALLET only)
3. Payment completes successfully
4. System automatically creates new group instance
5. User becomes first participant (initiator)
6. Group gets unique code (e.g., GP-A3X7K9)
7. Group status set to OPEN

8. Expiration set based on product's `groupTimeLimitHours`

2. Joining an Existing Group

When a user makes a GROUP_PURCHASE checkout with `groupInstanceId` in metadata:

Flow:

1. User finds available group (via product page or group code)
2. User creates checkout session with `sessionType: GROUP_PURCHASE`
3. User includes `groupInstanceId` in checkout metadata
4. User processes payment (WALLET only)
5. Payment completes successfully
6. System adds user to existing group
7. Group's `seatsOccupied` increases
8. Group's `totalParticipants` increases (if new member)

Note: User can join same group multiple times to buy more seats (Hybrid Approach)

3. Group Completion

When a group fills all seats:

Automatic Actions:

1. Group status changes to COMPLETED
2. `completedAt` timestamp recorded
3. Orders created for all participants
4. Inventory permanently deducted
5. No new participants allowed

4. Group Expiration

When a group reaches expiration time without filling:

Automatic Actions:

1. Group status changes to FAILED
2. All participants refunded
3. Inventory holds released
4. Participant status changes to REFUNDED

5. Transferring Between Groups

Users can transfer seats between compatible groups:

Compatibility Requirements:

- Same product
- Same shop
- Same group price
- Target group must be OPEN
- Target group not expired
- Target group has available seats

Transfer Types:

Partial Transfer:

- Transfer some seats, keep some in source
- User remains ACTIVE in both groups
- Source group seats reduced
- Target group seats increased

Full Transfer:

- Transfer all seats from source
- Source participant status → TRANSFERRED_OUT
- Source group participants count decreased
- If source group becomes empty → soft deleted

Group Status Definitions

Status	Description	Can Join?	Can Transfer From?	Can Transfer To?
OPEN	Active, accepting participants	Yes	Yes	Yes
COMPLETED	All seats filled, orders created	No	No	No
FAILED	Expired without filling, refunds issued	No	No	No
DELETED	Soft deleted (empty or admin action)	No	No	No

Participant Status Definitions

Status	Description	Still in Group?	Can Buy More?	Can Transfer?
ACTIVE	Currently participating in group	Yes	Yes	Yes
TRANSFERRED_OUT	Left this group via transfer	No	No	No
REFUNDED	Group failed, money refunded	No	No	No

Purchase History Tracking

Each participant maintains detailed purchase history:

Tracked Information:

- Checkout session ID
- Quantity purchased
- Amount paid
- Purchase timestamp
- Transaction ID

Use Cases:

- User buys 2 seats initially
- User joins same group again, buys 3 more seats
- Purchase history shows 2 separate purchases
- Total quantity: 5 seats
- Total paid: sum of both purchases

Transfer History Tracking

Each transfer is recorded in participant history:

Tracked Information:

- Source group ID and code
- Target group ID and code
- Transfer timestamp

- Transfer reason/description

Transfer Scenarios:

Scenario 1: Partial Transfer

- User has 5 seats in Group A
- Transfers 2 seats to Group B
- Group A participant: 3 seats, ACTIVE status
- Group B participant: 2 seats, ACTIVE status, transfer history added

Scenario 2: Full Transfer

- User has 3 seats in Group A
- Transfers all 3 seats to Group B
- Group A participant: 0 seats, TRANSFERRED_OUT status
- Group B participant: 3 seats, ACTIVE status, transfer history added

Scenario 3: Multiple Transfers

- User transfers from Group A to Group B
- Later transfers from Group B to Group C
- Full transfer history maintained in each participation

Group Expiration and Cleanup

Automatic Expiration

Scheduled Job runs periodically to:

1. Find groups with status=OPEN and expiresAt < now
2. Change status to FAILED
3. Initiate refunds for all participants
4. Update participant status to REFUNDED
5. Release inventory holds

Soft Deletion

Groups are soft-deleted when:

1. All participants transfer out (empty group)
2. Admin manually deletes group

Soft Delete Actions:

- `isDeleted` set to true
- `status` changed to DELETED
- `deletedAt` timestamp recorded
- `deletedBy` user ID recorded
- `deleteReason` stored
- Group still queryable but excluded from active lists

Business Rules

Maximum Seats Per Customer

If product has `maxPerCustomer` limit:

- Single purchase cannot exceed limit
- Multiple purchases in same group respect limit
- Transfer to group validates combined quantity

Example:

- Product has maxPerCustomer = 5
- User has 3 seats in Group A
- Tries to transfer to Group B where they have 3 seats
- Transfer rejected (3 + 3 = 6 > 5)

Group Price Lock

Group price is snapshot at creation:

- Price stored in group instance
- Transfers validate price match
- Product price changes don't affect existing groups

Inventory Management

During Group Lifecycle:

- Seats held in inventory when purchased
- Holds maintained until group completes or fails
- Completed: inventory permanently deducted
- Failed: inventory holds released

Transfer Impact:

- No inventory change during transfer
- Total inventory hold remains same
- Just moves between groups

Integration with Checkout Sessions

Creating New Group

Checkout Session Requirements:

- `sessionType: GROUP_PURCHASE`
- Exactly 1 item
- WALLET payment only
- No `groupInstanceId` in metadata
- Status: PAYMENT_COMPLETED

After Payment Success:

```
GroupPurchaseInstanceEntity group = groupPurchaseService.createGroupInstance(checkoutSession);
```

Joining Existing Group

Checkout Session Requirements:

- `sessionType: GROUP_PURCHASE`
- Exactly 1 item
- WALLET payment only
- `groupInstanceId` in metadata
- Status: PAYMENT_COMPLETED

After Payment Success:

```
UUID groupId = (UUID) checkoutSession.getMetadata().get("groupInstanceId");
GroupPurchaseInstanceEntity group = groupPurchaseService.joinGroup(groupId, checkoutSession);
```

Error Handling Best Practices

Common Error Scenarios

Product Not Available for Group Buying:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Group buying is not enabled for this product"
}
```

Action: Check product has `groupBuyingEnabled: true`

Group Expired:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Group has expired at: 2025-10-02T20:30:45"
}
```

Action: Find another available group or create new group

Group Full:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Group is full. Seats occupied: 10/10"
}
```

Action: Find another available group or create new group

Quantity Exceeds Group Size:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Quantity (8) exceeds group max size (5)"
}
```

Action: Reduce quantity or create multiple purchases

Transfer Between Incompatible Groups:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot transfer between groups from different shops"
}
```

Action: Only transfer between compatible groups

Quick Reference Guide

Endpoint Summary

Endpoint	Method	Purpose
`/group-purchases/product/{productId}/available`	GET	Get available groups for product
`/group-purchases/{groupId}`	GET	Get group details by ID
`/group-purchases/code/{groupCode}`	GET	Get group details by code
`/group-purchases/my-groups`	GET	Get user's groups
`/group-purchases/my-participations`	GET	Get user's participations
`/group-purchases/transfer`	POST	Transfer seats between groups

Common HTTP Status Codes

- `200 OK`: Successful operation
- `400 Bad Request`: Validation error or business rule violation
- `401 Unauthorized`: Authentication required
- `404 Not Found`: Group, product, or participation not found

Group Code Format

- Pattern: `GP-XXXXXX`
- Length: 9 characters (GP- + 6 alphanumeric)
- Example: `GP-A3X7K9`, `GP-B2Y8M5`
- Auto-generated at group creation

Participant Contribution Calculation

```
contributionPercentage = (participantQuantity / totalSeatsOccupied) × 100
```

Progress Calculation

```
progressPercentage = (seatsOccupied / totalSeats) × 100
```

Savings Calculation

```
savingsAmount = regularPrice - groupPrice
savingsPercentage = (savingsAmount / regularPrice) × 100
```

Testing

Test Scenarios

Scenario 1: Create and Complete Group

1. User A creates group (2 seats)
2. User B joins group (3 seats)
3. User C joins group (5 seats)
4. Group auto-completes
5. Orders created for all participants

Scenario 2: Transfer Between Groups

1. User A in Group 1 (3 seats)
2. User A transfers 2 seats to Group 2
3. User A remains in Group 1 (1 seat)
4. User A now in Group 2 (2 seats)

Scenario 3: Group Expiration

1. Create group with 1-minute expiration
2. Wait for expiration
3. Scheduled job processes
4. Status → FAILED
5. Participants refunded

Scenario 4: Hybrid Approach - Multiple Purchases

1. User A creates group (2 seats)
2. User A joins same group again (3 seats)
3. User A total: 5 seats
4. Purchase history shows 2 records

Revision #5
Created 2 October 2025 06:19:52 by Admin Qbit
Updated 9 July 2026 10:08:14 by Admin Qbit