



Protected Customer Attributes

General Knowledge Base Documentation

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-10-13

SOURCE
[Open this article online](#)

Protected Customer Attributes

SoftwareSilo modules can declare business-related customer and customer-address attributes as server-managed. The shared Extension Manager package (version 1.0.3 and later) then blocks customer- and guest-initiated API writes to those attributes, while admin and integration tokens keep Magento's normal behavior.

What is blocked

The protection hooks Magento's native API input handling and covers the standard customer paths:

API	Covered calls
REST	Account creation, customer repository saves, customer address repository saves
SOAP	The same account, customer and address service calls
GraphQL	Customer and address create/update mutations
Async and bulk REST	Customer- and guest-triggered queued updates

Setting, changing or clearing a protected attribute with a customer or guest token is rejected with Magento's input-validation error before any data is saved. Resubmitting an unchanged value during an update stays allowed, so read-then-write round trips keep working.

For asynchronous and bulk REST requests made with customer-level tokens the rules are stricter: protected attributes must be left out of the payload entirely, including nested customer-address attributes, or Magento rejects the request before it is queued.

Native storefront and admin forms are not intercepted — each attribute-owning module controls which fields its forms expose. Internal repository and account-management writes from module code also stay writable.

What this means for integrations

- Use an **admin or integration token** when a job legitimately writes a protected attribute; customer tokens and guest calls fail validation.
- In bulk or async customer-level payloads, omit protected attributes entirely.
- Custom API services or GraphQL mutations that bypass the native input paths are not covered — they must validate untrusted input themselves before mutating customer data.

The guard covers scalar attributes (text, numbers, selects, serialized settings). File and image attributes are outside its contract, and a declaration does not grant ACL

permissions by itself.

Protected attributes by module

Each module registers its own codes as arguments on

`MageB2B\ExtensionManager\Plugin\Customer\RestrictCustomAttributesPlugin` in its `etc/di.xml`. The current SoftwareSilo declarations are:

Package	Protected attributes
<code>mageb2b/customer-id</code>	<code>customer_id</code>
<code>mageb2b/customer-tfa</code>	<code>tfa_enabled</code>
<code>mageb2b/sublogin</code>	<code>can_create_sublogins</code> , <code>max_number_sublogins</code> , <code>sublogin_optional_email</code> , <code>sublogin_sales_report_config</code> ; address attribute <code>sublogin_id</code>
<code>mageb2b/sublogin-budget</code>	<code>budgets</code> , <code>can_manage_budgets</code>
<code>mageb2b/sublogin-role</code>	<code>can_manage_role</code>
<code>mageb2b/staff</code>	<code>primary_staff_id</code> , <code>inherit_restrict_sublogins</code>
<code>mageb2b/pricesystem-advancedconfig</code>	<code>priceselect</code> , <code>verified</code> , <code>price_formula</code> , <code>discountable</code>
<code>mageb2b/pricesystem-productcustomermatrix</code>	<code>procustmatrix_customer_group</code>

Only the declared attributes are guarded — installing a listed module does not protect unrelated customer attributes.