

Unit- IV

Shopping tools, Promotions, Communications, SEO & Search, The sales process, Point of Purchase, Order Management, Payment methods, Shipping methods, Configuring sales tax rules, Outgoing e-mails, Modules, Shipping Modules.

Shopping Tools:

Shopping tools are software applications that assist in various aspects of online shopping. They streamline, organize, and automate the processes of running and growing an online store

Here are some examples:

1. **Ecommerce Platforms:** These are platforms where you can set up your online store. Examples include Shopify, WooCommerce, and Wix
2. **Inventory Management Software:** This tool helps you manage your stock and orders
3. **Analytics Tools:** These tools help you track sales, orders, and measure your performance
4. **Marketing Tools:** These tools help you let the world know about your store, drive traffic, convert leads into customers, and build your brand
5. **Customer Service Tools:** These tools help you manage customer support
6. **Content Creation Tools:** These tools help you create engaging content for your store
7. **Communication and Internal Organization Tools:** These tools help you manage the day-to-day operations of your store.

These tools are important because they help business owners manage their online stores efficiently, keep costs down, and earn more money. They also help businesses grow and keep pace with consumer demand.

For example, a tool like Shopify can help you set up an online store, manage inventory, track sales, and even handle shipping. Another tool like Google Analytics can help you understand your customers better by providing insights into how they interact with your website. shopping tools are essential for anyone looking to start or grow an online business.

Shopping tools application:

1. **Ecommerce Platforms:** These are platforms where you can create and manage your online store. Examples include Shopify, BigCommerce, WooCommerce, Squarespace, and Wix.com.
2. **Inventory & Order Management:** These tools help you manage your stock and orders. An example is the Shopify Fulfillment Network.
3. **Analytics and Reporting:** These tools help you track sales, orders, and measure your performance. Google Analytics and Kissmetrics are examples.
4. **Marketing Tools:** These tools help you reach more customers online and make an exciting shopping experience. Examples include Adroll and AdEspresso.
5. **Customer Service Tools:** These tools help you manage customer support. Examples include Zendesk and LiveChat.
6. **Communication and Internal Organization:** These tools help in internal communication and organization. Examples include Asana and Trello.
7. **Content Creation Tools:** These tools help in creating content for your online store. Examples include Brandwatch Social Media Management and Hootsuite.

Remember, the right tools for your business will depend on your specific needs and goals. It's always a good idea to research each tool and consider its features, pricing, and reviews before making a decision.

Example: Magento2 tool

Magento2, now known as Magento Commerce and Magento Open Source, is a powerful E-Commerce platform that businesses use to create and manage their online stores. Here are some key features and examples:

1. **Wider Outreach:** Magento2 supports multiple languages and currencies, allowing businesses to reach a global audience
2. **Customer-friendly:** Magento2 offers many default themes and third-party apps for customization. This allows businesses to create well-structured and easy-to-navigate websites.
3. **Search Engine Optimization (SEO):** Magento2 has excellent SEO features that help businesses rank higher in search engine results
4. **Security:** Compared to other platforms, Magento2 is highly secure, making it a safe choice for E-Commerce websites
5. **Integration:** Magento2 can be integrated with other technologies

As for examples, many renowned brands have built their E-Commerce websites using Magento2. These websites stand out due to their load speed, structure, navigation, user experience (UX), and mobile responsiveness. However, the specific names of these websites are not provided in the search results

Promotions:

Promotions in Magento2 are a great way to increase customer awareness about your site and boost your conversion rate. There are two types of promotions you can configure in Magento2

1. **Catalog Price Rule:** This rule allows you to lower the product cost for all items on your Magento website. Once you set a rule, it adjusts product prices according to the special promotion. You can use it to set a special price for a single product, a group of products, or all products on your website. You can also schedule the start and end dates of the rule so that it automatically expires after a set time.
2. **Cart Price Rule:** This rule applies discounts to items in the shopping cart, based on a set of conditions. It's a great way to entice customers at the checkout.

This feature allows you to develop a well-planned and successful email marketing campaign. You can configure subscription options, create eye-catching newsletter templates, and manage your subscribers.

Here's a simple example of how to set up a Catalog Price Rule:

1. Navigate to the Admin Panel.
2. Expand the Marketing section, and find the catalog price rule tab.
3. Add a New Rule.
4. Set rule conditions for the new catalog price rule.
5. Define the Actions.
6. Save and test the new rule.

Remember, for a product attribute to appear in the drop-down configurable conditions, it must be previously configured to be used in promo rule conditions

Example: Promotions in Magento2

Magento promotions are an effective way to increase customer awareness and boost the conversion rate of your online store. Here are two main types of promotions in Magento:

Catalog Price Rule: This rule allows you to lower the product cost for all items on your Magento website. Once you set a rule, it adjusts product prices according to the special promotion. Here's how you can set new catalog price rules:

- Navigate to the Admin Panel.
- Expand the Marketing section, and find the catalog price rule tab.
- Add a New Rule.
- Set rule conditions for the new catalog price rule.
- Define the Actions.

- Save and test the new rule.

You can use **Magento Catalog price rules** to:

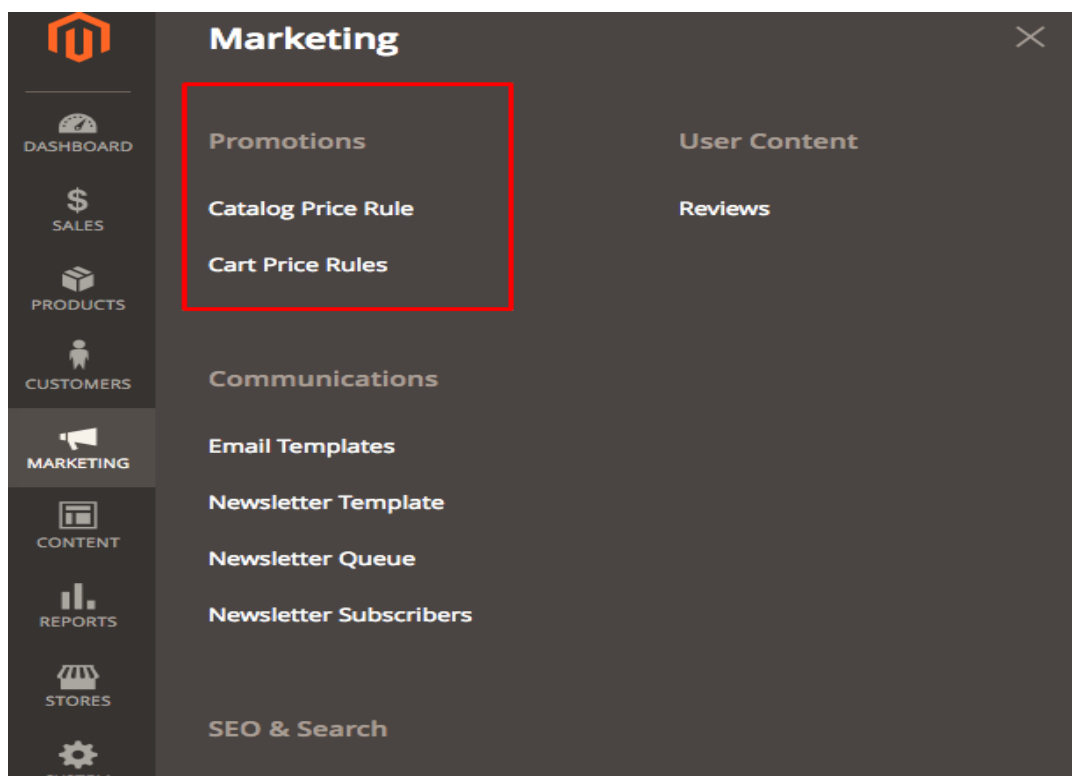
- Lay down a special price for a single product, a group of products, or all products on your website.
- Offer a fixed discount from the original price or apply percentage-based discounts on products purchased.
- Schedule the start and end dates of the rule so that it automatically expires after a set time.

Cart Price Rule: This rule applies discounts to items in the shopping cart, based on a set of conditions. Unlike the Catalog Price Rule, coupons can be used with Cart Price Rule. Remember, applying conditions to the promotions generated with catalog price rules effectively motivates customers to take desired actions. Once set, conditional rules ensure that customers can only get discounts once they meet the criteria.

Magento2 Promotions, you will be able to increase your sales and attract more customers by creating coupons for discounts on special products, scheduling a specific promotion for a period of time, or even allowing customers to receive Y items when purchasing X item.

In order to help you set up your own pricing rule, Magento2 gives you two types of Promotions programs:

- **Catalog Price Rule:** selectively offer products at a discounted price, based on a set of conditions
- **Cart Price Rule:** apply discounts to items in the shopping cart, based on a set of conditions



Catalog Price Rule:

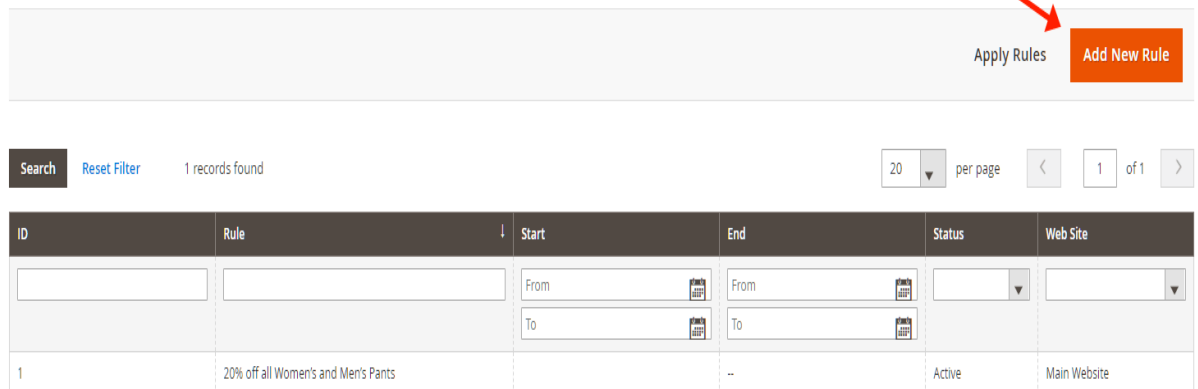
As the Catalog Price Rule applies to selected products, your customers will see the discounted price for each product on the product details pages. Coupons can not be used with Catalog Price Rule because this type of rule is applied before a product is placed into the shopping cart.

With the Catalog Price Rule, you can use it to:

- Set special fixed price for a single product/ a group of products, or all products
- Discount a fixed amount from the original price, or adjust to a percentage for a single product/ a group of products, or all products
- Schedule started date and ended date of the rule

To begin configuring your Catalog Price Rule, from your Admin dashboard select **Catalog Price Rule > Add New Rule**

Catalog Price Rule



The screenshot shows the 'Catalog Price Rule' admin interface. At the top right, there is a search icon, a notification bell with '26', and a user profile 'admin'. Below these are two buttons: 'Apply Rules' and 'Add New Rule'. A red arrow points to the 'Add New Rule' button. Below the buttons is a table with the following columns: ID, Rule, Start, End, Status, and Web Site. The table contains one record with ID 1, Rule '20% off all Women's and Men's Pants', Start and End dates set to 'From' and 'To' respectively, Status 'Active', and Web Site 'Main Website'. Above the table, there is a search bar, a 'Reset Filter' button, and a message '1 records found'. To the right of the table, there is a pagination control showing '20 per page' and '1 of 1'.

ID	Rule	Start	End	Status	Web Site
1	20% off all Women's and Men's Pants	From	From	Active	Main Website

Rule information

All the fields in this group are pretty much self-explanatory so you should be able to configure with ease.

Conditions

This is where it gets rough. Catalog price rules are based on condition combinations. You can set condition combinations by clicking on the corresponding bold words.

Conditions (don't add conditions if rule is applied to all products)

If **ALL** of these conditions are **TRUE** :

Features contains

Material contains Leather

For example, we have a master condition like so:

If **ALL** of these conditions are **TRUE**

And more subsequent rules following this condition, like so:

If **ALL** of these conditions are **TRUE**

Features **contains Lightweight**

Material **contains Leather**

These condition combinations will apply for all products that have Lightweight as its feature, and Leather as its material. Alternatively, to exclude Lightweight & Leather products from inheriting this catalog price rule, set:

if **ALL** of these conditions are **TRUE**


to



if **ALL** of these conditions are **FALSE**

Notes: For a product attribute to appear in the drop-down configurable conditions, it must be previously configured to be used in promo rule conditions.

With the above condition combinations, we can now set *Actions*.

Actions

Actions 

Apply	<input type="text" value="Apply as percentage of original"/>	
Discount Amount *	<input type="text" value="20"/>	
Discard subsequent rules	<input type="text" value="No"/>	

These are four actions for your catalog price rules:

- **Apply as percentage of original:** Add a percentage-based discount amount to the original price. A 20% discount, for example, can be applied by typing 20 to the discount amount
- **Apply as fixed amount:** Add a fixed amount of discount to the original price. For example, to set a \$50 discount (assuming that the configured currency is in USD), set the discount amount to 50
- **Adjust final price to this percentage:** Set the final price to a percentage of its original price. For example, if the product's original price is \$100 and you set the discount amount to 20 (20%), the product's price will be adjusted to 20.
- **Adjust final price to discount value:** Set the final price to a discount value. For example, if the product's original price is \$100 and you set the discount amount to 15, the product's price after discount will be adjusted to 15

Discard subsequent rules will prioritize this rule and disregard all the other rules from being processed. This will come in handy when you want to prevent multiple catalog price rules to apply on the same product. and that's it. Click on **Save** and see for yourself if the discount has been applied.

Cart Price Rule

Cart Price Rule takes effect on the items in the shopping cart and can be applied automatically as soon as the conditions are met, or when the customer enters a valid coupon code. When applied, the discount appears in the cart under the subtotal.



With Cart Price Rule in Magento2, you can create special discounts like **Buy X get Y free** (customers need to purchase X items to receive Y (quantity) items for free). Also, rules in Cart Price Rule can also apply to **Shipping** by allowing it to **Apply to Shipping Amount** or you can even apply **Free Shipping** for qualified carts.

Similarly to catalog price rule, you can set cart price rules with sets of condition combinations. Condition combinations in Cart Price Rules are more varied as you now get cart attributes and more product sub-selections.

To show you how to properly configure a cart price rule, we'll be creating a new cart price rule:

Rule information

All the fields in this section should be self-explanatory to you. In addition to all the basic fields like Rule Name, Description, you can choose to set Coupon Code and Uses per Coupon, as well as Uses per Customer in this section.

Coupon *	Specific Coupon ▼
Coupon Code *	LSDAA
	<input type="checkbox"/> Use Auto Generation
	<small>If you select and save the rule you will be able to generate multiple coupon codes.</small>
Uses per Coupon	1
Uses per Customer	1
	<small>Usage limit enforced for logged in customers only.</small>
From	01/13/2021 
To	01/27/2021 
Priority	1
Public In RSS Feed	<input checked="" type="radio"/> Yes

Configure coupon-related information in Rule Information

Conditions


Set the condition combinations that your Actions will be applying to. Similar to configuring conditions in Catalog Price Rule, you can configure conditions by clicking on bold words.

For example, here we have a condition for purchases over \$200.

Conditions

Apply the rule only if the following conditions are met (leave blank for all products).

If **ALL** of these conditions are **TRUE** :

Subtotal equals or greater than 200 



Actions

Set the discounts that you want your previously configured Conditions to take. Typically there should be four selectable action for you to take:

- **Percent of product price discount:** Add a percentage-based discount to the original price. For example, to set a 30% discount, enter 30 to Discount Amount.
- **Fixed amount discount:** Add a fixed amount of discount to the original price of each qualifying item in the cart. Note that the *currency* of the discount amount is based on your Magento configuration.
- **Fixed amount discount for the whole cart:** Add a fixed amount of discount to the price of the whole cart.
- **Buy X get Y free:** Set a X amount of products that customers must purchase to receive a Y number of products for free.

Labels

Set a label for the discount. The label is displayed in the Total section, enclosed in parentheses next to the word Discount. Labels can be created for all Store views or be made to be store-view-specific.

Subtotal	\$112.00
Shipping & Handling	\$0.00
Discount (<u>Special</u>)	-\$15.00
Tax	\$8.00
Grand Total	\$105.00

Special is a label

And that's it for a basic Cart Price Rule. Click on **Save** and test for yourself if the new cart price rule works.

Communications:

Magento2 offers several types of communication methods. Here are some of them:

1. **Product Types:** Magento2 supports various product types, each serving a different purpose . Here are the six main types:
 - **Simple Product:** These are sold as single items with no variations. Each product has its own SKU
 - **Configurable Product:** This type includes many simple products with different options of colors and sizes
 - **Virtual Product:** These are services, memberships, warranties, or subscriptions
 - **Grouped Product:** A group of Simple products or Virtual Product which shares the same characteristics or are related to each other
 - **Bundle Product:** These are customized products which include various Simple products or Virtual Product
 - **Downloadable Product:** These are products that can be downloaded, such as software, files, ebooks, videos, etc
2. **Message Queues:** Magento provides the Message Queue Framework (MQF) for publishing messages to queues and creating consumers to receive them asynchronously. MQF supports MySql and RabbitMQ queue systems.

3. **REST and SOAP APIs:** Magento2 supports REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) for communication. These APIs allow for data exchange between different systems in a reliable manner.
4. **HTTP Verbs:** Magento API calls incorporate HTTP verbs (GET, PUT, POST, DELETE) which are actions executed against an endpoint⁵.

Each of these communication types serves a different purpose and can be used in different scenarios. For example, product types are used to manage and categorize products in your store, message queues are used for asynchronous communication between different parts of your application, and APIs are used for data exchange between your Magento store and other systems.

SEO & Search:

Sure, SEO (Search Engine Optimization) is a crucial aspect of any E-commerce platform, including Magento2. It helps increase the quality and quantity of traffic to engage more customers' interest in stores' products. Here are some key SEO settings and features in Magento2:

1. **Magento Homepage SEO Setting:** Modify the default "Home Page" to become meaningful, such as the name of your store
2. **Magento Metadata SEO optimization:** This includes Meta Title, Meta Description, and Meta Keywords for Categories and Product settings
3. **Setting Canonical tags for SEO:** These help prevent duplicate content issues
4. **Setting Robots.txt for SEO:** This file tells search engine crawlers which pages or files the crawler can or can't request from your site
5. **Add Rich Snippets for product pages:** Rich snippets provide more information about your product directly in search results
6. **SEO XML sitemap generation:** An XML sitemap helps search engines better understand your website while crawling it
7. **Alt tags for SEO images:** Alt tags provide better image context/descriptions to search engine crawlers and help improve your SEO ranking
8. **URL Rewrite and Redirect:** URL rewriting can be used to convert complex URLs into simple, easy-to-read URLs, improving user experience and SEO
9. **Suffix for product and category URLs:** Clean, concise URLs can improve search engine optimization
10. **Advanced Magento2 SEO setting with SEO All-in-one:** This includes enabling the SEO module, removing duplicate content issues, SEO structured data configuration, business information and social profiles for local SEO, site verification for search

engine optimization, SEO tool XML and HTML sitemap integration, SEO redirect, SEO metadata template rules, SEO analysis, SEO crosslinks, SEO checklist, SEO report, SEO hreflang tag, SEO image optimizer

Remember, an SEO-friendly website improves the overall shopping experience and guarantees efficient website visibility. For example, when Internet users type “Magento2 SEO extension” on the search bar, they will get a search result that is more attractive and informative with Rating, Price, Stock status, and quick links included.

The sales process:

In Magento2, the sales processing involves managing orders, invoices, shipments, and other related tasks. Let’s break it down:

1. **Orders:** When a customer places an order on your Magento store, it gets recorded as an order. Orders include details such as products purchased, customer information, shipping address, and payment method. You can view and manage orders in the Magento admin panel.
2. **Invoices:** After an order is placed, you can create an invoice. Invoices are generated when you want to bill the customer for the purchased items. They include details like product quantities, prices, taxes, and shipping charges. Invoices are essential for accounting purposes.
3. **Shipments:** When you’re ready to ship the ordered items to the customer, you create a shipment. Shipments include information about the products being shipped, shipping carriers, tracking numbers, and delivery addresses. You can also print packing slips from the shipment details.
4. **Credit Memos:** Sometimes, you need to issue refunds or credit to customers. Credit memos allow you to handle returns, cancellations, or adjustments. They include details about refunded amounts, restocking fees, and reasons for the credit.
5. **Sales Reports:** Magento provides various sales reports to help you analyze your business performance. These reports cover aspects like order totals, tax summaries, shipping details, and more. You can access these reports in the Magento admin panel.

Example:

Suppose a customer named Alice visits your online store and places an order for two T-shirts. Here’s how the sales process would work:

1. Order Creation:

- Alice selects two T-shirts and adds them to her cart.
- She proceeds to checkout, enters her shipping address, and selects a payment method (e.g., credit card).

- Upon completing the order, Magento records it as an order in the system.

2. Invoice Generation:

- You review the order and decide to bill Alice immediately.
- You create an invoice for the order, which includes the T-shirt details, prices, taxes, and shipping charges.
- The invoice serves as a formal billing document.

3. Shipment Creation:

- After packing the T-shirts, you create a shipment.
- The shipment includes the product details, shipping carrier (e.g., FedEx), and tracking number.
- You mark the order as shipped.

4. Credit Memo (Refund):

- Later, Alice contacts you and requests a refund for one of the T-shirts.
- You create a credit memo, specifying the refunded amount and reason (e.g., size issue).
- The credit memo adjusts the order total and updates the inventory.

5. Sales Reports:

- In the Magento admin, you can generate sales reports to analyze trends, track revenue, and monitor inventory levels.
- These reports help you make informed business decisions.

Remember that this is a simplified example, but it demonstrates the essential steps involved in sales processing within Magento2. The actual process may vary based on your specific business requirements and configurations.

Point of Purchase:

In Magento2, the term “Point of Purchase” is often referred to as the Point of Sale (POS) system. The POS system is an organization’s method of maintaining inventory and data concerning goods or services. It’s a management tool that brings many benefits, including increased efficiency, reduced errors, better control over volume purchase items/units/quantities, and monitoring data across the business environment.

Here's an example of how a Magento2 POS system works:

1. **Keep Track of Inventory:** The Magento POS system helps retailers manage inventory wherever they have products. It makes tracking more manageable and order fulfillment faster when you know how many products are available in a particular location at any given time. It can automatically alert you when the inventory is low.
2. **Manage Customer and Sales Data:** This feature captures details and gives you a better understanding of your loyal shoppers. Businesses can collect contact information to build in-depth customer profiles to know more about customer shopping habits. With quick access to order history, it can support you cross-sell and up-sell efficiently by making unique, relevant offers based on purchase records

Example: Magento2 points of sale could show you how much they spent or how many times they visited your store or site, depending on your product or service. You can also produce any type of report that you need for strategic planning purposes.

Order Management:

Order management in Magento2 is a comprehensive process that involves managing, selling, and fulfilling inventory from any sales channel.

Here are some key components of the Magento2 Order Management System (OMS)

1. **Distributed Order Management:** Determines the most efficient methods and channels for fulfilling an order to best meet customer expectations.
2. **Sourcing:** Provides a bird's-eye view of global inventory, allowing merchants to define and optimize how products are sourced from all available inventory.
3. **Store Fulfillment:** Allows merchants to turn their stores into mini-distribution centers for easy and adaptive sourcing of multi-store inventory and online selling.
4. **Customer Service:** Enables easier order modifications and integration with other essential customer service tools.
5. **Mobile Point-of-Sale (POS):** Enhances the in-store experience and ensures a customer never leaves the store without the desired product.
6. **Business Intelligence:** Leverages order management analytics to gain valuable insight on orders, shipments, returns, and payments¹.

Every order in Magento2 goes through several stages before it is marked as "Complete". These stages are represented by different order statuses and states

Example: Here are some predefined order statuses in Magento2

- **Pending:** No shipments or invoices have been processed.
- **Processing:** Payment has been received or authorized. The order has been invoiced or shipped, but not both.
- **On Hold:** This order status is assigned manually when you need to hold it.
- **Complete:** Order is created, paid, and shipped.
- **Closed:** A credit memo was created for the order and the order was refunded.
- **Canceled:** Order is canceled by Magento admin user or payment gateway if a customer doesn't pay in a specified period of time.
- **Pending Payment:** Orders get this status if you use the payment gateway systems like PayPal. The status changes when a customer pays. You can also create custom order statuses and assign them to states.

Payment methods:

Magento2 supports a variety of payment methods out of the box, such as

- Check/Money Order
- Cash on Delivery
- Bank Transfer
- Zero Subtotal Checkout
- Purchase Order

You can also create your own custom payment methods

Here's a basic example of how to create a custom payment method in Magento2

1. **Create a new module:** You'll need to create a new module for your custom payment method. This involves creating a `registration.php` file and declaring your module in a `module.xml` file
2. **Add configuration settings:** In your new module, you'll need to add configuration settings for your payment method. This includes settings like whether the payment method is enabled, the title of the payment method, and which credit card types are accepted

3. **Implement the payment method:** After setting up the module and configuration, you'll need to implement the payment method. This involves creating a model for the payment method, defining the payment action, and implementing the necessary API logic
4. **Display the payment method in the checkout page:** Finally, you'll need to ensure that your new payment method is displayed on the checkout page. This involves creating a layout file and a template file for the payment method

Remember, this is a simplified overview. For a more detailed guide, you can refer to the resources provided. Please note that creating a custom payment method requires a good understanding of Magento2's architecture and PHP programming. If you're not comfortable with this, you might want to consider hiring a Magento developer or using a pre-built extension that meets your needs. Note that the above steps are for illustrative purposes and may not cover all the complexities involved in creating a custom payment method in Magento2. Always refer to the official Magento2 documentation and follow best practices when developing for Magento2.

Shipping methods:

Magento2 provides several shipping methods out of the box.

Here's a brief explanation of each:

1. **Flat Rate Shipping Method:** This is a standard Magento2 shipping method that can be applied per single item or per order regardless of product attributes such as size, weight, etc
2. **Free Shipping Method:** Magento2 allows store owners to provide free shipping based on specific conditions, such as order value, customer segments, minimum purchase, etc
3. **Table Rates Shipping Method:** The shipping costs in Table Rate are determined based on destination, weight, and order value. This option provides multiple shipping rules for distinct product sets or customer categories
4. **In-Store Pickup:** This method allows customers to buy products online and pick them up in a physical store or specific location

To configure these shipping methods, go to Stores > Configuration > Sales > Shipping Methods in your Magento2 admin panel.

Here's an example of how to configure the Flat Rate Shipping Method:

- Enable the method by setting Enable to Yes.
- Set the Title and Method Name.

- Define the way the shipping fee is added to the subtotal in the Type field. There are 3 possible options:
 - **None:** No change. In this case, the Flat rate shipping method is equivalent to Free.
 - **Per Order:** The charge is generated for the whole order and not every single item separately.
 - **Per Item:** The Flat Rate shipping fee is set for each item and multiplied by the number of items in the cart.

Remember, before configuring the shipping methods, you need to set up the default shipping settings in **Stores > Configuration > Sales > Shipping Settings**. This includes setting up the Origin (**Country, Region/State, ZIP/Postal Code, City, Street Address**) and Shipping Policy Parameters:

Configuring Sales tax rules:

Sure, here's how you can configure sales tax rules in Magento2:

1. **Provide Tax Rule Information:** Access the Admin panel and navigate to **Stores > Taxes > Tax Rules**. In the upper-right corner, click **Add New Tax Rule**. Enter the name of your new rule under **Tax Rule Information**. Next, select the applicable **Tax Rate** for this rule.
2. **Input Tax Rates:** You can choose between two methods to input tax rates
 - **Manual Entry:** Click **Add New Tax Rate**. Fill out the required form. Once the form is complete, click **Save**.
 - **Import Tax Rates:** Prepare a CSV file with your tax rates to speed up the process. Scroll down to the bottom of the **Add New Tax Rules** page. Click **Choose File** and select the CSV file containing the tax rates for import.

Here's an example of how to set up a tax rule:

Step 1: Go to **Stores > Taxes > Tax Rules** and press the **Add New Tax Rule** button.

Step 2: Set the **Tax Rule Information**. Give the tax rule a **Name**. Choose the **Tax Rate** from the list of available rates or **Add New Tax Rate** manually

Note: Remember, before setting up tax rules, you need to be well-aware of the tax requirements of your country. Also, Magento2 taxes are auto-calculated based on tax rules. Tax rules, on the other hand, depend on customer, product tax classes as well as customers' shipping and billing addresses, and store location

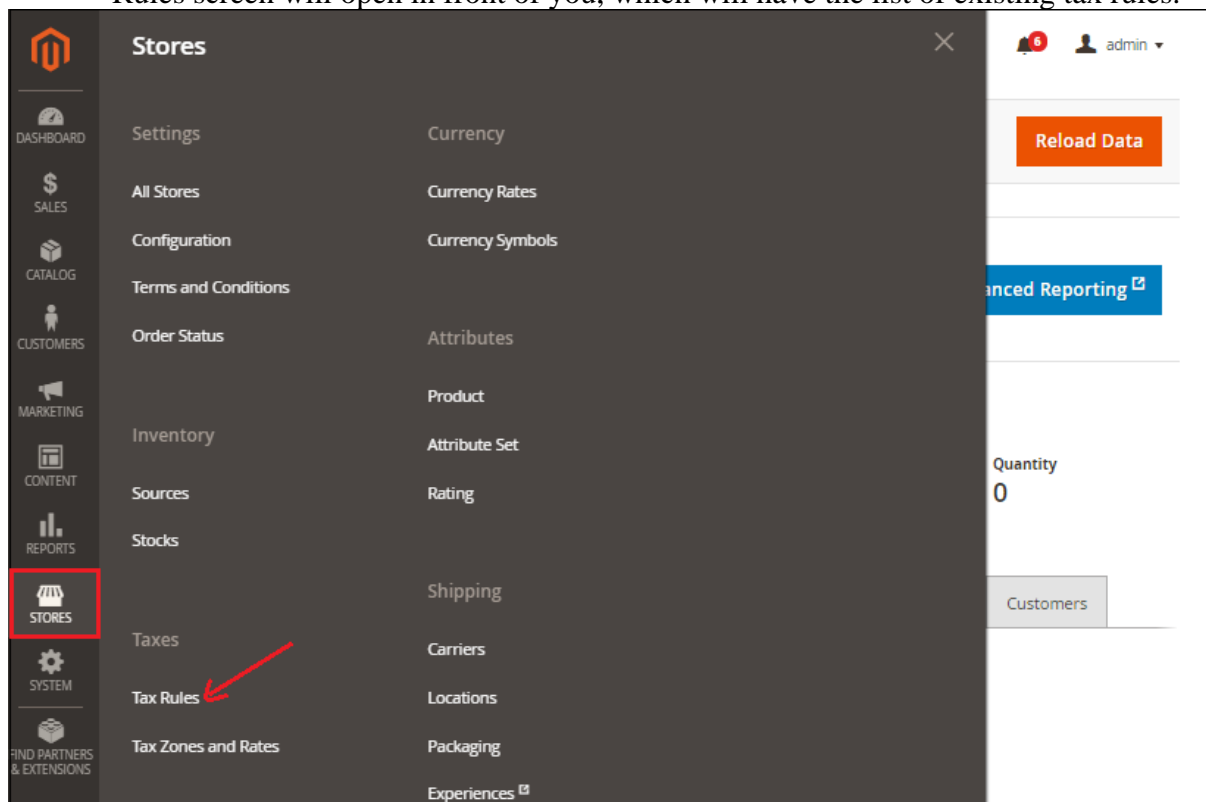
Example: Configuring sales tax rules:

The Tax Rule is a combination of **customer tax class**, **product tax class**, and tax rate. Each customer is assigned to a customer class, and each product is assigned to a product class. Magento analyzes each customer's shopping cart and calculates the tax according to the product, customer classes, and region.

To create the tax rule, follow the below steps:

Step 1: Login to the Magento2 Admin Panel.

Step 2: On the left sidebar of the Admin panel, Go to STORE >> Taxes >> Tax Rules. A Tax Rules screen will open in front of you, which will have the list of existing tax rules.



Step 3: In order to add a new tax rule, we need to click on the **Add New Tax Rule** button on at the top-right corner.

Tax Rules

0 records found

20
per page

1
of 1

Name	Customer Tax Class	Product Tax Class	Tax Rate	Priority	Subtotal Only	Sort Order
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

We couldn't find any records.

Step 4: Fill the required information in the given fields.

1. Complete the Tax Rule Information

New Tax Rule

Tax Rule Information

Name *

Tax Rate *

1 selected

☒ US-CA-*.Rate 1
☐ US-NY-*.Rate 1

Name: Provide the name of tax rule in the **Name** field, which is recommended to be named as an easy remembrance of what the rule contains.

Tax Rates: The Tax Rates field contains the collection of the tax rates, which are existing on our Magento2 store. We can select one or more tax rates according to our needs.

We can edit or delete the existing tax rate by clicking on the **Edit** or **Delete** icon corresponding to the tax rate. We can also define new tax rates by clicking on **Add new Tax Rate** button.

Tax Rate *

0 selected

- US-CA-*-Rate 1
- US-NY-*-Rate 1

Add New Tax Rate

2. Complete the Additional Settings

Expand the Additional Settings section.

▼ **Additional Settings**

Customer Tax Class *

Retail Customer

✓ Whole Sale

Add New Tax Class

Product Tax Class *

✓ Taxable Goods

Refund Adjustments

Gift Options

Order Gift Wrapping

Item Gift Wrapping

Printed Gift Card

Reward Points

Add New Tax Class

Priority *

0

Tax rates at the same priority are added, others are compounded.

Calculate Off Subtotal Only ☐

Sort Order *



0

Customer Tax Class - Select one or more customer tax class if you want to apply the tax rule for multiple customers. We can edit an existing customer tax class by clicking on the **Edit** icon and even create a new tax class from the **Add New Tax Class** button.

- Enter the name of the new Tax Class and click on the Right sign to save.

Customer Tax Class *

☒ Whole Sale

Here you can see how the tax rules are applying for the customer group. Go to **CUSTOMERS >> Customer Groups**.

DASHBOARD

SALES

CATALOG

CUSTOMERS

MARKETING

CONTENT

REPORTS

STORES

SYSTEM

FIND PARTNERS & EXTENSIONS

Customer Groups

4 records found

20
per page

1
of 1

ID	Group	Tax Class	Action
0	NOT LOGGED IN	Retail Customer	Edit
1	General	Retail Customer	Select
2	Wholesale	Retail Customer	Select
3	Retailer	Retail Customer	Select

Copyright © 2019 Magento Commerce Inc. All rights reserved.

Magento ver. 2.3.2
[Report an Issue](#)

Product Tax Class - It is the tax class that is associated with the product. Select one or more product tax classes to which the rule applies with the **Ctrl** key. We can also modify an existing product tax class by clicking on the **Edit icon** and even create a new tax class from the **Add New Tax Class** button.

Priority: Set the priority to determine this tax to be applied first when more than one tax is enabled. This will prioritize your tax rules with other existing tax rules anyhow shares the same conditions.

Calculate off Subtotal Only: Mark this field to calculate the tax according to the order subtotal. This means if you want to apply the tax on the subtotal, then mark this field.

Sort Order: It will determine the sequence in which we need to apply tax rules. In this field, enter a number to indicate the order of this tax rule when listed with others.

For completing the structure of tax rule, we have the import and export the tax rates. We can easily import or export our files from anywhere.

Import Tax Rates
No file chosen

Export Tax Rates

Note: Select the Import Tax Rates if you have several tax rates that need to set up but do not want to edit them manually.

Step 4: Click on **Save Rule** when completed. We can see the newly made tax rules on the main screen.

Tax Rules						
						<input type="button" value="Add New Tax Rule"/>
<div> <input type="button" value="Search"/> <input type="button" value="Reset Filter"/> 1 records found <div> 20 <div></div> per page <div><</div> <div>1</div> <div>of 1</div> <div>></div> </div> </div>						
Name	Customer Tax Class	Product Tax Class	Tax Rate	Priority	Subtotal Only	Sort Order
France R1	Whole Sale	Taxable Goods	US-CA-*.Rate 1	0	0	0

For additional settings regarding the Taxes calculation of your store, you need to visit the **STORES >> Configuration** section at backend and then access the **Sales >> Tax**.

Outgoing E- Mails:

In Magento2, outgoing emails are used for various purposes such as order confirmations, payment success or failure notifications, account creation confirmations, password resets, invoice creations, and more. These emails are sent through a protocol called Simple Mail Transfer Protocol (SMTP)

To set up outgoing SMTP email for Magento2, you can follow these steps.

1. Download the Custom SMTP for Magento2 extension.
2. Log into your Magento2 Admin
3. Go to Stores > Configuration > MagePal > SMTP Configuration
4. Configure your SMTP settings

SMTP allows your Magento2 store to send email through a specific third-party mail SMTP server. You can configure your Magento store to use your Gmail, Amazon, Microsoft, or any other mail server account to send all your emails. All you need is to configure the mail server settings in the Magento extension.

Remember, the way you present your email will be one of the evaluation criteria from the customer. They may decide whether or not to continue choosing your brand, partly depending on how expert your email is. So, it's essential to carefully set up emails on the Magento2 admin page.

Modules:

Magento2 is widely known for its extensibility. One of the extensibility concepts which Magento2 Open Source supports is called Modularity. **A Module in Magento2 is an independent component or a set of components that provide business logic and features.** Magento2 Open Source is a modular application and it supports modularity. **It means that all functionality is implemented and delivered in components that are known as Modules.** A Component is also known and called an Extension. **A module and an extension are the same things.**

Each extension or module in Magento2 provides a set of capabilities that support the business and functional logic of a module. It allows bringing additional features to the E-Commerce store which is based on Magento2.

Example:

Let's take a **Catalog module for example**. The Catalog module provides a different set of features which include **Category Management, Product Management, Product Types support for Simple and Virtual products, Catalog browsing capability**, and so on.

Also, the module provides extensibility points for other modules which allows extending the Catalog module with additional functionality. An example of such an additional module is a Downloadable Product module. A module in Magento2 could bring a complex set of features on one side and on the other side, it could be a tiny feature that adds an icon to a Product page for example.

A Module Name

Each module's name consists of two parts. A vendor name and the name of a module. These two parts are concatenated with a _(underscore) or a \ (backslash). As a result, the module name should follow the pattern VendorName_ModuleName.

For example: a Catalog module name is going to be the name of a Magento vendor name, and a Catalog module name, which is a Magento_Catalog or Magento\Catalog.

Example: Where should a Module be located in Magento2?

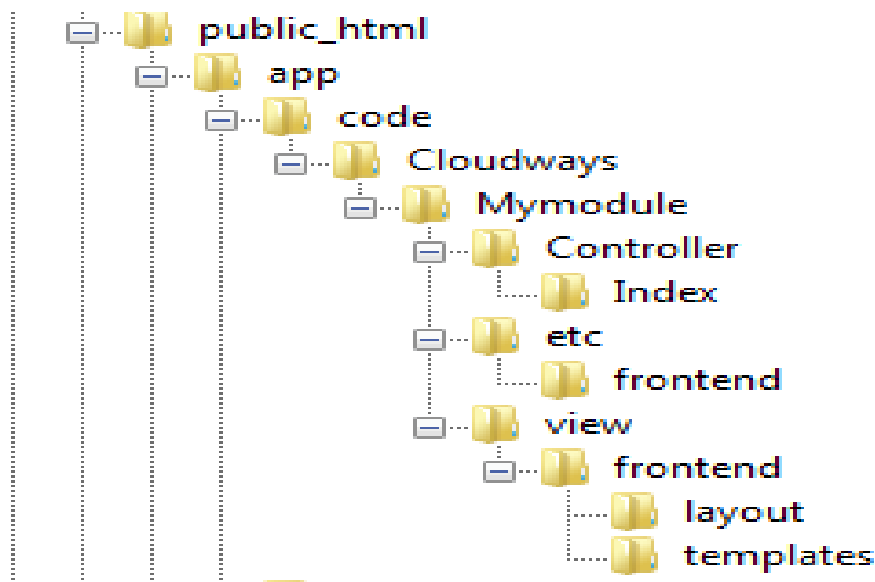
That's a very interesting question. There are two locations where a module can be located in a Magento2 application.

The first location is an app/code directory: This directory is used to add all custom and 3rd-party Magento2 modules. This location is usually used by development agencies, and internal or in-house developers to simplify a development process.

The second location is a vendor directory: This directory is used during the composer package management installation. A module installation in Magento2 with a composer package manager is the recommended way of installing a Magento2 module.

Example:

To create a Magento2 module, first, we need to set up the module's structure by creating folders.



- **Cloudways:** namespace name.
- **Mymodule:** module name.
- **Block:** it's a foundational building unit for layouts in Magento. It links a PHP block class (which contains logic) and a template (which renders content).
- **Model:** for Models and ResourceModels
- **Observer:** usually used when such an event is fired, the observer instantiates a Model to handle the necessary business logic for such an event.
- **Controller:** controls the flow of application execution.

- **Helper:** helper classes hold the code used in more than one application layer. For example, in the Cms module, helper classes are responsible for preparing HTML for presentation to the browser.
- **Setup:** contains the migration and upgradation classes for schema and data creation of the database.
- **Index:** controller name of the module and contains the action file.
- **etc:** contains the configuration file of the module.
- **etc/Frontend:** contains the router file.
- **view/Frontend:** Contains Layout and Templates Folder.
- **view/Frontend/Layout:** contain XML file.
- **view/Frontend/Templates:** contain .phtml file.
- **Composer.json:** used for the updating of product editions like the module. This file must be placed in the root directory of the module.

Below you'll find the structure of the vendor/magento core module responsible for handling the Magento core functionality. Note that how Magento2 module core code dependencies are inside the **composer** vendor folder.



Example: Creating a module in Magento2 involves several steps.

Here's a step-by-step guide:

1. **Create the Module Directory:** The first step is to create a directory for your module. The name of the module is defined as VendorName_ModuleName. For example, if your vendor name is MyVendor and your module name is MyModule, you would create the following directory: app/code/MyVendor/MyModule.
2. **Declare the Module:** Next, you need to declare your module by creating an etc/module.xml file in your module directory. This file should contain the following content.

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="MyVendor_MyModule" setup_version="1.0.0">
    </module>
</config>
```

AI-generated code. Review and use carefully.

3. **Register the Module:** After declaring your module, you need to register it by creating a registration.php file in your module directory. This file should contain the following content¹

PHP

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

AI-generated code. Review and use carefully.

4. **Enable the Module:** Finally, you need to enable your module. You can do this by running the following command in your Magento 2 root directory¹

```
php bin/magento module:enable MyVendor_MyModule
```

After running this command, your module should be enabled and ready to use. You can check the status of your module by running the following command

```
php bin/magento module:status
```

This is a basic guide to creating a module in Magento2. Depending on your specific needs, you may need to add additional files and configurations to your module, such as controllers, models, views, etc

Shipping Modules:

Magento2 (Adobe Commerce), the popular E-commerce platform, offers a range of shipping options..Table Rate Shipping Method and Flat Rate Shipping Methods are referred to as Magento2 modules because they are integral components of the Magento2 E-Commerce platform that provide specific functionalities

1. **Flat Rate Shipping Method:** This method allows merchants to apply a consistent rate for every order, regardless of their product attributes such as size or weight. It's particularly suitable for small businesses with a limited range of items
2. **Table Rate Shipping Method:** In contrast to the Flat Rate, the shipping costs in Table Rate are determined based on destination, weight, and order value. This option provides multiple shipping rules for distinct product sets or customer categories, giving retailers more flexibility. It's suitable for medium Magento stores.

These modules are part of the default shipping methods provided by Magento2, which also include Free Shipping and In-store Pickup. They help store owners to offer a variety of shipping options to their customers, enhancing the shopping experience and potentially influencing purchasing decisions. Therefore, they are called Magento2 modules due to their specific roles and functionalities within the Magento2 system.

Benefits of Shipping Modules:

1. **Modularity:** Modules are integral to Magento's architecture, enabling developers to add, remove, or modify functionality without affecting the core system¹². This makes Magento2 highly adaptable and customizable for e-commerce businesses.
2. **Customization:** By using modules, you can extend Magento's functionality, tweak its current capabilities, or connect it to external services. This allows you to create tailored solutions that align with your business needs and customer expectations.
3. **Cost and Time Efficiency:** Modules aid with customization, development, migration, and other tasks. This minimizes the time necessary for web development, lowers the cost of employing an experienced developer, and preserves time for future project developments.
4. **Improved Performance and Robustness:** Magento2 is a much more robust piece of software than its predecessor, with improvements such as unit and integration tests, a richer templating system, and a series of useful design patterns adopted throughout the code.

5. **Ease of Use:** Modules follow a structured directory hierarchy and can include controllers for handling user requests, layout XML files for page customization, blocks for managing content, and helpers for utility functions. This makes it easier for developers to understand and work with the system..

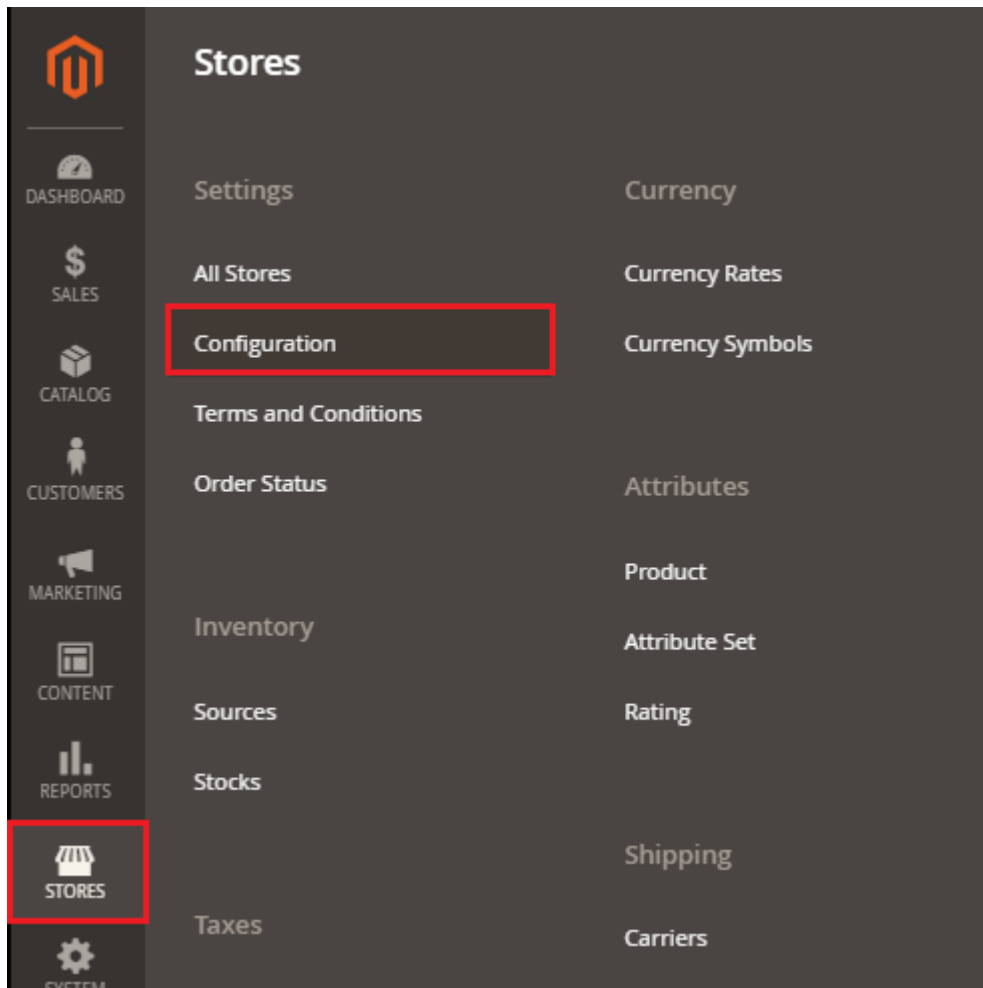
In summary, the use of modules in Magento2 enhances the flexibility, customization, and efficiency of e-commerce store development, making it a powerful tool for businesses.

Example 1: How to setup shipping methods in Magento2?

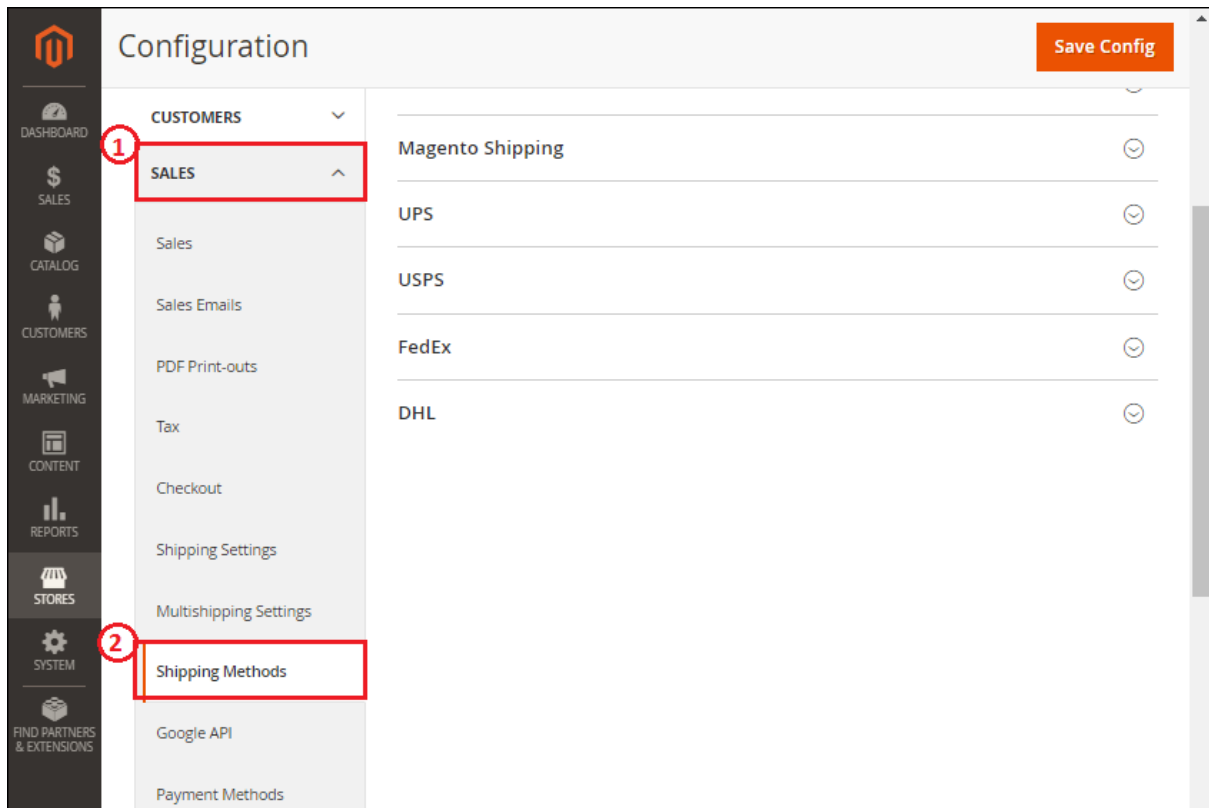
In online business, selling the products to the customers should have scalable and flexible shipping policies. Magento2 is a friendly system for business needs including shipment rates. The free shipping method is applied on the basis of minimum purchase or condition of cart price rule, which you set. Additionally, the shipping methods can be set for certain items or for an entire order that depends on your current strategy, which you set. If you want to charge the different shipping rates on your store, then you can do it with the Table Rate Shipping Method and Flat Rate Shipping Method.

To enable the shipping method follow the below steps:

Step 1: On the left sidebar of Magento2 Admin Panel, Navigate to Store >> Setting >> Configuration.



Step 2: Under the Sales section, click on the Shipping Methods tab.



Here you will get the shipping methods such as Flat Rate, Free Shipping, Table Rates, Magento Shipping, UPS, USPS, FedEx, and DHL.

Flat Rate	⌵
Free Shipping	⌵
Table Rates	⌵
Magento Shipping	⌵
UPS	⌵
USPS	⌵
FedEx	⌵
DHL	⌵

Flat Rate

Step 1: Expand the **Flat Rate** section and fill the required details in the given fields.

Configuration

Save Config

GENERAL

CATALOG

SECURITY

CUSTOMERS

SALES

Sales
Sales Emails
PDF Print-outs
Tax
Checkout
Shipping Settings
Multishipping Settings
Shipping Methods
Google API
Payment Methods
Fraud Protection

ENGAGEMENT CLOUD

SERVICES

ADVANCED

Flat Rate

Enabled [website] Yes ☒ Use system value

Title [store view] Flat Rate ☒ Use system value

Method Name [store view] Fixed ☒ Use system value

Type [website] Per Item ☒ Use system value

Price [website] 5.00 ☒ Use system value

Calculate Handling Fee [website] Fixed ☐ Use system value

Handling Fee [website] 2

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] Specific Countries ☐ Use system value

Ship to Specific Countries [website]

Heard & McDonald Islands
Honduras
Hong Kong SAR China
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland

Show Method if Not Applicable [website] No

Sort Order [website]

- **Enabled:** By default, Flat rate is enabled. So, it will be available in the Tax section of the shopping cart, and also in the Shipping section during checkout.

- **Title:** Enter the title of the flat rate shipping, which will be visible on the checkout page.
- **Method Name:** Set the method name in this field that displays in the shopping cart with the shipping rate. By default, "Fixed" is set.
- **Type:** Determine how the flat rate charges in shipping will apply - Per Item, Per Order, or None.

Type [website]

Per Item	▼
None	
Per Order	
Per Item	

☐ Use system value

- None is similar to the Free shipping method.
- Select the **Per Item** to apply the flat rate shipping for each item.
- Select the **Per Order** to apply the flat rate shipping for the entire order.
- **Price:** Set the Price that you want to charge for the flat rate shipping, e.g., Rs. 6.00. Means 6 rupees per item will apply.
- **Calculate Handling Fee:** If you want to apply some additional charges, set it to **Fixed** or **Percent**, and then enter the amount in **Handling Fee** field.

Calculate Handling Fee [website]

Fixed	▼
Fixed	
Percent	

☐ Use system value

- **Display Error Message:** Provide a short message if the flat rate shipping is unavailable.
- **Ship to Applicable Countries:** We have two options here:
 - All Allowed Countries: By selecting this option, free shipping is supported for every country.
 - Specific Countries: Choose the country to which the Flat Rate Shipping is supported with the Ctrl key.
- **Show Method if Not Applicable:** Determines if Flat Rate appears as an option during checkout if the method does not apply to purchase.
- **Sort Order:** If two or more shipping is enabled, then in which order it will display. So, set the number that determines the order.

Step 2: Click on the **Save Config** button to save the Flat shipping method.

Free Shipping:

Step 1: Expand the **Free Shipping** section and fill the required details in the given fields.

Configuration

Save Config

GENERAL

CATALOG

SECURITY

CUSTOMERS

SALES

Sales

Sales Emails

PDF Print-outs

Tax

Checkout

Shipping Settings

Multishipping Settings

Shipping Methods

Google API

Payment Methods

Fraud Protection

ENGAGEMENT CLOUD

SERVICES

Flat Rate

Free Shipping

Enabled
[website]

Yes

☐ Use system value

Title
[store view]

Free Shipping

☐ Use system value

Method Name
[store view]

Free

☒ Use system value

Minimum Order Amount
[website]

2500

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]

Specific Countries

☐ Use system value

Ship to Specific Countries
[website]

India

Indonesia

Iran

Iraq

Ireland

Isle of Man

Israel

Italy

Jamaica

Japan

Show Method if Not Applicable
[website]

No

Sort Order
[website]

- **Enabled:** Enable the shipping method by setting up the **Yes** for that. Free Shipping appears as an option in the shipping section during checkout when enabled.
- **Title:** Enter the Title for free shipping on the checkout page.
- **Method Name:** Provide a name to describe the shipping method in the **Method Name** field.
- **Minimum Order Amount:** Enter the minimum amount applicable for free shipping.
- **Displayed Error Message:** Provide an error message that will appear if the free rate shipping is unavailable.
- **Ship to Applicable Countries:** Specifies the countries where you offer Free Shipping. We have two options here:
 - All Allowed Countries: By selecting this option, free shipping is available for every country.
 - Specific Countries: Choose the country to which the Free Shipping is supported with the Ctrl key.
- **Show Method if Not Applicable:** Determines if Flat Rate appears as an option during checkout if the method does not apply to purchase.
- **Sort Order:** If two or more shipping is enabled, then in which order it will display. So, set the number that determines the order.

Step 2: Click on the Save Config button to save the Free shipping method.

Table Rates Shipping

Apply the Table Rates Shipping to calculate the shipping fee according to the combination of conditions such as Price vs. Destination, Weight vs. Destination, and # of items vs. Destination. All are listed in a table rate file that we can import and export in Magento2 settings. Now, we will learn to configure the table rate shipping method in Magento2.

Configuration

Save Config

SECURITY

CUSTOMERS

SALES

ENGAGEMENT CLOUD

SERVICES

ADVANCED

Table Rates

Enabled

[website]

Yes

Use system value

Title

[store view]

Best Way

Use system value

Method Name

[store view]

Table Rate

Use system value

Condition

[website]

Price vs. Destination

Use system value

Include Virtual Products in Price Calculation

[website]

Yes

Use system value

Calculate Handling Fee

[website]

Percent

Use system value

Handling Fee

[website]

3

Displayed Error Message

[store view]

This shipping method is not available.

Use system value

Ship to Applicable Countries

[website]

All Allowed Countries

Use system value

Ship to Specific Countries

[website]

Afghanistan

Albania

Algeria

American Samoa

Andorra

Angola

Anguilla

Antarctica

Antigua & Barbuda

Show Method if Not Applicable

[website]

No

Sort Order

[website]

- o **Enabled:** Enable the shipping method by setting up the **Yes** for that.

- **Title:** Enter the Title for table rate shipping on the checkout page. By default, we can use "Best Way" for free shipping.
- **Method Name:** Provide a name to describe the shipping method in the **Method Name** field.
- **Conditions:** Determines the condition on which the calculation is based. We have three options: Weight vs. Destination, Price vs. Destination, and # of Item vs. Destination.

Condition [website]

Price vs. Destination	▼
Weight vs. Destination	
Price vs. Destination	
# of Items vs. Destination	

☐ Use system value

- **Include Virtual Products in Price Calculation:** Determines if virtual products are included in the Table Rate price calculations.

Note: We can calculate the shipping on virtual products. Virtual products do not carry the weight, so it will apply only on Price vs. Destination and # of items vs. Destination.

- **Calculate Handling Fee:** If you want to apply some additional charges, set it to **Fixed** or **Percent**, and then enter the amount in **Handling Fee** field.

Calculate Handling Fee [website]

Fixed	▼
Fixed	
Percent	

☐ Use system value

- **Displayed Error Message:** Provide an error message that will appear if the table rate shipping is not available.
- **Ship to Applicable Countries:** Determines the countries where you offer Table Rate Shipping. We have two options here:
 - All Allowed Countries: By selecting this option, free shipping is available for every country.
 - Specific Countries: Choose the country to which the Free Shipping is supported.
- **Show Method if Not Applicable:** Determines if Flat Rate appears as an option during checkout if the method does not apply to purchase.

- **Sort Order:** If two or more shipping is enabled, then in which order it will display. So, set the number that determines the order.

Step 2: Click on the **Save Config** button to save the Table Rate shipping method.

Example 2: Shipping Modules

Creating a shipping module in Magento2 involves several steps.

Here's a step-by-step guide:

1. Create the Module Directory: The first step is to create a directory for your module. For example, if your vendor name is Mageplaza and your module name is Simpleshipping, you would create the following directory: `app/code/Mageplaza/Simpleshipping`.

2. Declare the Module: Next, you need to declare your module by creating an `etc/module.xml` file in your module directory. This file should contain the following content.

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">
    </module>
</config>
```

AI-generated code. Review and use carefully.

- 3. Register the Module:** After declaring your module, you need to register it by creating a `registration.php` file in your module directory. This file should contain the following content¹:

PHP

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

AI-generated code. Review and use carefully.

- 4. Create the Configuration File:** Now, create a `config.xml` file at `app/code/Mageplaza/Simpleshipping/etc/config.xml`. This file will contain the configuration for your shipping method¹:

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>
      <simpleshshipping>
        <active>1</active>
        <sallowsspecific>0</sallowsspecific>
        <model>Mageplaza\Simpleshipping\Model\Carrier\Shipping</model>
        <name>Mageplaza Sample Shipping Method</name>
        <price>10.00</price>
        <title>Mageplaza Sample Shipping Method</title>
        <specificerrormsg>This shipping method is not available. To use this shipping method, please
contact us.</specificerrormsg>
        <handling_type>F</handling_type>
      </simpleshshipping>
    </carriers>
  </default>
</config>
```

AI-generated code. Review and use carefully.

5. **Enable the Module:** Finally, you need to enable your module. You can do this by running the following command in your Magento2 root directory

```
php bin/magento module:enable Mageplaza_Simpleshipping
```

After running this command, your shipping module should be enabled and ready to use. You can check the status of your module by running the following command

```
php bin/magento module:status
```

This is a basic guide to creating a shipping module in Magento2. Depending on your specific needs, you may need to add additional files and configurations to your module

Part – A

1. What is shopping tool?
2. What is promotions?
3. What is Catalog price rule?
4. What is Cart price rule?
5. What is Message Queues?
6. What is SEO?
7. What is point of purchase?
8. What is distributed order management in Magento?
9. What is SMTP?
10. What is shipping?

Part – B

1. Explain the different types of shopping tools application with example.
2. What are the features of Magento2 tool
3. What are the components of the Magento2 Order Management System
4. Write a short notes about the Payment methods in Magento2.
5. Explain the shipping methods in Magento2.
6. Write a short notes about the configuring sales tax rules.
7. Write a short notes about the outgoing E- mails in Magento2

Part - C

1. Explain the features of SEO & Search in Magento2.
2. Explain the shipping modules and benefits with example