

amasty

For more details see how the [Order Attributes](#) extension works.

Guide for Order Attributes for Magento 2

Add an unlimited number of extra fields to your checkout page to easily collect and process additional order related data. Display all created order attributes on a handy backend grid to keep necessary information at hand.

- Create order attributes of different types
- Use flexible conditions for attribute fields display
- Show additional attributes info on the order grid
- Add custom attributes to emails and PDF documents
- Show extra attributes depending on a selected shipping method
- WCAG compliance (for the extension with Luma and Hyvä Theme)

Due to flexible settings and intuitive interface, the Order Attributes for Magento 2 extension will help you easily fine tune the checkout page according to your requirements. Create as many additional order attributes as you need. Add Magento custom field input types to orders and quickly process collected data on the backend.

General Settings

To configure the Magento 2 Order Attributes extension, please go to **Stores → Configuration → Amasty Extensions → Order Attributes**.

Checkout Settings

Checkout Settings

| | | |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|---|
| Display Attribute Values in the Order Summary <small>[store view]</small> | <input type="text" value="Yes"/> | ▼ |
| | Filled in order attributes will be displayed in the order summary section at the payment checkout step. | |
| Date Format <small>[store view]</small> | <input type="text" value="yyyy-mm-dd (2020-05-25)"/> | ▼ |
| | Format of date for outputting values | |
| Time Format <small>[store view]</small> | <input type="text" value="24-hour format"/> | ▼ |
| Include Order Attributes into Emails <small>[store view]</small> | <input type="text" value="Yes"/> | ▼ |
| Send Attributes on Shipping Step <small>[store view]</small> | <input type="text" value="Yes"/> | ▼ |
| | Set `No` to prevent validation on Shipping Step (One Step Checkout better compatibility) | |
| Newsletter Subscription <small>[global]</small> | <input type="text" value="subscribe_box"/> | |
| | Set an attribute code of order attribute (Yes/No or Checkbox) | |

Please, expand the Checkout Settings section:

Display Attribute Values in the Progress Sidebar — set the option to **Yes** if you want to display custom order attributes on the checkout progress bar;

Date Format — select the default date format for related attributes;

Time Format — choose the default time format for related attributes;

Include Order Attributes into Emails — enable this options to have the ability to include order attribute values into email.

Send Attributes on Shipping Step — set 'Yes' to display order attributes on the shopping step. Set 'No' to prevent validation on Shipping Step (One Step Checkout better compatibility).

Newsletter Subscription - enable the option to display Newsletter Subscription option on the checkout via an order attribute. Set an attribute code of order attribute (Yes/No or Checkbox).

PDF Documents

PDF Documents

| | |
|----------------------------------------------------------------------------------|-----|
| Include Order Attributes into Shipment (packing slip) <small>[global]</small> | Yes |
| Include Order Attributes into Invoice <small>[global]</small> | Yes |

Then, expand the **PDF Documents** section:

Include Order Attributes into Shipment (packing slip) — enable this option to include the completed attributes into the shipping documentation;

Include Order Attributes into Invoice — enable this option to include the completed attributes into the invoice documentation.

The Magento 2 Order Attributes extension is fully compatible with **Magento 2 PDF Customizer**. Easily create custom PDF invoices according to your business needs.

Invoices and Shipments

Order Attributes in the Invoices and Shipments

| | |
|------------------------------------------------------------------------------|-----|
| Display Order Attributes on the Invoice Grid <small>[global]</small> | Yes |
| Display Order Attributes when Creating an Invoice <small>[global]</small> | Yes |
| Display Order Attributes when Viewing an Invoice <small>[global]</small> | Yes |
| Display Order Attributes on the Shipments Grid <small>[global]</small> | Yes |
| Display Order Attributes when Creating a Shipment <small>[global]</small> | Yes |
| Display Order Attributes when Viewing a Shipment <small>[global]</small> | Yes |

Next, proceed to the Invoices and Shipments section:

Display Order Attributes on Invoices Grid — enable this option to display order attributes on the invoices grid;

Display Order Attributes on Invoices Create — enable this option to display order attributes when creating an invoice;

Display Order Attributes on Invoices View — enable this option to display order attributes when viewing an invoice;

Display Order Attributes on Shipments Grid — enable this option to display order attributes on the shipments grid;


Display Order Attributes on Shipments Create — enable this option to display order attributes when creating a new shipment;

Display Order Attributes on Shipments View — enable this option to display order attributes when viewing a shipment.

New Attribute Creation

To add custom Magento 2 attribute to order, please, go to **Sales → Manage Order Attributes**.

Manage Order Attributes

 demouser ▾

[Add New Order Attribute](#)

| Attribute Code | Attribute Label | Frontend Input Type | Visible | Checkout Step | Include Into PDFs | Show on Grid | Position | Action |
|---------------------|---------------------------------------|---------------------|---------|----------------------|-------------------|--------------|----------|----------------------|
| company_position | What is your position in the company? | Radio Buttons | Yes | Shipping Address | Yes | Yes | 1 | Edit |
| delivery_time | Delivery time | Date With Time | Yes | Shipping Methods | No | No | 0 | Edit |
| dpd | DPD | Html | Yes | Shipping Methods | Yes | Yes | 0 | Edit |
| favorite_products | Favorite Products | Text Field | Yes | Above Payment Method | Yes | Yes | 2 | Edit |
| join_us | Join us | Checkbox Group | Yes | Shipping Address | No | Yes | 3 | Edit |
| ups | UPS | Html | Yes | Shipping Methods | Yes | Yes | 0 | Edit |
| where_shop_is_found | Where did you find us | Radio Buttons | Yes | Order Summary | No | Yes | 4 | Edit |

All created attributes will be displayed on the special grid. Click the **Add New Order Attribute** button to create an attribute or select the existing one to edit it.

Add **One Step Checkout for Magento 2** and you will be able to sort order attributes relatively each other and relatively checkout fields on a 'Manage Checkout Fields' page.

Attribute Properties

General Configuration

Default Label *

Attribute Code *
This is used internally. Make sure you don't use spaces or more than 30 symbols.

Store View *

Main Website
Main Website Store
Default Store View

Customer Groups

NOT LOGGED IN
General
Wholesale
Retailer

Frontend Input Type

Values Required

Default Label — specify the attribute label that will be displayed on the frontend;

Attribute Code — specify the attribute code that will be used for internal needs;

Store View — select the store view for which the attribute will be available;

Customer Groups — select customer groups for which the attribute will be shown;

Frontend Input Type — choose the appropriate input type for an attribute. It's possible to choose among various input types depending on your specific needs:

- Text Field
- Text Area
- Date
- Date With Time
- Yes/No
- Dropdown

- Checkbox Group
- Radiobuttons
- HTML
- Single File Upload

For all input types except **Dropdown**, **Checkbox Group**, and **Radiobuttons** you can define the default value in the corresponding field.

For the **Dropdown**, **Checkbox Group**, and **Radiobuttons** input types you can add multiple options in the **Manage Label/Options** tab.

For the **Single File Upload** input type, you'll need to specify the list of allowed file types and max allowed size per file. A configuration example is shown on the screenshot below, with the frontend example on the right section of the image.

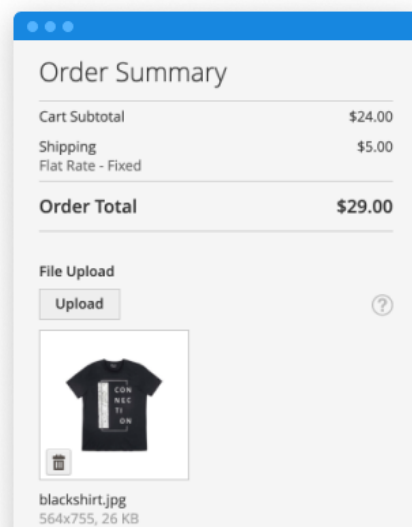
General Configuration

Frontend Input Type

Attribute Type Settings

Max File Size
In Mb

File Types
List comma-separated file types with no spaces, like: png,txt,jpg



NOTE: after the attribute code and the catalog input type are specified and saved they can't be changed.

Values Required — when the option is set to **Yes** customers won't be able to leave this attribute field blank while getting registered.

Attribute Type Settings

Attribute Type Settings

Visible on Front-end

Visible on Back-end

Visible on frontend — enable this option to display the attribute on the frontend;

Visible on backend — enable this option to display the attribute in the backend;

Default Value — specify the value which will be preselected by default if necessary;

Input Validation — enable this option to check the filled in information for being valid before submission;

Minimum Text Length — specify the minimum text length for text-related attributes;

Maximum Text Length — specify the maximum text length for text-related attributes;

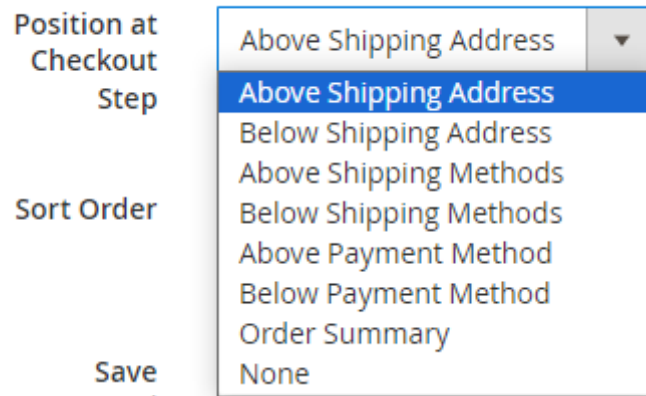
Input/Output Filter — this option is used for security reasons with the text and text-area input types. Choose the **Strip HTML Tags** option to exclude the code injections or the **Escape HTML Entities** option in order to screen the code injection. This option affects both frontend and backend sides.

Attribute Display Settings

Attribute Display Settings

| | |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Position at Checkout Step | <input type="text" value="Above Shipping Address"/> ▼ |
| Sort Order | <input type="text"/> Numeric, used in front-end to sort attributes |
| Save Entered Value For Future Checkout | <input type="text" value="Yes"/> ▼ If set to "Yes", previously entered value will be used during checkout. Works for registered customers only. |
| Show on Admin Grids | <input type="text" value="Yes"/> ▼ |
| Include Into HTML Print-out | <input type="text" value="Yes"/> ▼ Order attributes will be included in the document that a customer prints in his account |
| Include Into PDF Documents | <input type="text" value="Yes"/> ▼ Please, make sure that the 'attributes display in PDF files' option is enabled in the extension's general settings before modifying this option. |
| Include Into Transactional Emails | <input type="text" value="Yes"/> ▼ Please, make sure that the 'Include Order Attributes into Emails' option is enabled in the extension's general settings before modifying this option. |

Position at Checkout Step — specify at which place on checkout step the attribute should be displayed:



You can choose one of the following positions:

- Above Shipping Address
- Below Shipping Address
- Above Shipping Methods
- Below Shipping Methods
- Above Payment Method
- Below Payment Method
- Order Summary
- None

Display Sorting Order — set the sorting order for the attribute display on the frontend;

Save Entered Value For Future Checkout — if you set the option to **Yes**, the registered customers will need to fill in the field only once, after that the field will be automatically filled in (with the possibility to edit);

Show on Admin Grids — enable this option to display the attribute on the backend grids;

Include Into HTML Print-out — enable this option to include the attribute into the HTML print-outs;

Include Into PDF Documents — enable this option to include the attribute into the PDF documentation;

Include Into Transactional Emails — enable this option to include the attribute into the transactional emails.

Please, make sure that the **Include Order Attributes into Emails** [option](#) is enabled.

Manage Titles (Size, Color, etc.) ⌵

Default Store View

What is your position in the company?

Manage Tooltip ⌵

Manage Options (values of your attribute) ⌵

| Is Default (Unselect All) | Default Store View | Admin | |
|----------------------------------|------------------------------------------------|------------------------------------------------|---------------------------------------|
| <input checked="" type="radio"/> | <input type="text" value="Sales Manager"/> | <input type="text" value="Sales Manager"/> | <input type="button" value="Delete"/> |
| <input type="radio"/> | <input type="text" value="Marketing Manager"/> | <input type="text" value="Marketing Manager"/> | <input type="button" value="Delete"/> |
| <input type="radio"/> | <input type="text" value="Other"/> | <input type="text" value="Other"/> | <input type="button" value="Delete"/> |

Then, switch to the **Title, Tooltip, Options** tab:

Manage Titles — specify the attribute title for each store view for which it is enabled;

Manage Tooltip — set a tooltip to display near the attribute on the frontend.

Manage Conditions

Shipping Methods

Flat Rate
Fixed
Best Way
Table Rate

Please, note that if shipping methods are NOT selected in the field, order attributes will be displayed on the checkout page right after the page load. And if any is selected, order attributes will appear on the checkout page just when a user selects the shipping method

When done, please, proceed to the **Shipping Methods** tab:

Shipping Methods — specify particular shipping methods to show the extra attribute only to those customers who choose the specified methods.

Attribute Relations

To manage attribute relations please go to **Sales → Order Attributes → Manage Attribute Relations**.

Order Attribute Relations

[Add New Relation](#)

Filters | Default View | Columns

20 per page | 1 of 1

1 records found

| Relation Name ↓ | Parent Attribute | Dependent Attributes | Attribute Codes | Checkout Step | Action |
|---------------------------------|---------------------------------------|----------------------|---------------------------|------------------|----------------------|
| Join by position in the company | What is your position in the company? | Join us | company_position, join_us | Shipping Address | Edit |

On the Order Attribute Relations grid you'll see all created relations. It's possible to add an unlimited number of new relations and manage the already created one.

Please note, that attribute relations can be set for the attributes within the same Checkout Step only.

| Attribute Code ↓ | Attribute Label | Frontend Input Type | Visible | Checkout Step | Include Into PDFs | Show on Grid | Position | Action |
|------------------|---------------------------------------|---------------------|---------|------------------|-------------------|--------------|----------|----------------------|
| company_position | What is your position in the company? | Radio Buttons | Yes | Shipping Address | Yes | Yes | 1 | Edit |
| join_us | Join us | Checkbox Group | Yes | Shipping Address | No | Yes | 3 | Edit |

If the Checkout Steps are different, you will not be able to set the dependency.

Relation Information

Relation Name *
For internal use

Parent Attribute *

Attribute Options *

- Marketing Manager
- Other
- Sales Manager

Dependent Attributes *

- Delivery time
- Favorite Products
- Where did you find us
- Join us

Dependent attributes should be configured on the same checkout step as the parent attribute

Relation Name — specify a relation name that will be used for internal needs only;

Parent Attribute — choose the attribute that will be a parent;

Attribute Options — select attribute values that will trigger the display of the dependent attributes;

Dependent Attributes — specify which attributes will be dependent. These attributes will be displayed only if a customer selects one of the specified values of a parent attribute.

Please, mind that you can create relations only for the following types of attributes:

- Dropdown
- Checkbox Group
- Radio Buttons

Attributes Display in the Admin Panel

Order Grid

When you add a custom attribute to Magento order, it will appear on the **Orders Grid**.


Order View Page

Additional order attributes data will be displayed on the **Order info page**.

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Billing Address Edit Veronica Costello 6146 Honey Bluff Parkway Calder, Michigan, 49628-7978 United States T: (555) 229-3326 | Shipping Address Edit Veronica Costello 6146 Honey Bluff Parkway Calder, Michigan, 49628-7978 United States T: (555) 229-3326 |
| Custom Attributes Edit | |
| Delivery time | 2/28/17, 3:31 PM |
| Favorite Products | Order #2 |
| Where did you find us | Facebook |
| Join us | Affiliate Program |

Click **Edit** to modify custom attributes from the admin panel.

Attributes' Values

Delivery time 

Favorite Products

Where did you find us

- Google+
- Twitter
- Facebook
- YouTube
- Other

Join us * Affiliate Program

Business Club

HTML Printouts

Include custom attributes in **HTML** and **PDF** printouts.

| Product Name | SKU | Price |
|--------------------|---------|----------------|
| Joust Duffle Bag | 24-MB01 | \$34.00 |
| Overnight Duffle | 24-WB07 | \$45.00 |
| | | |
| Additional Info | | |
| Favorite products | | |
| Training Equipment | | |
| Date of birth | | |
| 4/25/86 | | |

Attributes on the Registration and Checkout Pages

See the example of custom order attributes with different input types (text fields, dropdowns, radio buttons, multiple checkboxes).

Choose delivery date and time

Favorite Products

What is your position in the company? *

Sales Manager
Marketing Manager
Other

Twitter

Facebook

YouTube

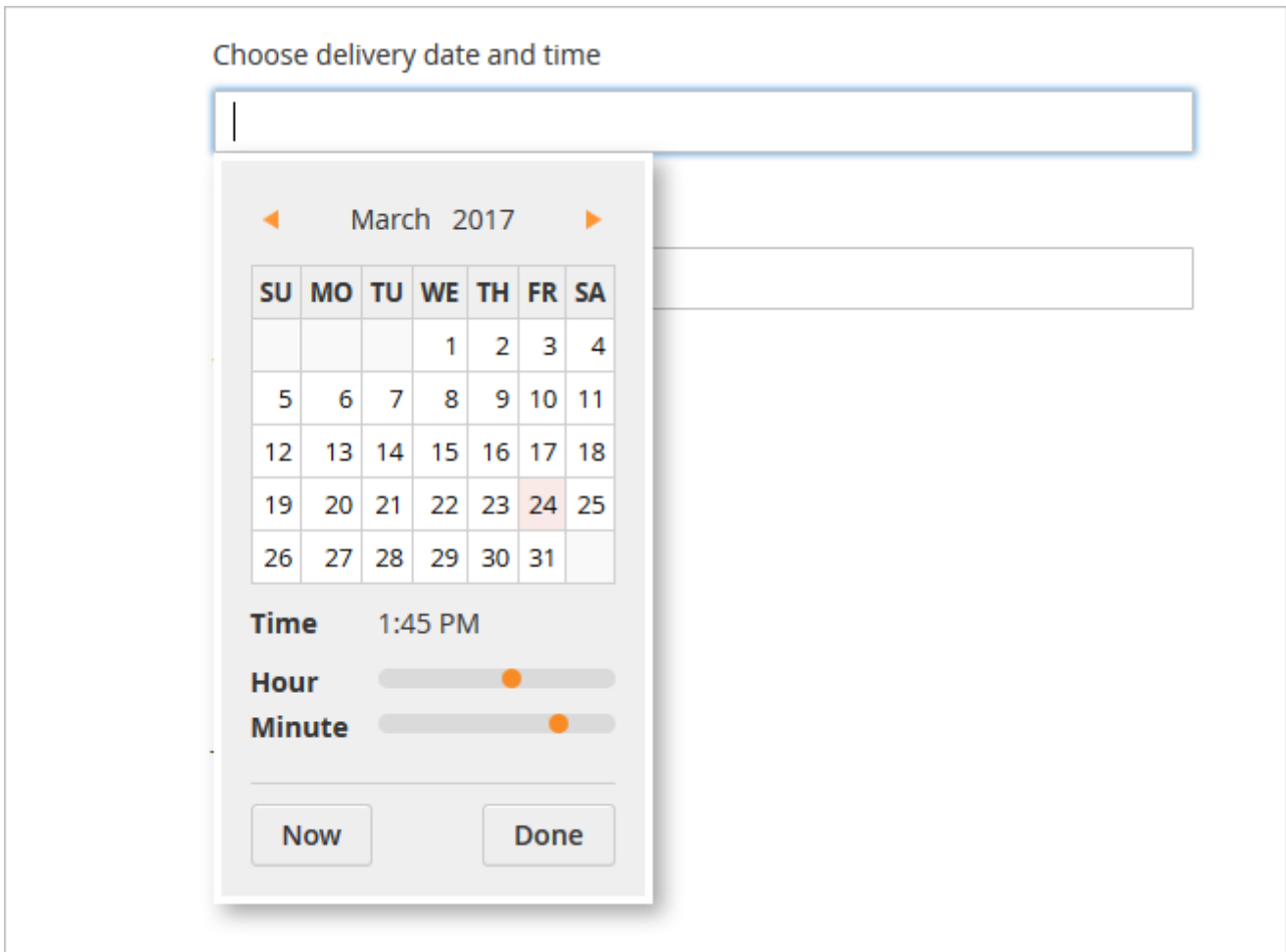
Other

Join us *

Affiliate Program

Business Club

Display a handy calendar for the **Date** and **Date and Time** input types. The calendar appears once a customer clicks on the attribute field.



Also, the customers will be provided with the selected attributes on the checkout page.

Attribute Variables in Email Templates

With the Magento 2 Order Attributes extension, you can include order attributes into the email templates.

Please, go to **Marketing** → **Email Templates** and hit the **Add New Template** button.

New Template

← Back Reset Convert to Plain Text Preview Template **Save Template**

Load default template

Template

Load Template

Template Information

Currently Used For Stores -> Configuration -> Sales Emails -> Order -> New Order Confirmation Template (Default Config)

Template Name *

Template Subject *

Insert Variable...

Template Content *

```
{{template config_path="design/email/header_template"}}  
  
<table>  
  <tr class="email-intro">  
    <td>  
      <p class="greeting">{{trans "%customer_name,"  
customer_name=$order.getCustomerName()}}</p>  
      <p>  
        {{trans "Thank you for your order from %store_name,"  
store_name=$store.getFrontendName()}}
```

Template Styles

Next, select the required template from the **Template** dropdown menu and click the **Load Template** button. Then, specify the **Template Name** and proceed to the **Template Content** field. You can add the order attributes here. Click the Insert Variable button and choose the attribute you need to insert it in the template.

Template Variables

[Billing Address](#)

[Email Order Note](#)

[Order Id](#)

[Order Items Grid](#)

[Payment Details](#)

[Shipping Address](#)

[Shipping Description](#)

[Shipping message](#)

[Amasty Order Attribute: What is your position in the company?](#)

[Amasty Order Attribute: Favorite Products](#)

[Amasty Order Attribute: Delivery time](#)

[Amasty Order Attribute: Join us](#)

[Amasty Order Attribute: Where did you find us](#)

When done, click the **Save Template** button.

GENERAL ▾

CATALOG ▾

CUSTOMERS ▾

SALES ▲

Sales

Sales Emails

PDF Print-outs

Tax

Checkout

Shipping Settings

General Settings

Order

Enabled [store view] Yes ▾

New Order Confirmation Email Sender [store view] Sales Representative ▾

New Order Confirmation Template [store view] New Order with Attributes ▾
Email template chosen based on theme fallback when "Default" option is selected.

New Order Confirmation Template for Guest [store view] New Order for Guest (Default) ▾
Email template chosen based on theme fallback when "Default" option is selected.

Send Order Email Copy To [store view]
Comma-separated

Send Order Email Copy Method [store view] Bcc ▾

Additionally, you may need to set the new template as the main one for emails. Please, go to **Stores** → **Configuration** → **Sales** → **Sales Emails**. Then, expand the required tab and select the newly created template.

Compatibility with REST/SOAP API

Order Attributes extension for Magento 2 supports API. To get order attributes using the API, please look through these steps and methods:

- To get order data you should be authorized as admin and have the API Authorization token
- Use the API URLs specified in webapi.xml file:

```
webapi.xml x
<?xml version="1.0"?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc/webapi.xsd">

  <route url="/V1/amasty_orderattr/entityData" method="POST">
    <service class="Amasty\Orderattr\Api\EntityDataRepositoryInterface" method="save"/>
    <resources>
      <resource ref="Amasty_Orderattr::order_attributes" />
    </resources>
  </route>
  <route url="/V1/amasty_orderattr/entityData/:entityId" method="PUT">
    <service class="Amasty\Orderattr\Api\EntityDataRepositoryInterface" method="save"/>
    <resources>
      <resource ref="Amasty_Orderattr::order_attributes" />
    </resources>
  </route>
  <route url="/V1/amasty_orderattr/entityData/:entityId" method="DELETE">
    <service class="Amasty\Orderattr\Api\EntityDataRepositoryInterface" method="deleteById"/>
    <resources>
      <resource ref="Amasty_Orderattr::order_attributes" />
    </resources>
  </route>
  <route url="/V1/amasty_orderattr/entityData" method="GET">
    <service class="Amasty\Orderattr\Api\EntityDataRepositoryInterface" method="getById"/>
    <resources>
      <resource ref="Amasty_Orderattr::order_attributes" />
    </resources>
  </route>
  <route url="/V1/amasty_orderattr/entityData/all" method="GET">
    <service class="Amasty\Orderattr\Api\EntityDataRepositoryInterface" method="getList" />
    <resources>
      <resource ref="Amasty_Orderattr::order_attributes" />
    </resources>
  </route>
  <route url="/V1/amasty_orderattr/checkoutData" method="POST">
    <service class="Amasty\Orderattr\Api\CheckoutDataRepositoryInterface" method="save"/>
    <resources>
      <resource ref="self" />
    </resources>
    <data>
      <parameter name="amastyCartId" force="true">%amasty_cart_id%/parameter>
    </data>
  </route>
  <route url="/V1/amasty_orderattr/guestCheckoutData" method="POST">
    <service class="Amasty\Orderattr\Api\GuestCheckoutDataRepositoryInterface" method="save"/>
    <resources>
      <resource ref="anonymous" />
    </resources>
  </route>
</routes>
```

This is how it works:

When you navigate to the second step of the checkout page or place an order (depending on the location of the attribute), the system creates an API query and sends it to the:

```
class="Amasty\Orderattr\Api\CheckoutDataRepositoryInterface" method="save"
```

This class takes the obligatory parameters:

```
int $amastyCartId
string $checkoutFormCode
string $shippingMethodCode
\Amasty\Orderattr\Api\Data\EntityDataInterface $entityData
```

Now the information about order attributes is recorded in database.

In order to pull the order attributes data out of DB, the following class was added:

```
class="Amasty\Orderattr\Api\EntityDataRepositoryInterface"
```

```
webapi.xml x EntityDataRepositoryInterface.php x CheckoutDataRepositoryInterface.php
<?php
namespace Amasty\Orderattr\Api;

/**
 * @api
 */
interface EntityDataRepositoryInterface
{
    /**
     * Save
     *
     * @param \Amasty\Orderattr\Api\Data\EntityDataInterface $entityData
     * @return \Amasty\Orderattr\Api\Data\EntityDataInterface
     */
    public function save(\Amasty\Orderattr\Api\Data\EntityDataInterface $entityData);

    /**
     * Get by id
     *
     * @param int $entityId
     * @return \Amasty\Orderattr\Api\Data\EntityDataInterface
     * @throws \Magento\Framework\Exception\NoSuchEntityException
     */
    public function getById($entityId);

    /**
     * Delete
     *
     * @param \Amasty\Orderattr\Api\Data\EntityDataInterface $entityData
     * @return bool true on success
     * @throws \Magento\Framework\Exception\CouldNotDeleteException
     */
    public function delete(\Amasty\Orderattr\Api\Data\EntityDataInterface $entityData);

    /**
     * Delete by id
     *
     * @param int $entityId
     * @return bool true on success
     * @throws \Magento\Framework\Exception\CouldNotDeleteException
     */
    public function deleteById($entityId);

    /**
     * Lists
     *
     * @param \Magento\Framework\Api\SearchCriteriaInterface $searchCriteria
     * @return \Magento\Framework\Api\SearchResultsInterface
     * @throws \Magento\Framework\Exception\NoSuchEntityException
     */
    public function getList(\Magento\Framework\Api\SearchCriteriaInterface $searchCriteria);
}
```

Steps:

1. Go to **Swagger** (<http://.../swagger>).
2. Find **integrationAdminTokenServiceV1** and log in with admin rights.

POST /V1/integration/admin/token

Create access token for admin given the admin credentials.

Parameters Try it out

| Name | Description |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| integrationAdminTokenServiceV1CreateAdminAccessTokenPostBody <i>(body)</i> | <p>Example Value Model</p> <pre>{ "username": "admin", "password": "a111111" }</pre> <p>Parameter content type</p> <p>application/json</p> |

Responses Response content type application/json

- 3. Take the token in **Response body**.
- 4. Enter the token in a search string and click **Apply**.

swagger [token] apply

Magento Enterprise ^{2.3}

[Base URL: 230ee.aivankov.ap71.sty/rest/all]
<http://230ee.aivankov.ap71.sty/rest/all/schema?services=all>

Schemes: HTTP

- directoryCurrencyInformationAcquirerV1 Currency information acquirer interface >
- directoryCountryInformationAcquirerV1 Country information acquirer interface >
- customerAccountManagementV1 Interface for managing customers accounts. >

- 5. Find a particular method (for example, **amastyOrderattrEntityDataRepositoryV1**)

GET /V1/amasty_orderattr/entityData

Get by id

Parameters Cancel

| Name | Description |
|--------------------------------------------------|--------------------------------|
| entityId * required integer (query) | <input type="text" value="9"/> |

Execute Clear

6. Enter **entityId** (it's in *amasty_order_attribute_entity* table)

Request URL

```
http://230ee.aivankov.ap71.sty/rest/all/V1/amasty_orderattr/entityData?entityId=9
```

Server response

| Code | Details |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 200 | <p>Response body</p> <pre>{ "entity_id": 9, "parent_id": 26, "parent_entity_type": 1, "custom_attributes": [{ "attribute_code": "order_test", "value": "Izivankov12" }, { "attribute_code": "order_at_2", "value": "225" }, { "attribute_code": "sss", "value": "229" }] }</pre> <p>Response headers</p> <pre>cache-control: no-store, no-cache, must-revalidate connection: Keep-Alive content-length: 214 content-type: application/json; charset=utf-8 date: Thu, 25 Jul 2019 07:35:37 GMT expires: Thu, 19 Nov 1981 08:52:00 GMT keep-alive: timeout=5, max=100 pragma: no-cache server: Apache/2.4.18 (Ubuntu) x-frame-options: SAMEORIGIN</pre> |

Responses

| Code | Description |
|------|--------------|
| 200 | 200 Success. |

Done! The result will show up in **Response body**.

Besides, all of these manipulations can be carried out with Remote Access simply by sending a query to your website specifying **url/rest/all/V1/amasty_orderattr/entityData?param=value**, for example:

```
http://.../rest/all/V1/amasty_orderattr/entityData?entityId=9
```

Troubleshooting

Compatibility with Magento 2 Multishipping

Please kindly note that the Order Attributes extension is not intended to use with Magento Multishipping functionality.

Find out how to install the **Order Attributes** extension for Magento 2 via [Composer](#).

From:
<https://stg.amasty.net/docs/> - **Amasty Extensions FAQ**

Permanent link:
https://stg.amasty.net/docs/doku.php?id=magento_2:order_attributes



Last update: **2023/12/19 11:59**