

EvoXup

MEMBERSHIP

Membership • Licensing • Integrations



MEMBERSHIP



LICENSING



INTEGRATIONS



Grow • Protect • Connect

BUILD A STRONGER DIGITAL BUSINESS



Evoxup Membership

Complete Administrator, Licensing, Integration & Extensions Handbook

A practical reference for administrators, support operators, product managers and developers. It explains every Core admin page and tab, every major field and dropdown, real operating examples, the frontend Membership Center, 37 customer-facing extensions, and three operator/infrastructure components documented separately.

Core documented

Evoxup Membership 1.6.0
Extension API 2.1.0
Core DB schema 3.0.0

Visual edition

Blue + crimson Evoxup identity
User-provided artwork used as the cover
Packaged add-on artwork used where available

Examples

Representative customer, plan, product, order, license and webhook records are deliberately realistic but not production secrets.

Navigation & viewing

The Table of Contents is clickable. Cover and screenshots fit the PDF page/container and remain available to the browser or PDF viewer zoom controls.

ABOUT THIS EDITION

How to use this handbook

This handbook is written as an operating manual rather than a marketing brochure. Start with the System Model chapter if you are deciding who owns products, plans, licenses and checkout. Use the page chapters while configuring WordPress. Chapter 16 separates the 37 customer-facing extensions from the three Evoxup operator/infrastructure packages that are not part of normal commercial customer distribution.

Official product name

The official product name is Evoxup Membership. Version 1.6.0 standardizes the UI, package identity, frontend center, editor blocks and documentation while preserving the existing database schema, Extension API, shortcodes and integration behavior.

Visual authenticity

The extension icons/artwork in the add-on chapter are taken from the packaged extension assets. The administrator figures reproduce the exact Core 1.6.0 page names, tab names, field labels and option sets from the plugin source, populated with representative example records. Because a live WordPress staging URL was not supplied inside this chat, these are source-faithful interface captures rather than photographs of a particular production database.

Key ownership rule

Membership Plans own access duration, tier, entitlements and included products. Products own price, purchase URL, delivery, license provider/type/policy, license duration and site/key capacity. Keeping this boundary intact prevents the licensing confusion that earlier builds suffered from.

Commercial distribution security boundary

A commercial customer never receives a Repository master/admin key, signing private key, or central pairing secret. Evoxup central infrastructure protects Evoxup add-ons and updates. The customer's own products, members, licenses and activations remain under that customer's Membership installation and, when needed, its own License API credentials.

Table of Contents

Click any topic to jump directly to it.

Core manual

1	System Model & Installation	6
2	Dashboard	9
3	Customers	10
4	Membership: Members	12
4.2	Membership: Plans	14
4.3	Membership: Plan Products	16
4.4	Membership: Statistics	18
4.5	Membership: Messages	19
4.6	Membership: Notifications	20
5	Products: All Five Source Tabs	21
5.2	Product Editor & Licensing Fields	23
5.3	WooCommerce Product Licensing	26
6	Orders	27
7	Licenses: Licenses	29
7.2	Licenses: License Types	31
7.3	Licenses: Generators	32
7.4	Licenses: Activations	33
8	Integrations: Platforms & IPN	34
8.2	Integrations: Product Routing	36
8.3	Integrations: EVO REST API	38
9	Webhook Inbox	39
10	Extensions	40
11	Marketplace	42
12	Access & Roles	44
13	Settings	45
14	Frontend Membership Center	47
15	End-to-End Examples	48

16	Official Extensions & Infrastructure Reference	49
17	Troubleshooting & Operational Checklist	93
18	Glossary	95

37 Customer / Commercial Extensions

01.	Evoxup Account	50	19.	Evoxup License API - Advanced	69
02.	Evoxup Unified Control Center	51	20.	Evoxup LMS	71
03.	Evoxup Affiliate	52	21.	Evoxup LMS Visual Builder	72
04.	Evoxup Reports & Analytics	53	22.	Evoxup MarketProducts	73
05.	Evoxup Automations	54	23.	Evoxup Notifications	74
06.	Evoxup Backup Manager	55	24.	Evoxup PayPal Provider	75
07.	Evoxup Billing & Subscriptions	56	25.	Evoxup Stripe Provider	76
08.	Evoxup Central Audit	57	26.	Evoxup WooCommerce Payments Bridge	77
09.	Evoxup Chat	58	27.	Evoxup Points & Reputation	78
10.	Evoxup Commerce & Offers	59	28.	Evoxup Ratings & Reviews	79
11.	Evoxup REST Connector	60	29.	Evoxup Release & Site Manager	80
12.	Evoxup Webhook Connector	61	30.	Evoxup Site Toolkit	81
13.	Evoxup Content Access	62	31.	Evoxup Social — Facebook	82
14.	Evoxup Extensions Marketplace	63	32.	Evoxup Social — GitHub	83
15.	Evoxup Forms	64	33.	Evoxup Social — Google	84
16.	Evoxup Groups & Referrals	65	34.	Evoxup Social Identity	85
17.	Evoxup Universal Integration Hub	66	35.	Evoxup Social — X	86
18.	Evoxup License API - Free	67	36.	Evoxup System Health & Diagnostics	87
			37.	Evoxup WooCommerce Integration	88

Operator / Infrastructure Reference - not customer distribution

I1.	Evoxup Repository Connector - legacy/private bridge	90	I2.	Evoxup Update Manager - operator infrastructure	91
			I3.	Evoxup Update Repository Server - central server	92

CHAPTER 1

System Model & Installation

Understand the authority boundaries before creating products or plans.

1.1 The authority model



The path is deliberately provider-neutral. A sale may originate in WooCommerce or an external platform. Product Routing resolves the external/native product to the canonical EVO product. A plan may include one or more products. Membership grants access. Products decide whether and how licenses are issued.

Product authority

- Price and sale price
- Purchase and upgrade URLs
- Delivery/download URL
- License required or not
- License provider and reusable License Type
- Verification policy: portable, site, domain, IP/CIDR, domain + IP
- License seats/sites and duration

Plan authority

- Tier: FREE, PRO, SUPER STAR or Custom
- Membership duration and grace days
- Included products: single or bundle
- Entitlements
- Auto-renew default
- Upgrade and downgrade targets
- Plan lifecycle status

1.2 Identity model

WordPress User ID is the login/account identity. **EVO Customer ID** is the commercial customer record. A **Membership Link ID** is the assignment row connecting a customer to a plan; it is not a user ID. This distinction is visible throughout Customers and Membership.

1.3 Commercial distribution & secret ownership

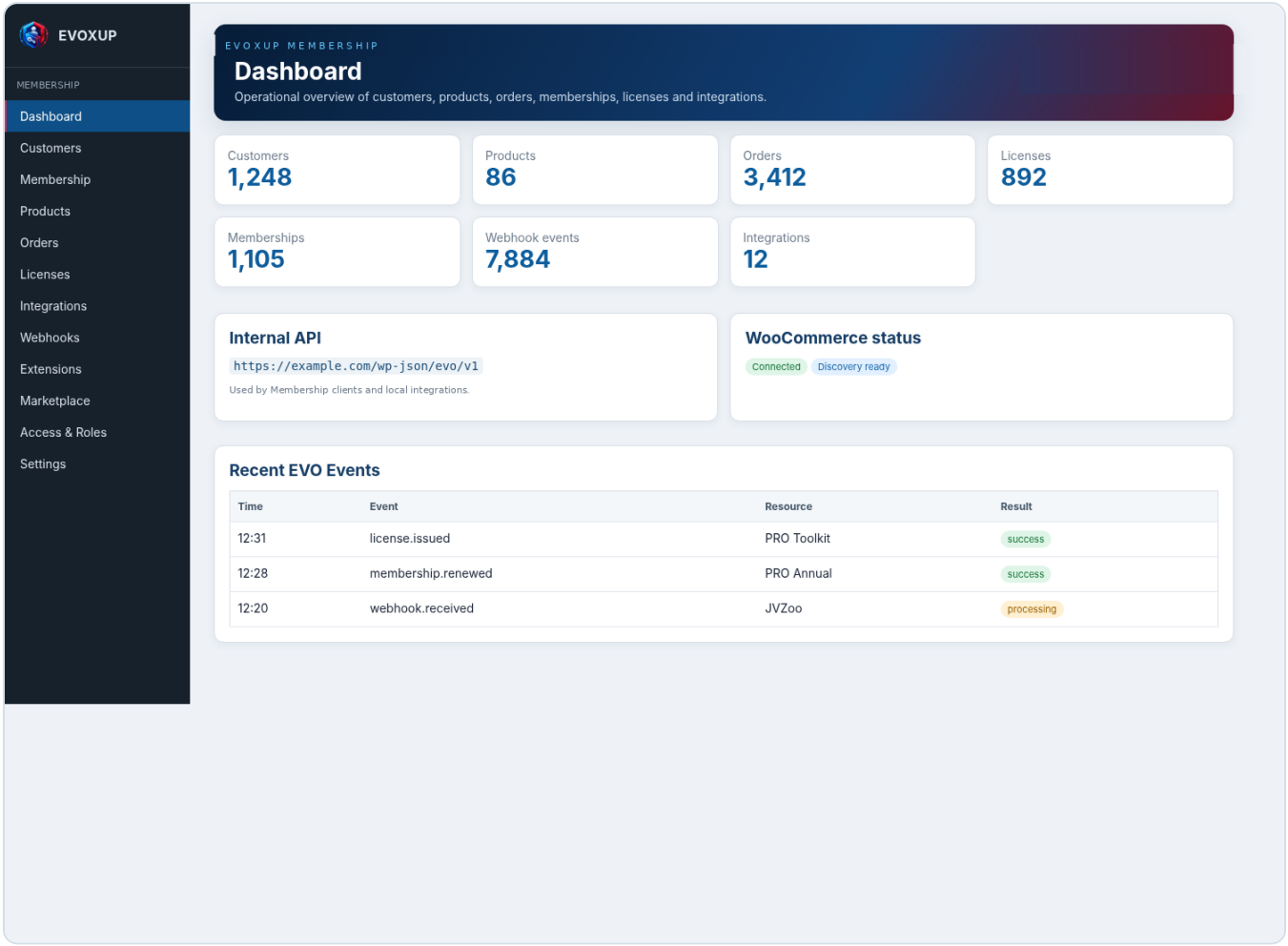
Trust domain	Owns	Credential rule
Evoxup central infrastructure	Evoxup extension repository, package/update releases, release signing, central site registry and platform telemetry.	Repository Master/Admin key, signing private key, pairing/publishing secrets stay on Evoxup-controlled servers and are never shipped to commercial customers.
Customer licensing authority	The customer's own products, members, memberships, licenses, activations and optional License API.	The customer creates and owns its own API clients/keys. These credentials are independent from Evoxup central repository secrets.

Official rule

The commercial package must not expose a field asking the customer for an Evoxup Repository Admin/Master/Pairing key. Customer-facing Marketplace/Extensions use only the supported internal update channel and site/license-scoped authorization. Operator-only infrastructure is documented separately in Chapter 16.

1.4 Installation and first-run order

1. Install and activate Evoxup Membership Core 1.6.0.
2. Open **EVOXUP -> Dashboard** and verify the internal REST base.
3. Open **Settings**, keep Internal API unless you intentionally use an external compatible API, then configure the Membership Center.
4. Create or sync Products before designing plans. Configure license ownership on each product.
5. Create Membership Plans and include the products they should grant.
6. If WooCommerce is used, run **Products -> WooCommerce -> Discover / Sync WooCommerce**.
7. If external platforms are used, create Integrations, then Product Routing, then test webhook receipt.
8. Install optional add-ons from Extensions/Marketplace only after Core is stable.



Source-faithful Dashboard capture with representative data.

CHAPTER 2

Dashboard

A fast operational view before drilling into individual records.

The Dashboard summarizes Customers, Products, Orders, Licenses, Memberships, Webhook events and Integrations, and exposes the internal API URL. Use it as a health/volume overview, not as a replacement for the detailed pages.

Field / control	Purpose	Example	Important values / behavior
Customers / Products / Orders / Licenses / Memberships	Counts of canonical Core records.	1,248 customers; 892 licenses	Use the dedicated pages for lifecycle operations.
Webhook events	Volume of stored inbound events.	7,884	Cleanup policy is controlled on Webhook Inbox.
Integrations	Configured external platforms.	12	Security and routes are managed under Integrations.
Internal API URL	Current local EVO REST base.	https://example.com/wp-json/evo/v1	External routing is configured under Settings.

EVOXUP MEMBERSHIP Dashboard
Operational overview of customers, products, orders, memberships, licenses and integrations.

Customers 1,248	Products 86	Orders 3,412	Licenses 892
Memberships 1,105	Webhook events 7,884	Integrations 12	

Internal API
<https://example.com/wp-json/evo/v1>
 Used by Membership clients and local integrations.

WooCommerce status
Connected Discovery ready

Time	Event	Resource	Result
12:31	license.issued	PRO Toolkit	success
12:28	membership.renewed	PRO Annual	success
12:20	webhook.received	JVZoo	processing

Dashboard with counters, Internal API, WooCommerce status and recent EVO events.

CHAPTER 3

Customers

The canonical commercial customer record, linked to WordPress identity and downstream activity.

3.1 Customer list

The list shows EVO Customer ID, name/email, WordPress link, phone/country, lifecycle status and last purchase. Open a customer to see memberships, licenses, orders and messages together.

3.2 Customer detail and actions

Field / control	Purpose	Example	Important values / behavior
EVO Customer ID	Internal customer record identity.	#1048	Never confuse this with WordPress User ID.
WordPress User	Login/account identity link.	WP User #304	Create/link an account when needed.
Status	Commercial customer state.	Active	Active, Inactive, Frozen, Archived.
Create/link WP account	Creates or connects the WordPress identity.	Link #304	Use when an external sale created the EVO customer first.
Send account access details	Sends member access/registration information.	Account access email	Useful after auto-provisioning.
Edit WP profile	Opens the WordPress user editor.	WP #304	WordPress remains the identity authority.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Customers

EVO customer identity linked to WordPress accounts, memberships, licenses and orders.

Customers
1,248

Linked WP accounts
1,104

Active
1,220

Customer list

EVO Customer ID	Customer	WP User	Phone / Country	Status	Last purchase	Actions
#1048	Amelia Stone amelia@example.com	WP #304	+1 555 0102 / US	active	2026-09-18	View
#1049	Omar Hassan omar@example.net	Not linked	+212 6... / MA	active	2026-09-18	View

Customer profile - #1048

Identity
EVO Customer ID: 1048
WordPress User ID: 304
Status: active

Quick actions
[Send account access details](#) [Edit WP profile](#)

Customer list and profile example.

CHAPTER 4.1

Membership - Members

Grant, renew, suspend or end customer-to-plan assignments.

Tabs: **Members** Plans Plan Products Statistics Messages Notifications

Field / control	Purpose	Example	Important values / behavior
Member	Existing EVO customer receiving access.	Amelia Stone - amelia@example.com	Required.
Plan	Active membership plan.	PRO Annual (pro-annual)	Plan determines products, tier, duration and entitlements.
Auto renew	Default renewal flag for this membership.	Checked	May originate from provider/sale policy.
Admin notes	Internal operational note.	Granted after annual purchase	Not a customer-facing entitlement.
Membership status	Lifecycle state.	Active	Pending, Active, Past due, Suspended, Cancelled, Refunded, Expired.
Renew duration days	Manual renewal length.	0	0 uses the plan duration; otherwise explicit days.
Permanent delete	Force deletion instead of normal lifecycle handling.	Unchecked	Use only when record removal is actually intended.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Membership

Membership controls access and entitlements; pricing and license rules remain on products.

Stable core rule: a plan may contain one product or a bundle. Plans do not issue licenses.

MembersPlansPlan ProductsStatisticsMessagesNotifications

Grant membership

Member

Amelia Stone — amelia@example.com

Select existing EVO customer

☒ Auto renew

Admin notes

Granted after annual subscription purchase.

Grant membership

Plan

PRO Annual (pro-annual)

Only active plans

Membership Link	Member identity	Plan	Status	Products	Source	Period	Renewals	Actions
#501 Plan/product assignment	Amelia Stone WP #304 · EVO Customer #1048	PRO Annual pro-annual	active	E000041, E000052	woocommerce	2026-09-18 → 2027-09-18	1	Apply · Renew · Delete

Members tab with grant form and membership identity columns.

CHAPTER 4.2

Membership - Plans

Define access packages; plans can include one product or a bundle.

Field / control	Purpose	Example	Important values / behavior
Plan code	Stable machine-friendly plan identifier.	pro-annual	Required; use a durable slug-like code.
Plan name	Human-readable plan.	PRO Annual	Required.
Tier	Commercial/access tier.	PRO	FREE, PRO, SUPER STAR, Custom.
Duration days	Membership access duration.	365	0 = lifetime.
Grace days	Grace period after due/ expiry logic.	7	0 = no additional grace.
Upgrade to	Next higher plan.	SUPER STAR Annual	None or another plan.
Downgrade to	Lower plan target.	FREE	None or another plan.
Status	Plan lifecycle.	Active	Active, Inactive, Frozen, Archived.
Default auto-renew	Default membership renewal behavior.	Checked	Can be overridden by provider/sale.
Description	Plan description.	Annual PRO membership...	Administrative/customer-facing context depending on frontend.
Entitlements	Core capability/feature tokens.	elementor_kits image_library	One per line or comma separated.
Included products	Products delivered by this plan.	E000041 + E000052	One = single mode; multiple = bundle; mode is calculated automatically.

EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Membership - Plans

Create single-product or bundle membership plans and connect upgrade paths.

MembersPlansPlan ProductsStatisticsMessagesNotifications

Create / Edit plan

Plan code

pro-annual

Plan name

PRO Annual

Tier

PRO

Duration days (0 = lifetime)

365

Grace days

7

Upgrade to

SUPER STAR Annual

Downgrade to

FREE

Status

Active

None or another plan

None or another plan

None or another plan

Active | Inactive | Frozen | Archived

Default auto-renew

☒

Description

Annual PRO membership with templates and media.

Entitlements

elementor_kits
image_library

E000041 Template Library

☒

E000052 Media Library

☒

Save plan changes

Plan responsibility

Plan: duration, tier, included products, entitlements, renewal and upgrade path.

Product: price, purchase URL, license provider/type, license duration, activations and delivery.

Mode: bundle (calculated automatically)

Plans tab: exact Core fields with a realistic PRO Annual bundle.

CHAPTER 4.3

Membership - Plan Products

Read-only matrix showing which products each plan contains.

Edit the plan to add/remove products. This tab intentionally does not edit price, license type, license provider or delivery because those belong to the product.

Field / control	Purpose	Example	Important values / behavior
Plan / Mode	Plan and calculated mode.	PRO Annual / Bundle	Mode is empty, single or bundle.
Included products	Canonical products linked to the plan.	E000041, E000052	Source labels remain visible.
Active members	Current active membership count.	714	Operational health/scale indicator.
Buy / Upgrade	Resolved purchase link.	Open	Shows a warning when no purchasable product can resolve a link.
Edit plan	Returns to plan editor.	Edit plan	All product membership links are changed from the plan.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Membership - Plan Products & Statistics

Read-only plan/product matrix plus membership health indicators.

Members

Plans

Plan Products

Statistics

Messages

Notifications

Products included in membership plans

Plan	Mode	Included products	Active members	Buy / Upgrade	
PRO Annual	Bundle	E000041 Template Library E000052 Media Library	714	Open	Edit plan
FREE	Single	E000010 Free Library	328	Open	Edit plan

Members

Plans

Plan Products

Statistics

Messages

Notifications

All memberships

1,105

Active

1,042

Pending

18

Past due

11

Suspended

7

Expired

27

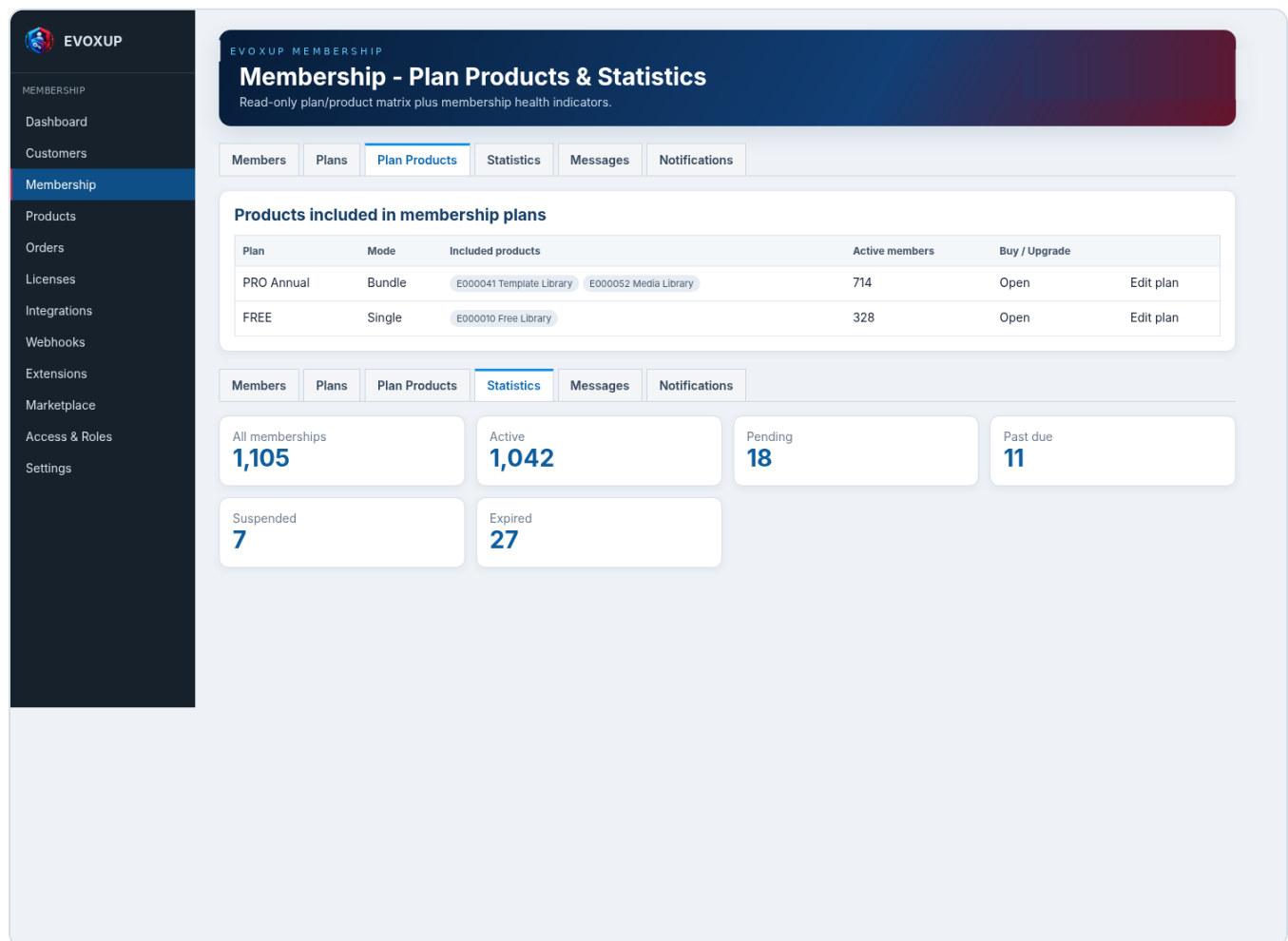
Plan Products matrix followed by the Statistics view.

CHAPTER 4.4

Membership - Statistics

Membership health by lifecycle status and plan.

Top cards show All memberships, Active, Pending, Past due, Suspended and Expired. Plan Health adds plan mode, product count and status-specific counts. Use this tab to spot a failing renewal/provider flow before opening individual rows.



EVOXUP MEMBERSHIP

Membership - Plan Products & Statistics

Read-only plan/product matrix plus membership health indicators.

Members Plans **Plan Products** Statistics Messages Notifications

Products included in membership plans

Plan	Mode	Included products	Active members	Buy / Upgrade
PRO Annual	Bundle	E000041 Template Library E000052 Media Library	714	Open Edit plan
FREE	Single	E000010 Free Library	328	Open Edit plan

Members Plans Plan Products **Statistics** Messages Notifications

All memberships
1,105

Active
1,042

Pending
18

Past due
11

Suspended
7

Expired
27

Statistics cards and plan health indicators.

CHAPTER 4.5

Membership - Messages

Send direct in-app/email messages tied to an EVO customer.

Field / control	Purpose	Example	Important values / behavior
Member	Message recipient.	Amelia Stone	Existing EVO customer.
Subject	Message subject.	Your EVO membership	Required.
Delivery	Channel.	In-app + Email	In-app, Email, Both.
Message	Body.	Your PRO membership has been renewed successfully.	Required; sent/stored according to channel.

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Membership - Messages & Notifications

Direct member communication and in-account notices.

Members

Plans

Plan Products

Statistics

Messages

Notifications

Send member message

Member

Amelia Stone — amelia@example.com

Select...

Subject

Your EVO membership

Delivery

In-app + Email

In-app | Email | Both

Message

Your PRO membership has been renewed successfully.

Send / create message

Members

Plans

Plan Products

Statistics

Messages

Notifications

Create notification

Member

Amelia Stone — amelia@example.com

Select...

Title

Important update

Type

Offer

Info | Success | Warning | Error | Offer | System

Action URL

https://example.com/account

Expires at

2026-12-31 23:59:59

Notification

Upgrade to SUPER STAR and unlock all premium libraries.

Messages and Notifications tabs with example content.

CHAPTER 4.6

Membership - Notifications

Create structured in-account notices with type, action and expiry.

Field / control	Purpose	Example	Important values / behavior
Member	Notification recipient.	Amelia Stone	Existing EVO customer.
Title	Short headline.	Important update	Required.
Type	Visual/semantic notification type.	Offer	Info, Success, Warning, Error, Offer, System.
Action URL	Optional destination.	https://example.com/account	Use for upgrade/detail actions.
Expires at	Optional expiry.	2026-12-31 23:59:59	Expired notices can stop being actionable.
Notification	Message body.	Upgrade to SUPER STAR...	Required.

CHAPTER 5.1

Products - All Five Source Tabs

One commercial authority with isolated source views.

Tabs: **All Products** **EVO Internal** **WooCommerce** **Member Products** **External** . The All Products tab is an operational view, not a copied product store.

EVO Internal

Products created directly in Membership.
Codes use the internal prefix and all fields are managed in EVO.

WooCommerce

Discovered from WooCommerce. Native Woo ID and name are synchronized; EVO licensing settings are layered on top.

Member Products

Products associated with a WordPress owner/member context while still using canonical ProductService records.

External

Products originating from an external platform. Provider product/native ID is stored for mapping and operational identity.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Products

Sources remain isolated while All Products provides one operational view.

All ProductsEVO InternalWooCommerceMember ProductsExternal

WooCommerce discovery: EVO stores the native Woo ID and current name together.

Image	Product	Actions
	E000041 Template Library PRO EVO Internal USD 79.00 active License: evo - 3 key/site - Plan: PRO Annual - Internal ID 41 - software	Edit
	W0000823 Woo Membership Annual WooCommerce USD 99.00 active License: evo - 1 key/site - Native ID 823 - membership	Edit

Source tabs

EVO InternalWooCommerceMember ProductsExternal

Each tab filters the same commercial authority without copying products between sources.

All Products compact view; all five source tabs remain visible.

CHAPTER 5.2

Product Editor & Licensing Fields

Configure commercial identity, delivery and the complete license policy in one canonical place.

Field / control	Purpose	Example	Important values / behavior
Product Code	Auto-generated source-aware code.	E000041	Read-only after generation; source prefix is automatic.
Product name	Human name.	EVO Pro Toolkit	Required.
Owner WordPress User ID	Product ownership/ operator context.	1	Defaults to current user.
Provider Product ID / Native ID	External-only source identity.	provider-product-123	Shown for External source.
Provider product name	External provider display name.	PRO Toolkit at Provider	Shown for External source.
Short / medium description	Compact summary.	Pro templates, media and updates	Used in product rows/previews.
Price	Normal price.	79.00	Decimal amount; product owns price.
Sale price	Optional reduced price.	59.00	Used when present.
Currency	Currency code.	USD	Example uses ISO-style code.
WordPress image ID	Media Library image attachment.	842	Upload / Select button can populate it.
External image URL	Remote visual when needed.	https://example.com/product.jpg	Optional.
Purchase URL	Primary buy destination.	https://example.com/buy/pro-toolkit	Used by catalog/plan link resolution.
Upgrade URL	Upgrade destination.	https://example.com/upgrade	Optional.
Product type	Functional category.	Software	Digital download, Software, Membership product, Course, Service, Other.
Digital file / delivery URL	Delivery/download target.	https://example.com/file.zip	Optional depending on product type.
Status	Product lifecycle.	Active	Active, Inactive, Frozen, Archived.
Requires a license	Enables license policy fields.	Checked	When off, license issuance is not required.

Field / control	Purpose	Example	Important values / behavior
Verification policy	Context required to verify.	Domain + server IP	Portable/key only, Exact site URL, Domain, Server IP/CIDR, Domain + server IP.
Exact site URL	Exact allowed URL for site policy.	https://customer.example.com	Used with Exact site URL policy.
Allowed domain	Allowed hostname.	customer.example.com	Used with Domain/Domain+IP.
Allowed server IP / CIDR	Allowed server address/range.	203.0.113.0/24	Used with IP/Domain+IP.
License provider	Who issues/manages the key.	EVO issues keys	EVO, External provider/adapter, No license, installed provider adapters, or a configured Integration.
License type	Reusable saved policy.	PRO Standard (pro-standard)	0 = product settings/defaults.
Independent license keys / sites	Seat/site capacity.	3	How many independent keys/sites the product grants.
License duration days	License validity length.	365	0 = lifetime.
Membership plans	Plans that include this product.	PRO Annual + SUPER STAR Annual	Many-to-many Product<->Plan.
Full description	Long product content.	Complete PRO toolkit...	Used in full preview/frontend where supported.
Internal notes	Operator-only note.	Migrated from 2025 catalog	Not customer-facing.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Products - Product Authority

Product Management owns pricing, delivery and licensing policy.

All ProductsEVO InternalWooCommerceMember ProductsExternal

Add product

Product Code

Generated automatically after save

Owner WordPress User ID

1

Price

79.00

Currency

USD

External image URL

https://example.com/product.jpg

Upgrade URL

https://example.com/upgrade

Digital file / delivery URL

https://example.com/file.zip

Verification policy

Domain + server IP

Portable | Exact site URL | Domain | Server IP/CIDR | Domain + server IP

Allowed server IP / CIDR

203.0.113.0/24

License type

PRO Standard (pro-standard)

Use product defaults or saved type

License duration days (0 = lifetime)

365

Full description

Complete PRO toolkit with 3 licensed sites, annual updates and member-only content.

Product name

EVO Pro Toolkit

Short / medium description

Pro templates, media and updates

Sale price

59.00

WordPress image ID

842

Purchase URL

https://example.com/buy/pro-toolkit

Product type

Software

Digital download | Software | Membership | Course | Service | Other

Status

Active

Active | Inactive | Frozen | Archived

Requires a license

Allowed domain

customer.example.com

License provider

EVO issues keys

EVO | External adapter | No license | Integration

Independent license keys / sites

3

PRO Annual

SUPER STAR Annual

Create product

EVO Internal product editor with commercial and licensing fields.

CHAPTER 5.3

WooCommerce Product Licensing

Keep WooCommerce as product/checkout identity while EVO owns optional licensing.

Run Discover / Sync WooCommerce first. Product Code, native Woo ID and name are synchronized/read-only in the Woo licensing form. Then choose whether the product requires a license, the verification policy, provider, type, seats/sites and duration, plus membership plan links.

EVOXUP MEMBERSHIP

Products - WooCommerce
Discover native WooCommerce products, then configure EVO licensing without changing WooCommerce identity.

[All Products](#)
[EVO Internal](#)
[WooCommerce](#)
[Member Products](#)
[External](#)

WooCommerce discovery
Native Woo ID is the technical link; the synchronized name is display/search data. [Discover / Sync WooCommerce](#)

WooCommerce product licensing

Product Code:
 Native Woo ID:

Name:

☒ Requires a license

Verification policy:
 License provider:

Portable | Site | Domain | Server IP | Domain + IP

 EVO / External / None / provider adapters

License type:
 Independent license keys / sites:

Saved license types:

License duration days:

☒ PRO Annual membership plan

[Save Woo product licensing](#)

WooCommerce discovery and EVO licensing layer.

CHAPTER 6

Orders

A provider-neutral commercial ledger for fulfillment, memberships and licenses.

Order rows preserve provider identity while assigning one EVO internal ledger ID. Status choices are **Received, Pending, Processing, Completed, Cancelled, Refunded, Failed, Archived**.

Field / control	Purpose	Example	Important values / behavior
EVO ID	Internal ledger identity.	#7102	Used for cross-system linking.
Source / Reference	Checkout/provider origin.	WooCommerce #9931 or JVZoo TX-7G6P9	Provider-native identity is preserved.
Customer	Linked EVO customer.	Amelia Stone	Customer email remains visible.
Items	Number of linked order items.	2	Expand routing to inspect item-to-product/plan resolution.
Total	Currency and total.	USD 99.00	Provider commercial value.
Status	EVO ledger lifecycle.	Completed	Received, Pending, Processing, Completed, Cancelled, Refunded, Failed, Archived.
Permanent delete	Force record removal.	Unchecked	Normal delete/archive behavior is safer for auditability.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Orders

Unified EVO ledger while checkout providers retain their native order identities.

Provider-neutral ledger: WooCommerce keeps native order ID; external providers keep their transaction reference; EVO links everything with one internal ledger ID.

EVO ID	Source / Reference	Customer	Items	Total	Status	Actions
#7102	WooCommerce #9931	Amelia Stone amelia@example.com	2	USD 99.00	processed	Status: Processed · Apply · Delete
#7103	JVZoo TX-7G6P9	Omar Hassan	1	USD 79.00	received	Apply

Order items / routing

Item	EVO Product	Plan	Qty	License provider	Keys/sites
PRO Annual	Woo Membership Annual	PRO Annual	1	EVO	1
Template Toolkit	Template Library PRO	PRO Annual	1	EVO	3

Unified order ledger and item routing example.

CHAPTER 7.1

Licenses - Licenses

Issue, verify and control key lifecycle independently from checkout.

Tabs: **Licenses** **License Types** **Generators** **Activations** . The Licenses tab contains both an administrator verification tester and the issue form.

Verify a license

Field / control	Purpose	Example	Important values / behavior
License key	Exact key presented by client.	EVO-8D2K...Z9LK	Required.
Site URL	Context for site/domain policies.	https://customer.example.com	Provide when policy needs it.
Domain	Explicit domain context.	customer.example.com	Optional when Site URL resolves domain.
Server IP	Server context for IP/CIDR policy.	203.0.113.10	Required by IP-based policies.
EVO Product ID	Optional expected product.	41	Prevents a valid key being accepted for the wrong product.
Client version	Client software version.	2.4.0	Useful for verification context/diagnostics.

Issue EVO license

Field / control	Purpose	Example	Important values / behavior
Product	Licensed active product.	EVO Pro Toolkit (E000041)	Product policy controls key behavior.
Customer email	Customer identity.	amelia@example.com	Can create/find customer/WP account.
First / Last name	Optional identity data.	Amelia / Stone	Useful when creating a new customer.
Existing EVO customer ID	Direct existing customer link.	1048	Optional alternative to lookup/create.
Number of independent keys / sites	How many keys/sites to issue.	3	Required, min 1.
Duration days	Explicit duration.	365	

Field / control	Purpose	Example	Important values / behavior
			0 = lifetime; product/type defaults may apply.
Create/find WordPress account	Identity provisioning.	Checked	Connects licensing to member account when appropriate.

Bulk actions: Activate, Stop, Disable / revoke, Freeze, Delete selected. A separate Delete all action removes all licenses and activations and should be reserved for controlled resets.

License verification, issuance and lifecycle example.

CHAPTER 7.2

Licenses - License Types

Reusable default policies for multiple products.

Field / control	Purpose	Example	Important values / behavior
Code	Stable type identifier.	pro-standard	Required.
Name	Human name.	PRO Standard License	Required.
Verification policy	Context binding rule.	Domain + server IP	Portable, Site, Domain, Server IP/CIDR, Domain + IP.
Default independent keys/sites	Default seat/site count.	3	Product can select/use type.
Default duration days	Default validity.	365	0 = lifetime.

MEMBERSHIP

- Dashboard
- Customers
- Membership
- Products
- Orders
- Licenses**
- Integrations
- Webhooks
- Extensions
- Marketplace
- Access & Roles
- Settings

EVOXUP MEMBERSHIP

Licenses - Types & Generators

Reusable license policies and configurable secure key formats.

Licenses
License Types
Generators
Activations

License Types

Code

pro-standard

Name

PRO Standard License

Verification policy

Domain + server IP

Portable | Site | Domain | Server IP/CIDR | Domain + IP

Default independent keys/sites

3

Default duration days

365

Add license type

Licenses
License Types
Generators
Activations

Default generator

Generation method

Secure letters + numbers

Alnum | Hex | Numeric | UUID v4

Prefix

EVO

Number of groups

3

Characters per group

10

Group separator

Dash (-)

Dash | Underscore | Dot

EVO-8D2K7P9F1A-Q4M6T8N2XR-W7B5C3Z9LK

License Types and Generator configuration.

CHAPTER 7.3

Licenses - Generators

Configure global key format and reusable generator profiles.

Field / control	Purpose	Example	Important values / behavior
Generation method	Random key algorithm/format.	Secure letters + numbers	Alnum, Hexadecimal, Numbers only, UUID v4.
Prefix	Readable prefix.	EVO	Optional branding/namespace.
Number of groups	Segment count.	3	For segmented formats.
Characters per group	Length per segment.	10	For segmented formats.
Group separator	Visual separator.	Dash (-)	Dash, Underscore, Dot.
Profile ID	Reusable generator identifier.	pro-generator	Required for profile.
Generator name	Profile name.	PRO Generator	Human label.

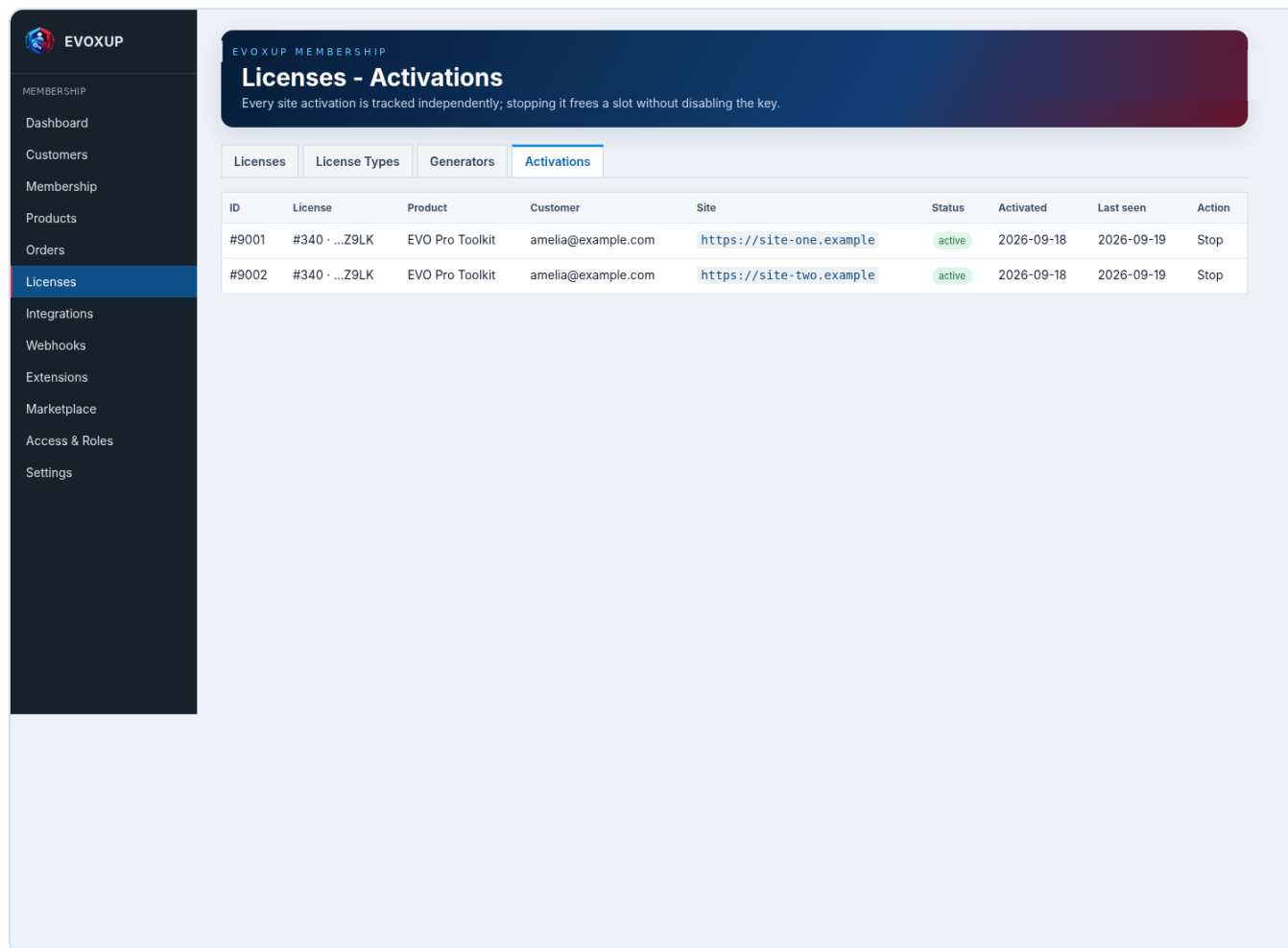
Example: EVO-8D2K7P9F1A-Q4M6T8N2XR-W7B5C3Z9LK

CHAPTER 7.4

Licenses - Activations

Track each licensed site independently.

Activation rows show ID, parent license, product, customer, site URL, status, activation time and last-seen time. **Stop** deactivates one site and frees a slot without disabling the parent license.



The screenshot displays the 'Licenses - Activations' page within the EVOXUP MEMBERSHIP system. The sidebar on the left contains the following navigation items: MEMBERSHIP, Dashboard, Customers, Membership, Products, Orders, Licenses (highlighted), Integrations, Webhooks, Extensions, Marketplace, Access & Roles, and Settings. The main content area features a header with the title 'Licenses - Activations' and a sub-header 'Every site activation is tracked independently; stopping it frees a slot without disabling the key.' Below this is a tabbed interface with 'Licenses', 'License Types', 'Generators', and 'Activations' (selected). The 'Activations' tab displays a table with the following data:

ID	License	Product	Customer	Site	Status	Activated	Last seen	Action
#9001	#340 · ...Z9LK	EVO Pro Toolkit	amelia@example.com	https://site-one.example	active	2026-09-18	2026-09-19	Stop
#9002	#340 · ...Z9LK	EVO Pro Toolkit	amelia@example.com	https://site-two.example	active	2026-09-18	2026-09-19	Stop

Two active site activations using a three-site license.

CHAPTER 8.1

Integrations - Platforms & IPN

Register external platforms and explicitly define webhook security.

Field / control	Purpose	Example	Important values / behavior
Platform name	Human provider name.	JVZoo	Required.
Platform slug	Endpoint/routing slug.	jvzoo	Required, stable.
Optional API URL	Provider API root if needed.	https://api.jvzoo.com/	Optional.
Status	Integration lifecycle.	Active	Active or Inactive.
Webhook/IPN security	Request verification mode.	HMAC SHA-256	None, Shared secret header, HMAC SHA-256, Bearer token, HTTP Basic, API key header, IP/CIDR allowlist, Custom hook/filter.
Signature / secret header	Header carrying signature/secret.	X-Signature	Depends on provider.
Shared secret / HMAC secret	HMAC/shared secret.	(secret)	Blank while editing keeps existing where supported.
Bearer token	Bearer credential.	(secret)	Security-mode specific.
Basic auth username/ password	HTTP Basic credentials.	api-user / (secret)	Security-mode specific.
API key header / Webhook API key	Header key authentication.	X-API-Key / (secret)	Security-mode specific.
Allowed IPs / CIDR	Source network restriction.	203.0.113.10 203.0.113.0/24	One per line.
Create/link WordPress user	Identity provisioning.	Checked	Used when external sale/customer needs an account.

Each platform card publishes a REST webhook URL and a direct fallback URL. Providers without signing can use security mode None, but the UI intentionally warns that the endpoint is unsigned.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Integrations

Connect external platforms, configure webhook security and product routing.

Platforms & IPN

Product Routing

EVO REST API

External platform

Platform name

JVZoo

Platform slug

javzoo

Optional API URL

https://api.javzoo.com/

Status

Active

Webhook/IPN security

HMAC SHA-256

Signature / secret header

X-JVZoo-Signature

Shared secret / HMAC secret

Allowed IPs / CIDR

203.0.113.0/24

Create/link WordPress user when appropriate

☒

Save platform

JVZoo Webhook

REST <https://example.com/wp-json/evo/v1/webhooks/javzoo>

Direct fallback https://example.com/?evo_webhook=javzoo

Platforms & IPN tab with JVZoo HMAC example and generated endpoints.

CHAPTER 8.2

Integrations - Product Routing

Map provider product identifiers to their real EVO/WooCommerce target.

Field / control	Purpose	Example	Important values / behavior
Platform	Configured external provider.	JVZoo	Required.
External product ID / permalink	Identifier arriving in webhook/IPN.	Ed3eHxDhgjp_nnDxmTb7Lw==	Required; use exact provider value.
Variant / price ID	Optional variant resolution.	annual-pro	Useful when one provider product has multiple variants/prices.
Target	Canonical target type.	EVO internal product	EVO internal product or WooCommerce product.
EVO product / WooCommerce product	Target record.	EVO Pro Toolkit (E000041)	Licensing is inherited from target product settings.

Critical routing rule

Product Routing does not own licensing. It only resolves provider identity to the target. Configure EVO product licensing under Products; configure WooCommerce licensing on the synchronized Woo product.

 EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Integrations - Routing & REST

Map provider product identifiers, then issue limited REST credentials.

Platforms & IPN

Product Routing

EVO REST API

Product Routing

Platform

JVZoo

External product ID / permalink

Ed3eHxDhgUp_nnDxmTb7LW==

Installed integrations

Variant / price ID

annual-pro

Target

EVO internal product

EVO product

EVO Pro Toolkit (E000041)

EVO internal product | WooCommerce product

Target product inherits licensing

Save route

Platforms & IPN

Product Routing

EVO REST API

Create REST key

Key name

Automation server

☒ Verify licenses

☒ Read products

☐ Manage products

☐ Submit webhooks

Allowed IPs / CIDR

198.51.100.20

Generate REST key

evo_live_7Q... (shown once)

Product Routing and REST API tabs.

Complete Administrator & Extensions Handbook - v1.6.0

Page 37

CHAPTER 8.3

Integrations - EVO REST API

Issue revocable, least-privilege credentials for external applications.

Field / control	Purpose	Example	Important values / behavior
Key name	Operator-friendly key label.	Automation server	Required.
Permissions	Allowed API actions.	Verify licenses + Read products	verify_license, read_products, manage_products, read_webhooks, write_webhooks, read_customers.
Allowed IPs / CIDR	Optional network restriction.	198.51.100.20	Blank = any allowed source.
Generated key	Secret shown once.	evo_live_7Q...	Copy immediately; stored list shows prefix, permissions, IP limit, status and last use.
Revoke	Stops one credential without affecting others.	Revoke	Prefer separate keys per application.

CHAPTER 9

Webhook Inbox

Inspect raw/normalized external events, verification, processing and errors.

Field / control	Purpose	Example	Important values / behavior
Received / Processed / Failed / Pending	Operational counters.	7,884 / 7,842 / 12 / 30	Pending corresponds to received/unprocessed events.
Keep days	Retention window for cleanup.	30	Clean old events affects stored history only.
Delete all history	Clears webhook history.	Confirm action	Does not remove integration/platform configuration.
View raw / normalized	Expands original payload and normalized EVO data.	seller_id=...	Use to diagnose mapping or verification mismatch.
Replay	Reprocesses an event.	Replay #7883	Useful after correcting product routing or transient failures.

EVOXUP

MEMBERSHIP
Dashboard
Customers
Membership
Products
Orders
Licenses
Integrations
Webhooks
Extensions
Marketplace
Access & Roles
Settings

EVOXUP MEMBERSHIP
Webhook Inbox
Inspect received webhook/IPN data and replay failed processing.

Received
7,884

Processed
7,842

Failed
12

Pending
30

Inbox maintenance
Cleanup affects history only, not platform configuration.

Keep 30 days - Clean old events
Delete all history

ID	Platform	Event	Received	Verify	Process	Source IP	Error	Data	Actions
#7884	jvzoo	sale_completed	12:22	verified	processed	203.0.113.22	—	View raw / normalized	Replay
#7883	gumroad	sale	12:18	not_required	failed	198.51.100.7	product_not_mapped	View raw / normalized	Replay

Webhook Inbox with verification/processing status, raw data and Replay.

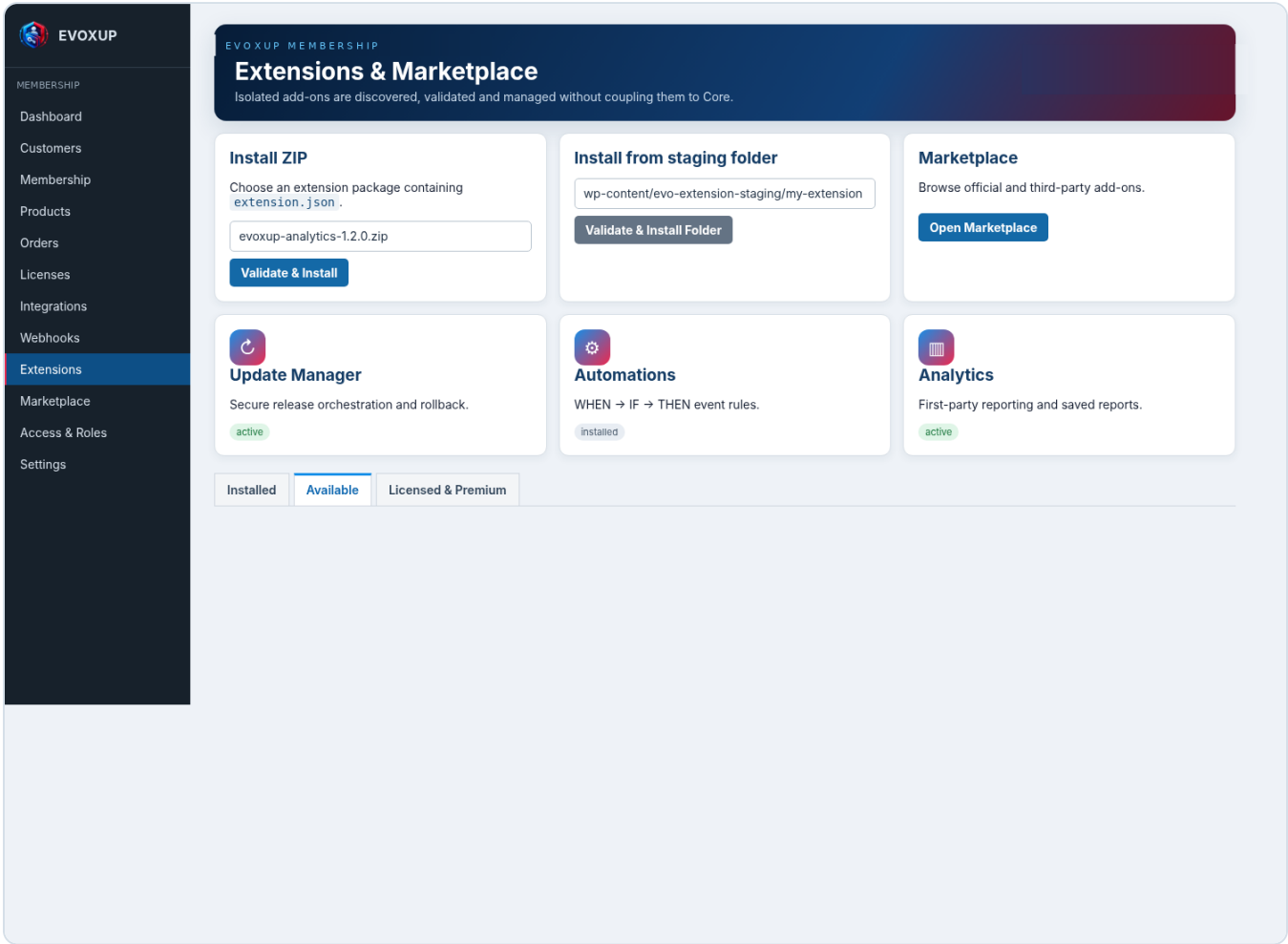
CHAPTER 10

Extensions

Install, validate, activate, update, remove and isolate optional capabilities on the customer site.

New packages use extension.json; legacy module.json remains supported. Installation starts disabled. Compatibility checks protect Core, and a boot failure isolates/deactivates the failing extension rather than taking Core down. In the commercial distribution profile, Extensions is the local lifecycle surface and does not ask the customer for a central Repository administration secret.

Field / control	Purpose	Example	Important values / behavior
Install ZIP	Upload extension package.	evoxup-analytics-1.2.0.zip	Validate & Install checks package format/compatibility.
Install from staging folder	Server-side installation source.	wp-content/evo-extension-staging/my-extension	Legacy staging is also accepted.
Marketplace	Opens add-on discovery UI.	Open Marketplace	Marketplace discovers/classifies releases and hands approved actions to the internal release engine; no central admin key is entered by the customer.
Activate / Deactivate	Turns compatible extension runtime on/off.	Activate Analytics	Recovery details are shown after isolated failures.
Format / Installed path	Shows package contract/location.	extension.json / wp-content/evoxup-extensions/...	Useful for support diagnostics.



Source-faithful current-build Extensions/Marketplace capture. In the commercial distribution profile documented here, the standalone Update Manager package/card is operator infrastructure and is not shipped as a normal customer add-on.

CHAPTER 11

Marketplace

Discover Evoxup add-ons and resolve public/licensed availability without exposing central repository secrets.

The dedicated Extensions Marketplace classifies releases into Installed, Available and Licensed & Premium. The commercial package uses a managed repository channel and site/license-scoped authorization internally. The customer never enters an Evoxup Repository Master/Admin/Pairing key.

Field / control	Purpose	Example	Important values / behavior
Managed repository channel	Evoxup package/update source.	Configured by the product, not by a customer secret.	May use site/license-scoped authorization internally. Never request the central repository admin/master key.
Installed	Already present locally.	Analytics 1.2.0	Manage under Extensions.
Available	Public repository extension not installed.	Forms 1.1.0	Can be installed when compatible.
Licensed & Premium	Protected extension available to entitled site.	Advanced add-on	Entitlement precedes package/runtime access.

Do not confuse two APIs

Marketplace access protects Evoxup products. License API Free/Advanced protects the customer's own products and uses credentials owned by that customer. The two trust domains are independent.

EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Extensions & Marketplace

Isolated add-ons are discovered, validated and managed without coupling them to Core.

Install ZIP

Choose an extension package containing extension.json.

evoxup-analytics-1.2.0.zip

Validate & Install

Install from staging folder

wp-content/evo-extension-staging/my-extension

Validate & Install Folder

Marketplace

Browse official and third-party add-ons.

Open Marketplace

Update Manager

Secure release orchestration and rollback.

active

Automations

WHEN → IF → THEN event rules.

installed

Analytics

First-party reporting and saved reports.

active

Installed

Available

Licensed & Premium

Extensions and Marketplace example.

CHAPTER 12

Access & Roles

Use WordPress roles plus granular EVO capabilities to enforce least privilege.

Built-in Evoxup role blueprints include **Evoxup Manager**, **Evoxup Support**, **Evoxup Finance**, **Evoxup Moderator** and **Evoxup Analyst**. Customer and Subscriber are also selectable as base roles.

Administrators are never downgraded from this screen.

Core capability groups

Dashboard, Customers, Memberships, Products, Product ownership actions, Plans, Orders, Licenses, Activations, Integrations, Webhooks, Extensions, Marketplace, Messages, Events, Logs, Settings and Access.

Shared extension capabilities

Analytics, Affiliate, Finance, Moderation, Diagnostics, Audit and Updates can register into the same central capability registry.

EVOXUP

MEMBERSHIP

Dashboard

Customers

Membership

Products

Orders

Licenses

Integrations

Webhooks

Extensions

Marketplace

Access & Roles

Settings

EVOXUP MEMBERSHIP

Access & Roles

Assign Evoxup roles and granular capabilities to WordPress users.

WordPress users

User	Email	Roles	EVO access	
Site Manager #12 · manager	manager@example.com	evox_manager	Products, Memberships, Orders, Licenses...	Manage access
Support Agent #18 · support	support@example.com	evox_support	Customers, Messages, Webhooks	Manage access

Manage access - Support Agent

Base role

Evoxup Support

Keep current role | Evoxup roles | Customer | Subscriber

Customer & Membership

☒ View customers ☒ Manage messages ☐ Manage plans

Licensing & Integrations

☒ View licenses ☐ Issue licenses ☒ Read webhooks

Save EVO access

WordPress user access list and fine-grained EVO capability editor.

CHAPTER 13

Settings

Choose API routing and shape the generated frontend Membership Center.

13.1 API routing

Field / control	Purpose	Example	Important values / behavior
Internal API	Use the current WordPress EVO REST API.	https://example.com/wp-json/evo/v1	Recommended for a normal single-site installation.
External API	Route supported calls to compatible external base.	https://api.example.com/evo/v1	Select only when architecture requires a remote EVO API.
Current effective API	Resolved base after selection.	https://example.com/wp-json/evo/v1	Shown for verification.
Test selected API	Connectivity/configuration test.	Test selected API	Use before relying on external routing.

13.2 Membership Center structure

Field / control	Purpose	Example	Important values / behavior
Hero title	Main frontend heading.	Welcome to Evoxup	Editable.
Products section title	Catalog heading.	Choose your product	Editable.
Plans section title	Membership heading.	Membership plans	Editable.
Member account title	Account section heading.	My Evoxup	Editable.
Hero text	Introductory copy.	Manage your products...	Editable.
Layout	Section order/density.	Catalog first	Catalog first, Member account first, Compact catalog.
Catalog visibility	Who can browse catalog.	Public	Public or Signed-in members only.
Primary blue	Theme color.	#0b66b0	Color input.
Crimson accent	Theme accent.	#d61f45	Color input.

Field / control	Purpose	Example	Important values / behavior
Visible center sections	Module toggles.	All enabled	Products, Plans, Account, Memberships, Licenses, Activations, Orders, Downloads, Messages.

MEMBERSHIP

- Dashboard
- Customers
- Membership
- Products
- Orders
- Licenses
- Integrations
- Webhooks
- Extensions
- Marketplace
- Access & Roles
- Settings**

EVOXUP MEMBERSHIP

Settings

Configure API routing and the generated Membership Center structure.

API routing

☒ Internal API
<https://example.com/wp-json/evo/v1>

☐ External API
 Use another EVO-compatible API endpoint.

External API base URL

Current effective API: <https://example.com/wp-json/evo/v1>

Membership Center structure

Hero title

Products section title

Plans section title

Member account title

Hero text

Layout

Catalog visibility

Catalog first | Member account first | Compact catalog

Public | Signed-in members only

Primary blue

Crimson accent

☒ Products catalog
 ☒ Membership plans
 ☒ Member account
 ☒ Memberships
 ☒ Licenses
 ☒ Activations
 ☒ Orders
 ☒ Downloads
 ☒ Messages

API routing and Membership Center configuration.

13.3 Center page and WordPress blocks

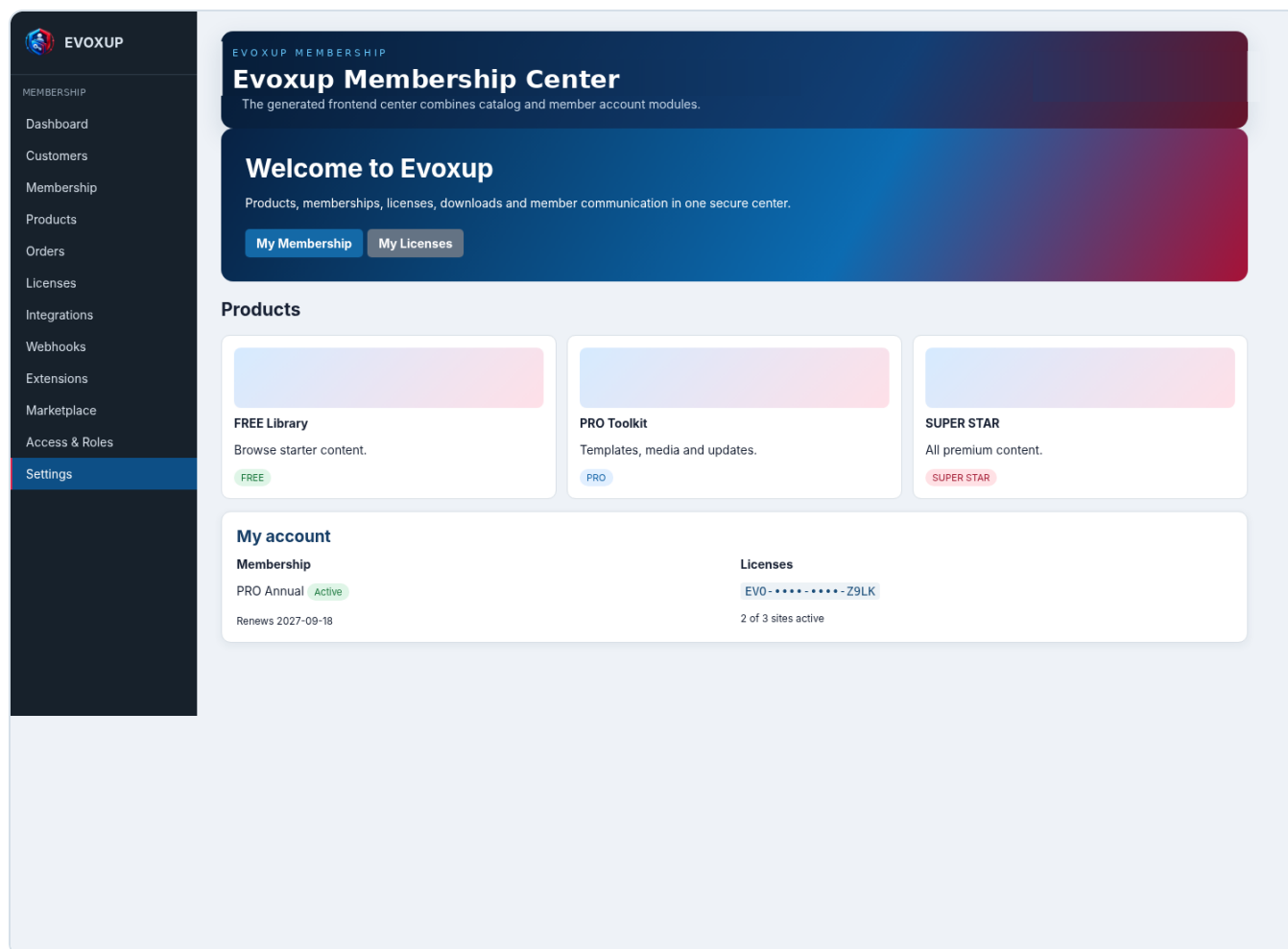
Full center shortcode: `[evoxup_membership_center]`. Independent shortcodes: `[evoxup_products]`, `[evoxup_plans]`, `[evoxup_my_membership]`. Gutenberg blocks: **EVO Membership Center**, **EVO Products**, **EVO Membership Plans**, **EVO My Membership**. The Repair / recreate center action restores the generated center page if its WordPress page was removed or damaged.

CHAPTER 14

Frontend Membership Center

The member-facing surface for catalog, account, memberships, licenses, orders, downloads and messages.

The Center is not a second database. It reads the same canonical Product, Plan, Membership, License, Activation, Order and Message services that the admin pages manage. You can use the generated page or compose a custom WordPress page from the independent blocks/shortcodes.



Representative frontend Center using the configured blue/crimson identity.

CHAPTER 15

End-to-End Examples

Concrete flows that show how the pages work together.

A. WooCommerce membership + EVO license

1. Woo order #9931 is paid.
2. WooCommerce product #823 has been discovered and linked to PRO Annual.
3. Order ledger stores provider ID + EVO ID.
4. Membership is created/renewed for the EVO customer.
5. Because the product requires an EVO license, key(s) are issued using the product License Type/policy.
6. Member sees membership/license in the Center; activation is tracked per site.

B. External JVZoo sale

1. Create JVZoo under Integrations -> Platforms & IPN and choose the strongest available security.
2. Product Routing maps exact JVZoo product ID to EVO product E000041.
3. Webhook Inbox receives, verifies and normalizes the sale.
4. EVO resolves/creates the customer and provider-neutral order.
5. Target product/plan determines membership and license fulfillment.
6. If mapping was missing, fix Product Routing and Replay the stored webhook.

C. Domain + IP protected software license

1. Product requires license with verification policy Domain + server IP.
2. Allowed domain = customer.example.com; allowed IP/CIDR = 203.0.113.0/24.
3. Client calls verification with key + product_id + site_url/domain + server_ip + client_version.
4. Core checks lifecycle, expiry, product match, context and activation capacity.
5. A successful activation consumes one site slot; stopping the activation frees it without revoking the key.

D. Bundle membership

1. Plan PRO Annual includes E000041 Template Library + E000052 Media Library.
2. Plan duration = 365 days; entitlements include elementor_kits and image_library.
3. Each product keeps its own price, purchase URL and license policy.
4. Granting the plan gives access to both products while license issuance follows each product independently.

37 Customer Extensions + Infrastructure Reference

The commercial catalog contains 37 customer-facing extensions. Three additional packages are documented in a separate operator/infrastructure reference and are not normal customer-distribution add-ons.



EXTENSION 1 OF 37

Evoxup Account

Package ID: evoxup-account Version: 1.4.0 Category: identity

Service definition

Frontend Identity authority for Evoxup with Identity Provider Contract 1.1, provider-neutral external account provisioning, visual-form actions, onboarding and secure local credential flows.

How it operates

Owns the member-facing identity experience. It uses WordPress accounts as the authentication identity, can create the My Account page, exposes login/register/password-recovery shortcodes, and lets external identity providers plug into one neutral contract.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

[Account Settings](#)[Frontend Account](#)[Identity Providers / Visual Form Actions](#)

Typical operating example

Example: Create a My Account page, enable local login, and add a Google provider. A user signing in with Google is linked to WordPress User #304; the customer profile continues to reference that same WP identity.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 2 OF 37

Evoxup Unified Control Center

Package ID: evoxup-admin-center Version: **3.0.0** Category: administration

Service definition

Unified role-aware Evoxup administration with Customer 360, Product 360, global search and the existing Operations Center.

How it operates

Aggregates cross-system data without replacing the source services. Customer 360 brings memberships, licenses, orders and messages together; Product 360 brings product identity, plans and entitlements together; role-aware views hide actions the operator cannot perform.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Overview

Customer 360

Product 360

Global Search

Operations

Typical operating example

Example: Search `amelia@example.com`, open Customer 360, verify the active PRO membership, license #340 and Woo order #9931 from one role-aware view.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 3 OF 37

Evoxup Affiliate

Package ID: evoxup-affiliate Version: **1.0.6** Category: commerce

Service definition

Affiliate program with isolated accounting, public Extension API 2.1 integration, extension-owned payout encryption, Reports Source and Account Summary Source contracts.

How it operates

Records referral attribution, applies commission rules, creates commission ledger entries and moves them through payout/fraud states. Financial records stay isolated from Core while public Extension API contracts expose safe summaries and report sources.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Dashboard

Affiliates

Clicks

Commissions

Commission Rules

Payouts

Coupons

Fraud Review

Settings

Audit

Typical operating example

Example: Rule: 30% of eligible product net revenue. Referral `AFF-102` sends a buyer to PRO Annual; after the order is completed, a pending commission is created and later moved to payable.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 4 OF 37

Evoxup Reports & Analytics

Package ID: evoxup-analytics Version: 1.2.0 Category: analytics

Service definition

First-party analytics with capability-gated cross-extension report sources, saved reports, CSV export and scheduled delivery.

How it operates

Collects and reads registered report sources, then turns them into time-window reports. Operators can save report definitions, export CSV data and schedule delivery without giving the analytics extension direct ownership of other extensions data.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Overview / Sources

Saved Reports

Exports

Scheduled Delivery

Typical operating example

Example: Build a 30-day report with Sales / conversions + Affiliate clicks, save it as `Growth Weekly`, export CSV and schedule a Monday delivery.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 5 OF 37

Evoxup Automations

Package ID: evoxup-automations Version: **1.1.0** Category: automation

Service definition

Event-driven WHEN → IF → THEN automation engine for Evoxup.

How it operates

Listens to Evoxup events and executes WHEN -> IF -> THEN rules. The queue isolates delayed work; run history records the trigger, condition decisions, actions and failures so a rule can be audited and retried safely.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Rules

Run History

Queue

Settings

Typical operating example

Example: WHEN `membership.status\_changed` IF new status = `past\_due` THEN create an in-app warning and email the member. Inspect Run History if an action fails.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 6 OF 37

Evoxup Backup Manager

Package ID: evoxup-backup-manager Version: 1.1.0 Category: management

Service definition

Portable configuration backup, selective import, release restore points and safe target-site recovery certification for Evoxup.

How it operates

Exports portable Evoxup configuration, imports selected sections and creates restore points before risky release work. Recovery certification verifies that a backup is usable on the intended target instead of treating file creation as proof of recoverability.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Export

Import

Restore Points

Typical operating example

Example: Before Update Manager upgrades Analytics, create a release restore point. If postflight health fails, restore the previous package/configuration from Restore Points.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 7 OF 37

Evoxup Billing & Subscriptions

Package ID: evoxup-billing Version: 1.0.1 Category: commerce

Service definition

Provider-neutral payments, billing plans, subscriptions, transactions, refunds, disputes and dunning.

How it operates

Provides a provider-neutral billing model for recurring plans, subscriptions, payments, refunds, disputes and dunning. Stripe, PayPal and WooCommerce adapters report payment state into the shared billing model rather than duplicating membership/license fulfillment.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Plans

Transactions

Subscriptions

Providers

Typical operating example

Example: Plan `PRO Annual` bills USD 99/year through Stripe. A successful transaction activates the subscription; a failed renewal enters dunning without manually editing the Core membership.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 8 OF 37

Evoxup Central Audit

Package ID: evoxup-central-audit Version: 1.1.0 Category: security

Service definition

Central privacy-safe audit history with request correlation, incident evidence, retention, observability/SLO source, filters and CSV export.

How it operates

Receives privacy-safe audit facts from participating components, correlates them by request/incident, applies retention policy and exposes evidence for operations and SLO review. It is an evidence source, not a replacement for extension-owned operational data.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Audit Events

Filters / Correlation

Incident Evidence

Retention / Export

Typical operating example

Example: Filter by correlation ID from a failed webhook flow and export privacy-safe incident evidence containing webhook verification, routing and license issuance decisions.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 9 OF 37

Evoxup Chat

Package ID: evoxup-chat Version: 1.0.3 Category: communication

Service definition

Isolated member-to-admin messaging with expandable sidebar, unread state, moderation, safe attachments and support inbox for Evoxup.

How it operates

Provides member-to-admin conversation threads, unread state, moderation, assignments and controlled attachments. The frontend sidebar and support inbox share the same isolated conversation data while permissions protect staff-only actions.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Inbox

Reports

Settings

Typical operating example

Example: Member opens support chat about a license activation. Staff assigns the thread, answers, attaches a safe file, and closes the conversation after resolution.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 10 OF 37

Evoxup Commerce & Offers

Package ID: evoxup-commerce-offers Version: 1.0.2 Category: commerce

Service definition

Provider-neutral offers, coupons, bundles, trials, upgrades, renewals and campaign attribution for Evoxup products and membership plans.

How it operates

Defines offers that reference existing EVO products/plans, then applies coupon, bundle, trial, upgrade, renewal and campaign-attribution rules. It never becomes the owner of product price/licensing records; it adds a commercial offer layer above them.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Offers

Coupons / Eligibility

Campaign Attribution

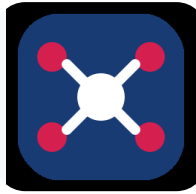
Typical operating example

Example: Offer `PRO-UPGRADE-20`: 20% upgrade discount from FREE to PRO, eligible only for active FREE members, attributed to campaign `sept-upgrade`.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 11 OF 37

Evoxup REST Connector

Package ID: evoxup-connector-rest Version: **1.0.0** Category: integrations

Service definition

Safe generic REST/HTTP connector with encrypted authentication for the Evoxup Universal Integration Hub.

How it operates

Registers a reusable generic HTTP/REST connector with encrypted authentication for Integration Hub. Connections supply endpoint and credentials; workflows use the connector without each integration implementing its own transport code.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Connector registration / endpoint settings

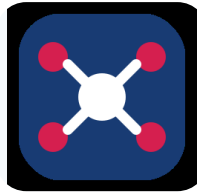
Typical operating example

Example: Connection `CRM API` uses Base URL `https://crm.example.com/api/` with Bearer authentication; Integration Hub calls `/contacts` through this connector.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 12 OF 37

Evoxup Webhook Connector

Package ID: evoxup-connector-webhook Version: **1.0.0** Category: integrations

Service definition

Signed inbound and outbound generic webhook connector for the Evoxup Universal Integration Hub.

How it operates

Provides signed inbound and outbound webhook transport for Integration Hub, including verification and delivery behavior. It is useful when a provider communicates through events rather than a polling/API client.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Inbound / outbound webhook connector settings

Typical operating example

Example: Register inbound path `partner-order` with an HMAC secret and outbound event `license.issued`; Hub tracks each delivery and retry.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 13 OF 37

Evoxup Content Access

Package ID: evoxup-content-access Version: 1.0.0 Category: content-access

Service definition

Granular WordPress content restriction and protected digital assets powered by Evoxup products, plans, memberships and entitlements.

How it operates

Evaluates access rules against products, plans, memberships and entitlements before rendering protected WordPress content or serving protected files. Reports explain allowed/denied/expired requests and diagnostics show configuration problems.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Access Rules

Protected Files

Reports

Settings

Diagnostics

Typical operating example

Example: Rule: page `Premium Downloads` requires entitlement `image\_library`. An active PRO member is allowed; a signed-in FREE member receives the configured denial/upgrade response.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 14 OF 37

Evoxup Extensions Marketplace

Package ID: evoxup-extension-marketplace Version: **2.1.0** Category: customer platform / marketplace

Service definition

Customer-side discovery and entitlement-aware Marketplace for Evoxup extensions. It presents Installed, Available and Licensed & Premium releases while keeping central repository administration private.

How it operates

Classifies releases, resolves whether the current site/license is entitled, and hands approved install/update actions to the commercial build's internal release engine / Extensions lifecycle. It never asks the customer for an Evoxup Repository Master/Admin/Pairing key.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.5

Admin surfaces / tabs

Installed

Available

Licensed & Premium

Typical operating example

Example: Open Available, select Analytics 1.2.0, verify site entitlement, then install through the managed customer-side Extensions/release engine. No central repository administration credential is entered.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 15 OF 37

Evoxup Forms

Package ID: evoxup-forms Version: 1.1.0 Category: experience

Service definition

Visual frontend form builder with registered action providers, secure password fields, identity templates and private-action data isolation.

How it operates

Builds frontend forms visually and dispatches submissions to registered action providers. Private actions such as passwords are kept out of generic submission data, while diagnostics reveal missing providers or invalid form configuration.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Forms

Submissions

Diagnostics

Typical operating example

Example: Build `Member Onboarding` with name, company and private password fields; action provider creates/updates the account while the password never enters the generic submission payload.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 16 OF 37

Evoxup Groups & Referrals

Package ID: evoxup-groups-referrals Version: 1.0.1 Category: community

Service definition

Groups and referral hierarchy with public Extension API usage and Account Summary Source integration.

How it operates

Creates groups and referral hierarchy records, assigns members and publishes safe account-summary data through the public extension contract. It can support team structures, sponsor trees and group membership without changing Core customer identity.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Groups

Members

Referrals

Settings

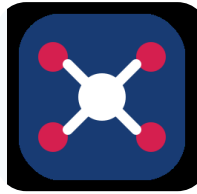
Typical operating example

Example: Create group `Agency Alpha`, add three members, assign one as sponsor, and expose the group/referral summary inside the member account.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 17 OF 37

Evoxup Universal Integration Hub

Package ID: evoxup-integration-hub Version: 1.0.0 Category: integrations

Service definition

Provider-neutral connection, credentials, webhook, queue, health and sync orchestration for Evoxup integrations.

How it operates

Centralizes provider-neutral connections, credentials, connector selection, outbound queueing, inbound webhook activity, health checks and synchronization. Connectors provide transport; Hub owns orchestration and operational visibility.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Connections

Connectors

Activity

Webhooks

Diagnostics

Typical operating example

Example: Create connection `Helpdesk`, choose REST Connector, store encrypted credentials, run Test, then observe outbound sync jobs and failed retries in Activity.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 18 OF 37

Evoxup License API - Free

Package ID: evoxup-license-api Version: **1.0.0** Category: membership-licensing

Service definition

Self-hosted license verification API for Evoxup Membership. Verify, activate, deactivate and check license status on the customer's own server, with strict private storage outside the web root.

How it operates

Adds a self-hosted public licensing endpoint layer around Membership license verification/activation/deactivation/status. Secrets and private state are stored outside the public web root where possible, while clients call stable REST endpoints.

Requirements / placement

Evoxup 1.6.0 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Overview

Endpoints

Private Storage

Security

Customer-owned trust domain

This API belongs to the customer who installed Membership and is used to verify that customer's own products/licenses. Its API clients and secrets are created by that customer and are completely separate from Evoxup central repository/update credentials.

Typical operating example

Example: Client calls `/verify`` with a license key, product ID, site URL/domain and server IP when policy requires it; response reports validity, product and activation capacity.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 19 OF 37

Evoxup License API - Advanced

Package ID: evoxup-license-api-advanced Version: 1.0.0 Category: membership-licensing

Service definition

Advanced self-hosted license API security and policy layer: HMAC API clients, replay protection, membership/entitlement policies, device binding, offline signed grants, callbacks and auditable request analytics.

How it operates

Adds signed API clients, replay protection, policy checks, device binding, offline signed grants, callback controls and security analytics above the Free License API. Use it when third-party software requires a hardened machine-to-machine contract.

Requirements / placement

Evoxup 1.6.0 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

API Clients

Offline Grants

Security Analytics

Signing Guide

Settings

Customer-owned trust domain

This API belongs to the customer who installed Membership and is used to verify that customer's own products/licenses. Its API clients and secrets are created by that customer and are completely separate from Evoxup central repository/update credentials.

Typical operating example

Example: Create client `Desktop App`, sign requests with HMAC, allow a 5-minute replay window, and issue an offline signed grant for a device that may be disconnected temporarily.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 20 OF 37

Evoxup LMS

Package ID: evoxup-lms Version: 1.1.2 Category: learning

Service definition

Integrated learning management for Evoxup with courses, lessons, quizzes, progress, drip content, prerequisites, certificates and a public visual-builder mutation contract.

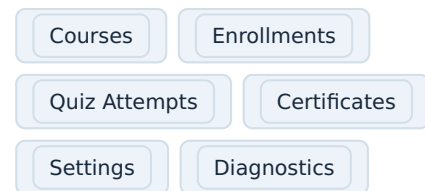
How it operates

Adds courses, lessons, quizzes, enrollments, progress, drip rules, prerequisites and certificates linked to Evoxup access. Course ownership remains in the LMS, while eligibility can reuse products/plans/memberships and public builder contracts.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs



Typical operating example

Example: Course `Evoxup Admin Mastery` requires PRO plan, drips Lesson 2 after 3 days, uses a quiz pass mark, and issues a certificate after completion.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 21 OF 37

Evoxup LMS Visual Builder

Package ID: evoxup-lms-builder Version: 1.0.0 Category: learning

Service definition

Visual drag-and-drop curriculum builder for Evoxup LMS using only the public LMS Builder Contract.

How it operates

Provides drag-and-drop curriculum editing through the public LMS Builder Contract. It mutates courses/lessons through the LMS contract rather than directly reading or writing private LMS tables.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Visual Builder

Course Settings

Typical operating example

Example: Open `Evoxup Admin Mastery`, drag Lesson 3 before Lesson 2, create a new quiz lesson and save through the LMS Builder Contract.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 22 OF 37

Evoxup MarketProducts

Package ID: evoxup-marketproducts Version: 1.0.1 Category: commerce

Service definition

Frontend product marketplace for EVO, WooCommerce, member and external products managed by Evoxup Membership Core.

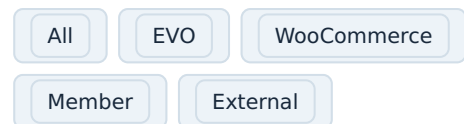
How it operates

Publishes a frontend marketplace from Core ProductService across EVO, WooCommerce, Member and External sources. Per-product visibility, featured/priority rules and display options affect presentation only; pricing/licensing remain in Core products.

Requirements / placement

Evoxup 1.2.0 · Extension API 1.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs



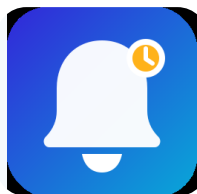
Typical operating example

Example: Set `Template Library PRO` to Public + Featured priority 10, display source badge and code, and show 12 products per marketplace page.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 23 OF 37

Evoxup Notifications

Package ID: evoxup-notifications Version: 1.1.0 Category: communication

Service definition

Queued member notifications and email delivery for Evoxup events with editable templates, retries and an in-account notification center.

How it operates

Converts selected Evoxup events into in-app and/or email notifications through a queue with retry behavior. Templates remain editable, multi-seat license notices can be grouped, and wp_mail keeps SMTP plugins compatible.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Settings

Templates

Delivery Queue

Typical operating example

Example: Template `license.issued` sends one grouped email for a 3-seat issuance and an in-account notification; a transient mail error is retried by the delivery queue.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 24 OF 37

Evoxup PayPal Provider

Package ID: evoxup-payment-paypal Version: 1.0.0 Category: payments

Service definition

PayPal Orders, Subscriptions, refunds and verified webhooks for Evoxup Billing.

How it operates

Implements PayPal Orders/Subscriptions/refunds and verified webhooks as a Billing provider. Billing consumes provider events and state; downstream Evoxup fulfillment remains controlled by the shared billing and Core services.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

PayPal Provider Settings

Typical operating example

Example: Connect PayPal, create the Billing checkout, verify PayPal webhook events, then let Billing update subscription and transaction state.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 25 OF 37

Evoxup Stripe Provider

Package ID: evoxup-payment-stripe Version: **1.0.0** Category: payments

Service definition

Stripe Checkout, subscriptions, refunds and verified webhooks for Evoxup Billing.

How it operates

Implements Stripe Checkout/subscriptions/refunds and verified webhooks as a Billing provider. API credentials are provider settings; webhook verification protects incoming state changes.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Stripe Provider Settings

Typical operating example

Example: Use Stripe Checkout for `PRO Annual`; verified `invoice.payment\_succeeded` updates Billing, while refunds are reported back through the provider adapter.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 26 OF 37

Evoxup WooCommerce Payments Bridge

Package ID: evoxup-payment-woocommerce Version: **1.0.0** Category: payments

Service definition

Observe-only payment adapter that reports WooCommerce payment/subscription state to Evoxup Billing without duplicating WooCommerce fulfillment.

How it operates

Observes WooCommerce payment/subscription state and reports it to Evoxup Billing without taking over WooCommerce checkout or duplicating Core fulfillment. Use it when WooCommerce remains the merchant-of-record interface.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Observe-only WooCommerce payment bridge

Typical operating example

Example: A Woo order paid via an existing gateway is observed by the bridge; Billing receives payment state while normal Woo checkout/email behavior remains unchanged.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 27 OF 37

Evoxup Points & Reputation

Package ID: evoxup-points-reputation Version: 1.0.1 Category: engagement

Service definition

Points and reputation with public Extension API usage and Account Summary Source integration.

How it operates

Awards or adjusts points from defined events, calculates reputation levels and exposes member summaries. Rules determine the event-to-points mapping; Core membership/license data remains untouched.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Overview

Members

Event Rules

Levels

Settings

Typical operating example

Example: Rule: +50 points for course completion; level `Expert` begins at 2,000 points. The member account displays the updated level via Account Summary Source.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 28 OF 37

Evoxup Ratings & Reviews

Package ID: evoxup-ratings-reviews Version: 1.0.1 Category: commerce

Service definition

Product ratings, moderated reviews and verified buyer/member badges for Evoxup products and MarketProducts.

How it operates

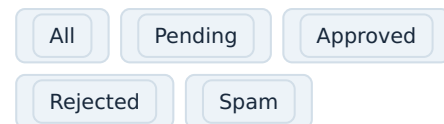
Collects product ratings/reviews, moderates publication status and can mark verified buyers/members.

MarketProducts can display the resulting rating signal without owning review moderation.

Requirements / placement

Evoxup 1.4.0 · Extension API 2.0.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs



Typical operating example

Example: Buyer of product E000041 submits 5 stars and a review. Moderator approves it; MarketProducts displays the rating with a verified-buyer badge.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 29 OF 37

Evoxup Release & Site Manager

Package ID: evoxup-release-site-manager Version: 1.2.0 Category: infrastructure

Service definition

Site identity, release policy, phased rollout, maintenance, rollback and Release Governance Contract 1.0.0 enforcement for Evoxup.

How it operates

Owns site identity, release channel, rollout bucket, maintenance policy and rollback policy. Update Manager asks it whether a file-changing release is allowed, allowing phased rollout and maintenance windows to be enforced consistently.

Requirements / placement

Evoxup 1.5.4 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Site Identity

Release Policy

Rollout / Maintenance

Rollback / History

Typical operating example

Example: Site is on Stable channel with rollout bucket 37. A release at 25% rollout is deferred; at 50% it becomes eligible during the configured maintenance window.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 30 OF 37

Evoxup Site Toolkit

Package ID: evoxup-site-toolkit Version: **1.0.1** Category: operations-admin

Service definition

Central directory for Evoxup pages, shortcodes and design components, plus public EVO product pages and purchase-link management.

How it operates

Provides one directory for Evoxup pages, shortcodes and design components, and can expose public product pages with purchase links. It helps administrators locate/render platform surfaces without manually tracking WordPress page IDs.

Requirements / placement

Evoxup 1.6.0 · Extension API 2.1.0 ·
PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Pages

Shortcodes

Components

Product Pages / Purchase Links

Typical operating example

Example: Find the auto-created Membership Center page, copy `[evoxup\_membership\_center]`, and set a public product page purchase link to the WooCommerce checkout URL.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 31 OF 37

Evoxup Social — Facebook

Package ID: evoxup-social-facebook Version: 1.0.0 Category: identity

Service definition

Official Facebook Login identity provider.

How it operates

Registers Facebook Login as an identity provider under the Social Identity/Account contract. Configure provider credentials and redirect URI; the neutral identity layer handles account linking instead of the provider creating a separate user system.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Facebook Identity Provider Settings

Typical operating example

Example: Enter App ID/secret and callback URL, enable Facebook, then link the returned provider identity to an existing member account.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 32 OF 37

Evoxup Social — GitHub

Package ID: evoxup-social-github Version: 1.0.0 Category: identity

Service definition

Official GitHub OAuth identity provider.

How it operates

Registers GitHub OAuth as an identity provider under the Social Identity/Account contract. It is suited to developer audiences and keeps the provider-specific OAuth exchange separate from account ownership.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

GitHub Identity Provider Settings

Typical operating example

Example: Configure GitHub OAuth app credentials, enable provider, then use GitHub sign-in to resolve the same WordPress account on return.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 33 OF 37

Evoxup Social — Google

Package ID: evoxup-social-google Version: 1.0.0 Category: identity

Service definition

Official Google OAuth 2.0/OpenID identity provider.

How it operates

Registers Google OAuth 2.0/OpenID as an identity provider under the Social Identity/Account contract. Successful provider identity is linked to the existing WordPress/Evoxup account according to the account-link policy.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Google Identity Provider Settings

Typical operating example

Example: Configure Google Client ID/secret and authorized redirect URI; a successful OpenID email is linked according to the account-link policy.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 34 OF 37

Evoxup Social Identity

Package ID: evoxup-social-identity Version: 1.0.0 Category: identity

Service definition

Provider-neutral social login, secure account linking and connected-account management for Evoxup.

How it operates

Provides the provider-neutral social-login and connected-account layer. Provider add-ons register with it; members can sign in/link/unlink while one WordPress account remains the identity authority.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Social Login

Connected Accounts

Provider Registry

Typical operating example

Example: Enable Google and GitHub providers, allow connected-account management, and require an authenticated local session before unlinking the last external identity.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 35 OF 37

Evoxup Social — X

Package ID: evoxup-social-x Version: 1.0.0 Category: identity

Service definition

Official X OAuth 2.0 + PKCE identity provider.

How it operates

Registers X OAuth 2.0 with PKCE as an identity provider. Credentials and callback settings are provider-specific; resulting identity is consumed through the common social/account contract.

Requirements / placement

Evoxup 1.5.3 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

X Identity Provider Settings

Typical operating example

Example: Configure X client credentials and PKCE callback; provider token exchange returns identity to the shared Social Identity layer.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 36 OF 37

Evoxup System Health & Diagnostics

Package ID: evoxup-system-health Version: 1.6.0 Category: infrastructure

Service definition

Read-only platform diagnostics with Production, Recovery, Security, Performance, Observability and Release Governance readiness gates.

How it operates

Runs read-only readiness gates across production, recovery, security, performance, observability, governance, Core/database, services/queues and extensions. Its support report consolidates diagnostic evidence without changing operational state.

Requirements / placement

Evoxup 1.5.4 · Extension API 2.1.0 · PHP 8.3
· WordPress 6.2

Admin surfaces / tabs



Typical operating example

Example: Before a production update, check Production, Recovery, Security, Performance, Observability and Governance readiness. A blocking Recovery failure prevents release execution.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.



EXTENSION 37 OF 37

Evoxup WooCommerce Integration

Package ID: evoxup-woocommerce Version: 1.0.2 Category: commerce

Service definition

Deep WooCommerce integration for Evoxup products, plans, customers, orders, licenses, entitlements, refunds and variable products.

How it operates

Synchronizes WooCommerce product/customer/order identity into Evoxup, maps products/plans and performs fulfillment through Core memberships/licenses/entitlements. Refund/cancel settings decide when access or licenses should be revoked or suspended.

Requirements / placement

Evoxup 1.5.2 · Extension API 2.1.0 · PHP 8.3 · WordPress 6.2

Admin surfaces / tabs

Overview

Mappings

Fulfillment

Settings

Diagnostics

Typical operating example

Example: Map Woo product #823 to PRO Annual, enable EVO licensing, process an order, create membership + license, then revoke access on refunded quantity according to settings.

Recommended activation procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

This entry is based on the packaged manifest and current extension source. Add-ons intentionally own only their isolated responsibility; Core remains authoritative for customer/product/membership/license records unless the component is explicitly the owner of a separate domain such as billing, LMS or audit.

Private / Operator Infrastructure

These packages are documented for Evoxup operators, controlled deployments and legacy compatibility. They are not normal commercial customer add-ons and central secrets must never be distributed to customers.



INFRASTRUCTURE 11 OF 3 - NOT CUSTOMER DISTRIBUTION

Evoxup Repository Connector

Package ID: repository-connector Version: 1.1.1 Category: legacy / private infrastructure

Service definition

Historical server-to-server bridge used by older Evoxup Repository API deployments for pairing, entitlement resolution and runtime diagnostics. It is retained only for controlled legacy/private compatibility.

How it operates

Older deployments paired Membership and Repository using a repository admin/pairing key, then exchanged a dedicated connector secret. This mechanism is not part of the commercial customer distribution model and no Evoxup central admin/pairing key should be given to customers.

Requirements / placement

Evoxup operator-controlled / legacy split-server deployments only. Do not ship in the normal commercial customer package.

Admin surfaces / tabs

Legacy Pairing

Entitlement Bridge

Runtime Diagnostics

Typical operating example

Operator-only example: Maintain an older controlled Repository API 0.4.x deployment during migration. The central pairing/admin secret remains under Evoxup operator control.

Deployment rule

1. Do not include this package in the normal commercial customer distribution.
2. Do not provide an Evoxup Repository Admin/Master/Pairing key to a customer.
3. Use only for a controlled legacy/private deployment that still requires the old bridge.
4. Prefer the current central repository + managed customer update channel for new deployments.

Legacy/private compatibility reference. This package is intentionally outside normal commercial customer distribution; central pairing credentials remain operator-controlled.

**INFRASTRUCTURE 12 OF 3 - NOT CUSTOMER DISTRIBUTION**

Evoxup Update Manager

Package ID: evoxup-update-manager Version: **2.2.0** Category: operator release/security infrastructure

Service definition

Expanded operator release/security component for managed Evoxup infrastructure: release orchestration, recovery, health gates, rollback, repository security controls and operational history/statistics.

How it operates

The standalone package is reserved for Evoxup-managed/operator deployments where repository credentials and security controls belong to the platform owner. In the commercial customer build, the necessary client-side install/update behavior belongs behind Core Extensions/Marketplace and must not expose central repository pairing/admin secrets.

Requirements / placement

Evoxup operator/managed infrastructure. Not a normal commercial customer add-on.

Admin surfaces / tabs

Updates

History & Rollback

Recovery

Security & Repository

Typical operating example

Operator-only example: Manage a signed Evoxup release, run preflight/backup/update/postflight, record telemetry and rollback history on an Evoxup-controlled site.

Deployment rule

1. Keep the standalone Update Manager package on Evoxup-controlled/operator deployments.
2. Do not expose central Repository pairing/admin/master credentials in the commercial customer UI.
3. Customer sites use Core Extensions + Marketplace + the managed internal release engine for Evoxup package lifecycle.
4. Central security statistics, repository administration and private credentials stay on Evoxup infrastructure.

Operator infrastructure reference. The standalone package is not a normal commercial customer add-on; central repository/security credentials remain on Evoxup-controlled systems.



INFRASTRUCTURE 13 OF 3 - CENTRAL SERVER ONLY

Evoxup Update Repository Server

Package ID: evoxup-update-repository-server Version: **3.1.0** Category: central private repository server

Service definition

Central repository server for publishing Evoxup releases, authorizing protected distribution, and serving signed Local ZIP, Remote Runtime and Cloud Service releases.

How it operates

Runs on the central repository site, not an ordinary customer Membership site. It publishes signed versioned releases, authorizes protected downloads/runtime/service access and enforces package/version/channel/distribution rules before artifacts or cloud results are served.

Requirements / placement

Evoxup central server role · PHP 8.1+ · WordPress server.

Server-side component

Install only on Evoxup central repository/server infrastructure. Never distribute this package or its administration/signing credentials as a customer-site add-on.

Admin surfaces / tabs

Repository Dashboard

Packages / Releases

Signing & Access

Distribution / Runtime

Cloud Services

Private secret boundary

Signing private keys, Repository Master/Admin credentials and publishing secrets remain on this central server. Commercial customer sites receive neither these secrets nor a configuration field asking for them.

Typical operating example

Example: Publish package `evoxup-analytics` 1.2.0 on Stable, attach SHA-256/signature and distribution type, then issue an exact site/version/channel access grant when entitlement is valid.

Operator deployment procedure

1. Install/validate the package through Evoxup Extensions or the appropriate server installation flow.
2. Activate only after dependency and capability checks pass.
3. Open the listed admin surface and save explicit production settings rather than relying on placeholders.
4. Run one small controlled workflow using a test customer/product/event.
5. Review logs, queue/audit/diagnostics where the component provides them, then enable production use.

Central server reference. Repository publishing, signing private keys, release storage, telemetry and administration remain on Evoxup-controlled infrastructure.

CHAPTER 17

Troubleshooting & Operational Checklist

Fast diagnosis patterns for the most common membership/licensing/integration problems.

A license verifies as invalid after issuance

Check product Requires a license, license status/expiry, expected product ID and verification policy context. For Site/Domain/IP policies, the client must send matching site/domain/server IP; Portable can verify with key only.

External sale is received but no membership/license is created

Open Webhook Inbox: confirm verification and processing. Inspect normalized payload. Ensure Product Routing exactly matches the provider product ID/variant. After fixing routing, use Replay.

WooCommerce product is missing from EVO

Open Products -> WooCommerce and run Discover / Sync WooCommerce. Then open the synchronized product to configure EVO licensing and plan links.

Plan grants the wrong license behavior

Do not configure license behavior on the plan. Open each included Product and verify license provider, License Type/policy, site capacity and duration.

Member sees no purchase/upgrade button

Ensure the product has a Purchase URL / Upgrade URL or the WooCommerce product can resolve a purchase permalink. Check Plan Products for the No purchasable product warning.

REST application suddenly fails

Check that the REST key is Active, has the required permission, and the source IP matches Allowed IPs/CIDR. Regenerate separate credentials instead of sharing one broad key.

Extension causes an error

Core isolates failing extensions where possible. Open Extensions for compatibility/recovery information, deactivate the add-on, review System Health/Audit, then use the managed Extensions release/rollback path.

Marketplace shows package but cannot install

Verify Evoxup package entitlement, compatibility, release policy, System Health and recovery/backup readiness. Do not enter or request a Repository Admin/Master/Pairing key on the customer site.

Pre-production checklist

- One test EVO internal product and one test membership plan work.
- At least one license verifies with the intended policy context.
- WooCommerce discovery works if WooCommerce is used.
- External platform security and exact product routing are verified if webhooks/IPN are used.
- Membership Center page renders and all enabled sections are visible.
- Operator roles have only the capabilities they need.
- System Health production/recovery/security/performance/observability/governance gates are reviewed before major releases.
- Backup/restore point and the customer-side managed Extensions/internal release rollback path are tested before production updates.

CHAPTER 18

Glossary

Canonical vocabulary used throughout Evoxup Membership.

EVO Customer ID

Internal commercial customer record. It may link to a WordPress user but is not itself the login identity.

WordPress User ID

The authentication/account identity used by WordPress and the member-facing account system.

Membership Link ID

The assignment row connecting a customer to a plan/product access relationship.

EVO Product

Canonical commercial/licensing product record. It may represent internal, WooCommerce, member-owned or external-source identity.

Membership Plan

Access package containing one or more products plus tier, duration, entitlements and renewal/upgrade relationships.

Entitlement

A capability/feature token granted by a plan/product relationship and evaluated by consumers such as repository/content access.

License Type

Reusable license policy defaults: verification mode, seats/sites and duration.

Verification policy

The context required for a key to be valid: Portable, exact Site URL, Domain, Server IP/CIDR, or Domain + IP.

Activation

A specific site/device usage record consuming license capacity.

Integration

External platform configuration for webhook/IPN/API identity and security.

Product Routing

Mapping from external provider product ID/variant to the real EVO or WooCommerce target.

Webhook Inbox

Stored inbound event evidence including raw payload, normalized payload, verification and processing state.

Extension

Isolated optional component loaded through extension.json / Extension API contracts.

Extensions lifecycle

Customer-side installation, activation/deactivation, update, deletion and recovery surface. Managed Evoxup package actions should not require a central repository admin/pairing secret.

Marketplace

Customer-facing discovery and Evoxup-package entitlement UI. It does not expose central Repository administration credentials.

Update Manager

Standalone Evoxup operator release/security infrastructure package. It is not part of normal commercial customer distribution; required client-side update behavior belongs behind Core Extensions/Marketplace.

Repository Connector

Legacy/private server-to-server pairing bridge for older Repository API deployments. Not a normal commercial customer add-on.

Customer License API

Optional self-hosted API (Free/Advanced) used by a Membership customer to verify that customer's own products/licenses. Its credentials belong to that customer and are independent from Evoxup central repository secrets.

Membership Center

Generated/customizable frontend account/catalog surface backed by canonical Core services.

End of handbook

This edition documents Evoxup Membership Core 1.6.0, 37 customer-facing extensions, and three separately classified Evoxup operator/infrastructure components. Central repository secrets are explicitly outside the commercial customer trust boundary.