



Cardlink Checkout Payment Gateway v.1.3.0
for Prestashop 1.7.x-9.x

Date: 14/06/2026

Cardlink Checkout Payment Gateway

Contributors: cardlink

Tags: payments, payment-gateway

Requires at least: 1.7.x

Tested up to: 9.x

Requires PHP: 7.x - 8.x

License: GPLv2 or later

License URI: <http://www.gnu.org/licenses/gpl-2.0.html>

Changelog

- **1.3.0**
 - Compatibility with version 9.x.
- **1.2.0**
 - Support for background confirmations.
 - Support secondary transactions (capture, refund, cancel) through VPOS XML API calls.
- **1.1.3**
 - Remove requirement for IRIS customer code configuration setting. If you already use IRIS with your own seller ID, DO NOT install the new version!
- **1.1.2**
 - Add Cardlink and Worldline as IRIS business partners.
 - Support MySQL versions older than 5.6.5.
 - Fix guest checkout redirect to order confirmation bug.
 - Tested compatibility with version 8.2.
- **1.1.1**
 - Minor fixes.
 - Added translations.
 - Update README.
- **1.1.0**
 - Added support for IRIS payments.
- **1.0.14**
 - Changed flow to create orders only after successful payment.
 - Removed unpaid order cancellation cron job.
- **1.0.13**
 - Minor bug fixes.
- **1.0.12**
 - Support backorder related order states for sending order confirmation/notification emails only when payment is successful.
- **1.0.11**
 - Enforced creation of pending credit card payment order state.
- **1.0.10**
 - Minor bug fixes.
- **1.0.9**
 - Minor bug fixes.
- **1.0.8**
 - Bug fixes in checkout response controller.

- **1.0.7**
 - Bug fixes.
 - Validate Alpha Bonus digest.
- **1.0.6**
 - Removed billing state from data sent to the payment gateway.
- **1.0.5**
 - Fixed data-module-name value in checkout.
- **1.0.4**
 - Added payment success/denied/canceled/error pages.
- **1.0.3**
 - Added support for Prestashop version 8.0.x, added missing order state, fixed minor bugs.
- **1.0.2**
 - Added hook to disable sending order confirmation email to customer for the waiting payment state.
- **1.0.1**
 - Added settings for order states after successful capture/authorize payments.
 - Added cron job script and HTTP endpoint for canceling orders that have been abandoned (left in waiting for payment for more than 1 hour).
- **1.0.0**
 - Initial release

Support tickets

If you encounter any technical issues during the installation process, please contact the Cardlink e-commerce team at ecommerce_support@cardlink.gr.

Description

Cardlink Payment Gateway allows you to accept payment through various schemes such as Visa, Mastercard, Maestro, American Express, Diners, Discover cards on your website, with or without variable installments.

This module aims to offer new payment solutions to Cardlink merchants for their PrestaShop 1.7.x online store without requiring web development knowledge. However, for the initial module installation some technical knowledge will be required.

Merchants with e-shops will be able to integrate the Cardlink Payment Gateway to their checkout page using the CSS layout that they want. Also, they can choose between redirect or IFRAME option for the payment environment. Once the payment is made, the customer is returned to the online store and the order is updated.

Once you have completed the requested tests and any changes to your website, you can activate your account and start accepting payments.

Features

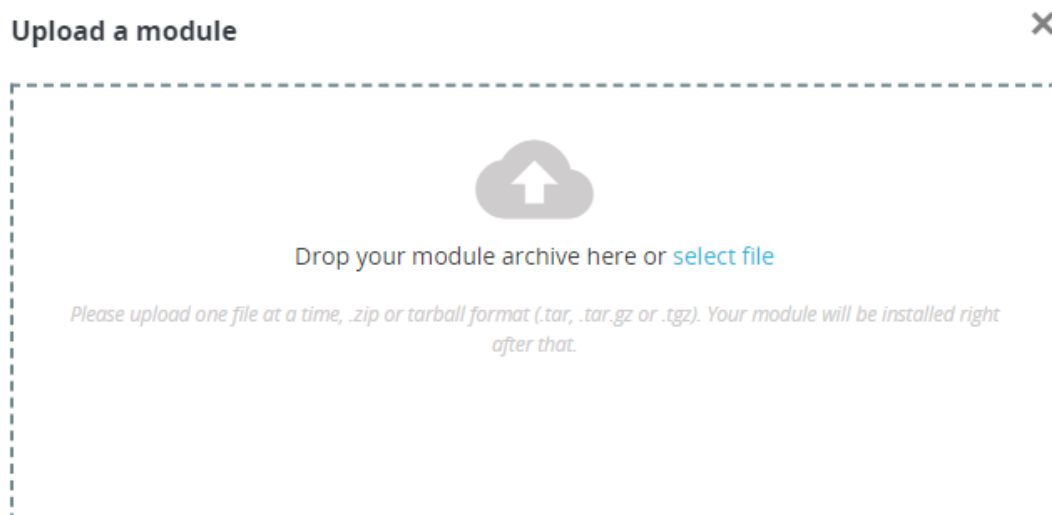
1. A dropdown option for instance between Worldline, Nexi and Cardlink.
2. Option to enable test environment (sandbox). All transactions will be re-directed to the endpoint that represents the production environment by default. The endpoint will be different depending on which acquirer has been chosen from instance dropdown option.

3. Ability to define the maximum number of installments regardless of the total order amount.
4. Ability to define the ranges of the total order amounts and the maximum number of installments for each range.
5. Option for pre-authorization or sale transactions.
6. Option to set different order statuses for transactions with a captured or authorized payment.
7. Option for a user tokenization service. The card token will be stored at the merchant's e-shop database and will be used by customers to auto-complete future payments.
8. In-store checkout option: the merchant can configure the payment process to take place inside a pop up with IFRAME to mask the redirection process from the customers.
9. A text field for providing the absolute or relative (to Cardlink Payment Gateway location on server) URL of custom CSS stylesheet, to apply custom CSS styles in the payment page.
10. Translation ready for Greek & English languages.
11. Support for IRIS payments.
12. The IFRAME feature is not supported for IRIS payments.
13. Background payment confirmation webhook for reliable order creation when the customer closes the browser before returning from the payment gateway.
14. Secondary transaction support (capture, void, refund) directly from the order page in the administration panel.

Installation

Download the archive at the [Releases page](https://github.com/Cardlink-SA/cardlink-payment-gateway-prestashop/releases).

From your administration, go to **Modules > Module Manager** and click on the **Upload a module** button. A popup will appear.



Drag and drop your **.zip** file or find it and select it through the **select file** link. Once the module's **.zip** file has been uploaded and the module installed, you will be asked if you want to configure the module.

Administration

Recommended Modules and Services

Help

General

- * Automatically check for module updates Yes
Choose a stability level for the modules downloaded from the Addons Marketplace. All zips pushed on Addons are in stable state unless stated otherwise.
- * Check the cookie's IP address Yes
Check the IP address of the cookie in order to prevent your cookie from being stolen.
- * Lifetime of front office cookies
Set the amount of hours during which the front office cookies are valid. After that amount of time, the customer will have to log in again.
- * Lifetime of back office cookies
When you access your back office and decide to stay logged in, your cookies lifetime defines your browser session. Set here the number of hours during which you want them valid before logging in again.
- * Cookie SameSite
Allows you to declare if your cookie should be restricted to a first-party or same-site context.

[Save](#)

Cookie Security Settings

For security reasons, Web browsers will not send target domain cookies when the referrer website is on another domain and data are POSTed unless the SameSite option of these cookies is set to the value None. If you fail to properly configure the required hosting settings, customers returning from the payment gateway will be automatically logged out from their accounts. The following configuration instructions will manipulate all cookies set by your Prestashop store to allow customer sessions to persist after returning from the payment gateway. If you fail to properly set these, customers returning from the payment gateway will be automatically logged out from their accounts.

Through the administration panel

This setting is currently available in Prestashop version 1.7.8.7. If this setting is not available in your installed Prestashop version, follow the manual method. From your administration panel, go to [Configure > Advanced Parameters > Administration > SameSite cookies](#) and change it from [Strict](#) or [Lax](#) to [None](#). You will need to be running your entire store under SSL protection otherwise, the panel will inform you of the error.

Manually

Apache Web Server

For hosting solutions running the Apache web server software, you will need to add the following lines to your web site's root `.htaccess` file. Make sure the `mod_headers` Apache module is installed and active.

```
<IfModule mod_headers.c>
Header always edit Set-Cookie ^(.*)$ $1;SameSite=None;Secure
</IfModule>
```

Nginx Web Server

If your hosting provider uses the Nginx web server instead, you will need to add/edit the following lines of code to your virtual host's configuration file.

```
location / {
    proxy_cookie_path / "/; SameSite=None; Secure";
    ...
}
```

Plesk Hosting Control Panel

If you are using Plesk and nginx in proxy mode, under `Apache & nginx Setting for ... > Additional nginx directives` add only the following line:

```
proxy_cookie_path / "/; SameSite=None; Secure";
```

If you are only using Apache, add the following configuration lines in the `Additional Apache directives` section on the same page. By default, Plesk has the Apache `mod_headers` module installed and active; however, verify that this is the case for your Plesk installation.

```
<IfModule mod_headers.c>
Header always edit Set-Cookie ^(.*)$ $1;SameSite=None;Secure
</IfModule>
```

If you are unsure or unfamiliar with the actions described above, please ask a trained IT person or contact your hosting provider to do them for you.

Screenshots

15. The Cardlink Payment Gateway settings screen used to configure the main Cardlink gateway (`Improve > Modules > Module Manager > Cardlink Checkout > Configure`). The bottom panel is only visible when the `Accept Installments` configuration option is set and stored to `Based on Order Amount`.

PrestaShop 1.7.8.7 Quick Access View my shop

Modules / cardlink_checkout / Configure

Configure
Cardlink Checkout

Back Translate Check update Manage hooks

Settings

* Title

The title of the payment method to be displayed during the checkout.

Description

A short description of the payment method to be displayed during the checkout.

* Business Partner

Identify the business partner that will handle payment transactions as agreed with Cardlink.

* Merchant ID

The merchant ID provided by Cardlink.

* Shared Secret

The shared secret code provided by Cardlink.

Transaction Environment

Identify the working environment for payment transactions.

Payment Action

Identify the type of transaction to perform. By selecting the "Authorize" option, you will need to manually capture the order amount on Cardlink's merchant dashboard.

Accept Installments

Enable installment payments and define the maximum number of installments.

Fixed Maximum Installments

The maximum number of installments available for all orders.

Allow Tokenization

Allow customers to select whether they want to secure store their payment cards for future checkouts.

Checkout without Leaving Your Store

Perform the payment flow without having the customers leave your website for Cardlink's payment gateway. You will need to have a valid SSL certificate properly configured on your domain.

Force Store Language

Instruct Cardlink's Payment Gateway to use the language of the store that the order gets placed.

Display Cardlink Logo


Display the Cardlink logo next to the payment method title.

CSS URL

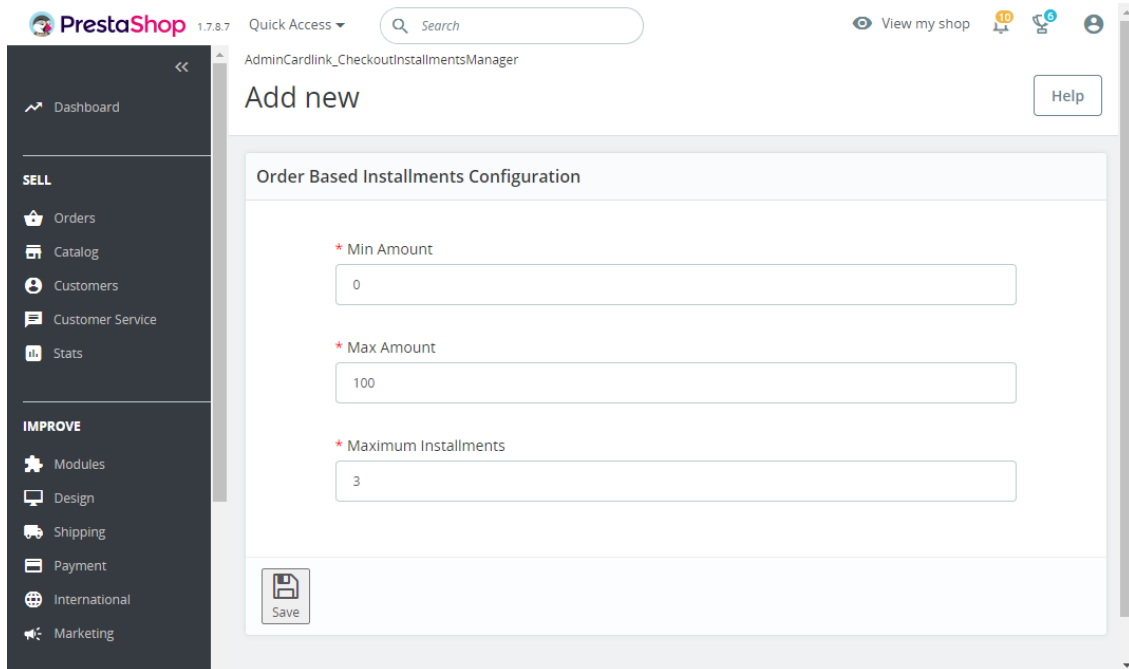
Full URL of custom CSS stylesheet, to be used to display payment page styles.

Save

Order Amount Range Installments

Min Amount	Max Amount	Maximum Installments
 No records found		

16. This is the admin page for configuring an order amount range to apply a specified maximum number of installments limit (only accessible when the “Accept Installments” configuration option is set to “Based on Order Amount”). A range of minimum to maximum order amount can be set that will limit the maximum selectable number of installments for the order.



17. This is the front-end of Cardlink Payment Gateway plugin located in checkout page.

✓ PERSONAL INFORMATION ✎ edit

✓ ADDRESSES ✎ edit

✓ SHIPPING METHOD ✎ edit

4 PAYMENT

Pay through Cardlink



Pay Via Cardlink: Accepts Visa, Mastercard, Maestro, American Express, Diners, Discover.

Installments

No Installments ▾

Stored Cards



xxxx-xxxx-xxxx-0011

05/2025 - [Remove](#)

New Card

I agree to the [terms of service](#) and will adhere to them unconditionally.

PLACE ORDER

18. To set up IRIS payments, you will need to have the Merchant ID and Shared Secret specifically issued for use with IRIS. Other settings are similar to the ones for Card Payments.

Pay through IRIS

Pay through IRIS	<input type="text" value="Enabled"/>	
	<i>Allow customers to pay using IRIS.</i>	
* Title	<input type="text" value="Pay through IRIS"/>	<input type="text" value="en"/>
	<i>The title of the payment method to be displayed during the checkout.</i>	
Description	<input type="text" value="Pay via your bank's web banking application."/>	<input type="text" value="en"/>
	<i>A short description of the payment method to be displayed during the checkout.</i>	
* IRIS Business Partner	<input type="text" value="Nexi"/>	
	<i>Identify the business partner that will handle payment transactions as agreed with Cardlink.</i>	
* IRIS Merchant ID	<input type="text" value="70000186"/>	
	<i>The merchant ID provided by Cardlink.</i>	
* IRIS Shared Secret	<input type="text" value="Cardlink1"/>	
	<i>The shared secret code provided by Cardlink.</i>	
Transaction Environment	<input type="text" value="Sandbox"/>	
	<i>Identify the working environment for payment transactions.</i>	
Seller ID	<input type="text" value="90002"/>	
	<i>The seller ID for the DIAS network.</i>	
Display IRIS Logo	<input type="text" value="Display Logo"/>	
	<i>Display the IRIS logo next to the payment method title.</i>	
CSS URL	<input type="text"/>	
	<i>Full URL of custom CSS stylesheet, to be used to display IRIS payment page styles.</i>	

Background Confirmations

Background confirmations allow the payment gateway to notify the store directly when a payment result is available, independently of the customer's browser session. This ensures that orders are created even when a customer closes the tab or loses connectivity before being redirected back from the payment gateway.

Webhook URL

Configure the following URL as the **Background Confirmation URL** in your Cardlink / Worldline / Nexi merchant dashboard:

```
https://{your-store-domain}/index.php?fc=module&module=cardlink_checkout&controller=backgroundconfirmation
```

Replace `{your-store-domain}` with your actual store domain. The endpoint accepts **POST** requests and must be reachable from the payment gateway servers over HTTPS in production.

How It Works

19. After processing a payment, the gateway sends a signed **POST** request with the transaction result to the webhook URL.
 20. The module validates the request:
 - The **User-Agent** header must identify the gateway client (**Modirum HTTPClient**).
 - The HMAC digest included in the request body is verified against the configured shared secret.
 - The merchant ID in the request must match the configured merchant ID.
 - In production, requests over plain HTTP are rejected.
 21. If a PrestaShop order already exists for the cart (e.g. the customer returned in time), the webhook updates the order state if needed and exits without creating a duplicate order.
 22. For **AUTHORIZED** or **CAPTURED** transactions where no order exists yet, the module creates the order, records the transaction details, and returns an HTTP 200 JSON response to acknowledge receipt.
 23. For failed transactions (**CANCELED**, **REFUSED**, **ERROR**) no order is created.
- The endpoint returns a JSON response. On success the HTTP status is **200**; on validation failure it returns **403**.

Secondary Transactions

For orders that were placed using a **pre-authorization** (the payment was authorized but not immediately captured), the module adds a panel to the order detail page in the administration area that allows merchants to perform the following operations:

Operation	Description	Available when
Capture	Finalises the pre-authorization and transfers the funds. Partial amounts are supported.	Payment status is AUTHORIZED .
Void	Cancels the pre-authorization and releases the hold on the cardholder's account without transferring funds.	Payment status is AUTHORIZED .
Refund	Returns captured funds to the cardholder. Partial refunds are supported.	Funds have been captured. Cannot exceed the total captured amount minus any prior refunds.

Where to Find It

In the administration panel, open any order paid through Cardlink Checkout. The **Cardlink Payment Actions** panel appears below the order details. It shows the current transaction status, the captured and refunded amounts, and input fields for the desired action and amount.

Behavior Notes

- All operations call the Cardlink / Worldline / Nexi VPOS XML API and the order state is updated in PrestaShop to reflect the result (**Payment accepted**, **Cancelled**, or **Refunded**).
- The correct credentials (Merchant ID, Shared Secret, acquirer) are determined automatically from the payment method recorded on the original transaction (card, IRIS).
- A successful capture creates a payment record and links it to the order invoice.
- A successful full refund creates a credit slip in PrestaShop.
- If the gateway voids a transaction instead of refunding it (because it has not yet settled), the order is marked as cancelled and the response message indicates that a void was performed.