

# Denticon Integration

## What the AI can do

### Summary

**Denticon Integration** connects **Denticon** (PlanetDDS dental practice management system) with **Newo Platform**.

It enables:

- Checking real-time availability for dental appointments with appointment type matching
- Booking appointments with automatic new/existing patient detection
- Cancelling appointments with intelligent reason code selection
- Looking up existing patients and their upcoming appointments
- Multi-location dental office support with LLM-powered location matching

This integration is designed for **dental clinic administrators and front-desk teams** who need **automated appointment scheduling through a conversational AI agent** (voice or chat).

### Common use cases

- When a patient asks about available slots → Check Denticon availability with appointment type resolution and return open time slots
- When a patient wants to book an appointment → Search/create patient record and schedule appointment in Denticon
- When a patient wants to cancel an appointment → Cancel with automatically matched cancellation reason code
- When a clinic has multiple locations → Agent uses LLM to match patient's preferred location from natural language
- When a patient asks about their upcoming visits → Look up patient by name/DOB and retrieve future appointments
- When a new patient books → Automatically detect new vs existing patient and handle accordingly

### Features at a glance

| Feature                    | Supported | Notes                                            |
|----------------------------|-----------|--------------------------------------------------|
| Check Availability         | ✓         | With LLM-based appointment type matching         |
| Book Appointment           | ✓         | Auto-detects new vs existing patient             |
| Cancel Appointment         | ✓         | With LLM-based cancellation reason selection     |
| Patient Lookup             | ✓         | By name + date of birth                          |
| Upcoming Appointments      | ✓         | Future appointments filtered by timezone         |
| Multi-Location Support     | ✓         | LLM-powered location matching (Gemini)           |
| Single-Location Mode       | ✓         | Simplified setup with pre-selected location      |
| Timezone Handling          | ✓         | Converts slots to clinic's local timezone        |
| OAuth 2.0 Token Management | ✓         | Auto-refresh with retry logic (up to 3 attempts) |

|                        |   |                               |
|------------------------|---|-------------------------------|
| Reschedule Appointment | ✗ | Cancel + rebook as workaround |
| Historical Data Import | ✗ | Only events after activation  |

## Before You Start

Before installation:

- Active **Denticon** (PlanetDDS) account with API access
- **OAuth 2.0 Client ID** and **Client Secret**
- **API Subscription Key** (PDDS-Subscription-Key)
- Denticon API base URL (default: `https://api.planetdds.com/denticon` )

## Setup

### 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).



Create New Project menu in Newo Builder

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.



Create Project form showing IDN, Title, Registry, and Module fields 5. Leave **Module version** on **Latest**

**version** unless support tells you to pin a specific version, then click **Create**.

### 3.1 Step 1 — Generate Credentials in Denticon

1. Log in to your **Denticon** / PlanetDDS account
2. Navigate to the API management or developer section
3. Create an **OAuth 2.0 application** to obtain:
  - **Client ID**
  - **Client Secret**
4. Obtain your **PDDS-Subscription-Key** from the API portal
5. Store all credentials securely

### 3.2 Step 2 — Connect in Platform

1. Open **Newo** → **Projects**
2. Set the following attributes in **Builder / Attributes**:

| Attribute | Required | Description |
|-----------|----------|-------------|
|-----------|----------|-------------|

|                                |   |                                                                                                              |
|--------------------------------|---|--------------------------------------------------------------------------------------------------------------|
| denticon_client_id             | ✓ | OAuth 2.0 Client ID                                                                                          |
| denticon_client_secret         | ✓ | OAuth 2.0 Client Secret                                                                                      |
| denticon_subscription_key      | ✓ | API Subscription Key (PDDS-Subscription-Key)                                                                 |
| denticon_base_url              | ✗ | API base URL (default: <a href="https://api.planetdds.com/denticon">https://api.planetdds.com/denticon</a> ) |
| denticon_multilocation         | ✗ | Enable multi-location support (default: True)                                                                |
| denticon_enable_booking        | ✗ | Enable booking capability (default: True)                                                                    |
| denticon_enable_cancellation   | ✗ | Enable cancellation capability (default: True)                                                               |
| denticon_enable_slot_check     | ✗ | Enable availability check (default: True)                                                                    |
| denticon_check_existing_client | ✗ | Enable patient lookup (default: True)                                                                        |
| denticon_enable_client_history | ✗ | Enable appointment history (default: False)                                                                  |

3. Click **Save + Publish All**

4. On publish, the SetupFlow automatically:

- Authenticates via OAuth 2.0 (client credentials grant)
- Fetches clinic locations from Denticon
- Fetches available providers
- Fetches cancellation reason codes
- Configures booking tools for the agent

## How to use the integration

This section explains how the integration works, how to configure automation, and how to test it.

### 4.1 How the Integration Works

The integration operates based on **Triggers and Actions** via the event-driven system.

- A **Trigger** is a system event that starts a flow
- An **Action** is the API operation performed in Denticon

### Where to test

Testing can be done through the **Newo conversation interface** (chat or voice channel) by interacting with the agent. Use the chat transcript and external-system console to confirm the result. Cancellation can also be tested via the `test_integration` webhook event.

## How to test that everything works

**To test the integration:**

1. **Setup:** Publish the project and verify logs show successful OAuth authentication, office fetch, and cancellation codes fetch
2. **Availability:** Ask the agent "What slots are available for a cleaning on Monday?" — verify slots returned with correct appointment type
3. **Booking:** Complete a booking conversation — verify appointment appears in Denticon dashboard

4. **Cancellation:** Request cancellation of a booked appointment — verify status changed in Denticon with correct reason code
5. **Patient Lookup:** Ask "Do I have any upcoming visits?" — verify patient is found and future appointments returned

**If no action occurs:**

- Ensure OAuth credentials are set correctly ( `denticon_client_id` , `denticon_client_secret` , `denticon_subscription_key` )
- Verify `denticon_base_url` points to the correct endpoint
- Check that OAuth token was obtained (look for `denticon_access_token` in attributes)
- Confirm the integration was re-published after configuration changes
- Check that the relevant feature flags are enabled
- In multi-location mode: verify `denticon_offices` was populated during setup

*Note: This integration performs real operations in Denticon. Appointments created or cancelled through the agent are reflected in the live Denticon system.*

## FAQ

**Q: Does the integration import historical appointments?** A: No. Only appointments created or managed after activation are processed. Patient lookup shows future appointments only.

**Q: Can I connect multiple Denticon accounts?** A: Each integration instance supports one set of OAuth credentials. For multiple accounts, create separate integration instances.

**Q: What happens if a patient already exists in Denticon?** A: The booking flow searches for existing patients by first name, last name, and date of birth. If a match is found, the existing patient record is used with `isNewPatient=false` . If not, the appointment is created with `isNewPatient=true` .

**Q: How does appointment type matching work?** A: When a patient describes their visit reason (e.g., "teeth cleaning", "checkup"), an LLM (Gemini) matches it to available production types configured for the selected office in Denticon. For generic requests, it defaults to the general exam option.

**Q: How does multi-location mode work?** A: When enabled ( `denticon_multilocation=True` ), the agent uses an LLM to match the patient's natural language location description to available clinic offices. The matched location is cached for the session.

**Q: What happens if the OAuth token expires?** A: The integration automatically detects HTTP 401 responses and refreshes the token using the client credentials grant. It retries the failed request up to 3 times.

**Q: How are cancellation reasons selected?** A: During setup, all available cancellation codes are fetched from Denticon. When a cancellation occurs, an LLM analyzes the cancellation notes and selects the most appropriate reason code. If notes are empty, a neutral default reason is used.