

# Ingenico ePayments (Connect platform) for Magento® 2

This extension allows processing payments in Magento® via the Ingenico ePayments Connect platform.

# End user documentation

<b>1 Requirements</b>	<b>4</b>
1.1 Magento®	4
1.2 PHP	4
1.3 Ingenico contract	4
<b>2 Hints for using the module</b>	<b>5</b>
2.1 Language support	5
2.2 Difference between Cancel and Void operations	5
2.3 Compatibility with OneStepCheckout Magento Extension	5
<b>3 Installation and configuration</b>	<b>6</b>
3.1 Before installation	6
3.2 Installation options	6
3.2.1 Installation from package using Composer	6
3.2.2 Installation from Marketplace using Composer	6
3.3 Module configuration	8
3.3.1 General Settings	8
3.3.2 Account Settings	9
3.3.3 Capture Settings	10
3.3.4 Checkout Settings	11
3.3.5 Webhooks	12
3.3.6 Cancellation of Pending Orders	16
3.3.7 Fraud notification	17
3.3.8 Payment statuses frontend output	18
3.3.9 Payment methods frontend output	19
3.3.10 Customer notification Email Settings	20
<b>4 Workflow and features</b>	<b>21</b>
4.1 Order workflow	21
4.1.1 Order placement	21
4.1.2 Capture (create invoice)	22
4.1.3 Refund (create creditmemo)	25
4.1.4 Cancel orders	27
4.2 Additional features and tips	28
4.2.1 Transaction information	28
4.2.2 Order update	29
4.2.3 Fraud handling	31

4.2.4 E-Mail Notifications	32
<b>5 Uninstalling or disabling the module</b>	<b>33</b>
<b>6 Troubleshooting</b>	<b>34</b>
6.1 Payment status not accurate	34
6.2 Check log file	35
6.3 Technical support	35

# 1 Requirements

The following requirements must be met for the smooth operation of the module:

## 1.1 Magento®

The following Magento® versions are supported:

- Community Edition 2.1.7 and later
- Enterprise Edition 2.1.7 and later

## 1.2 PHP

These PHP versions are supported:

- PHP 7.1
- PHP 7.0
- PHP 5.6

## 1.3 Ingenico contract

To use this extension, you need an Ingenico ePayments account (contract).

Please visit <https://www.ingenico.com/epayments> for more information or contact the Ingenico team.

## 2 Hints for using the module

### 2.1 Language support

The module supports the locale `en_US`. Translations are stored in CSV translation files and can therefore be modified by third-party modules.

### 2.2 Difference between Cancel and Void operations

On the order detail page, you will often have the ability to either "Cancel" or "Void" an order:

- To **cancel** an order means to cancel every transaction related to the order. This makes it impossible to create any further transactions (invoices, creditmemos, etc.).
- The **void** action only cancels the invoice of the order (i.e. a capture transaction). This leaves the order itself intact for further capture operations.

### 2.3 Compatibility with OneStepCheckout Magento Extension

The module is compatible with the OneStepCheckout Magento 2 extension (onestepcheckout.com) in version 1.2.022. Newer versions may work as well, but are not officially supported.

Before you use the module with OneStepCheckout, please make sure to check the OneStepCheckout configuration. In the section **Billing address fields and settings**, the field `country_id` must be set to *Enabled* and *Required*.

#### Compatibility

We can only guarantee compatibility with the OneStepCheckout Magento 2 extension version 1.2.022. However, feel free to report any issues with newer versions to our [Technical support](#).

## 3 Installation and configuration

### 3.1 Before installation

We strongly recommend making a complete backup of the shop's files and database before any module installation or update.

Also, the installation and module functionality should be completely verified on a testing or staging system before moving it into a production system.

### 3.2 Installation options

#### 3.2.1 Installation from package using Composer

If you got the module as a `.zip` or `.tgz` package, and the package **contains a 'composer.json' file**, it is a so-called *Artifact* which can be installed with Composer.

Follow these steps:

- Copy (do not unzip) the package into a folder that is accessible by Composer.
- On the command line, navigate into the root folder of your Magento® 2 installation.
- Execute the following commands:

::

```
composer config repository.epayments artifact /path/to/folder/with/zip composer require nrepayments/module-epayments-m2 composer update
```

- Make sure that no Composer errors occurred. If there are any, they must be resolved.
- **Important:** the path in the above command is absolute, i.e. starting at the server (Linux) root.
- Then log out from the Magento® 2 admin panel, and log in again.
- After this, clear all Magento caches.
- Now proceed with the [Module configuration](#).

#### 3.2.2 Installation from Marketplace using Composer

You can get the module from the Magento® Marketplace and install it with Composer. The module files will be downloaded from the public repository during the installation.

Follow the instructions on the [`Marketplace documentation<http://docs.magento.com/marketplace/user\\_guide/quick-tour/install-extension.html>`](http://docs.magento.com/marketplace/user_guide/quick-tour/install-extension.html).

If you have access to the CLI of your installation, you can also use the steps described [here<https://devdocs.magento.com/guides/v2.2/comp-mgr/install-extensions.html>](https://devdocs.magento.com/guides/v2.2/comp-mgr/install-extensions.html) `\_\_`.

- Make sure that no Composer errors occurred. If there are any, they must be resolved.
- Then log out from the Magento® 2 admin panel, and log in again.
- After this, clear all Magento caches.
- Now proceed with the [Module configuration](#).

### 3.3 Module configuration

The extension adds a new configuration sections which must be configured to process payments with Ingenico ePayments:

Stores → Configuration → Sales → Ingenico ePayments

#### Please note

Make sure that all fields in *Account Settings* and *Fraud Notification* are configured before saving the configuration.

Flush the Magento caches after making changes to the configuration.

#### 3.3.1 General Settings

In the configuration section *General Settings*, you can configure the name of the Ingenico payment method that will be displayed to customers during checkout (see also [Order placement](#)).

Set *Enabled* to *Yes* to display Ingenico ePayments in the checkout. If you set it to *No*, the payment method is hidden from the checkout, but the module will still be active.

Use the *Group methods* setting to decide whether to sort payment methods in the checkout view by type (choose *Yes*) or use the default sorting (choose *No*). Possible Payment types are *card*, *redirect*, *bank transfer*, *invoice* and *cash*.

General



<b>Version</b> <small>[website]</small>	1.2.0
<b>Enabled</b> <small>[website]</small>	<input type="text" value="Yes"/> <span style="float: right;">▼</span>
	<small>Show payment method in checkout</small> <span style="float: right;">?</span>
<b>Title</b> <small>[store view]</small>	<input type="text" value="Ingenico ePayments"/>

### 3.3.2 Account Settings

The section *Account Settings* holds your access data (API endpoint, credentials, etc.) for the Ingenico ePayments platform. You will receive this information from the Ingenico team.

The *secondary* api endpoint is used as fallback if authentication with the *primary* endpoint fails. This allows to safely migrate endpoints.

#### Account Settings



<b>API Endpoint</b> <small>[website]</small>	<input type="text" value="https://world.preprod.api-ingenico.com"/>	
	<small>Available endpoints are listed in the <a href="#">API Docs</a>.</small>	
<b>API Endpoint (Secondary)</b> <small>[website]</small>	<input type="text" value="https://eu.preprod.api-ingenico.com"/>	
	<small>Fallback if primary endpoint fails.</small>	
<b>API Key</b> <small>[website]</small>	<input type="password" value="*****"/>	
<b>API Secret</b> <small>[website]</small>	<input type="password" value="*****"/>	
<b>MID (Merchant ID)</b> <small>[website]</small>	<input type="text"/>	
<b>Soft Descriptor</b> <small>[website]</small>	<input type="text"/>	
<b>Hosted Checkout Subdomain</b> <small>[website]</small>	<input type="text" value="https://payment."/>	
<b>Log All Requests</b> <small>[website]</small>	<input type="button" value="Yes"/>	
	<small><a href="#">Download log file</a></small>	

#### Please note

If you enable *Log All Requests*, you should clear or rotate the log files regularly.

### 3.3.3 Capture Settings

The section *Capture Settings* is used to configure what to do when a new payment is created. Here you have two options:

- on setting *Direct Capture*, payments will automatically be captured and will be reported as paid as soon as the capture is complete on the gateway. An invoice is created automatically
- with *Delayed Settlement* you will have to manually capture new payments by creating an invoice from the order details page.

#### Capture Settings



Configuration of payment capture operations

Capture mode

[store view]



#### Delayed settlements

For more information about delayed settlements, please refer to the [Ingenico ePayments API Documentation](#), section 'What is delayed settlement?'.

### 3.3.4 Checkout Settings

The section *Checkout Settings* is used to configure how payment information is going to be captured by the customer.

If you enable inline payments the payment data (e.g. credit card information) is requested directly in your checkout and encrypted via the Connect JavaScript SDK. The encrypted data is then sent to your shop and forwarded to the Connect platform through the Connect Server SDK. Not all payment methods support payment data capture through this way - for those who don't the HostedCheckout page is being used.

Should you disable inline payments, the customer will get redirected to the HostedCheckout page for all available payment products after completing the Magento® checkout.

#### Checkout Settings

Configuration of the Magento checkout

Enable inline payments [store view]

Note that the use of inline payments requires your Magento store to be PCI compliant on level SAQ-A EP.

#### PCI compliance

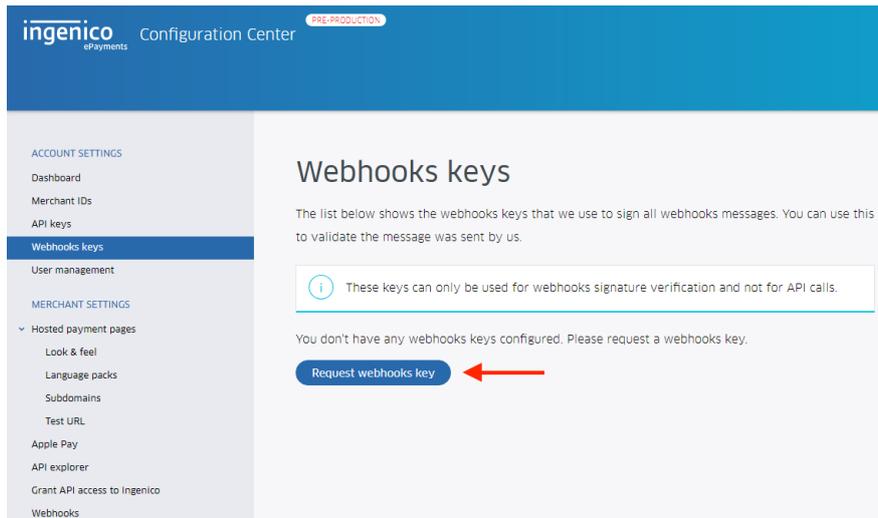
When using inline payments you will need to fill PCI SAQ A-EP, otherwise you should use PCI SAQ A.

### 3.3.5 Webhooks

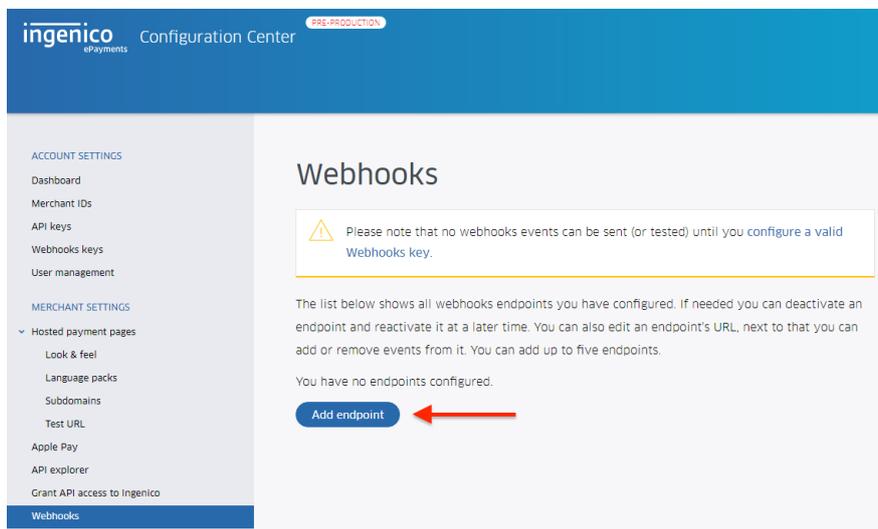
In the section *Webhooks* you can set up the Ingenico Webhooks integration. If required, you will receive Webhooks credentials from the Ingenico ePayments team.

Intructions for the Webhooks configuration:

Please login to the Configuration Center and click on **Webhooks keys** on the left menu under **ACCOUNT SETTINGS** and then click the button **Request webhooks key**:



To configure the endpoint, click on **Webhooks** on the left menu under **MERCHANT SETTINGS** and then click the button **Add endpoint**:



You will need to add two endpoints, one for the payments and one for the refunds. When you create the endpoint for the payment, enter <https://shop.domain/epayments/webhooks/payment> (replace shop.domain with the correct domain) as the Endpoint URL and subscribe to the payment events:

Edit Endpoint ✕

Here you can create a new endpoint and configure which events it should be subscribed to.

Endpoint URL

Subscriptions  Send all events  Send selected events

**payment**

✓	payment.created	The transaction has been created. This is the initial state once a new payment is created.
✓	payment.redirected	The consumer has been redirected to a 3rd party to complete the authentication/payment.
✓	payment.pending_payment	Instructions have been provided and we are now waiting for the money to come in.
✓	payment.account_verified	The account has been verified using a validation services like 0\$ authorization.
✓	payment.pending_fraud_approval	The transaction is awaiting approval from you to proceed with the capturing of the funds.
✓	payment.authorization_requested	We have requested an authorization against an asynchronous system and is awaiting its response.
✓	payment.pending_approval	There are transactions waiting for your approval.
✓	payment.pending_completion	There are transactions waiting for you to complete them.
✓	payment.pending_capture	There are transactions waiting for you to capture them.
✓	payment.capture_requested	The transaction is in the queue to be captured (For Cards, this means that the transaction has been authorized).
✓	payment.captured	The transaction has been captured and we have received online confirmation.
✓	payment.paid	We have matched the incoming funds to the transaction.
✓	payment.reversed	The transaction has been reversed.
✓	payment.chargeback_notification	We have received a notification of chargeback and this status informs you that your account will be debited for a particular transaction.
✓	payment.chargebacked	The transaction has been chargebacked.
✓	payment.rejected	The transaction has been rejected.
✓	payment.rejected_capture	We or one of our downstream acquirers/providers have rejected the capture request.
✓	payment.cancelled	You have cancelled the transaction.
✓	payment.refunded	The transaction has been refunded.

**payout**

Then create another endpoint for the refunds, entering <https://shop.domain/epayments/webhooks/refund> (replace shop.domain with the correct domain) as the Endpoint URL and subscribe to the refund events.

Edit Endpoint
×

Here you can create a new endpoint and configure which events it should be subscribed to.

---

Endpoint URL

Subscriptions  Send all events  Send selected events

payment

refund

Select all Deselect all

<input checked="" type="checkbox"/>	refund.created	The refund has been created.
<input checked="" type="checkbox"/>	refund.pending_approval	There are refunds waiting for your approval.
<input checked="" type="checkbox"/>	refund.rejected	The refund request has been rejected.
<input checked="" type="checkbox"/>	refund.refund_requested	The transaction is in the queue to be refunded.
<input checked="" type="checkbox"/>	refund.captured	The transaction has been captured and we have received online confirmation.
<input checked="" type="checkbox"/>	refund.refunded	The transaction has been refunded.
<input checked="" type="checkbox"/>	refund.cancelled	You have cancelled the refund.

token

Select all Deselect all

<input type="checkbox"/>	token.created	A token has been created (if done by a user).
<input type="checkbox"/>	token.card.expired	A token references a card that has expired.
<input type="checkbox"/>	token.card.expiring_soon	A token references a card that will expire next month.
<input type="checkbox"/>	token.deleted	A token has been deleted (if done by a user).

---

For security reasons, we need to verify your password.

Your password

On the Magento extension, you will have to enter the Webhooks Key ID and Secret Key in the following screen:

Webhook Settings ⊙

Webhooks Key ID (set 1) <small>(webhook)</small>	<input type="text"/>	?
Webhooks Secret Key (set 1) <small>(webhook)</small>	<input type="text"/>	?
Webhooks Key ID (set 2) <small>(webhook)</small>	<input type="text"/>	?
<small>Fallback if primary key (set 1) fails.</small>		
Webhooks Secret Key (set 2) <small>(webhook)</small>	<input type="text"/>	?
<small>Fallback if primary key (set 1) fails.</small>		

Again, the *secondary* credentials are used if authentication with the *primary* credentials fails. You can use this to safely migrate keys without missing any Webhooks requests.

### 3.3.6 Cancellation of Pending Orders

In the section *Cancellation of Pending Orders*, you can configure the extension to automatically cancel abandoned orders after a period of days.

Cancellation of pending orders



Number of days before cancellation  
[global]



This requires working Magento Cronjobs!

#### Please note

Automatic cancellations are performed by a periodically running *Cronjob*.

For this, the standard Magento® 2 Cronjobs must be configured correctly on your system. See also section [Order update](#).

Please note that questions about the basic setup of Cronjobs in Magento® 2 are not covered by our technical support.

### 3.3.7 Fraud notification

In the section *Fraud notification* you can configure a destination email address for notifications when the Ingenico ePayments Fraud Detection is triggered.

#### Fraud Notification

**Manager Email**  
[store view]  

Destination email address for Fraud notifications.

#### Please note

*Manager Email* is a required field.

If Magento® 2 doesn't allow saving the configuration, check if the field contains an email address.

### 3.3.8 Payment statuses frontend output

In the section *Payment statuses* you will find a list of payment statuses and their definition.

These definition texts will be displayed to the customer after making a payment in the Magento® 2 checkout.

You can modify the status definition texts on the Store View level according to your needs.

The default definitions are provided by Ingenico ePayments. If you need to know the original values, you can find them in the [Ingenico ePayments API Documentation](#).

**Payment statuses** ⌵

This is list of **payment statuses** and their **definition**.  
 Source for **default definitions**: [API Documentation](#): **Get payment**  
 You can specify custom definition for desired statuses.

<b>ACCOUNT_VERIFIED</b> <small>[store view]</small>	The account has been verified using a validation services like 0\$ auth
<b>CREATED</b> <small>[store view]</small>	The transaction has been created. This is the initial state once a new payment is created

### 3.3.9 Payment methods frontend output

In the section *Payment methods* you will find a list of payment methods groups and their display name in the checkout.

These definition texts will be displayed to the customer for grouping the offered payment methods by type in the Magento® checkout.

You can modify the payment method group label texts on the Store View level according to your needs.

#### Payment Methods Frontend Output



This is a list of **payment methods** by which payment options are grouped in the checkout. You can specify **custom titles** for these methods.

<b>token</b> <small>[store view]</small>	Saved payment options
<b>bankTransfer</b> <small>[store view]</small>	Bank transfer
<b>card</b> <small>[store view]</small>	Credit cards
<b>cash</b> <small>[store view]</small>	Cash
<b>directDebit</b> <small>[store view]</small>	Direct debit
<b>eInvoice</b> <small>[store view]</small>	e-Invoice
<b>invoice</b> <small>[store view]</small>	Invoice
<b>redirect</b> <small>[store view]</small>	Redirect

### 3.3.10 Customer notification Email Settings

You can configure when to notify a customer about the status of their payment in the *Customer notification Email Settings* section.

If one of the options is set to *Yes*, the customer will automatically receive a notification email when their payment gets the corresponding status.

#### Email Settings ⌵

<b>Payment Successful</b> <small>[store view]</small>	<input type="text" value="No"/>	▼
<b>Fraud Suspicion</b> <small>[store view]</small>	<input type="text" value="No"/>	▼
<b>Customer Action Needed</b> <small>[store view]</small>	<input type="text" value="Yes"/>	▼
<b>Delayed Settlement</b> <small>[store view]</small>	<input type="text" value="Yes"/>	▼
<b>Slow 3rd Party</b> <small>[store view]</small>	<input type="text" value="Yes"/>	▼

## 4 Workflow and features

### 4.1 Order workflow

This section describes the standard workflow of an order when using Ingenico ePayments. Special cases and actions will be documented in their respective chapters.

#### 4.1.1 Order placement

If the Ingenico ePayments payment method is enabled in the section [Account Settings](#), the shop customer can select the payment method in the checkout.

The name of the payment method can be configured in the [General Settings](#) section.

If this payment method is selected, it offers a dynamic list of payment products, depending on the customer's billing address country, order amount, and the configured Ingenico ePayments endpoint.

Ingenico ePayments

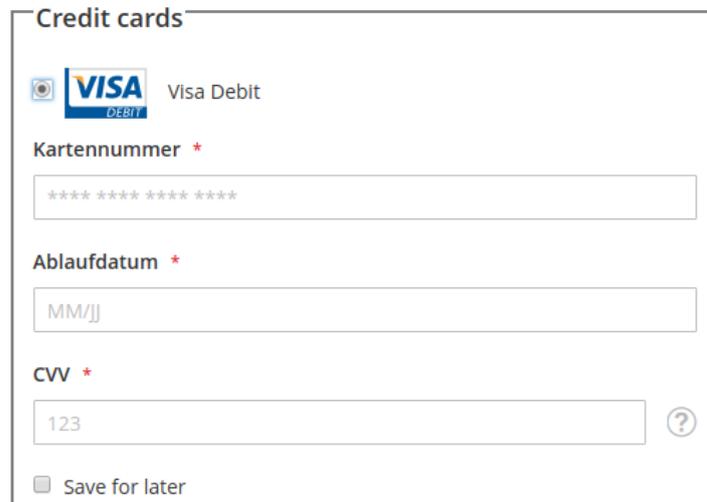
My billing and shipping address are the same

John Doe  
 Samplestreet 123  
 New York, New York 10880  
 Vereinigte Staaten  
 0123465798

**Redirect**

-  paysafecard
-  PayPal
-  WebMoney
-  Skrill
-  Alipay

If inline payments has been enabled in the [Checkout Settings](#), depending on the payment product, input fields will show up upon selection.



The entered data gets sent to your Magento instance as encrypted string and will be forwarded to the Connect Server API upon order placement.

After placing an order with the Ingenico ePayments payment method, the customer is either redirected to the checkout success page or to an external page to enter the payment information. The external page is hosted by Ingenico ePayments.

Afterwards, the customer is redirected back to Magento® and will see the status message from the [Payment statuses frontend output](#) section, depending on the status of the payment.

If everything went right, the payment status will be `PENDING_CAPTURE` or `PENDING_APPROVAL` in case of `Delayed settlement` on [Capture Settings](#). Otherwise `CAPTURE_REQUESTED` or `CAPTURED` indicate a successful transaction.

The next section explains how to deal with those statuses.

### Styling the external payment page

If you want to customize the look and feel of the external payment page, have a look at Ingenico's [Hosted Payment Page documentation](#).

## 4.1.2 Capture (create invoice)

If you are using the default *Delayed Settlement* capture mode (see section [Capture Settings](#)), Orders that are processed with Ingenico ePayments have to be manually captured (approved) by the merchant.

This is done by creating an invoice in Magento®.

### Important: use *online capture*

Make sure that *Capture Online* is selected in the *Amount* box when creating the invoice. Otherwise, the capture will not be registered by the Ingenico ePayments platform.

**If you don't use online capture, the payment flow will be interrupted!**

#### Invoice Totals

Subtotal	€59.00
<b>Grand Total</b>	<b>€59.00</b>

Amount

Capture Online ▼

Append Comments

Email Copy of Invoice

**Submit Invoice**

Creating the invoice in Magento® (with *capture online*) starts the capturing process on the Ingenico ePayments platform. However, the funds will most likely not be transferred immediately, which means:

- the order state will be `PAYMENT_REVIEW`,
- the invoice status will be `PENDING`.

This will change after some time, as soon as Ingenico ePayments reports the capture as complete. When the payment status is set to `CAPTURED` by Ingenico, these things will happen:

- the order state will be set to `PROCESSING`,
- the invoice will be marked as `PAID`,
- the funds are transferred to the merchants account.

From now on, it is possible to perform a *Refund*. This is explained in the next section.

## Refresh payment status

You can quickly "pull" the current payment status from the Ingenico ePayments platform using the *Refresh Payment Status* button on the order details page. See also the section [Order update](#).

### 4.1.3 Refund (create creditmemo)

You can trigger refunds on the Ingenico ePayments platform using Magentos® standard refund function.

To see if a payment can be refunded, you can check its [Transaction information](#). Look at the value *isRefundable* in the table.

Follow these steps to make a refund:

- Go to the order details page.
- **Important:** You must open the invoice you want to refund.
- Then select *Creditmemo* from the top menu.
- Adjust the refund amount if needed, and select *Refund* at the bottom of the page.

#### Refund Totals

Subtotal	€45.00
Refund Shipping	<input type="text" value="0"/>
Adjustment Refund	<input type="text" value="0"/>
Adjustment Fee	<input type="text" value="0"/>
<b>Grand Total</b>	<b>€45.00</b>

Append Comments  
 Email Copy of Credit Memo

Refund Offline
Refund

#### Important: use *online* refunds

Do **not** choose *Refund Offline* when creating a credit memo. Only online refunds will be registered with the ePayments platform.

If you select *Refund offline*, the buyer will not get their money back.

Similar to captures, refunds will not be processed immediately. There will be a slight time delay.

Refunds through the GlobalCollect endpoint must be explicitly approved (confirmed) on the Credit Memo details page.

As long as the refund is in status `PENDING`, it can be canceled on the Credit Memo details page.

#### 4.1.4 Cancel orders

Orders that were created using the Ingenico ePayments module can be cancelled with Magento's® standard cancel function.

If possible, this will cancel the order on the Ingenico ePayments platform as well.

To see if a payment or refund can be cancelled, you can check its [Transaction information](#). Look at the value *isCancelable* in the table.

Denying a *Suspected Fraud* order will also result in a cancellation, as explained in the section [Fraud handling](#).

#### **Difference between Cancel and Void**

For details on what *Cancel* and *Void* mean, have a look at the section [Difference between Cancel and Void operations](#).

## 4.2 Additional features and tips

### 4.2.1 Transaction information

You can look up detailed information about the status of a payment in the Transaction Details:

- Open an order.
- Select *Transactions* from the sidebar.
- Click on a transaction entry in the list.

This shows in-depth details about the transaction status. The data received from the Ingenico ePayments platform is shown in the *Transaction Details* table.

Transaction Details	
Key	Value
status	PAID
isAuthorized	Yes
isRefundable	Yes
isCancellable	No
statusCategory	COMPLETED
statusCode	1050
statusCodeChangeDateTime	20171214201503

Depending on whether you created an order using the GlobalCollect or Ogone platform, you might see multiple transactions for each stage of the payment, or just one *Authorization Transaction* that holds the current status of the payment.

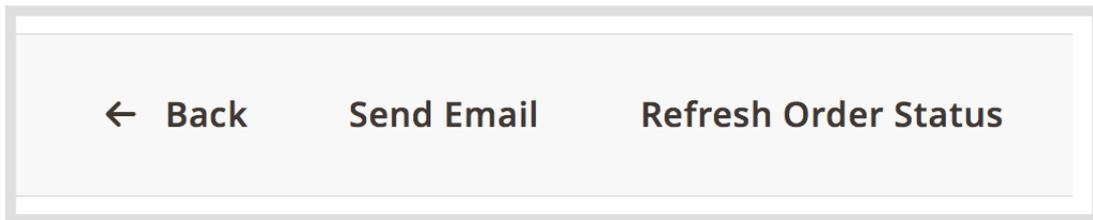
Refunds create separate transactions which show the current status of the refund.

## 4.2.2 Order update

A payment or refund status can get updated automatically via [Webhooks](#). Alternatively, there are these additional ways to get status updates from the Ingenico ePayments platform:

### Update an order manually

Click the *Refresh Payment Status* button at the top of the order details view. You will see a message if any changes were found, and the order status will be updated accordingly.



### Update order automatically

A *Cronjob* is executed every 5 minutes and will query the Ingenico ePayments platform for updates to your orders. The process behind this is similar to the manual order update.

You can configure the Magento® Cron functionality here:

System → Advanced → System → Cron (Scheduled Tasks)

This also lets you configure a log file that will be stored in the Magento® folder `var/log/`. The log shows information about past automatic order updates.

The Cronjob can also cancel unpaid orders that have not been updated in a certain time. See section [Cancellation of Pending Orders](#) on how to configure this feature.

### Update by .wx file polling

If this is enabled, the `.wx` files with order update information are downloaded daily at 4:30am from the Ingenico ePayments platform. See section [`WX file SFTP Settings`](#) on how to configure this feature.

#### Only for Global Collect merchants

Updating the order status by `.wx` file polling is only available if you are using Global Collect.

This will update the order and transaction status, so you are able to handle orders accordingly - however some meta information is not available via WX file update (e.g. if the order can actually be canceled or

refunded through the platform). A relatively empty [Transaction information](#) can indicate an update via WX File. To rectify this you can always use the [Order update](#) functionality.

### 4.2.3 Fraud handling

Ingenico ePayments offers automatic Fraud Detection.

If Ingenico has detected a possible fraud:

- The transaction will show a `FRAUD_SUSPECTED` order status.
- A [Fraud notification](#) email will be sent to the fraud manager.

Payments that are in *Suspected Fraud* can only be processed further after being *approved* or *denied* by the merchant.

To approve or deny a payment, go to the order details page and use the buttons in the top menu.



The result will be:

- A denied payment will cancel the order, as explained in the section [Cancel orders](#).
- An approved payment will make the order ready for [Capture \(create invoice\)](#).

#### 4.2.4 E-Mail Notifications

You can configure if and when to inform a customer about the order status.

This can be used to keep customers up-to-date and manage their expectations. Please see the section [Customer notification Email Settings](#) on how to use this feature.

## 5 Uninstalling or disabling the module

To **uninstall** the module, execute this command from your Magento 2 installation's root directory:

```
php ./bin/magento module:uninstall module-epayments-m2
```

If you only want to **disable** the module without uninstalling it, execute this command:

```
php ./bin/magento module:disable module-epayments-m2
```

## 6 Troubleshooting

### 6.1 Payment status not accurate

If you believe that the payment status of an order isn't correct or up-to-date, you should first try to refresh the order status with the *Refresh Payment Status* button (see section [Order update](#)).

If you need more information for debugging the problem, you can check the [Transaction information](#) from the order details page. Select a transaction and look at the *Transaction Details* received from the Ingenico ePayments platform.

Transaction Details	
Key	Value
status	PAID
isAuthorized	Yes
isRefundable	Yes
isCancellable	No
statusCategory	COMPLETED
statusCode	1050
statusCodeChangeDateTime	20171214201503

## 6.2 Check log file

If you enabled logging in section [Account Settings](#), you can download the log file using the link:

Stores → Configuration → Ingenico ePayments → Account Settings → Log All Requests → Download log file

The log file contains all messages to and from the Ingenico servers and may help finding the cause.

## 6.3 Technical support

In case of questions or problems, have a look at the Support Portal (FAQ):

<http://ingenico.support.netresearch.de/>

If the problem cannot be resolved, you can contact the Ingenico support team:

<https://www.ingenico.com/epayments/support>