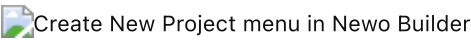


Clover Integration

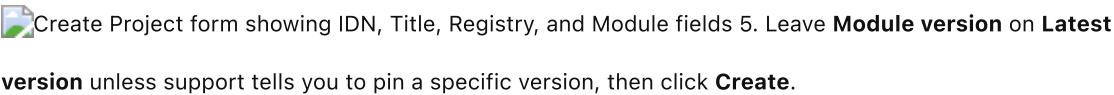
Create the Newo project

If this project already exists in **Newo Builder**, skip this subsection and continue with the integration-specific settings below.

1. In **Newo Builder**, open the projects list and click **Create Project** (or **Create New Project** from the top-right menu).



2. Fill **IDN** and **Title**. The exact names do not matter; use any clear names your team will recognize.
3. In **Registry**, choose the release channel:
 - **staging** — the newest module fixes appear here first. Use it when you need the latest fix, but expect possible unfinished changes.
 - **production** — the final stable version for live projects.
4. In **Module**, select the module for this integration.



5. Leave **Module version** on **Latest version** unless support tells you to pin a specific version, then click **Create**.

Overview

The Clover integration connects a Newo conversational agent to Clover so the agent can:

- create orders from live Clover menu items
- check the status of recent orders
- cancel existing orders

This integration is for order-taking workflows, not appointment scheduling. In NAF terms, the old generic "availability" path is repurposed here for order-status lookup.

Supported Features

Feature	Supported	Notes
Create order	Yes	Uses synchronized Clover items and creates a Clover order
Check order status	Yes	Looks up recent orders for the customer
Cancel order	Yes	Cancels an existing Clover order
OAuth authentication	Yes	Via Clover connector flow
Merchant API token authentication	Yes	Direct token mode for one merchant
Payment capture	No	The integration creates and manages orders, not payment checkout

Multi-merchant selection UI	No	clover_merchant_id is still set explicitly per project
-----------------------------	----	--

Authentication Modes

OAuth mode

Use OAuth when:

- you want Clover connector-based authentication
- you have `clover_app_id` , `clover_app_secret` , and `clover_oauth_code`

Main attributes:

- `clover_auth_type=OAuth`
- `clover_app_id`
- `clover_app_secret`
- `clover_oauth_code`

Token mode

Use token mode when:

- you have a merchant-generated Clover API token
- you want direct single-merchant authentication without the connector bootstrap

Main attributes:

- `clover_auth_type=Token`
- `clover_api_token`
- `clover_merchant_id`

In token mode, connector setup is skipped by design and requests are sent directly with `Authorization: Bearer <token>` .

Main Attributes

User-facing Clover attributes

Attribute	Purpose
<code>clover_auth_type</code>	Selects OAuth or Token mode
<code>clover_api_token</code>	Merchant API token for token mode
<code>clover_oauth_code</code>	One-time OAuth code for connector authorization
<code>clover_app_id</code>	Clover OAuth app id
<code>clover_app_secret</code>	Clover OAuth app secret
<code>clover_base_url</code>	Clover API base URL
<code>clover_merchant_id</code>	Merchant id used in Clover API paths
<code>clover_order_type</code>	Order type label for created orders

<code>clover_availability_days</code>	Recent-order lookback window for status lookup
<code>clover_enable_booking</code>	Enables order creation tool
<code>clover_enable_slot_check</code>	Enables order-status lookup tool
<code>clover_enable_cancellation</code>	Enables order cancellation tool
<code>clover_setup_scenarios</code>	Controls Workflow Builder scenario setup

Hidden or system-managed Clover attributes

Attribute	Purpose
<code>clover_static_data_update_interval</code>	Menu cache refresh interval in seconds
<code>clover_available_items</code>	Serialized Clover item cache used for item matching
<code>clover_menu_prompt_context</code>	Readable menu snapshot for debugging
<code>clover_static_data_updated_at</code>	Timestamp of the last successful menu refresh
<code>clover_last_menu_sync_status</code>	Latest sync result
<code>clover_order_type_id</code>	Resolved Clover order type id
<code>clover_override_agent_attributes</code>	Enables Clover setup overrides for standard NAF settings
<code>setup_persona_id</code>	Internal setup persona id used during connector and tool setup

Legacy Note

Current Clover code no longer uses `clover_check_availability_on_conversation_start`.

If you still see that attribute on an older deployed project, it is a leftover from the legacy Clover implementation and can be treated as deprecated compatibility state rather than an active runtime setting.

Tooling Model

The integration uses custom tools registered into `project_attributes_settings_platform_tools`:

- `clover_create_order_tool`
- `clover_check_order_status_tool`
- `clover_cancel_order_tool`

At the same time, SetupFlow overrides these generic NAF tools so they are not used accidentally:

- `check_availability_tool`
- `create_booking_tool`
- `cancel_booking_tool`

All Clover custom tools are configured with `useConversationHistory: false`.

Operational Notes

- Exact menu matching is driven by the synchronized `clover_available_items` cache.

- Menu grounding is appended into project RAG context rather than being hardcoded in static prompts.
- The integration assumes one Clover merchant per project.
- `project_publish_finish` runs setup, and setup triggers the internal static-data refresh path.

Testing and Verification

Recommended verification sequence after setup or publish:

1. Confirm Clover attributes are present and populated correctly.
2. Confirm `project_attributes_settings_platform_tools` contains Clover custom tools.
3. Trigger a menu refresh and verify `clover_available_items` and `clover_last_menu_sync_status` .
4. Test create-order flow with exact menu item names.
5. Test order-status lookup for the created order.
6. Test cancellation for that same order.