

# Wishlist Management

**Author:** Josh S. Sakweli, Backend Lead Team

**Last Updated:** 2026-06-04

**Version:** v1.0

**Base URL:** `{base_url}/api/v1/e-commerce/wishlist`

**Short Description:** The Wishlist API lets authenticated users save products for later, organize them into named groups, transfer items between groups, and move items directly to the cart. Every wishlist operation is strictly private — users can only read and modify their own wishlist.

## Hints:

- All endpoints require a valid JWT Bearer token — there are no public wishlist endpoints
- When adding a product, pass either `groupId` (existing group) or `groupName` (creates a new group) — passing both returns `400`
- Group names are unique per user — attempting to create a duplicate name returns `400`
- Deleting a group with `?deleteProducts=false` (default) moves all items in that group to Ungrouped; `?deleteProducts=true` permanently removes those items from the wishlist
- `PATCH /{itemId}/group` with `groupId: null` moves an item to Ungrouped without deleting it
- `isInWishlist`, `wishlistItemId`, `wishlistGroupId`, and `wishlistGroupName` are populated on the single product detail response (`GET /api/v1/e-commerce/products/{slug}`) for authenticated users

---

## Standard Response Format

### Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-09-23T10:30:45",
  "data": {}
}
```

# Error Response Structure

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2025-09-23T10:30:45",
  "data": "Error description"
}
```

## Standard Response Fields

Field	Type	Description
success	boolean	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 result
action_time	string	ISO 8601 timestamp of when the response was generated
data	object/string	Response payload on success, error details on failure

## Standard Error Types

- 400 BAD\_REQUEST: Business rule violation (duplicate product, duplicate group name, ambiguous group fields)
- 401 UNAUTHORIZED: Missing, expired, or invalid JWT token
- 404 NOT\_FOUND: Product, wishlist item, or group not found
- 422 UNPROCESSABLE\_ENTITY: Validation errors with field-level detail
- 500 INTERNAL\_SERVER\_ERROR: Unexpected server error

## Endpoints

### 1. Add Product to Wishlist

**Purpose:** Adds a product to the authenticated user's wishlist. Optionally places it in an existing group (via `groupId`) or creates a new group on the fly (via `groupName`). If neither is provided the item lands in Ungrouped.

**Endpoint:** `POST` `{base_url}/api/v1/e-commerce/wishlist/add`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	<code>Bearer &lt;token&gt;</code>

**Request JSON Sample — add to Ungrouped (no group):**

```
{
  "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
}
```

**Request JSON Sample — add to an existing group:**

```
{
  "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "groupId": "f9e8d7c6-b5a4-3210-fedc-ba9876543210"
}
```

**Request JSON Sample — add and create a new group:**

```
{
  "productId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "groupName": "Birthday Gifts"
}
```

**Request Body Parameters:**

Parameter	Type	Required	Description	Validation
<code>productId</code>	UUID	Yes	ID of the product to add	Must be a valid, non-deleted product
<code>groupId</code>	UUID	No	Add to an existing group	Must belong to the authenticated user. Cannot be combined with <code>groupName</code>

Parameter	Type	Required	Description	Validation
<code>groupName</code>	string	No	Create a new group with this name and add the product to it	Must not already exist for this user. Cannot be combined with <code>groupId</code>

### Success Response JSON Sample — added to Ungrouped:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product added to wishlist successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

### Success Response JSON Sample — added to existing group:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product added to wishlist in group 'Electronics'",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

### Success Response JSON Sample — new group created and product added:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product added to wishlist in group 'Birthday Gifts'",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

### Success Response Fields:

Field	Description
<code>message</code>	Confirms the product was added; includes group name when a group is involved

### Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "'Sony WH-1000XM5' is already in your wishlist",
  "action_time": "2026-06-04T14:22:10",
  "data": "'Sony WH-1000XM5' is already in your wishlist"
}
```

## 2. Get Wishlist (Flat)

**Purpose:** Returns the authenticated user's full wishlist as a flat list. Each item includes its group ID and group name (or `null` if Ungrouped). Useful for list views where grouping is handled client-side.

**Endpoint:** `GET` `{base_url}/api/v1/e-commerce/wishlist`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

### Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <token>

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Wishlist retrieved successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": {
    "user": {
      "userId": "uuid",
      "userName": "josh_dev",
      "name": "Josh Sakweli"
    },
    "wishlistSummary": {
```

```

    "totalItems": 3,
    "totalValue": 749.97,
    "inStockItems": 2,
    "outOfStockItems": 1
  },
  "wishlistItems": [
    {
      "wishlistId": "uuid",
      "productId": "uuid",
      "productName": "Sony WH-1000XM5",
      "productSlug": "sony-wh-1000xm5",
      "productImage": "https://cdn.example.com/img.jpg",
      "unitPrice": 349.99,
      "isOnSale": false,
      "shop": {
        "shopId": "uuid",
        "shopName": "Tech Haven",
        "shopSlug": "tech-haven",
        "logoUrl": "https://cdn.example.com/logo.jpg"
      },
      "availability": {
        "inStock": true,
        "stockQuantity": 12
      },
      "groupId": "uuid",
      "groupName": "Birthday Gifts",
      "addedAt": "2026-06-04T10:00:00"
    }
  ],
  "updatedAt": "2026-06-04T10:00:00"
}

```

### Success Response Fields:

Field	Description
<code>user.userId</code>	UUID of the authenticated user
<code>user.userName</code>	System username
<code>user.name</code>	Full name

Field	Description
wishlistSummary.totalItems	Total number of items in the wishlist
wishlistSummary.totalValue	Sum of unit prices of all items
wishlistSummary.inStockItems	Count of items currently in stock
wishlistSummary.outOfStockItems	Count of items out of stock
wishlistItems[].wishlistId	UUID of the wishlist entry (used for remove/transfer)
wishlistItems[].productId	UUID of the product
wishlistItems[].productName	Product display name
wishlistItems[].productSlug	URL-friendly product identifier
wishlistItems[].productImage	URL of the primary product image
wishlistItems[].unitPrice	Current price of the product
wishlistItems[].isOnSale	Whether the product is currently on sale
wishlistItems[].shop	Shop that sells the product (id, name, slug, logo)
wishlistItems[].availability.inStock	Whether the product is in stock
wishlistItems[].availability.stockQuantity	Current stock count
wishlistItems[].groupId	UUID of the group this item belongs to ( <code>null</code> if Ungrouped)
wishlistItems[].groupName	Name of the group ( <code>null</code> if Ungrouped)
wishlistItems[].addedAt	Timestamp when the product was added
updatedAt	Timestamp of the most recently added item

### 3. Get Wishlist (Grouped)

**Purpose:** Returns the wishlist organized into sections — one section per named group plus a separate Ungrouped section. Useful for grouped display views.

**Endpoint:** `GET` `{base_url}/api/v1/e-commerce/wishlist/grouped`

**Access Level:** `🔒` Protected (Requires valid JWT)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

## Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Grouped wishlist retrieved successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": {
    "user": {
      "userId": "uuid",
      "userName": "josh_dev",
      "name": "Josh Sakweli"
    },
    "wishlistSummary": {
      "totalItems": 3,
      "totalValue": 749.97,
      "inStockItems": 2,
      "outOfStockItems": 1
    },
    "groups": [
      {
        "groupId": "uuid",
        "groupName": "Birthday Gifts",
        "itemCount": 2,
        "items": [ ]
      }
    ],
    "ungrouped": {
      "groupId": null,
      "groupName": "Ungrouped",
      "itemCount": 1,
      "items": [ ]
    },
    "updatedAt": "2026-06-04T10:00:00"
  }
}
```

## Success Response Fields:

Field	Description
-------	-------------

<code>user</code>	Same user summary as flat response
<code>wishlistSummary</code>	Same summary totals across all items
<code>groups</code>	Array of named group sections, ordered by creation date (oldest first)
<code>groups[].groupId</code>	UUID of the group
<code>groups[].groupName</code>	Name of the group
<code>groups[].itemCount</code>	Number of items in this group
<code>groups[].items</code>	Array of wishlist item responses (same shape as flat list items)
<code>ungrouped</code>	Section for items with no group assigned
<code>ungrouped.groupId</code>	Always <code>null</code>
<code>ungrouped.groupName</code>	Always <code>"Ungrouped"</code>
<code>ungrouped.itemCount</code>	Count of items with no group
<code>ungrouped.items</code>	Array of ungrouped wishlist items
<code>updatedAt</code>	Timestamp of the most recently added item

## 4. Remove Item from Wishlist

**Purpose:** Permanently removes a specific item from the authenticated user's wishlist.

**Endpoint:** **DELETE** `{base_url}/api/v1/e-commerce/wishlist/{itemId}`

**Access Level:**  Protected (Requires valid JWT — owns the item)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <token>

**Path Parameters:**

Parameter	Type	Required	Description	Validation
<code>itemId</code>	UUID	Yes	The <code>wishlistId</code> of the item to remove	Must belong to the authenticated user

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product removed from wishlist successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

## 5. Clear Wishlist

**Purpose:** Permanently removes all items from the authenticated user's wishlist. Groups are not deleted — only the items inside them.

**Endpoint:** **DELETE** `{base_url}/api/v1/e-commerce/wishlist/clear`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <code>&lt;token&gt;</code>

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Wishlist cleared successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

## 6. Move Item to Cart

**Purpose:** Adds a wishlist item to the user's cart at the specified quantity. The item remains in the wishlist — it is not automatically removed.

**Endpoint:** **POST** `{base_url}/api/v1/e-commerce/wishlist/move-to-cart/{itemId}`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

**Path Parameters:**

Parameter	Type	Required	Description	Validation
itemId	UUID	Yes	The wishlistId of the item to move	Must belong to the authenticated user

**Query Parameters:**

Parameter	Type	Required	Description	Validation	Default
quantity	integer	No	Quantity to add to cart	Min: 1	1

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Product moved to cart successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

## 7. Transfer Item to Group

**Purpose:** Moves a wishlist item to a different group, or removes it from its current group by setting `groupId` to `null` (moves to Ungrouped). The item stays in the wishlist — only its group assignment changes.

**Endpoint:** **PATCH** `{base_url}/api/v1/e-commerce/wishlist/{itemId}/group`

**Access Level:**  Protected (Requires valid JWT — owns the item)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

**Path Parameters:**

Parameter	Type	Required	Description	Validation
itemId	UUID	Yes	The wishlistId of the item to transfer	Must belong to the authenticated user

**Request JSON Sample — move to a group:**

```
{
  "groupId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
}
```

**Request JSON Sample — remove from group (move to Ungrouped):**

```
{
  "groupId": null
}
```

**Request Body Parameters:**

Parameter	Type	Required	Description	Validation
groupId	UUID or null	Yes	Target group UUID, or null to move to Ungrouped	When a UUID, must be a group that belongs to the authenticated user

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Item moved to group 'Electronics'",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

# 8. Create Group

**Purpose:** Creates a new named wishlist group for the authenticated user. Group names must be unique per user.

**Endpoint:** **POST** `{base_url}/api/v1/e-commerce/wishlist/groups`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

**Request JSON Sample:**

```
{
  "name": "Electronics"
}
```

**Request Body Parameters:**

Parameter	Type	Required	Description	Validation
name	string	Yes	Display name for the group	Must not be blank. Must be unique for the authenticated user

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Wishlist group created successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": {
    "groupId": "uuid",
    "name": "Electronics",
    "itemCount": 0,
  }
}
```

```
"createdAt": "2026-06-04T14:22:10"
}
}
```

### Success Response Fields:

Field	Description
<code>groupId</code>	UUID of the newly created group
<code>name</code>	Name of the group as saved
<code>itemCount</code>	Always <code>0</code> on creation
<code>createdAt</code>	Timestamp of group creation

### Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "A group named 'Electronics' already exists",
  "action_time": "2026-06-04T14:22:10",
  "data": "A group named 'Electronics' already exists"
}
```

## 9. Get Groups

**Purpose:** Returns all wishlist groups created by the authenticated user, ordered by creation date (oldest first). Each group includes a live item count.

**Endpoint:** **GET** `{base_url}/api/v1/e-commerce/wishlist/groups`

**Access Level:**  Protected (Requires valid JWT)

**Authentication:** Bearer Token

### Request Headers:

Header	Type	Required	Description
<code>Authorization</code>	string	Yes	Bearer <token>

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Wishlist groups retrieved successfully",
  "action_time": "2026-06-04T14:22:10",
  "data": [
    {
      "groupId": "uuid",
      "name": "Birthday Gifts",
      "itemCount": 3,
      "createdAt": "2026-06-01T09:00:00"
    },
    {
      "groupId": "uuid",
      "name": "Electronics",
      "itemCount": 1,
      "createdAt": "2026-06-03T11:30:00"
    }
  ]
}
```

### Success Response Fields:

Field	Description
<code>[].groupId</code>	UUID of the group
<code>[].name</code>	Display name of the group
<code>[].itemCount</code>	Current number of wishlist items in this group
<code>[].createdAt</code>	Timestamp of group creation

## 10. Delete Group

**Purpose:** Deletes a wishlist group. Controls what happens to the items inside via the `deleteProducts` query parameter.

**Endpoint:** **DELETE** `{base_url}/api/v1/e-commerce/wishlist/groups/{groupId}`

**Access Level:**  Protected (Requires valid JWT — owns the group)

**Authentication:** Bearer Token

## Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer <token>

## Path Parameters:

Parameter	Type	Required	Description	Validation
groupId	UUID	Yes	UUID of the group to delete	Must belong to the authenticated user

## Query Parameters:

Parameter	Type	Required	Description	Validation	Default
deleteProducts	boolean	No	true → permanently delete all items in the group from wishlist. false → move items to Ungrouped, then delete group	true or false	false

## Success Response JSON Sample — items moved to Ungrouped:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Group deleted, products moved to Ungrouped",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

## Success Response JSON Sample — items permanently deleted:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Group and all its products deleted from wishlist",
  "action_time": "2026-06-04T14:22:10",
  "data": null
}
```

# Quick Reference

## All Endpoints Summary

#	Method	Path	Description
1	POST	<code>/wishlist/add</code>	Add product to wishlist
2	GET	<code>/wishlist</code>	Get flat wishlist
3	GET	<code>/wishlist/grouped</code>	Get grouped wishlist
4	DELETE	<code>/wishlist/{itemId}</code>	Remove item from wishlist
5	DELETE	<code>/wishlist/clear</code>	Clear entire wishlist
6	POST	<code>/wishlist/move-to-cart/{itemId}</code>	Move item to cart
7	PATCH	<code>/wishlist/{itemId}/group</code>	Transfer item to group
8	POST	<code>/wishlist/groups</code>	Create group
9	GET	<code>/wishlist/groups</code>	Get all groups
10	DELETE	<code>/wishlist/groups/{groupId}</code>	Delete group

## Group Behavior Rules

Scenario	Behavior
Add with no group	Item lands in Ungrouped
Add with <code>groupId</code>	Item added to that existing group
Add with <code>groupName</code> (new)	New group created, item added to it
Add with <code>groupName</code> (exists)	400 — duplicate group name
Add with both <code>groupId</code> and <code>groupName</code>	400 — ambiguous request
Delete group <code>?deleteProducts=false</code>	Items move to Ungrouped, group deleted
Delete group <code>?deleteProducts=true</code>	Items permanently removed, group deleted
Transfer item with <code>groupId: null</code>	Item moved to Ungrouped

Revision #5

Created 23 September 2025 13:32:54 by Admin Qbit

Updated 4 June 2026 10:58:16 by Admin Qbit