



easybill Integration for WooCommerce Version 1.0.0

User Documentation

TABLE OF CONTENTS

Overview	3
What It Does	3
How the Plugin Works	4
Quick Setup	5
Install	5
Upload and Activate	6
Upload	6
Activate	6
Configuration	7
The Transfer Flow	9
Customer Creation & Linking	11
Invoice Mapping	12
PDF: Storage, Email & Download	13
Manual Resend & Error Handling	15
Architecture (SPBILL Core)	16
Notes From the easybill API	17
Filters	18
Hooks for Developers	19
File Map	19
Troubleshooting	20
Requirements & Compatibility	21
Roadmap	Fehler! Textmarke nicht definiert.
Changelog	22
1.0.0 – 2026-06-30	22

OVERVIEW

easybill Integration for WooCommerce transfers WooCommerce orders to easybill via its REST API after checkout: it creates the customer, creates and finalizes the invoice, fetches the PDF, stores it on the order and sends it to the buyer.

It is built on a small, reusable REST billing-connector core (SPBILL) designed to grow into a family of source $\rightleftarrows$ billing-platform connectors — further platforms (Fastbill, Lexware, WISO, Papierkram, SevDesk) and further sources (GravityForms, MemberPress) can reuse the same abstraction.

Version	1.0.0 (V1)
Author	Ralf Fuhrmann
License	GPL-2.0-or-later
Requirements	WordPress 6.9+, WooCommerce 8.0+ (tested up to 10.9), PHP 7.4+
Also needs	An easybill account with an API key (Bearer token)
Languages	English and German (fully translated)

WHAT IT DOES

When a WooCommerce order reaches the configured status, the plugin:

- creates an easybill customer from the buyer's billing data (and links it to the WordPress user for logged-in buyers),
- creates an invoice document from the order items,
- finalizes it so it receives its invoice number,
- downloads the PDF, stores it on the order and keeps a link to the easybill document,
- attaches the PDF to the configured WooCommerce customer email.

HOW THE PLUGIN WORKS

- Trigger — an order reaches the configured status (default processing).
- Customer — the plugin creates / reuses the easybill customer from the buyer data.
- Invoice — it creates an invoice from the order items and finalizes it (invoice number).
- PDF — it downloads the PDF, stores it on the order, keeps the easybill link and number.
- Email — the PDF is attached to the WooCommerce customer email sent to the buyer.

Built on a reusable “SP REST Billing” core: a generic REST client, an abstract billing provider and provider-agnostic data objects. New billing platforms and new sources plug in without rewriting the flow.

QUICK SETUP

INSTALL

- Copy the woocommerce_rest-easybill folder to wp-content/plugins/.
- Make sure WooCommerce is already installed and active.
- Activate “easybill Integration for WooCommerce”.
- Configure the plugin under WooCommerce → Settings → EasyBill (see below).

You will need an easybill account and an API key (easybill → Settings → API) before you configure the plugin.

UPLOAD AND ACTIVATE

UPLOAD

After you have downloaded the 'easybill Integration for WooCommerce' plugin, you need to upload it to your WordPress installation.

Please go to your 'Plugins / Add New' menu, select 'Upload Plugin' from the top links and choose the downloaded ZIP file at the upload field.

After selecting the file, please press the "Install Now" button.

ACTIVATE

After the plugin is uploaded you will get a success screen.

Please use the link "Activate Plugin" to activate the plugin.

CONFIGURATION

Go to WooCommerce → Settings → EasyBill:

 Screenshot: The "EasyBill" settings tab under WooCommerce → Settings

EasyBill REST Connection

After an order reaches the selected status, it is transferred to easybill to create an invoice. The PDF is fetched, stored on the order and attached to the customer email.

Enable transfer ☒ Automatically create an easybill invoice for new orders.

API key (Bearer token) 

Trigger order status  

Finalize invoice ☒ Mark the document as "done" so it receives its final invoice number (recommended). If disabled, a draft is created.

Attach PDF to email  

Enable logging ☒ Write API errors and results to the WooCommerce logs (WooCommerce → Status → Logs, source "wcreb").

Save changes

Setting	Description
Enable transfer	Turn the automatic invoicing on/off.
API key (Bearer token)	Your easybill API key (easybill → Settings → API). Sent as Authorization: Bearer ...
Trigger order status	The status at which the invoice is created (default processing).
Finalize invoice	Mark the document done so it gets its final number (recommended). If off, a draft is created.
Attach PDF to email	The WooCommerce customer email the PDF is attached to (or do not attach).

Enable logging

Log API errors/results to WooCommerce →
Status → Logs (source wcneb).

THE TRANSFER FLOW

The flow runs on the `woocommerce_order_status_{trigger}` action at an early priority, so the order meta and the PDF are ready before WooCommerce sends the matching transactional email. Each step is guarded; an error is logged and added as an order note without breaking the order or email processing.

- Order reaches the configured status.
- Ensure customer — create or reuse the easybill customer.
- Create invoice — from the order items.
- Finalize — the document receives its invoice number.
- Fetch PDF — store it on the order and keep the easybill link and number.
- Attach — the PDF is attached to the WooCommerce customer email.

CUSTOMER CREATION & LINKING

The buyer's billing data is mapped to an easybill customer. easybill requires both a `last_name` and a `company_name`; for private buyers without a company the plugin falls back to the person's name (and vice-versa), so customer creation never fails.

- For logged-in buyers, the resulting easybill customer id is stored as user meta `_easybill_customer_id` and reused on future orders (no duplicate customers).
- The id is also stored on the order (`_easybill_customer_id`).

V1 reuses an existing linked customer as-is and does not overwrite remote customer data; a future version may push updates.

INVOICE MAPPING

- Each line item becomes an invoice position: description = item name, number = SKU, quantity, net unit price (after discounts) and the VAT percentage derived from the line tax.
- Shipping and fees are added as positions with their own net price and VAT percentage.
- Coupon discounts are already reflected in the per-line net totals, so the invoice matches the order total.
- easybill computes the document totals from the positions; the order id is stored as the document external_id and the order number as order_number.

PDF: STORAGE, EMAIL & DOWNLOAD

- The PDF is downloaded as binary application/pdf and stored under wp-content/uploads/easybill-invoices/ in a folder protected by .htaccess + index.php.
- It is attached to the configured WooCommerce customer email.
- The order details page (thank-you page / My Account) shows the invoice number and a key-protected download link; access is granted by the WooCommerce order key, by the order owner, or by a shop manager.
- The order keeps _easybill_document_id, _easybill_invoice_number, _easybill_pdf_path and _easybill_invoice_url.

 Screenshot: The invoice number and secure download link on the order details page

specialPress - Weilerweg 31 - 53881 Euskirchen

Ralf Fuhrmann
Ralf Fuhrmann
Weilerweg 31
53881 Euskirchen
DE-NW

So erreichen Sie uns

E-Mail admin@specialpress.de
 Telefon 015780812732

USt-IdNr. **DE253286535**

Datum **02.07.2026**
 Kunde **10001**
 Rechnung **202610001**

Order 119

Rechnung 202610001

Das Rechnungsdatum entspricht dem Leistungsdatum

Pos	Art-Nr.	Bezeichnung	Menge	Einzelpreis	Betrag
1	woo-fashion-jacket	Jacket	1	0,25 €	0,25 €
Nettobetrag					0,25 €
Umsatzsteuer 0%					0,00 €
Rechnungsbetrag					0,25 €

Vielen Dank für Ihren Auftrag!

Bitte begleichen Sie den offenen Betrag bis zum 16.07.2026.

Mit freundlichen Grüßen

specialPress

MANUAL RESEND & ERROR HANDLING

- The order edit screen offers an order action “Create / resend easybill invoice”.
- The flow is idempotent: once the document id is stored it is never created twice — a resend only (re)fetches the PDF for the existing document.
- Errors (missing API key, API/HTTP errors, invalid PDF) are written to the WooCommerce log (source wcweb) and added as an order note. HTTP 429 (rate limit) is retried with a short back-off.



Screenshot: The “Create / resend easybill invoice” order action

ARCHITECTURE (SPBILL CORE)

This plugin is the first of a planned series connecting source tools to billing platforms. The reusable part lives in `includes/lib/` as the SP REST Billing core:

Class	Responsibility
SPBILL_REST_Client	Generic JSON REST client (bearer auth, 429 handling, binary downloads)
SPBILL_Provider (abstract)	The billing-platform contract: <code>ensure_customer</code> , <code>create_invoice</code> , <code>finalize_invoice</code> , <code>get_invoice_pdf</code> , <code>get_invoice_url</code>
SPBILL_Customer / SPBILL_Invoice / SPBILL_Invoice_Item	Provider-agnostic DTOs

- A provider implements the contract for one platform — here `SPBILL_Provider_EasyBill`.
- A source adapter maps one source to the DTOs — here `WCREB_Source_WooCommerce` (a WooCommerce order).
- The orchestrator `WCREB_Plugin` drives the flow and the WooCommerce integration.

Adding `Fastbill` / `Lexware` / `SevDesk` / ... = a new provider. Adding `GravityForms` / `MemberPress` / ... = a new source adapter. Both reuse the DTOs and the REST client.

NOTES FROM THE EASYBILL API

Derived from the bundled swagger.json (REST API 1.98.x):

- Auth: Authorization: Bearer <API key>, base <https://api.easybill.de/rest/v1>.
- Customer requires last_name and company_name → handled by the B2C fallback above.
- Invoice numbering only happens on finalize (PUT /documents/{id}/done) — hence the Finalize invoice setting.
- PDF is binary (GET /documents/{id}/pdf).
- Rate limiting (429) is handled with a short retry/back-off.
- EU OSS (is_oss, vat_country, ...) and credit notes / cancellations are out of scope for V1 (see the roadmap).

FILTERS

- `wcreb_customer` (SPBILL_Customer, WC_Order) — adjust the customer DTO.
- `wcreb_invoice` (SPBILL_Invoice, WC_Order) — adjust the invoice DTO.
- `wcreb_vat_id_meta_keys` (array) — order meta keys checked for the buyer VAT id.
- `spbill_easybill_customer_payload` (array, SPBILL_Customer) — adjust the easybill customer payload.
- `spbill_easybill_document_payload` (array, SPBILL_Invoice) — adjust the easybill document payload.

HOOKS FOR DEVELOPERS

FILE MAP

File	Purpose
woocommerce_rest-easybill.php	Bootstrap (constants, HPOS, loader, init)
includes/lib/	Reusable SP REST Billing core (REST client, abstract provider, DTOs)
includes/providers/class-spbill-provider-easybill.php	easybill implementation
includes/sources/class-wcreb-source-woocommerce.php	WooCommerce order → DTOs
includes/class-wcreb-settings.php	WooCommerce settings tab
includes/class-wcreb-plugin.php	Orchestrator + WooCommerce hooks
swagger.json	easybill REST API spec (reference)

TROUBLESHOOTING

- **No invoice is created.** Check that the transfer is enabled, the API key is valid, and the order actually reaches the configured trigger status. Check WooCommerce → Status → Logs (source wcreb) for errors.
- **Customer creation fails.** easybill requires both a last name and a company name; the plugin applies a B2C fallback automatically, so this should not normally occur — check the log for the exact API error.
- **The PDF is missing from the email.** Verify the “Attach PDF to email” setting points to an email that is actually sent for the order's status, and that Finalize invoice is on (a PDF is still available for drafts, but confirm the setting).
- **A duplicate invoice was created.** This should not happen — the flow is idempotent once a document id is stored. Use the “Create / resend easybill invoice” order action rather than re-triggering the status change.
- **Requests fail with HTTP 429.** This is easybill's rate limit; the plugin retries with a short back-off automatically. Persistent 429s usually mean too many orders are being processed in a short time.

REQUIREMENTS & COMPATIBILITY

- WordPress 6.9+
- WooCommerce 8.0+ (tested up to 10.9)
- PHP 7.4+
- An easybill account with an API key (Bearer token)
- Declares compatibility with WooCommerce High-Performance Order Storage (HPOS)

CHANGELOG

1.0.0 — 2026-06-30

- Initial release (V1).
- After-checkout transfer of WooCommerce orders to easybill: customer create/link, invoice create + finalize, PDF download, storage on the order, attachment to the customer email and a secure customer download link.
- Reusable SPBILL core (REST client, abstract provider, DTOs); easybill provider; WooCommerce source adapter.
- WooCommerce settings tab, manual resend order action, logging, HPOS compatibility.