

WhatsApp Notify & Promotions

for Magento 2

Connect Your WhatsApp API with Magento store and send dynamic WhatsApp templates with ease

Overview

Transform your customer communication with our WhatsApp Notify and Promotion extension. The extension lets you connect the WhatsApp API with Magento 2, and is designed ! Designed to enhance engagement and boost retention, this feature-rich extension lets you send dynamic, personalised messages to your customers. Seamlessly support all default Magento 2 user groups and effortlessly pass dynamic variables to Meta WhatsApp Templates. While direct template creation or editing isn't available, you'll love the versatility of our variety of trigger types including event-based triggers for automatic messages on order updates or shipment creation, manual triggers for scheduled messages at your chosen frequency, and abandoned cart reminders to win back lost sales. Elevate your customer experience with timely, relevant communication that keeps your customers coming back for more

Prerequisite:

In order to use this extension you need the following things:

- Verified Meta Business Account
- Meta Developer Account
- Phone number with no active WhatsApp account
- Meta App with WhatsApp Instance

You can follow the below guide to setup Meta App to access the WhatsApp APIs
<https://developers.facebook.com/docs/whatsapp/cloud-api/get-started>

Features

- Fully Customizable Campaigns
- Add Dynamic Values to Templates using Variables
- Supports Default Magento Customer Groups
- Support for Abandoned Cart Notifications
- Setup Event Based Trigger (Send Notification based on Specific Events like Order Updates, Order Shipments, Refunds, etc)
- Schedule a manual trigger for WhatsApp Campaigns
- Setup Repeat Trigger and Message Loops based on specific criteria
- Browser and Device Compatibility

How To Install

Method 1 – Zip file

1. Unzip the zip file in app/code/Azguards
2. Enable the module by running `php bin/magento module:enable Azguards_WhatsappNotify`
3. Apply database updates by running `php bin/magento setup:upgrade`
4. Apply static deploy by running `php bin/magento setup:static-content:deploy -f`
5. Flush the cache by running `php bin/magento cache:flush``

Method 2 – Composer

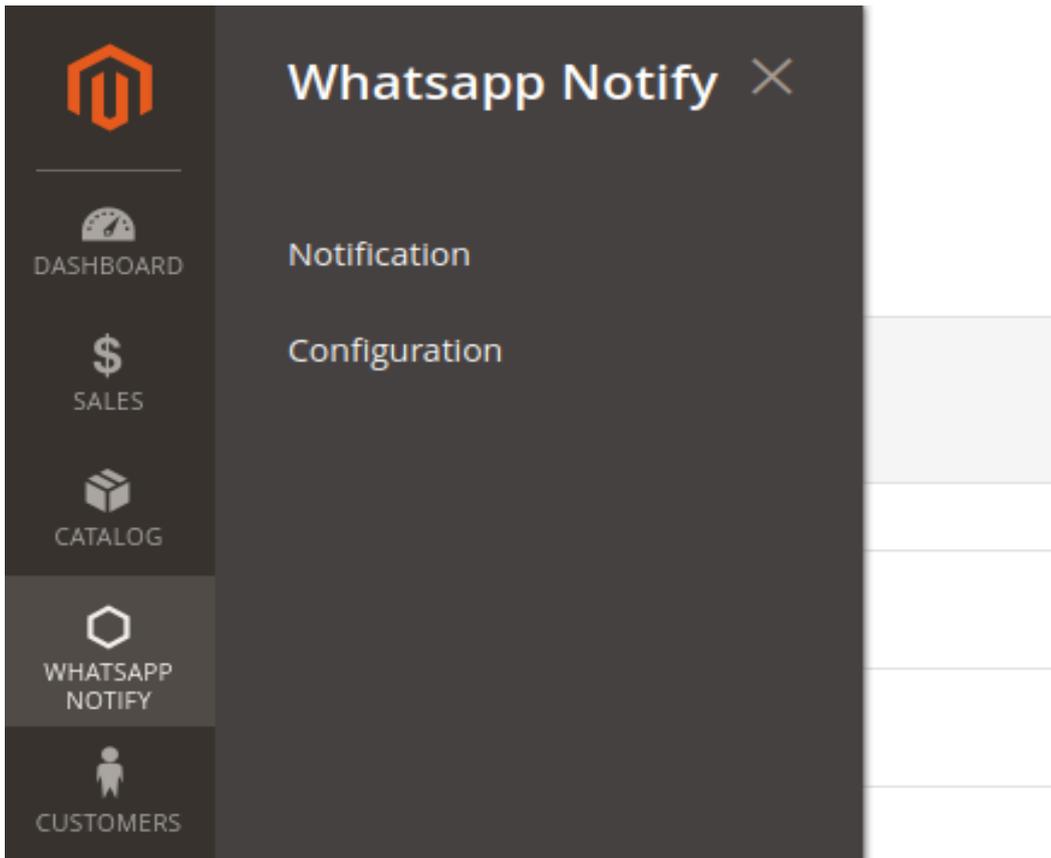
1. Add the composer repository to the configuration by running
``composer config repositories.repo.magento.com composer https://repo.magento.com/``
2. Install the module composer by running ``composer require azguards/module-whatsappnotify``
3. Enable the module by running ``php bin/magento module:enable Azguards_WhatsappNotify``
4. Apply database updates by running ``php bin/magento setup:upgrade``
5. Apply static deploy by running `php bin/magento setup:static-content:deploy -f``
6. Flush the cache by running ``php bin/magento cache:flush``

How to Configure:

Configuring and using the WhatsApp Notifications feature is a straightforward process. After installing the extension, follow these simple steps to make the most of its benefits.

Step – 1: Navigate to Store Configuration

Menu ->WhasApp Notify



Step – 2: Added value for Configuration

Task "Rule processing: 2": 1 item(s) have been scheduled for update. [View Details](#) System Messages: 4

Configuration

Scope: Default Config [?](#) [Save Config](#)

GENERAL	General
SECURITY	
CATALOG	
CUSTOMERS	
SALES	
AZGUARDS	
Whatsapp Notify	
SpamBuster	
SERVICES	
ADVANCED	

Access Token (store view)

Phone Number Id (store view)

Whatsapp Business Account Id (store view)

App Id (store view)

WhatsApp Number (store view)
Add Whatsapp Number with Country Code.

Phone Number (store view)
Add Phone Number with Country Code.

Website Url (store view)

Find the following values from the meta developer portal inside the app you created for WhatsApp API.

1. Access Token
2. Phone Number ID
3. WhatsApp Business Account ID
4. App ID

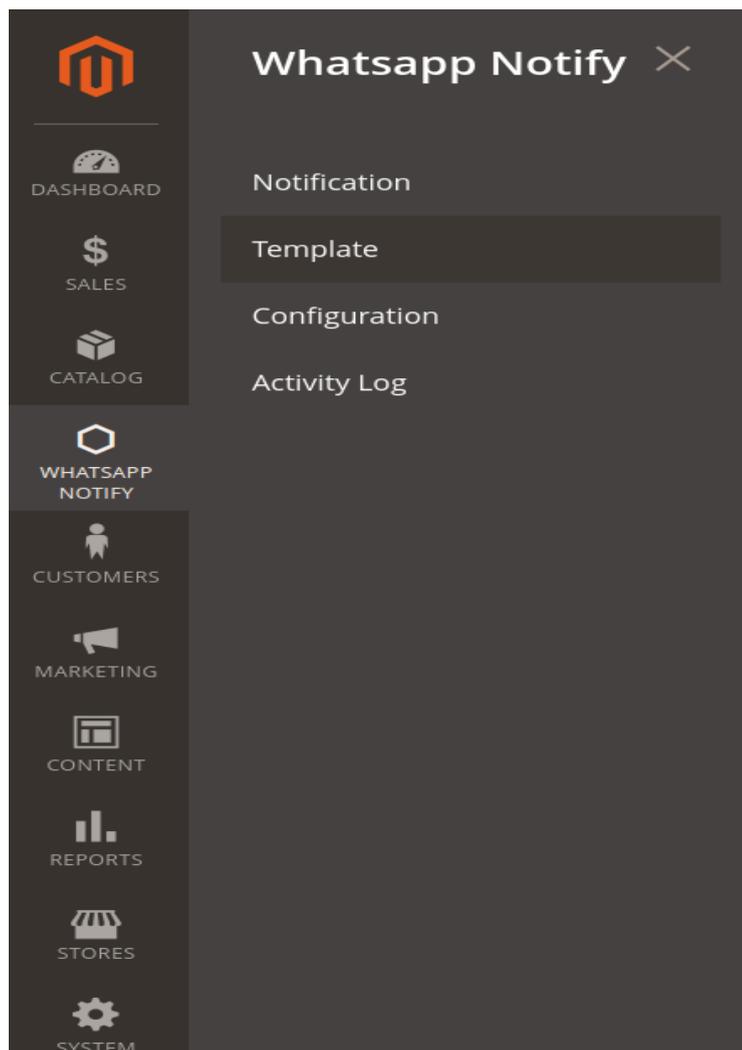
The other fields are for template button-

1. WhatsApp Number
2. Phone Number
3. Website Url

(*Note:All above config fields are required)

Step – 3: Select Whatsapp Notify from the menu and select Template.

WhasApp Notify -> Template



Step – 3.1: How to Create a New Template

WhatsApp Notify -> Templates , and press Add New Template button. You will be brought to the creation page - as shown on the screenshot below.

The screenshot shows the 'New Template' creation interface. At the top, there's a header with 'New Template' on the left and three buttons: 'Back', 'Save and Continue Edit', and 'Save Template'. The form fields are as follows:

- Title ***: Text input field containing 'Orderplaced text template'.
- Template Type ***: Dropdown menu with 'Text' selected.
- category ***: Dropdown menu with 'Marketing' selected.
- Header Text**: Text area containing 'Dear {{customer_name}},'.
- Body ***: Text area containing a longer message with placeholders like {{order_number}} and {{items_ordered}}.
- Destination Uri**: Text input field containing 'https://dev.magento/'.
- Destination Label**: Text input field containing 'Dev Site'.
- Phone No**: Toggle switch set to 'No'.
- Website**: Toggle switch set to 'Yes'.

Title - a sensical title, which will be displayed on the grid.

Template Type - select template type dropdown value (text, Image).

Header Text - a short message for the header of the main article or page.

Body - a short message for the customer or a header of the main article or Page.

Custom button - Phone No , Website , Destination url

Custom Variables

To enhance and personalise your messages, you can use special variables. However note that these variables will only work for logged in Users.

Here is their shortlist:

- To add customer name use {{customer_name}}
- To add Order Id use {{order_number}}
- To add name of item ordered use {{item_ordered}}

Step – 3.2: How to Create a New Notification

WhatsApp Notify -> Notification , and press Add New Notification button. You will be brought to the creation page - as shown on the screenshot below.

The screenshot displays the 'Edit Notification 1' page. At the top, there is a yellow notification bar stating 'Task "Rule processing: 2": 1 item(s) have been scheduled for update.' and a 'View Details' link. The page title is 'Edit Notification 1' and the user 'admin' is logged in. The main content area is titled 'General' and contains the following fields:

- Enable Notification:** A toggle switch set to 'Yes'.
- Name:** A text input field containing 'Order notification'.
- Customer Group:** A dropdown menu with options: ALL GROUPS, NOT LOGGED IN, General, Wholesale, and Retailer.
- Template:** A dropdown menu set to 'test_text_template_0001'.
- Notification Type:** A dropdown menu set to 'Event Based Trigger'.

Below the 'General' section is the 'Triggered Notification' section, which includes a 'Trigger Event' dropdown menu set to 'Order Success'.

Name – is the sensical name of the notification which helps you identify the notification from the admin grid. It will not be seen by the customer

Enable Notification – Whether notification is active.

Customer Group: You can select specific customer groups to whom you want to send the Whatsapp message

Template – Load a template which contains a message that you want to push to the customer. You can review the template section for further information.

Trigger Types

The Whatsapp Extension Gives your two different kind of triggers, event based trigger and time based trigger

Event Based Trigger

The event based trigger allows the admin to set up a push notification whenever a specific event occurs. For example if you set up a push notification for order shipment, whenever the order status is changed to shipped, the customer will automatically get the notification about the same.

Notification Type *

Triggered Notification

Trigger Event

Note: Event based triggered notifications only work with the logged in users, the guest users will not be getting these notifications.

Time Based Trigger

Time based trigger will allow you to send push notifications on specific time. Azguards Whatsapp Notifications Gives you two options in time based trigger

One Time

You can set up a specific date and time when the notification will be sent to the selected group of the users.

Notification Type * Time Based Trigger ▼

Manual Notification

Manual Type One Time ▼

Select Date 

Repeated

One of the best features of Azguards Whatsapp Notifications is that you can also set up automatic repeated notifications or notification loops based on following criterias

Manual Notification

Manual Type Repeated ▼

Repeat Type -- Please Select -- ▼

Frequency

Enter the frequency of Repeat. Note: Please add frequency grater than 1

From 

To 

Frequency: Enter the frequency of Repeat value 1 or more.

Repeat Type: When you want to repeat the notification, hour, day, week or month.

From: Mention the date and time when you want to start the campaign i.e. when you want to send your first notification

To: Date and time when you want to stop sending the specific notification.

Abandoned Cart

Before To Send cart abandoned Notification - Before sending the cart abandonment notification, set the time for when the notification will be sent after the item is added to the cart.

Notification Type * Abandoned Cart

Abandoned Cart Notification

Before To Send cart abandoned Notification 1 Hours

Custom Variables

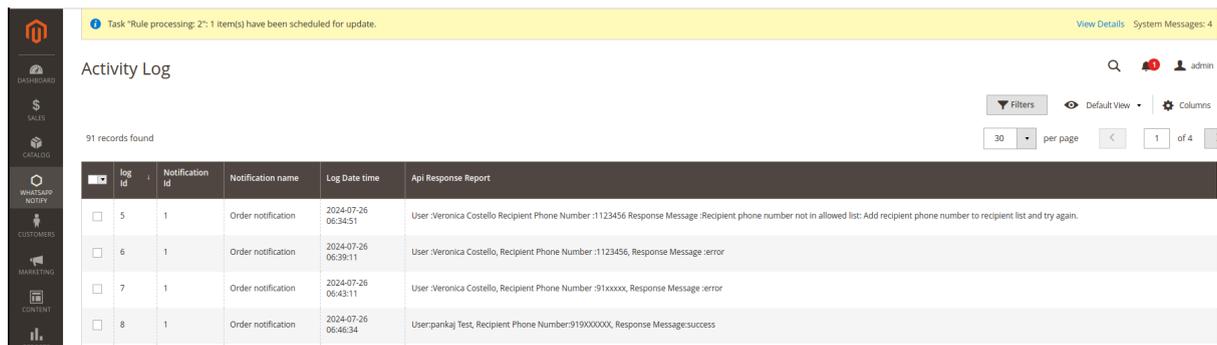
To enhance and personalise your messages, you can use special Variables in Abandoned cart template. However note that these variables will only work for logged in Users.

Here is their shortlist:

- To add customer name use `{{customer_name}}`
- To add cart total use `{{cart_total}}`
- To add name of cart item use `{{cart_items}}`

Activity Log

WhatsApp Notify -> Activity Log , as shown on the screenshot below.



The screenshot shows a web application interface with a sidebar on the left containing icons for Dashboard, Sales, Catalog, WhatsApp Notify, Customers, Marketing, and Content. The main area displays the 'Activity Log' for 'WhatsApp Notify'. A yellow banner at the top indicates a task: 'Task *Rule processing: 2*: 1 item(s) have been scheduled for update.' with links for 'View Details' and 'System Messages: 4'. Below the banner, the 'Activity Log' title is followed by a search icon, a notification bell, and the user 'admin'. There are controls for 'Filters', 'Default View', and 'Columns'. A pagination bar shows '91 records found', '30 per page', and '1 of 4' pages. The table below has columns for 'log id', 'Notification id', 'Notification name', 'Log Date time', and 'Api Response Report'.

log id	Notification id	Notification name	Log Date time	Api Response Report
5	1	Order notification	2024-07-26 06:34:51	User :Veronica Costello, Recipient Phone Number :1123456, Response Message :Recipient phone number not in allowed list: Add recipient phone number to recipient list and try again.
6	1	Order notification	2024-07-26 06:39:11	User :Veronica Costello, Recipient Phone Number :1123456, Response Message :error
7	1	Order notification	2024-07-26 06:43:11	User :Veronica Costello, Recipient Phone Number :91xxxxx, Response Message :error
8	1	Order notification	2024-07-26 06:46:34	User:pankaj Test, Recipient Phone Number:919XXXXXX, Response Message:succes

Note: Configure cron jobs

The time based trigger depends on Crons, so please configure cron jobs using the following guide

<https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cli/configure-cron-jobs.html?lang=en>