

1. Payment Method Overview

Magento 2 supports multiple payment methods:

- **Online Payment Methods:** Integrated with payment gateways (e.g., PayPal, Stripe, Authorize.Net).
- **Offline Payment Methods:** Check/Money Order, Cash on Delivery, etc.

Key Concepts:

- **Payment Method Configuration:** Configured in the admin panel under **Stores > Configuration > Sales > Payment Methods**.
- **Integration Points:**
 - Orders
 - Invoices
 - Credit Memos (Refunds)

2. Order-Related Tables

Magento 2 has a robust table structure for handling orders, payments, invoices, and refunds. Here's an overview of the key tables:

Table Name	Description
sales_order	Stores basic order information like order ID, status, total, and customer details.
sales_order_payment	Stores payment-related details, such as payment method, transaction ID, and payment status.
sales_order_item	Contains item-level details for orders, like quantity, price, and SKU.
sales_invoice	Stores invoice data, including totals, tax details, and payment links.
sales_invoice_item	Links invoice items to specific products and quantities.
sales_creditmemo	Holds refund data, including refunded amounts and linked invoices or orders.
sales_creditmemo_item	Stores product-level details for refunds.
sales_order_status	Defines custom statuses for orders.
sales_order_status_state	Links order states to statuses.
sales_payment_transaction	Tracks payment transactions (authorization, capture, refund).

3. Invoice Workflow

How Invoices Work:

1. Order Placement:

- a. Payment is either authorized or captured based on the configured payment method.
- b. Payment details are stored in the sales_order_payment table.

2. Invoice Generation:

- a. Admin or API generates an invoice for the order.
- b. Data flows from sales_order to sales_invoice and sales_invoice_item.
- c. Payment capture occurs at this stage (if not already captured during order placement).

Relevant Tables for Invoices:

- sales_invoice
- sales_invoice_item
- sales_order_payment

4. Refund (Credit Memo) Workflow

How Refunds Work:

1. Initiating a Refund:

- a. Admin creates a credit memo linked to an invoice.
- b. Refund amount and items are specified.
- c. Refund can be issued online (via the payment gateway) or offline.

2. Data Storage:

- a. Refund data is stored in the sales_creditmemo and sales_creditmemo_item tables.
- b. Payment gateway interactions (for online refunds) are tracked in the sales_payment_transaction table.

3. Updating Order Status:

- a. Order status and state are updated based on the refund.

Relevant Tables for Refunds:

- sales_creditmemo
- sales_creditmemo_item
- sales_payment_transaction

5. Payment Modules and Transactions

Magento processes payments using a combination of built-in and custom modules. Each payment method module implements:

- **Payment Interface:** Defines core functionalities like authorization, capture, and refund.
- **Payment Model:** Implements specific logic for payment gateway interactions.

Modules Involved:

- **Magento_Payment:** Base module for payment methods.
- **Custom Payment Modules:** Extend the `Magento\Payment\Model\Method\AbstractMethod` class.
- **Magento_Sales:** Handles order, invoice, and refund workflows.
- **Magento_PaymentGraphQL:** Supports GraphQL APIs for payments.

Transaction Management:

- Transactions (e.g., capture, refund) are stored in the `sales_payment_transaction` table with a parent-child hierarchy for related operations.

6. Payment Method Integration with Invoices and Refunds

1. Invoice Workflow Integration:

- a. For online methods: Capture payment via the payment gateway during invoice creation.
- b. For offline methods: Mark as paid manually.

2. Refund Workflow Integration:

- a. For online methods: Trigger a refund transaction via the payment gateway.
- b. For offline methods: Refund is processed manually without gateway interaction.

Key APIs for Development:

- **Payment Method Implementation:**
 - Extend `\Magento\Payment\Model\Method\AbstractMethod`.
- **Invoice Operations:**
 - Use `\Magento\Sales\Model\Order\Invoice`.
- **Refund Operations:**
 - Use `\Magento\Sales\Model\Order\Creditmemo`.

7. Debugging and Testing

To debug and test payment, invoice, and refund workflows:

1. Enable Logging:

- a. Use the var/log/payment.log file for payment-specific logs.
- b. Enable debug logging for the specific payment module.

2. Use Database Queries:

- a. Check the sales_* tables for accurate data storage.
- b. Verify the sales_payment_transaction table for transaction records.

3. Test Scenarios:

- a. Place orders with different payment methods.
- b. Generate invoices and test payment captures.
- c. Process refunds and validate data integrity.

✔ Task "Trigger recollect totals for quotes by rule ID 1": 1 item(s) have been successfully updated.

[View Details](#) System Messages: 3

Credit Memos

Search by keyword

Filters

Default View

Columns

Export

Actions

2 records found

20 per page1 of 1

	Credit Memo	Created	Order #	Order Date	Bill-to Name	Status	Refunded	Payment Method	Action
<input type="checkbox"/>	000000001	Apr 19, 2024 7:35:37 AM	000000002	Apr 19, 2024 7:35:37 AM	Veronica Costello	Refunded	\$39.64	Check / Money order	View
<input type="checkbox"/>	000000002	Jan 20, 2025 10:12:34 PM	000000095	Jan 19, 2025 10:43:11 PM	vicky s	Refunded	\$45.00	Test Payment	View

Copyright © 2025 Magento Commerce Inc. All rights reserved.

Magento ver. 2.4.6-p4
[Privacy Policy](#) | [Report an Issue](#)

✔ Task "Trigger recollect totals for quotes by rule ID 1": 1 item(s) have been successfully updated.

[View Details](#) System Messages: 3

Invoices

Search by keyword

Filters

Default View

Columns

Export

Actions

9 records found

20 per page1 of 1

	Invoice	Invoice Date	Order #	Order Date	Bill-to Name	Status	Grand Total (Base)	Grand Total (Purchased)	Payment Method	Action
<input type="checkbox"/>	000000009	Jan 20, 2025 1:29:13 AM	000000095	Jan 19, 2025 10:43:11 PM	vicky s	Paid	\$105.00	\$105.00	Test Payment	View
<input type="checkbox"/>	000000008	Dec 19, 2024 1:21:03 AM	000000071	Nov 13, 2024 12:21:51 AM	Ragu ram	Paid	\$58.00	\$58.00	Check / Money order	View
<input type="checkbox"/>	000000007	Jul 2, 2024 2:31:50 AM	000000018	Jul 2, 2024 2:29:57 AM	shanmuga NewLastName	Paid	\$43.40	\$43.40	Check / Money order	View
<input type="checkbox"/>	000000006	Jun 20, 2024 12:49:51 AM	000000013	Jun 7, 2024 4:26:20 AM	vicky s	Paid	\$29.00	\$29.00	Check / Money order	View
<input type="checkbox"/>	000000005	May 30, 2024 1:19:21 AM	000000011	May 30, 2024 1:14:54 AM	ragu ram	Paid	\$64.00	\$64.00	Check / Money order	View
<input type="checkbox"/>	000000004	May 28, 2024 12:05:21 AM	000000008	May 28, 2024 12:03:24 AM	ragu ram	Paid	\$108.00	\$108.00	Check / Money order	View

✔ Task "Trigger recollect totals for quotes by rule ID 1": 1 item(s) have been successfully updated.

[View Details](#) System Messages: 3

Orders

Search by keyword

Filters

Default View

Columns

Export

Actions

54 records found

20 per page1 of 3

	ID	Purchase Point	Purchase Date	Bill-to Name	Ship-to Name	Grand Total (Base)	Grand Total (Purchased)	Status	Payment Method	Action
<input type="checkbox"/>	000000095	Main Website Main Website Store Default Store View	Jan 19, 2025 10:43:11 PM	vicky s	vicky s	\$105.00	\$105.00	Processing	Test Payment	View
<input type="checkbox"/>	000000094	Main Website Main Website Store Default Store View	Jan 2, 2025 3:28:39 AM	shanmuga NewLastName	shanmuga NewLastName	\$17.00	\$17.00	Pending	Check / Money order	View

Create New Order

Default Payment Methods:

- **Cash On Delivery (COD):** Allows customers to pay for their orders with cash upon delivery.
- **Bank Transfer (Wire Transfer):** Provides bank account details to customers for making a payment via bank transfer.
- **Check / Money Order:** Allows customers to pay via check or money order.
- **Credit Card (Saved CC):** Lets customers save their credit card details for future use and facilitates faster checkout.
- **PayPal:** Integrates PayPal payment processing into the checkout process.

- **Authorize.net:** Enables payment processing through the Authorize.net payment gateway.
- **Braintree:** Integrates Braintree payment processing into the checkout process.
- **Stripe:** Allows payment processing via the Stripe payment gateway.
- **Cash On Pickup:** same as COD, but customers pay with cash when picking up the order in person.

Core Magento 2 Payment Modules

Magento 2 comes with several built-in payment methods, which are defined in core modules. These include:

1. **Magento_Payment:**

- This is the core module responsible for the basic infrastructure for payment methods.
- It provides the interfaces and abstract classes for integrating custom payment methods.
- The primary interface here is `Magento\Payment\Model\MethodInterface`, which all payment method classes must implement.
- `Magento\Payment\Model\Method\AbstractMethod`: Base class for creating custom payment methods.
- `Magento\Payment\Model\MethodInterface`: Interface that all payment methods should implement.
- `Magento\Payment\Model\Method\Free`: Provides a free payment method for testing.

2. **Magento_Paypal:**

- Provides PayPal integration for Magento, supporting different types of PayPal payment methods like Express Checkout, PayPal Credit, and more.
- Express Checkout:** A fast checkout option that allows customers to pay using their PayPal account.
- Standard:** Redirects customers to the PayPal site for completing the payment.
- Payflow Pro:** Allows for advanced payment options and integrates with the Payflow API.

3. **Magento_BankTransfer:**

- A payment method that supports bank transfer as the payment option.
- It provides an option for customers to pay via bank transfer after placing the order.

4. **Magento_CashOnDelivery:**

- Provides a "Cash on Delivery" (COD) payment method for customers who prefer paying in cash when their order is delivered.

5. **Magento_Checkout:**

- Handles the checkout process, including the interaction with payment methods, but does not provide specific payment methods itself.

- b. It integrates with `Magento_Payment` to display payment methods during the checkout process.

1. Payment-Related Vendor Files

The payment system in Magento 2 is implemented in the `Magento_Payment` module and related modules for specific payment methods.

Core Payment Module (`Magento_Payment`):

- **Location:** `vendor/magento/module-payment/`
- **Key Files:**
 - **Config Files:**
 - `etc/config.xml` - Global configuration for the payment module.
 - `etc/adminhtml/system.xml` - Admin configuration for payment methods.
 - `etc/frontend/di.xml` - Dependency injection for frontend payment services.
 - `etc/webapi.xml` - API definitions for payment methods.
 - **Model Files:**
 - `Model/Method/AbstractMethod.php` - Base class for payment methods.
 - `Model/Info.php` - Stores payment-related information.
 - **Controller Files:**
 - `Controller/Transparent/Response.php` - Handles payment gateway responses for transparent redirect methods.
 - **View Files:**
 - `view/frontend/web/js/view/payment/` - JavaScript components for payment methods.

Specific Payment Method Modules:

Magento 2 includes some default payment method modules:

- **Magento_Braintree:** `vendor/magento/module-braintree/`
- **Magento_Paypal:** `vendor/magento/module-paypal/`
- **Magento_Authorizenet:** `vendor/magento/module-authorizenet/`

Each of these modules contains:

- `etc/config.xml` for method-specific configuration.
- `Model/` directory for method-specific logic.
- `view/frontend/web/js/` for frontend components.

2. Invoice-Related Vendor Files

Invoices are managed by the Magento_Sales module.

Core Sales Module (Magento_Sales):

- **Location:** vendor/magento/module-sales/
- **Key Files:**
 - **Config Files:**
 - etc/config.xml - Invoice and credit memo configurations.
 - etc/webapi.xml - API definitions for managing invoices.
 - etc/di.xml - Dependency injection for invoice-related services.
 - **Model Files:**
 - Model/Order/Invoice.php - Core model for invoices.
 - Model/Order/Invoice/Item.php - Handles item-level invoice details.
 - Model/Order/InvoiceRepository.php - Repository for managing invoice data.
 - **API Interfaces:**
 - Api/InvoiceRepositoryInterface.php - Defines methods for invoice data access.
 - **Controller Files:**
 - Controller/Adminhtml/Order/Invoice/ - Admin controllers for invoice actions.
 - **View Files:**
 - view/adminhtml/templates/order/invoice/ - Admin templates for invoice views.

3. Refund (Credit Memo)-Related Vendor Files

Refunds (credit memos) are also managed by the Magento_Sales module.

Core Sales Module (Magento_Sales):

- **Location:** vendor/magento/module-sales/
- **Key Files:**
 - **Model Files:**
 - Model/Order/Creditmemo.php - Core model for credit memos.
 - Model/Order/Creditmemo/Item.php - Handles item-level refund details.
 - Model/Order/CreditmemoRepository.php - Repository for managing credit memo data.
 - **API Interfaces:**
 - Api/CreditmemoRepositoryInterface.php - Defines methods for refund data access.
 - **Controller Files:**
 - Controller/Adminhtml/Order/Creditmemo/ - Admin controllers for refund actions.
 - **View Files:**

- `view/adminhtml/templates/order/creditmemo/` - Admin templates for refund views.

4. Payment Transactions (Magento_Sales and Magento_Payment):

Payment transactions are tracked in the Magento_Sales and Magento_Payment modules.

- **Key Files:**
 - `Magento_Payment\Model\Info.php` - Stores and manages payment information.
 - `Magento_Sales\Model\Order\Payment\Transaction.php` - Core model for payment transactions.
 - `Magento_Sales\Model\ResourceModel\Order\Payment\Transaction.php` - Resource model for database operations.

5. Commonly Used APIs

Payment:

- `Magento\Payment\Api\Data\PaymentInterface`
- `Magento\Payment\Api\PaymentMethodInterface`

Invoice:

- `Magento\Sales\Api\Data\InvoiceInterface`
- `Magento\Sales\Api\InvoiceRepositoryInterface`

Refund:

- `Magento\Sales\Api\Data\CreditmemoInterface`
- `Magento\Sales\Api\CreditmemoRepositoryInterface`

6. Frontend and Admin UI Components

Payment, invoice, and refund features rely heavily on Magento's UI components:

- **Location:** `vendor/magento/module-sales/view/`
 - `adminhtml/ui_component/sales_order_invoice_grid.xml`
 - `adminhtml/ui_component/sales_order_creditmemo_grid.xml`

1. Magento_Braintree

The Magento_Braintree module integrates Braintree, a payment gateway that supports multiple payment methods, including credit/debit cards, PayPal, and wallets like Google Pay and Apple Pay.

Key Features:

- Support for credit and debit card payments.
- Integration with PayPal through Braintree.
- Tokenization for secure payment data storage.
- Support for 3D Secure (for additional fraud protection).
- Vault integration for storing customer payment methods.
- Hosted Fields for PCI compliance.

2. Magento_Paypal

The Magento_Paypal module integrates various PayPal payment solutions, including Express Checkout, PayPal Credit, and PayPal Payments Pro.

Key Features:

- Support for PayPal Express Checkout.
- Integration with PayPal Credit (deferred payments).
- Support for in-context checkout (reduces redirection).
- Comprehensive reporting and refund capabilities.
- Fraud management tools (via PayPal).

3. Magento_Authorizenet

The Magento_Authorizenet module integrates Authorize.Net, a payment gateway primarily used for credit and debit card transactions.

Key Features:

- Support for credit and debit card payments.
- Advanced fraud detection suite (AFDS).
- Tokenization for secure storage of payment details.
- Support for Authorize.Net CIM (Customer Information Manager) for saving cards.
- Refunds and voids.

Comparison of Modules

Feature	Braintree	PayPal	Authorize.Net
Payment Types	Cards, PayPal, Wallets	PayPal, PayPal Credit	Credit/Debit Cards
PCI Compliance	Hosted Fields, Tokenization	In-context checkout, Tokenization	Tokenization, CIM
Fraud Protection	3D Secure, Braintree fraud tools	PayPal fraud tools	Advanced Fraud Detection Suite (AFDS)
Vault Integration	Yes	Yes	Yes
Refund Capabilities	Yes	Yes	Yes