

Integration Guide Shopware 6

1

Requirements

An active PAYONE - Account is required. If you do not have one yet, please [contact us](#).

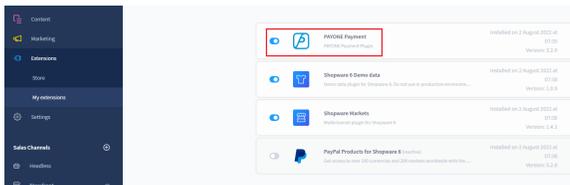
2

Plugin Installation

You have 3 ways to install our plugin:

Installation directly in Shopware - Admin

1. "Buy" the free PAYONE plugin in the Shopware Store.
2. Download our plugin in your shop under *Extensions My extensions*.
3. Install and activate the plugin.



Installation via Composer

```
composer require payone-gmbh/shopware-6
php bin/console plugin:install PayonePayment
php bin/console plugin:activate PayonePayment
php bin/console cache:clear
```

Installation via Github

Down

3

PAYONE - Portal configuration

To send the transaction status to the correct address, the log in to the PAYONE Merchant Interface (PMI).

Under *Configuration Payment Portals YOUR_PORTAL Advanced TransactionStatus URL*

<SHOPURL>/payone/webhook

Quick Facts

Payment methods + Features

Payment methods

Features

Transactionstatus URL:

<SHOPURL>/payone/webhook

Latest Release

v3.3.0



Shopware 6
Plugin: Release v3.3.0
PAYONE Admin posted on
Sep 15, 2022
New Features

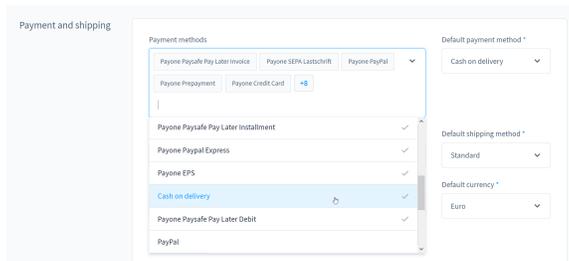
- New payment method: Ratepay Open Invoice
- New payment method: Ratepay Direct Debit
- New payment method: Ratepay Installments

Maintenance

- Integrate a sales landingpage in backend
- Tested with version 6.4.14
- [integrations](#)
- [shopware6](#)

Download

Download from



- Each payment method can be used with separate access data. These can be entered directly in the payment type.

Special adjustments

Some payment methods require special adjustments. The next points are only relevant to you if you use the specific payment method:

PAYONE Secure Invoice

PAYONE Secure Invoice

This payment method requires its own payment portal. Please enter a separate portal ID under *Settings > Extensions > PAYONE > PAYONE Secure Invoice*

A rule created during the plugin installation for the Rule Builder ensures that the payment method can only be used for the currency "Euro". If you have not installed the currency "Euro" in your store during the plugin installation, a random, non-existing currency ID will be linked to the condition, so that the rule will always fail. If you then create the currency "Euro" afterwards, you have to select the currency "Euro" in the condition "Currency is one of" under *Settings > Rule Builder > Payone secure invoice*.

PAYONE Prepayment

PAYONE Prepayment

To inform customers about the bank details, we have included a block with billing data in the invoice template. It makes sense to include this data in the order confirmation email template as well. We do not have a detailed instruction available currently.

PAYONE Unzer Rechnungkauf + PAYONE Unzer Ratenkauf

PAYONE Unzer Rechnungkauf + PAYONE Unzer Ratenkauf

To be able to offer Unzer, the payment methods must be additionally configured in the backend:

- **PAYONE Unzer Rechnungkauf**

Under *Settings > Extensions > PAYONE > PAYONE Unzer Rechnungkauf* the following data must be stored: Company name (for the supplementary T&C), IBAN and BIC (both will be displayed next to the payment reference). Activate the B2B slider if you want to accept B2B orders.

Company Name ?

IBAN ?

BIC ?

Sets the b2b flag if the customer's company name is present.

- **PAYONE Unzer Ratenkauf**

Under *Settings Extensions PAYONE PAYONE Unzer Ratenkauf* the following data must be stored: Company name (for the supplementary GTC), channel name and channel password (both you get from Unzer).

Company Name ?

Channel Name ?

Channel Password ?

Apple Pay

PAYONE Apple Pay

Under *Settings Extensions PAYONE PAYONE Apple Pay* the following data must be stored: Supported Cards, ApplePay Merchant ID, ApplePay Display Name und Certificate Passphrase (the last three values you get from Apple).

Create the following files in the config folder: *merchant_id.pem* and *merchant_id.key*.

You will receive the content of the files in the process of [Apple Onboardings](#) under point 3.

In addition, a subfolder *.well-known* must be created in the *public* folder. In this subfolder please put the file: *apple-developer-merchantid-domain-association.txt*

You will receive the contents of the file after domain certification from Apple.

Supported Cards ?

ApplePay Merchant ID

ApplePay Display Name ?

The ApplePay merchant authentication requires a certificate/key-pair. Further information: <https://docs.payone.com/display/public/PLATFORM/Special+Remarks+-+Apple+Pay+Special+Remarks+ApplePay+Onboarding>

Create a pem-File afterwards by using the following command:

```
openssl x509 -inform der -in merchant_id.cer -out merchant_id.pem
```

Copy certificate (**merchant_id.pem**) and key (**merchant_id.key**) file into the following folder:

```
%shopwareRoot%/config/apple-pay-cert
```

Certificate Passphrase

Apple Pay is only visible and usable on Apple devices.

In addition to the configuration in Shopware, it is also necessary to set up the corresponding certificates to operate Apple Pay as a payment method. You can find instructions for this here: [Special Remarks - Apple Pay](#)

Additionally, Apple requires validation of the store domain for Apple Pay. To do this, follow the instructions in the Apple Developer Portal:

Apple Pay Payment Processing on the Web

To configure Apple Pay Payment Processing on the web for this merchant ID, you must register and verify the domains that will process transactions. You must also create a Apple Pay Merchant Identity, which authenticates your web sessions with the Apple Pay Payment Processing servers.

Incorporation of Apple Pay Payment Processing into your website is subject to these [Apple Pay Payment Processing Web Merchant Terms and Conditions](#) and [Acceptable Use Guidelines](#). Failure to comply with any of these Terms and Conditions or guidelines may result in deactivation of Apple Pay Payment Processing transactions on your website.

Merchant Domains

Domain: [redacted]	Remove	Verify
Status: Verified		
Verification Expires: Dec 22, 2021		
Domain: [redacted]	Remove	Verify
Status: Verified		
Verification Expires: Dec 22, 2021		
Domain: [redacted]	Remove	Verify
Status: Verified		
Verification Expires: Jan 24, 2022		
Domain: [redacted]	Remove	Verify
Status: Verified		
Verification Expires: Jan 8, 2022		
Domain: [redacted]	Remove	Verify
Status: Verified		
Verification Expires: Dec 26, 2021		

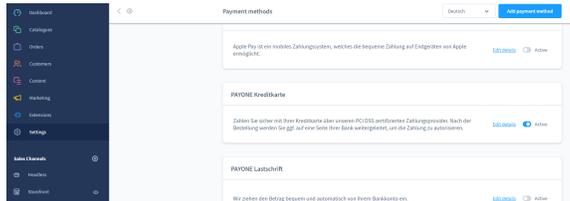
Add a domain for use with this Merchant ID.

Add Domain

5

Payment methods setup

Activate the requested payment methods under *Settings Shop Payment methods*



6

Editing the orders

The PAYONE plugin sets the payment status of an order according to the set status assignment.

Order number	Sales channel	Customer name	Billing address	Amount	Order status	Payment status	Order date
10001	10001	Paul Tester	Testweg 123, 24110 Kiel	2.000,00 €	offen	offen	2019-12-18T18:28:58+00:00
10002	10001	Paul Tester	Testweg 123, 24110 Kiel	400,00 €	offen	offen	2019-12-18T18:27:13+00:00

Capture

Unauthorized transactions can be captured using the "Capture" button in the order details.

PAYONE

TXID
379606102

Sequence Number
0

State
appointed

Capture

Refund

Refund

The "Refund" button can be used to refund an order (in the "paid" status).

PAYONE

TXID
379606102

Sequence Number
1

State
capture

Capture

Refund

Partialcapture and -refund

You can partially enter or refund an order by selecting an order item and editing the quantity. The entry amount is adjusted dynamically.

If you want to fully capture or refund the order, click "Complete Capture".

You can also directly enter the requested amount in the "Capture amount" field.

Captures and refunds are only possible after the store has received a transaction status for an order, therefore, the buttons for captures and refunds are grayed out directly after an order.

Capture

<input type="checkbox"/>	Product	Quantity	Price
<input type="checkbox"/>	Hauptprodukt mit erweiterten Preisen	3	2.400,00 €
<input checked="" type="checkbox"/>	Hauptartikel	2	991,90 €

Order amount
3.391,90 €

Captured amount
0,00 €

Remaining amount
3.391,90 €

Capture amount *
991,9

Close Capture Full capture

7

Extended configurations

Here you will find separate setting options that allow you to customize our plugin more precisely to your needs.

Detailed status mapping (global)

Detailed status mapping (global)

Under *Settings Extensions PAYONE Status Mapping* you can individually assign the states to which the orders should be set as soon as the respective information is sent by the PAYONE platform.

Status Mapping

Status for Appointed	Offen
	Offen
	Bezahl
	Erinnert
	Erstattet
	Teilweise erstattet
	Teilweise bezahlt
	Abgebrochen
Status for Cancellation	Offen
Status for Refund	Offen
Status for Debit	Offen

Detailed status mapping (by payment method)

Detailed status mapping (by payment method)

Under *Settings Extensions PAYONE* you can individually assign the states for every payment method separately. Please choose the payment method and activate "Display state mapping configuration".

Display state mapping configuration

Creditcard - CSS adjustments

Creditcard - CSS adjustments

The following customizations are not necessary to offer credit card payments. The possibility to customize the payment type visually in the frontend is described in the following:

You can change the CSS by creating a new plugin that depends on this plugin via Composer. Here you can then define an overriding JavaScript plugin like this one:

```
import PayonePaymentCreditCard from '../../../../../../../../PayonePayment/src/Resources/storefront/credit-card/payone-payment.credit-card';

export default class MyPayonePaymentCreditCard extends PayonePaymentCreditCard
{
  getFieldStyle() {
    const style = super.getFieldStyle();

    style.push('height: 300px');

    return style;
  }
  getSelectStyle(){
    const style = super.getSelectStyle();

    style.push('background-color: black');

    return style;
  }
}
```

Then you can register the overriding plugin in your `custom/plugins/MyPlugin/src/Resources/storefront/main.js`:

```
import MyPayonePaymentCreditCard from './my-payone-payment
/my-payone-payment.credit-card.plugin';

const PluginManager = window.PluginManager;
PluginManager.override('PayonePaymentCreditCard',
MyPayonePaymentCreditCard, '[data-is-payone-credit-card]');

// Necessary for the webpack hot module reloading server
if (module.hot) {
  module.hot.accept();
}
```

Right management

Right management

If you use Shopware's rights management, please give permission to all users. Under *Settings System Users&permissions Role* create a new role or update an existing role. Activate the PAYONE transaction management right.

PAYONE

PAYONE Transaktionsmanagement
