

Magento2 Coupon Auto Share at Checkout Module User Guide

Introduction of the module:

Coupon Auto Share at Checkout module allows admin to share the coupon codes to the registered and guest customers via order confirmation email. This module will automatically create the cart rule based on the configuration, and the admin can manage that cart rule based on the needs.

Installation Steps:

Manual Installation Steps (Installing the extension by copying code)

Step 1: Go to the Magento root directory, then go inside the /app/code directory.

Step 2: Create the folder Wagento/CouponShare

Step 3: Download the extension zip and extract all files and folders inside the CouponShare directory.

Step 4: Open your server terminal and go to your store root directory, then RUN below command step by step:

1. `php bin/magento setup:upgrade`
2. `php bin/magento setup:di:compile`
3. `php bin/magento setup:static-content:deploy`
4. `php bin/magento cache:flush`

Configuration:

Login to your admin panel and go to the [Menu > Stores > Settings > Configuration > Wagento > Coupon Share](#) section.

General Configuration

Enable Module [website]	Yes
Cart Rule Name [website]	10% Off For Second Order
	Entered the cart rule name automatically created/update while saving configuration. You can update the settings from Marketing >> Promotions >> Cart Price Rules
Dynamically Coupons Generated [website]	No
	Enable this option to generate dynamically coupons.
Import Coupons [website]	<input type="button" value="Choose file"/> No file chosen
	Used Coupons: 4 Total Coupons: 4 Sample CSV To Import: DOWNLOAD
Export All Coupons [website]	<input type="button" value="Download"/>
	You will get the CSV file with all coupons code in the cart price rule.

General Configuration:

1. **Enable Module:** Set as Yes to enable module
2. **Cart Rule Name:** Define any Rule Name which is created in the Marketing > Promotion > Cart Price Rule
Example: 10% OFF ON SECOND ORDER
3. **Dynamically Coupons Generated:** Enable this option when you don't have any coupons so that coupons will be created automatically when it's needed. A newly

created coupon will be shown on the cart price rule as well.

4. **Import Coupons:** Just import CSV files with only coupon codes. All imported coupons will be imported on the cart price rule as well.
5. **Export All Coupons:** You are able to download CSV files with all coupons which are imported by admin and you can also see which coupons are used or not.

Admin Notification Configuration

Active Notification
[website] Yes

Email Address
[website] rajan.sonl@wagento.com

Enter email address to get notification about How many coupons are remaining?

Notify for Remaining Coupons Below
[website] 5

Enter the only number to get a notification when unused coupons qty reach below from your entered number.

Notification Email Template
[website] Admin Notification Template (Default)

Admin Notification Configuration:

1. **Active Notification:** Enable this option to get notification about how many coupons are left.
 2. **Email Address:** Enter the email address to get the notification.
 3. **Notify for Remaining Coupons Below:** Enter the number so admin will get the notification when coupon qty reaches below of your entered number.
 4. **Notification Email Template:** By default, it will take the default email template for notification.
-

Setup Short Code In The Order Email Template.



Please follow the below steps to set coupon code in email templates:

- Step 1: Go to the **Marketing >> Communications >> Email Templates**
- Step 2: Click on the **Add New Template** button.
- Step 3: Load default template >> **Magento_Sales** >> select **New Order** and click on the Load Template button.
- Step 4: Copy the shortcode: **{{var couponcode|raw}}**
- Step 5: At the specific location you can paste the above shortcode in the **Template Content** section
- Step 6: Save the template with a specific email template name and you can click on the Preview Template button to check the layout of the template.
- Step 7: Go to the **Stores >> Settings >> Configurations >> Sales >> Sales Email >> Order** >> Select your recently created template in the **New Order Confirmation Template** field.
- Step 8: Save the configurations.

If you have enabled the guest checkout option. Do the above 1 to 8 same steps for **Guest Order Email** In the above steps you just need to take create below points:

- In step 3, select **New Order For Guest**
- In step 7, select your recently created template in the **New Order Confirmation Template for Guest**

Setup Short Code In The Order Email Template:

Note: Please follow the below steps to set coupon code in email templates:

Step 1: Go to the **Marketing >> Communications >> Email Templates**

Step 2: Click on the **Add New Template** button.

Step 3: Load default template >> **Magento_Sales** >> select **New Order** and click on the Load Template button.

Step 4: Copy the shortcode: **{{var couponcode|raw}}**

Step 5: At the specific location you can paste the above shortcode in the **Template Content** section

Step 6: Save the template with a specific email template name and you can click on the Preview Template button to check the layout of the template.

Step 7: Go to the **Stores > Settings > Configurations > Sales > Sales Email > Order >** Select your recently created template in the **New Order Confirmation Template** field.

Step 8: Save the configurations.

Note: If you have enabled the guest checkout option. Do the above 1 to 8 same steps for **Guest Order Email** In the above steps you just need to take create below points:

In **step 3**, select **New Order For Guest**

In **step 7**, select your recently created template in the **New Order Confirmation Template for Guest**

To uninstall module:

Step 1: Remove the module directory CouponShare from the Magento Root/app/code/Wagento path.

Step 2: Open your server terminal and go to your store root directory, then RUN below command step by step:

1. `php bin/magento setup:upgrade`
2. `php bin/magento setup:di:compile`
3. `php bin/magento setup:static-content:deploy`
4. `php bin/magento cache:flush`