



**SOFTWARESILOS**

# Customer ID Extension

PDF Manual

MODULE

customer-id

LAST UPDATED

2026-03-12

# Table of Contents

*Customer ID Extension · customer-id*

- [Overview](#)
- [Customer ID Extension](#)
- [Getting Started](#)
  - [Installation](#)
  - [Configuration](#)
- [Features](#)
  - [API Integration](#)
  - [Login Integration](#)
  - [Range Management](#)
  - [Template Formatting](#)

# Customer ID Extension

Customer ID Extension · /customer-id

---

## Overview

The **Customer ID Extension** (Kundennummer) allows you to assign unique, customizable customer numbers to your Magento 2 customers. Perfect for B2B stores requiring traditional customer numbering systems with auto-increment, custom templates, and login integration.

## Key Features

### Auto-Increment System

Automatically assign sequential customer IDs with configurable starting points and ranges. Support for multiple websites with separate ID ranges.

### Template Formatting

Create custom ID formats using sprintf templates (e.g., "CUST-%05d", "B2B-%06d"). Flexible formatting for different business requirements.

### Login Integration

Allow customers to log in using their customer ID instead of email address. Seamless integration with Magento's authentication system.

### Range Management

Define ID ranges per website or customer group. Prevent overlaps and maintain organized numbering across multi-store setups.

### Duplicate Validation

Automatic duplicate detection prevents ID conflicts. Validates uniqueness before saving customer records.

### API Integration

Customer IDs are exposed via REST/SOAP API through extension attributes. Perfect for ERP and CRM integrations.

## What You Can Do

- Assign unique customer numbers automatically
- Customize ID format with templates
- Enable login with customer ID
- Define ID ranges per website

- Import existing customer numbers
- Multi-website support with separate ranges
- Access customer IDs via API

## Use Cases

### B2B Customer Management

Traditional customer numbering for B2B businesses requiring structured customer identification.

### Legacy System Migration

Import existing customer numbers from legacy systems while maintaining continuity.

### ERP Integration

Synchronize customer IDs between Magento and ERP systems (SAP, Dynamics, etc.) via API.

### Multi-Website Ranges

Separate customer ID ranges for different websites or brands (e.g., 1000-2000 for Website A, 3000-4000 for Website B).

### Compliance & Audit

GoBD/SOX compliance with structured customer identification and audit trails.

### Custom ID Formats

Flexible templates for different customer types (e.g., "B2B-00001" for business, "RET-00001" for retail).

## System Requirements

- Magento 2.4+
- PHP 8.0+

## Quick Start

1. **Installation Guide** - Install the extension
2. **Configuration** - Configure ID templates and ranges
3. **Login Integration** - Enable login with customer ID

**Ready to get started?** Follow our [Installation Guide](#) to begin using Customer ID Extension.

# Installation

Customer ID Extension · </customer-id/getting-started/installation>

Install the Customer ID extension for Magento 2.

## Prerequisites

- Magento 2.4.0 or higher
- PHP 8.0+
- Composer installed
- SSH access to your server
- Backup of your Magento installation

## Installation Steps

### 1. Add Repository Access

Contact your account manager to receive Composer repository credentials.

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
```

### 2. Require the Package

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/kundennummer:*
```

### 3. Enable the Module

```
php bin/magento module:enable MageB2B_Kundennummer
```

### 4. Run Setup Upgrade

```
php bin/magento setup:upgrade
```

### 5. Compile Code

```
php bin/magento setup:di:compile
```

### 6. Deploy Static Content

```
php bin/magento setup:static-content:deploy
```

## 7. Clear Cache

```
php bin/magento cache:flush
```

## Verify Installation

Check module status:

```
php bin/magento module:status MageB2B_Kundennummer
```

## Database Changes

The installation creates a new customer attribute:

- **customer\_id\_number** - Stores the customer ID

## Access Configuration

Navigate to **Stores > Configuration > MageB2B > Customer ID** to configure the extension.

## Troubleshooting

### Module Not Found

- Verify repository credentials
- Run `composer clear-cache`
- Check `composer.json` for correct repository URL

### Permission Errors

```
chmod -R 755 var/ pub/static/ generated/
```

### Attribute Not Created

If the `customer_id_number` attribute is missing:

```
php bin/magento setup:upgrade --keep-generated
```

## Next Steps

Proceed to [Configuration](#) to set up customer ID templates and ranges.

# Configuration

*Customer ID Extension · /customer-id/getting-started/configuration*

Configure the Customer ID extension to match your business requirements.

## Access Configuration

Navigate to **Stores > Configuration > MageB2B > Customer ID**

## General Settings

### Enable Customer ID

Enable or disable the customer ID functionality.

#### Options:

- Yes - Enable customer ID assignment
- No - Disable (existing IDs remain)

### ID Template

Define the format for customer IDs using sprintf syntax.

#### Examples:

- %d - Simple number: 1, 2, 3
- %05d - Zero-padded 5 digits: 00001, 00002
- CUST-%05d - With prefix: CUST-00001, CUST-00002
- B2B-%06d - Business format: B2B-000001
- %d-%s - With suffix: 1-A, 2-B

**Default:** %d

### Starting Number

The first customer ID to assign.

**Example:** 1000 (IDs will start from 1000)

**Note:** Only applies to new customers. Existing customers keep their IDs.

### ID Range (Optional)

#### Minimum ID

Minimum allowed customer ID number.

**Example:** 1000

#### Maximum ID

Maximum allowed customer ID number.

**Example:** 9999

**Use Case:** Separate ranges for different websites:

- Website A: 1000-4999
- Website B: 5000-9999

## Login Settings

### Enable Login with Customer ID

Allow customers to log in using their customer ID instead of email.

**Options:**

- Yes - Enable login with customer ID
- No - Email login only

**How it works:**

1. Customer enters customer ID in email field
2. System detects numeric format
3. Looks up customer by ID
4. Authenticates with password

### Show Customer ID in Account

Display customer ID in customer account dashboard.

**Options:**

- Yes - Show in "My Account"
- No - Hide from frontend

## Advanced Settings

### Auto-Assign on Registration

Automatically assign customer ID when customer registers.

**Options:**

- Yes - Assign immediately (recommended)
- No - Manual assignment only

### Allow Manual Override

Allow admin to manually set customer ID.

**Options:**

- Yes - Admin can override auto-generated ID
- No - Always use auto-increment

**Warning:** Manual override can cause duplicates if not careful.

## Validate Uniqueness

Check for duplicate customer IDs before saving.

### Options:

- Yes - Prevent duplicates (recommended)
- No - Allow duplicates (not recommended)

## Multi-Website Configuration

### Per-Website Ranges

To configure different ID ranges per website:

1. Change **Configuration Scope** to specific website
2. Set different **Starting Number** and **ID Range**
3. Save configuration

### Example Setup:

#### Website A (B2B):

- Starting Number: 1000
- Minimum ID: 1000
- Maximum ID: 4999
- Template: B2B-%04d

#### Website B (Retail):

- Starting Number: 5000
- Minimum ID: 5000
- Maximum ID: 9999
- Template: RET-%04d

## API Configuration

### Expose in API

Include customer ID in REST/SOAP API responses.

### Options:

- Yes - Available via extension attributes
- No - Not exposed in API

**API Field:** `customer_id_number` in extension attributes

## Save Configuration

After configuring, click **Save Config** and clear cache:

```
php bin/magento cache:flush
```

---

## Configuration Examples

### Simple Sequential IDs

```
Enable: Yes  
Template: %d  
Starting Number: 1
```

Result: 1, 2, 3, 4...

### Zero-Padded IDs

```
Enable: Yes  
Template: %05d  
Starting Number: 1
```

Result: 00001, 00002, 00003...

### B2B Customer IDs

```
Enable: Yes  
Template: B2B-%06d  
Starting Number: 100000
```

Result: B2B-100000, B2B-100001...

### Range-Based IDs

```
Enable: Yes  
Template: %d  
Starting Number: 1000  
Minimum ID: 1000  
Maximum ID: 2500
```

Result: 1000-2500 (then stops)

## Troubleshooting

### IDs Not Assigned

- Check "Enable Customer ID" is set to Yes
- Verify "Auto-Assign on Registration" is enabled
- Clear cache after configuration changes

### Duplicate IDs

- Enable "Validate Uniqueness"
- Check for manual overrides

- Verify range settings don't overlap between websites

## **Login Not Working**

- Ensure "Enable Login with Customer ID" is Yes
- Customer ID must be numeric (template affects this)
- Clear cache and test

## **Range Exceeded**

- Increase Maximum ID
- Or start new range for new website

## **Next Steps**

- [Login Integration](#) - Configure login with customer ID
- [Range Management](#) - Advanced range configuration
- [API Integration](#) - Use customer IDs in API

# API Integration

Customer ID Extension · /customer-id/features/api-integration

Access customer IDs via REST and GraphQL APIs.

## REST API

### Get Customer ID

```
GET /rest/V1/customers/{customerId}
```

Response includes:

```
{
  "id": 123,
  "email": "customer@example.com",
  "custom_attributes": [
    {
      "attribute_code": "customer_number",
      "value": "CUST-12345"
    }
  ]
}
```

### Search by Customer ID

```
GET
/rest/V1/customers/search?searchCriteria[filter_groups][0][filters][0][field]=customer_number&searchCriteria[filter_groups][0][filters][0][value]=CUST-12345
```

### Update Customer ID

```
PUT /rest/V1/customers/{customerId}
```

```
{
  "customer": {
    "custom_attributes": [
      {
        "attribute_code": "customer_number",
        "value": "CUST-99999"
      }
    ]
  }
}
```

# GraphQL

## Query Customer

```
{
  customer {
    customer_number
    email
    firstname
    lastname
  }
}
```

## ERP Integration

Common integration patterns:

- Sync customer IDs to ERP
- Import IDs from legacy system
- Real-time ID lookup

## Related

- [Login Integration](#)
- [Configuration](#)

# Login Integration

Customer ID Extension · /customer-id/features/login-integration

Allow customers to log in using their customer ID instead of email address.

## Overview

The login integration feature enables customers to use their customer ID as an alternative to email for authentication. This is particularly useful for B2B scenarios where customers prefer using their assigned customer numbers.

## How It Works

### Technical Implementation

The extension uses an Observer on the `controller_action_predispatch_customer_account_loginPost` event:

1. Customer enters customer ID in the email field
2. Observer (`PreControllerLogin.php`) intercepts the login request
3. System detects if input is a customer ID (numeric check)
4. Looks up customer by `customer_id_number` attribute
5. Replaces customer ID with email address
6. Magento's standard authentication proceeds

### Code Flow

```
// Observer/PreControllerLogin.php
if (is_numeric($login['username'])) {
    $customer = $customerRepository->getByCustomerId($login['username']);
    if ($customer) {
        $login['username'] = $customer->getEmail();
        $request->setPostValue('login', $login);
    }
}
```

## Configuration

### Enable Login with Customer ID

Navigate to **Stores > Configuration > MageB2B > Customer ID > Login Settings**

Set **Enable Login with Customer ID** to **Yes**

### Clear Cache

After enabling, clear cache:

## Customer Experience

### Login Form

Customers can now use either:

- **Email:** `customer@example.com`
- **Customer ID:** 12345 (or CUST-00123 depending on template)

### Login Process

1. Go to customer login page
2. Enter customer ID in "Email" field
3. Enter password
4. Click "Sign In"
5. System automatically recognizes customer ID and logs in

## Template Considerations

### Numeric IDs (Recommended for Login)

#### Templates that work well:

- `%d` - Pure numbers: 1, 2, 3
- `%05d` - Zero-padded: 00001, 00002

**Why:** Easy for customers to remember and type

### Alphanumeric IDs

#### Templates with prefixes:

- `CUST-%05d` - CUST-00001
- `B2B-%06d` - B2B-000001

**Note:** Customers must enter the full ID including prefix

### Non-Numeric Templates

#### Templates that DON'T work for login:

- `%d-%s` - Contains non-numeric suffix
- Custom formats with letters

**Why:** Login detection relies on numeric check

## Best Practices

**Use simple numeric templates** - Easier for customers to remember

**Communicate to customers** - Inform them they can use customer ID to login

**Display customer ID** - Show it in account dashboard and emails

**Test thoroughly** - Verify login works with your template format

**Keep email login** - Don't disable email login, offer both options

**Don't use complex templates** - Avoid formats that are hard to type

**Don't change templates** - Once set, changing affects existing customers

## Troubleshooting

### Login Not Working

**Check configuration:**

```
php bin/magento config:show mageb2b_kundenummer/login/enable_login
```

Should return `1` (enabled)

**Check customer has ID:**

```
SELECT entity_id, email, customer_id_number  
FROM customer_entity  
WHERE customer_id_number IS NOT NULL;
```

**Check template format:**

- Numeric templates work best
- Alphanumeric requires exact match including prefix

### Customer ID Not Recognized

**Possible causes:**

1. Customer ID not assigned yet
2. Template includes non-numeric characters
3. Customer entered ID incorrectly
4. Cache not cleared after configuration

**Solution:**

```
# Clear cache  
php bin/magento cache:flush  
  
# Verify customer ID exists  
php bin/magento customer:info customer@example.com
```

## Multiple Customers with Same ID

This should never happen if validation is enabled.

**Check for duplicates:**

```
SELECT customer_id_number, COUNT(*) as count
FROM customer_entity
WHERE customer_id_number IS NOT NULL
GROUP BY customer_id_number
HAVING count > 1;
```

**Fix duplicates:**

1. Enable "Validate Uniqueness" in configuration
2. Manually reassign duplicate IDs in admin
3. Run validation script

## Security Considerations

### Password Still Required

Customer ID login does NOT bypass password authentication:

- Customer ID only replaces email in lookup
- Password validation remains unchanged
- All security measures still apply

### Brute Force Protection

Magento's built-in brute force protection applies:

- Failed login attempts are tracked
- Account lockout after X failed attempts
- CAPTCHA can be enabled

### Customer ID Privacy

**Consider:**

- Customer IDs are less private than emails
- Sequential IDs are predictable
- Use ranges to obscure total customer count

**Recommendations:**

- Start numbering at higher values (e.g., 10000)
- Use non-sequential ranges
- Don't expose customer IDs publicly

## Frontend Display

### Show Customer ID in Account

Enable **Show Customer ID in Account** to display the ID in:

- Customer account dashboard
- Order confirmation emails
- Invoice PDFs
- Account information page

### Custom Template Display

Add customer ID to custom templates:

```
<?php
$customerId =
$block->getCustomer()->getCustomAttribute('customer_id_number');
if ($customerId) {
    echo 'Your Customer ID: ' . $customerId->getValue();
}
?>
```

## API Integration

Customer IDs are available via API for external systems:

```
{
  "customer": {
    "id": 123,
    "email": "customer@example.com",
    "extension_attributes": {
      "customer_id_number": "CUST-00123"
    }
  }
}
```

See [API Integration](#) for details.

## Related Features

- [Range Management](#) - Configure ID ranges
- [Template Formatting](#) - Customize ID formats
- [API Integration](#) - Access IDs via API

# Range Management

Customer ID Extension · /customer-id/features/range-management

Configure and manage customer ID number ranges.

## Overview

Define separate ID ranges for different customer types or stores.

## Configuration

**Stores > Configuration > Customers > Customer ID**

Setting	Description
<b>Start Number</b>	First ID in range
<b>End Number</b>	Last ID in range
<b>Current Number</b>	Next ID to assign
<b>Prefix</b>	Optional prefix
<b>Suffix</b>	Optional suffix

## Multiple Ranges

### Per Store View

Each store can have its own range:

- DE Store: 10000-19999
- US Store: 20000-29999
- UK Store: 30000-39999

### Per Customer Group

Different ranges by group:

- Retail: 100000-199999
- Wholesale: 200000-299999
- VIP: 300000-399999

## Range Exhaustion

When a range is exhausted:

1. Warning notification sent
2. Admin alert displayed
3. Configure new range or expand existing

## Best Practices

- Plan ranges with growth in mind
- Leave gaps between ranges
- Document range assignments
- Monitor usage regularly

## Related

- [Template Formatting](#)
- [Configuration](#)

# Template Formatting

Customer ID Extension · /customer-id/features/template-formatting

Customize the format of generated customer IDs.

## Template Variables

Variable	Description	Example
{id}	Sequential number	12345
{year}	Current year	2024
{month}	Current month	01
{store}	Store code	DE
{group}	Customer group code	WS

## Example Templates

### Simple Numeric

```
Template: {id}
Result: 12345
```

### With Prefix

```
Template: CUST-{id}
Result: CUST-12345
```

### Year-Based

```
Template: {year}-{id}
Result: 2024-12345
```

### Store-Specific

```
Template: {store}-{id}
Result: DE-12345
```

### Complex Format

```
Template: {store}{year}{id}
Result: DE202412345
```

# Padding

Configure number padding:

Setting	Value	Result
Pad Length	6	012345
Pad Character	0	012345

# Configuration

Stores > Config > Customers > Customer ID > Format

- **Template:** Format string
- **Pad Length:** Minimum digits
- **Pad Character:** Fill character

# Related

- [Range Management](#)
- [Login Integration](#)