



Test Technique — Bot + Rebrand ChatWoot (3 Paliers)

 **Début** : Dimanche 10/08/2025

 **Remise** : jeudi 14/08/2025 à 12h00

 **Soutenance** : Jeudi 14/08/2025 après-midi (15 min / candidat)

Pour toute question: 0532267440 - Aaron

בהצלחה

Objectif

Construire un **bot de service client fonctionnel** pour **Mandelboim** et réaliser un **rebranding basique de ChatWoot**, en travaillant uniquement à partir des ressources publiques disponibles.

PALIER 1

But : Créer un bot n8n qui gère une FAQ et la prise de lead.

But : Créer un bot n8n fonctionnel, basé sur un **prompt-système** pour un agent de service client, capable de gérer :

- **FAQ** en hébreu.
- **Prise de lead** (enregistrement d'informations clients).
- **Prise de rendez-vous** (stockage des créneaux choisis).

Ce palier met en place les bases, mais **le catalogue produit ne sera pas complet tant que le Palier 2 ne sera pas atteint.**

Tâches

1. déployer n8n:

- github: <https://github.com/n8n-io/n8n>
- Docker hub: <https://hub.docker.com/r/n8nio/n8n>

2. Catalogue produits

- Scraper **≥ 20 / 30 produits** (bonus catalogue complet + méthode de scraping) le site Mandelboim: <https://www.mandelboim.com/>
- Champs obligatoires : title_he, url, price?, sizes, description_he, image_url.
- Format : Google Sheet ou CSV.
- **⚠ Remarque :** Le catalogue complet et la recherche avancée seront implémentés au **Palier 2.** (objectif palier 1: livrer sans RAG sous Google sheet a ce niveau)

3. FAQ

- **≥ 8 Q/R** en hébreu issues du site Mandelboim (horaires, livraison, retours, etc.).
- Format : Google Sheet ou CSV.

4. Prompt Système

Créer un prompt-système clair et structuré (en hébreu - markdown si possible) pour guider l'agent virtuel :

- Identifier si l'utilisateur pose une question de FAQ, cherche un produit (catalogue partiel) ou souhaite laisser ses coordonnées.
- Formuler les réponses en langage naturel, courtois et clair.
- Demander les informations nécessaires pour la prise de lead ou de rendez-vous.

5. Workflow n8n

- Webhook entrant → détection intent (FAQ, produit, lead).
- FAQ → réponse directe.
Lead → enregistrement dans Google Sheets (colonnes : timestamp, full_name, phone, city, interest_note, preferred_time, source, consent, status).

Livrables

- Export .json du workflow n8n.
- Sheets/CSV FAQ & catalogue.
- Google Sheet leads avec données de test.

PALIER 2

But : Étendre le bot n8n avec un module de **recherche produit avancée (RAG)** et un **catalogue complet**, permettant à l'agent de trouver rapidement les produits pertinents ou de proposer la prise de contact si aucun résultat fiable n'est trouvé.

⚠ important: Pensez à dupliquer votre workflow avant de continuer

Tâches

1. Catalogue Produits (complet)

- Compléter le **scraping intégral** du site Mandelboim : <https://www.mandelboim.com/>
- Champs obligatoires (identiques au Palier 1) :
 - title_he
 - url
 - price (ou N/A)
 - sizes
 - description_he
 - image_url
- Ajouter un champ **tags** (mots-clés) pour améliorer la recherche.
- Format : Google Sheet ou CSV.

2. Base pgvector (RAG)

- Mettre en place **PostgreSQL avec l'extension pgvector** pour stocker les embeddings des produits.
- **Optionnel** : Utiliser un service externe (ex. Supabase, Railway Postgres) si la mise en place locale n'est pas possible.
- Générer les embeddings pour title_he, description_he et tags.

3. Module de Recherche Produit

- Intégrer un **module RAG** dans n8n :
 - Recherche par **mots-clés** dans le catalogue complet (titre, description, tags).
 - Retourner **2 à 3 produits pertinents** au format :
 - Titre
 - Prix
 - Lien produit

- Améliorer le **prompt-système** pour que l'agent sache utiliser la recherche produit avant de répondre.

4. Fallback intelligent

- Si **aucun résultat fiable** → pensez à switcher sur un fallback intelligent. ex:
 - proposer à l'utilisateur de laisser ses coordonnées (prise de lead).
 - Enregistrer dans Google Sheets (même structure que Palier 1).

5. Jeu de test

- Fournir un fichier .md avec **5 scénarios de questions produit** et leurs **réponses attendues** (incluant cas où aucun produit n'est trouvé).

Livrables

- Workflow n8n mis à jour (export .json).
- Google Sheet / CSV **catalogue complet**.
- Base PostgreSQL avec embeddings (dump SQL ou lien).
- Fichier .md avec scénarios de test.
- Fichier texte **prompt-système mis à jour** (incluant la logique de recherche).

PALIER 3

But : Rebrand et déployer ChatWoot.

Liens utiles

- **ChatWoot GitHub** : <https://github.com/chatwoot/chatwoot>
- **Docker Hub ChatWoot** : <https://hub.docker.com/r/chatwoot/chatwoot>

Tâches

1. Fork & build

- Fork du dépôt officiel ChatWoot sur votre GitHub.
- Modifier le branding :
 - Nom affiché : “Chatwoot rebranded”
 - Logo, favicon, couleurs CSS/SCSS
palette de couleur et logo de votre choix. soyez créatifs 😊

2. Déploiement

- **Option A** : Docker Compose local (Postgres + Redis + App).
- **Option B** : Déploiement Railway (captures config).

Livrables

- Lien du fork GitHub.
- docker-compose.yml ou instructions Railway.
- Screenshots écran login + dashboard rebrandés.
- README-rebrand.md (setup, build, run).