How to Create Shipping Method in Magento 2

Vinh Jacker | 12-18-2024

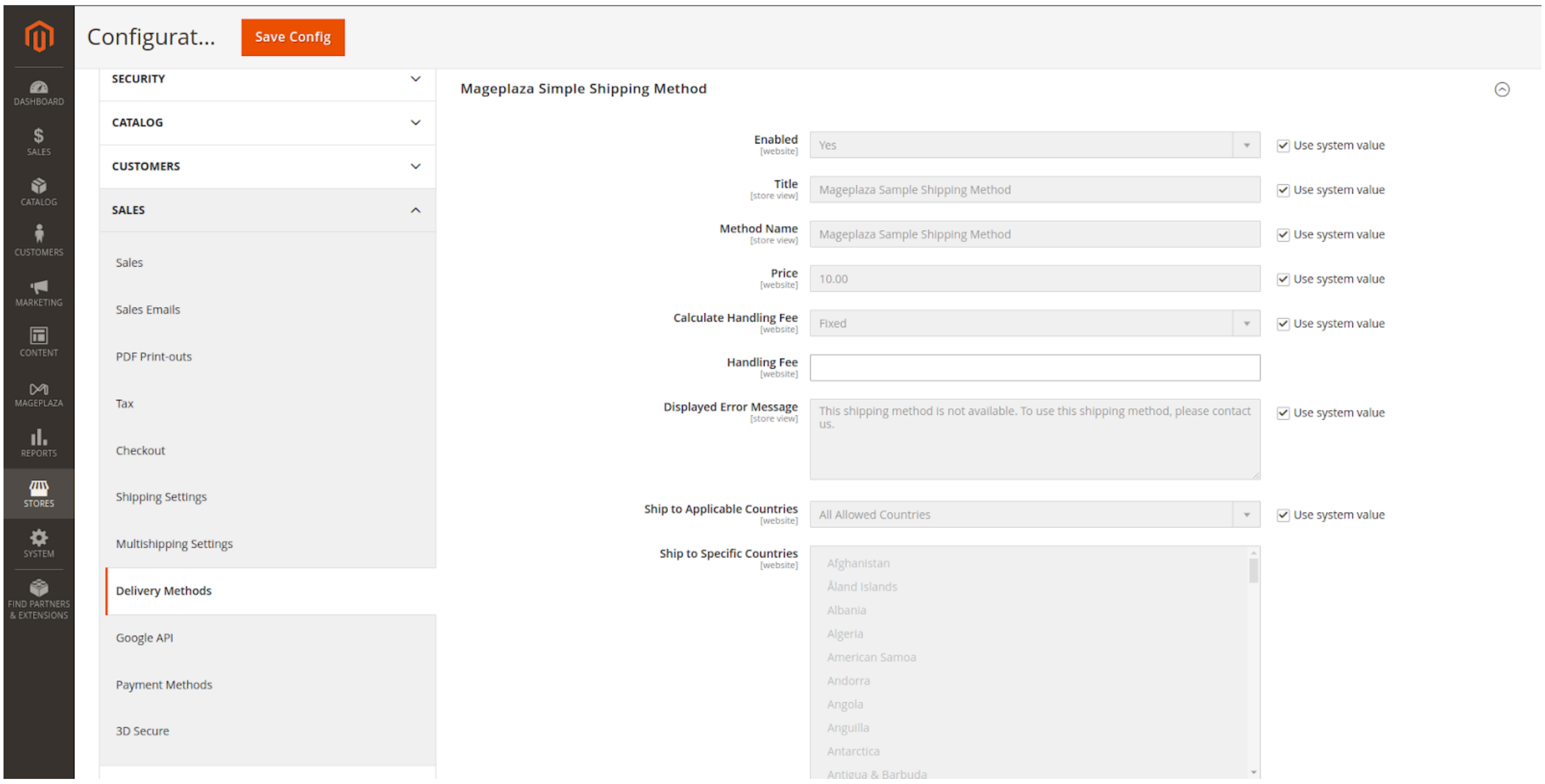
E-commerce success is strongly dependent on customer convenience, and a seamless shipping experience is vital for any online store. Of course, Magento 2 is a rich eCommerce platform and it also supports few shipping methods in the checkout process. Magento 2 provides default shipping methods like Free Shipping, Flat Rates, Table Rates, UPS, and FedEx, among others.

However, as e-commerce grows, businesses often need custom solutions. **Programmatic creating a custom shipping method** that is suitable to your business requirements is essential. Real-time cost calculation during checkout will enhance user experience, and offering **free shipping can boost sales**.

There are several pieces of evidence proving this with data. According to 99firms, 61% of shoppers abandon transactions due to additional costs like shipping fees, and 46.5% of small to mid-sized businesses said that providing free shipping increases their profits.

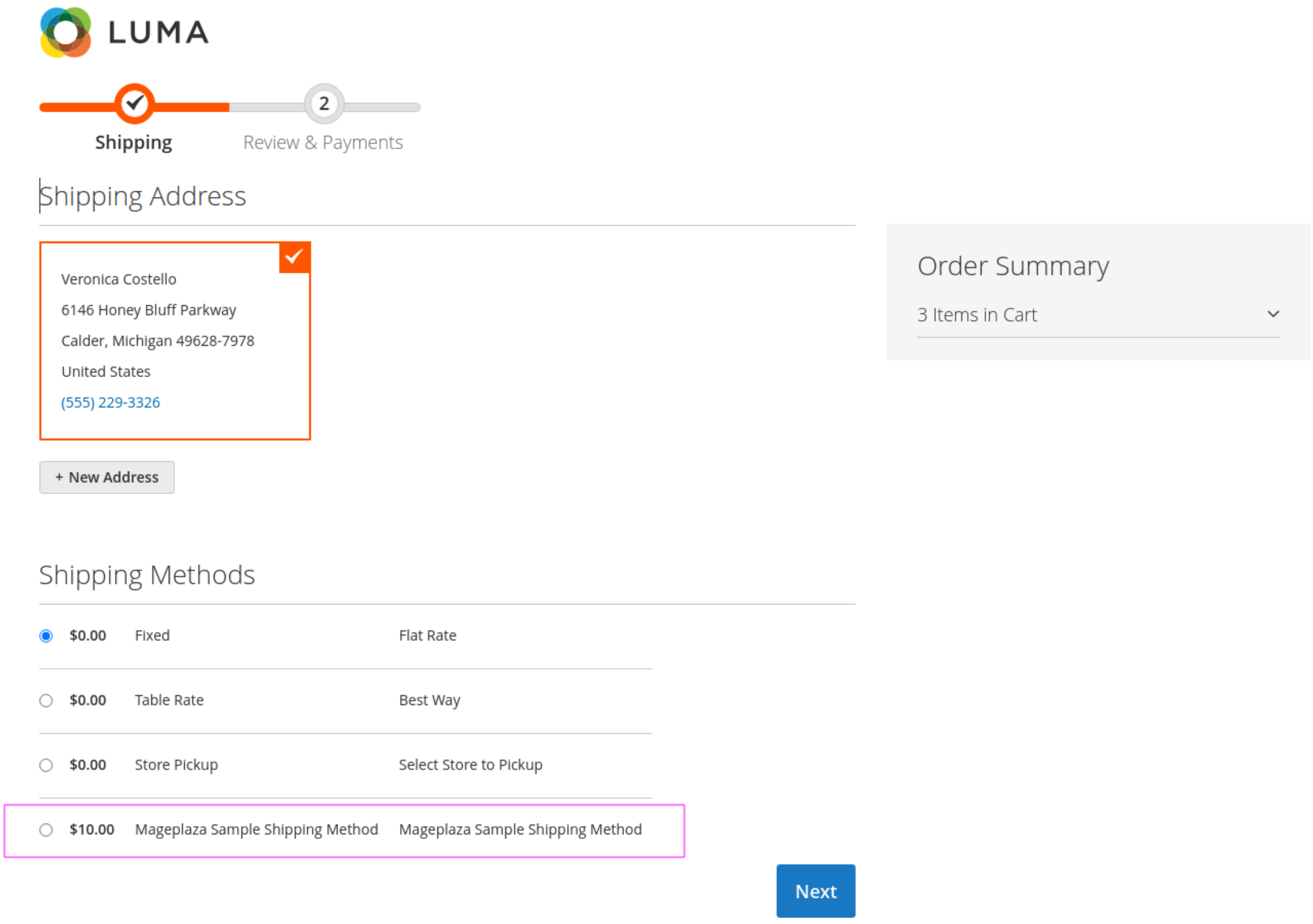
Remember, an optimized shipping strategy brings **benefits to both customers and profits**. Let's explore how to create custom shipping methods in Magento 2!

With the simple explanation, it is accessible to follow step-by-step and complete the creation of new shipping methods. All generated shipping methods are stored in Magento Admin Panel. Please go to `Stores > Settings > Configuration > Sales > Delivery Methods` to find and enable it on the storefront. But hold on, access the file `/Model/Carries/Generatedshippingmethod.php` in which you can set the specific shipping cost for each shipping method. Namely to create the shipping method, please keep tracking on the following steps.



Mageplaza Simple Shipping Method

Enabled [website]	Yes	☑ Use system value
Title [store view]	Mageplaza Sample Shipping Method	☑ Use system value
Method Name [store view]	Mageplaza Sample Shipping Method	☑ Use system value
Price [website]	10.00	☑ Use system value
Calculate Handling Fee [website]	Fixed	☑ Use system value
Handling Fee [website]		
Displayed Error Message [store view]	This shipping method is not available. To use this shipping method, please contact us.	☑ Use system value
Ship to Applicable Countries [website]	All Allowed Countries	☑ Use system value
Ship to Specific Countries [website]	Afghanistan Åland Islands Albania Algeria American Samoa Andorra Angola Anguilla Antarctica Antigua & Barbuda	



LUMA

Shipping Address

Veronica Costello
6146 Honey Bluff Parkway
Calder, Michigan 49628-7978
United States
(555) 229-3326

+ New Address

Order Summary

3 Items in Cart

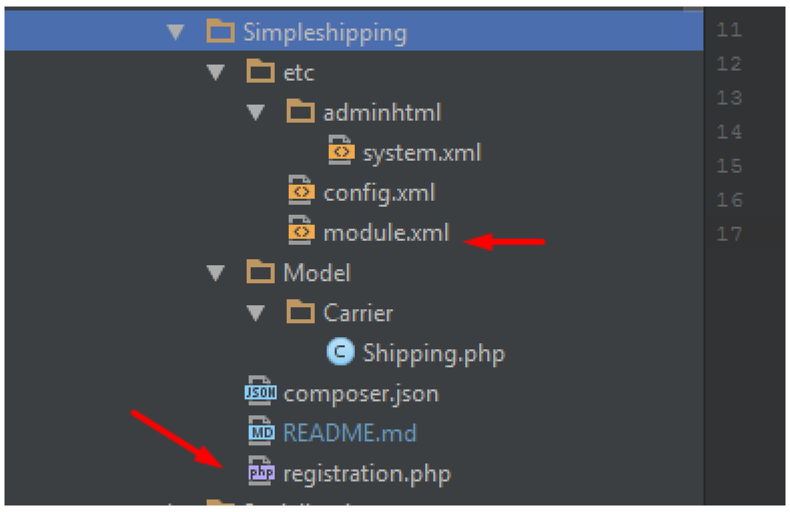
Shipping Methods

●	$0.00	Fixed	Flat Rate
○	$0.00	Table Rate	Best Way
○	$0.00	Store Pickup	Select Store to Pickup
○	$10.00	Mageplaza Sample Shipping Method	Mageplaza Sample Shipping Method

How to create simple shipping method in Magento 2

Step 1: Declare a Shipping module

Create a module call: `Simpleshipping` located in `app/code/Mageplaza/Simpleshipping`



```
▼ Simpleshipping
    ▼ etc
        ▼ adminhtml
            system.xml
        config.xml
        module.xml
    ▼ Model
        ▼ Carrier
            Shipping.php
    composer.json
    README.md
    registration.php
```

We have at least 2 files:

File: app/code/Mageplaza/Simpleshipping/registration.php

```
<?php
\Magento\Framework\Component\ComponentRegistrar::register(
    \Magento\Framework\Component\ComponentRegistrar::MODULE,
    'Mageplaza_Simpleshipping',
    __DIR__
);
```

File: app/code/Mageplaza/Simpleshipping/etc/module.xml

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
    <module name="Mageplaza_Simpleshipping" setup_version="1.0.0" />
</config>
```

Now create `config.xml` at `app/code/Mageplaza/Simpleshipping/etc/config.xml`

File: config.xml

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Store:etc/config.xsd">
    <default>
        <carriers>
            <simpleshipping>
                <active>1</active>
                <sallowspecific>0</sallowspecific>
                <model>Mageplaza\Simpleshipping\Model\Carrier\Shipping</model>
                <name>Mageplaza Sample Shipping Method</name>
                <price>10.00</price>
                <title>Mageplaza Sample Shipping Method</title>
                <specificerrmsg>This shipping method is not available. To use this shipping method,
please contact us.</specificerrmsg>
                <handling_type>F</handling_type>
            </simpleshipping>
        </carriers>
    </default>
</config>
```

The shipping method code should be child of `default > carriers`. And this code should be exactly same with `$_code` in `app/code/Mageplaza/Simpleshipping/Model/Carrier/Shipping.php:13`

Step 2: Create shipping model

Next, we need to create a model class for this **shipping method in Magento 2**

Create a Shipping.php at `app/code/Mageplaza/Simpleshipping/Model/Carrier/Shipping.php`

```
<?php
namespace Mageplaza\Simpleshipping\Model\Carrier;

use Magento\Quote\Model\Quote\Address\RateRequest;
use Magento\Shipping\Model\Rate\Result;

class Shipping extends \Magento\Shipping\Model\Carrier\AbstractCarrier implements
    \Magento\Shipping\Model\Carrier\CarrierInterface
{
    /**
     * @var string
     */
    protected $_code = 'simpleshipping';

    /**
     * @var \Magento\Shipping\Model\Rate\ResultFactory
     */
    protected $_rateResultFactory;

    /**
     * @var \Magento\Quote\Model\Quote\Address\RateResult\MethodFactory
     */
    protected $_rateMethodFactory;

    /**
     * Shipping constructor.
     *
     * @param \Magento\Framework\App\Config\ScopeConfigInterface          $scopeConfig
     * @param \Magento\Quote\Model\Quote\Address\RateResult\ErrorFactory  $rateErrorFactory
     * @param \Psr\Log\LoggerInterface                                    $logger
     * @param \Magento\Shipping\Model\Rate\ResultFactory                  $rateResultFactory
     * @param \Magento\Quote\Model\Quote\Address\RateResult\MethodFactory $rateMethodFactory
     * @param array                                                       $data
     */
    public function __construct(
        \Magento\Framework\App\Config\ScopeConfigInterface $scopeConfig,
        \Magento\Quote\Model\Quote\Address\RateResult\ErrorFactory $rateErrorFactory,
        \Psr\Log\LoggerInterface $logger,
        \Magento\Shipping\Model\Rate\ResultFactory $rateResultFactory,
        \Magento\Quote\Model\Quote\Address\RateResult\MethodFactory $rateMethodFactory,
        array $data = []
    ) {
        $this->_rateResultFactory = $rateResultFactory;
        $this->_rateMethodFactory = $rateMethodFactory;
        parent::__construct($scopeConfig, $rateErrorFactory, $logger, $data);
    }

    /**
     * get allowed methods
     * @return array
     */
    public function getAllowedMethods()
    {
        return [$this->_code => $this->getConfigData('name')];
    }

    /**
     * @return float
     */
    private function getShippingPrice()
    {
        $configPrice = $this->getConfigData('price');

        $shippingPrice = $this->getFinalPriceWithHandlingFee($configPrice);
        return $shippingPrice;
    }

    /**
     * @param RateRequest $request
     * @return bool|Result
     */
    public function collectRates(RateRequest $request)
    {
        if (!$this->getConfigFlag('active')) {
            return false;
        }

        /** @var \Magento\Shipping\Model\Rate\Result $result */
        $result = $this->_rateResultFactory->create();

        /** @var \Magento\Quote\Model\Quote\Address\RateResult\Method $method */
        $method = $this->_rateMethodFactory->create();

        $method->setCarrier($this->_code);
        $method->setCarrierTitle($this->getConfigData('title'));

        $method->setMethod($this->_code);
        $method->setMethodTitle($this->getConfigData('name'));

        $amount = $this->getShippingPrice();

        $method->setPrice($amount);
        $method->setCost($amount);

        $result->append($method);

        return $result;
    }
}
```

Step 3: Create configuration file

Now create system.xml file at `app/code/Mageplaza/Simpleshipping/etc/adminhtml/system.xml`

Related: How to Create a system.xml configuration for Magento 2

with the following content:

```
<?xml version="1.0"?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Config:etc/system_file.xsd">
    <system>
        <section id="carriers" translate="label" type="text" sortOrder="320" showInDefault="1"
showInWebsite="1" showInStore="1">
            <group id="simpleshipping" translate="label" type="text" sortOrder="0" showInDefault="1"
showInWebsite="1" showInStore="1">
                <label>Mageplaza Simple Shipping Method</label>
                <field id="active" translate="label" type="select" sortOrder="1" showInDefault="1"
showInWebsite="1" showInStore="0" canRestore="1">
                    <label>Enabled</label>
                    <source_model>Magento\Config\Model\Config\Source\Yesno</source_model>
                </field>
                <field id="name" translate="label" type="text" sortOrder="3" showInDefault="1"
showInWebsite="1" showInStore="1" canRestore="1">
                    <label>Method Name</label>
                </field>
                <field id="price" translate="label" type="text" sortOrder="5" showInDefault="1"
showInWebsite="1" showInStore="0" canRestore="1">
                    <label>Price</label>
                    <validate>validate-number validate-zero-or-greater</validate>
                </field>
                <field id="handling_type" translate="label" type="select" sortOrder="7"
showInDefault="1" showInWebsite="1" showInStore="0" canRestore="1">
                    <label>Calculate Handling Fee</label>
                    <source_model>Magento\Shipping\Model\Source\HandlingType</source_model>
                </field>
                <field id="handling_fee" translate="label" type="text" sortOrder="8"
showInDefault="1" showInWebsite="1" showInStore="0">
                    <label>Handling Fee</label>
                    <validate>validate-number validate-zero-or-greater</validate>
                </field>
                <field id="sort_order" translate="label" type="text" sortOrder="100"
showInDefault="1" showInWebsite="1" showInStore="0">
                    <label>Sort Order</label>
                </field>
                <field id="title" translate="label" type="text" sortOrder="2" showInDefault="1"
showInWebsite="1" showInStore="1" canRestore="1">
                    <label>Title</label>
                </field>
                <field id="sallowspecific" translate="label" type="select" sortOrder="90"
showInDefault="1" showInWebsite="1" showInStore="0" canRestore="1">
                    <label>Ship to Applicable Countries</label>
                    <frontend_class>shipping-applicable-country</frontend_class>

<source_model>Magento\Shipping\Model\Config\Source\Allspecificcountries</source_model>
                </field>
                <field id="specificcountry" translate="label" type="multiselect" sortOrder="91"
showInDefault="1" showInWebsite="1" showInStore="0">
                    <label>Ship to Specific Countries</label>
                    <source_model>Magento\Directory\Model\Config\Source\Country</source_model>
                    <can_be_empty>1</can_be_empty>
                </field>
                <field id="showmethod" translate="label" type="select" sortOrder="92"
showInDefault="1" showInWebsite="1" showInStore="0">
                    <label>Show Method if Not Applicable</label>
                    <source_model>Magento\Config\Model\Config\Source\Yesno</source_model>
                    <frontend_class>shipping-skip-hide</frontend_class>
                </field>
                <field id="specificerrmsg" translate="label" type="textarea" sortOrder="80"
showInDefault="1" showInWebsite="1" showInStore="1" canRestore="1">
                    <label>Displayed Error Message</label>
                </field>
            </group>
        </section>
    </system>
</config>
```

Step 4: Enable module

This is the final step, you need to run the upgrade command to enable Simple Shipping method.

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

Flush Magento cache if any.

Sample Shipping method module

Download Simple Shipping Module

Shipping Extensions for your Magento 2 Stores

Mageplaza knows what your store needs to be flexible with your delivery options. That is why we highly recommend these extensions for you to start with:

- Create rules for your shipping methods via Shipping Rules extension
- Configure flat rates according to countries with Multiple Shipping Flat Rates module
- Set limits for shipping application based on various conditions by Shipping Restrictions
- Display free delivery notice on Free Shipping Bar

The bottom line

Shipping methods can affect stores' costs and customers' purchase decisions due to the convenience, popularity, and required fees. That is why creating additional delivery methods is essential for all online merchants. This guide hopefully brings you useful instructions for your shipping method settings. Besides that, you may also be interested in Magento 2 Create Payment Methods to customize the payment methods as expected. Every question and comment are welcomed in the below section, or feel free to contact us for direct support.

Vinh Jacker

With over a decade of experience crafting innovative tech solutions for ecommerce businesses built on Magento, Jacker is the mastermind behind our secure and well-functioned extensions. With his expertise in building user-friendly interfaces and robust back-end systems, Mageplaza was able to deliver exceptional Magento solutions and services for over 122K+ customers around the world.

Share this post:

Related Post

JanDecTopImpWhTop
20252024 andare10+
PrinExploreWebShopify
2 5 01- 12-
mins mins | 30 | 30 on-OrdersStore

readread20252024

[demand Gym Clothing Providers in 2025]()

13 mins read

[...ia Ma... 2]()

6 mins read | 12-18-2024

[...id Manga 2]()

| 12-13-2024

[Name Generators: Ultimate Review Guide]()

15 mins read | 12-10-2024

19 mins read | 12-27-2024

Website Support & Maintenance Services

Make sure your store is not only in good shape but also thriving with a professional team yet at an affordable price.