

# Documentation fonctionnelle - Module WhatsApp b-2-b (Odoo 16)

## 1. Objectifs du module

Le module **WhatsApp b-2-b** permet de :

- **Centraliser les échanges WhatsApp** avec les clients directement dans Odoo.
- **Automatiser les notifications** liées au cycle de vente et de facturation (commandes, factures, rappels).
- **Proposer des scénarios interactifs** (boutons, menus) pour valider des commandes, envoyer des liens de paiement, etc.
- **Suivre l'historique des conversations** par client (traçabilité commerciale et comptable).

Le module s'adresse principalement :

- **Aux équipes commerciales** (suivi des commandes et relances clients).
  - **Aux équipes facturation/comptabilité** (envoi de factures, rappels d'impayés).
  - **Aux administrateurs fonctionnels** (paramétrage des scénarios, templates, actions automatiques).
- 

## 2. Prérequis fonctionnels

- **Compte Meta WhatsApp Business Cloud** opérationnel.
  - Au moins un **numéro WhatsApp Business** configuré dans Meta.
  - Accès administrateur à Odoo pour paramétrer :
    - o `web.base.url`
    - o la configuration WhatsApp (`whatsapp.config`)
    - o le webhook dans Meta Business Suite.
- 

## 3. Menus et navigation dans Odoo

- **Menu principal WhatsApp**
  - o **WhatsApp ► Configuration**
    - Fiche de configuration WhatsApp (`whatsapp.config`)
    - Actions de test : vérification des paramètres, synchronisation des templates, diagnostics.
  - o **WhatsApp ► Messages**
    - Liste de tous les messages WhatsApp (`whatsapp.message`).
    - Filtres par direction (entrant/sortant), type, statut.
    - Bouton *Répondre* pour ouvrir un wizard d'envoi.

- **WhatsApp ► Conversations**
    - Vue consolidée des conversations par numéro / partenaire (`whatsapp.conversation`).
  - **WhatsApp ► Templates**
    - Liste des templates approuvés par Meta (`whatsapp.template`).
  - **WhatsApp ► Envoyer un message**
    - Wizard `whatsapp.send.message` : envoi manuel à un numéro ou un partenaire.
  - **WhatsApp ► Envoyer un message partenaire**
    - Wizard `whatsapp.send.partner.message` : envoi ciblé à des partenaires Odoo.
  - **WhatsApp ► Actions de boutons**
    - Définition des actions exécutées lors des clics sur les boutons interactifs (`whatsapp.button.action`).
  - **WhatsApp ► Scénarios interactifs**
    - Configuration des scénarios multi-étapes avec boutons (`whatsapp.interactive.scenario`).
  - **Intégrations dans les objets métier**
    - **Contacts** (`res.partner`)
      - Bouton *Envoyer WhatsApp* (si activé dans la config).
    - **Commandes clients** (`sale.order`)
      - Bouton WhatsApp (création, validation, détails).
      - Champs d'indicateurs (messages déjà envoyés, validation via WhatsApp, etc.).
    - **Factures clients** (`account.move`)
      - Bouton WhatsApp (envoi facture, détails, rappels).
      - Champs d'indicateurs (facture envoyée, rappel impayé, message résiduel, etc.).
- 

## 4. Scénarios fonctionnels principaux

### 4.1. Envoi manuel de messages

- **Depuis le menu WhatsApp**
  - Aller dans **WhatsApp ► Envoyer un message WhatsApp**.
  - Renseigner :
    - la configuration (si plusieurs),

- le numéro ou le partenaire,
  - le texte du message,
  - éventuellement la prévisualisation des liens.
- Cliquer sur **Envoyer**.
- Le message est loggé dans `whatsapp.message` et rattaché à une conversation.
- **Depuis un contact**
  - Ouvrir un partenaire.
  - Cliquer sur le bouton **Envoyer WhatsApp**.
  - Le wizard est pré-rempli avec le numéro du partenaire.

## 4.2. Réponse à un message entrant

- Aller dans **WhatsApp ► Messages**.
- Ouvrir un message **entrant** (direction = *Entrant*).
- Cliquer sur **Répondre**.
- Un wizard d'envoi s'ouvre avec :
  - le numéro,
  - le contact,
  - la configuration.
- Saisir la réponse et envoyer.

---

## 5. Intégration avec les commandes clients

### 5.1. Envoi automatique à la création de commande

- Condition : `auto_send_order_creation` **activé** dans la configuration WhatsApp.
- À chaque création de commande (`sale.order`) :
  - le module vérifie si le partenaire a un numéro de téléphone,
  - envoie un **message interactif** avec 3 boutons :
    - **Valider** la commande,
    - **Annuler** la commande,
    - **Voir détail** (et éventuellement lien PDF).
  - Les indicateurs suivants sont mis à jour :
    - `x_whatsapp_creation_sent`
    - `x_whatsapp_creation_sent_date`.

Fonctionnellement, cela permet au client de **confirmer ou refuser sa commande** directement depuis WhatsApp sans se connecter à un portail.

## 5.2. Validation / rejet via boutons WhatsApp

- Lorsque le client clique sur un bouton (ex. `btn_validate_order_{id}` ou `btn_cancel_order_{id}`) :
  - o Le webhook reçoit l'événement.
  - o Le module retrouve la commande concernée.
  - o Les actions configurées dans `whatsapp.button.action` sont exécutées :
    - changement d'état de la commande (ex. en *Confirmé*),
    - envoi d'un message de confirmation ou d'annulation,
    - mise à jour des champs :
      - `x_whatsapp_validated` / `x_whatsapp_rejected`,
      - `x_whatsapp_validation_sent`,  
`x_whatsapp_validation_sent_date`.

Ce mécanisme permet d'implémenter un **flux de validation 100 % par WhatsApp**.

## 5.3. Envoi de détails de commande

- Via un bouton ou une action dédiée, le module peut :
  - o envoyer un **récapitulatif détaillé** de la commande :
    - client, montant, date, état,
    - liste des lignes produits (désignation, quantité, prix, sous-total),
  - o éventuellement inclure un **lien de téléchargement PDF** de la commande.
- L'indicateur `x_whatsapp_details_sent` est mis à jour.

---

## 6. Intégration avec les factures clients

### 6.1. Envoi automatique de la facture à la validation

- Lors du passage d'une facture client à l'état **Postée** :
  - o Le module détecte le changement d'état dans `write()`.
  - o Si la configuration est active et qu'un numéro est disponible :
    - Génération du **PDF facture**.
    - Création d'un attachement public.
    - Construction d'une URL de téléchargement.
    - Envoi via WhatsApp (document + message explicatif).
  - o Mise à jour des champs :

- `x_whatsapp_invoice_sent`
- `x_whatsapp_invoice_sent_date`.

Fonctionnellement, le client reçoit automatiquement sa facture par WhatsApp dès sa validation.

## 6.2. Envoi de détails de facture et liens de paiement

- Depuis une facture, l'action **Envoyer détails facture WhatsApp** :
  - Vérifie la présence du partenaire et du numéro.
  - Construit un **message détaillé** :
    - informations générales (numéro, dates, montants),
    - liste des lignes (produit, quantité, prix, sous-total),
    - montant total, montant restant.
  - Si le système de paiement est intégré (Wave, Orange Money, etc.) :
    - génère ou récupère les **liens de paiement**,
    - insère les liens (ou boutons) dans le message ou via templates/actions associées.

Ce flux permet d'orienter le client vers un **paiement immédiat** depuis son téléphone.

## 6.3. Notification de changement du montant résiduel

- À chaque modification qui impacte `amount_residual` :
  - le module compare l'ancien et le nouveau montant,
  - si le montant change de façon significative et reste > 0 :
    - envoi d'un message récapitulatif (montant payé, montant restant),
    - éventuellement lien vers la facture.
  - Mise à jour des champs :
    - `x_whatsapp_residual_sent`
    - `x_whatsapp_residual_sent_date`.

Cela permet d'**informer le client** en cas de paiement partiel, d'avoir un suivi clair du reste à payer.

## 6.4. Rappels automatiques de factures impayées (Cron)

- Condition : `auto_send_unpaid_invoices` activé dans `whatsapp.config`.
- Un **cron quotidien** (défini dans les données `whatsapp_cron_data.xml`) :
  - recherche les factures impayées dépassant de `unpaid_invoice_days` jours la date d'échéance,
  - envoie un **message de rappel** au client via WhatsApp,
  - met à jour :
    - `x_whatsapp_unpaid_reminder_sent`

- `x_whatsapp_unpaid_reminder_sent_date`.

Fonctionnellement, cela automatise la **relance des impayés** sans intervention manuelle.

---

## 7. Scénarios interactifs et actions de boutons

### 7.1. Scénarios interactifs (`whatsapp.interactive.scenario`)

Un scénario interactif permet de définir un **parcours conversationnel** composé de :

- un **message initial**,
- jusqu'à 3 **boutons** (ID, titre, réponse textuelle),
- des **enchaînements** vers d'autres scénarios (multi-étapes).

Exemples de cas d'usage :

- Validation de commande en plusieurs étapes (confirmation → choix du mode de paiement).
- Menu d'accueil (ex. 1 : Infos commande, 2 : Factures, 3 : Assistance).

### 7.2. Actions de boutons (`whatsapp.button.action`)

Pour chaque ID de bouton (ex. `btn_validate_order`, `btn_pay_wave`, etc.), on peut définir :

- un **type d'action** :
  - simple message de réponse,
  - exécution de **code Python** (mise à jour d'un document, création d'un paiement, etc.),
- un **message de réponse par défaut**.

Fonctionnellement, cela permet de lier les clics sur les boutons WhatsApp à des **actions métier Odoo** :

- validation de commande,
  - annulation,
  - envoi de facture,
  - génération de liens de paiement,
  - ouverture de nouveaux scénarios interactifs.
- 

## 8. Gestion des conversations et de l'historique

- **Conversations** (`whatsapp.conversation`)
  - Une conversation est créée par numéro / contact.
  - Elle regroupe tous les `whatsapp.message` liés.
  - Permet à un commercial de voir **tout l'historique WhatsApp** d'un client en un coup d'œil.
- **Messages** (`whatsapp.message`)

- Chaque message stocke :
    - direction (entrant/sortant),
    - type (texte, image, template, interactif, etc.),
    - statut logique (reçu, envoyé, lu, erreur),
    - contenu, pièces jointes, réponses API, aide sur l'erreur.
  - Les messages peuvent être filtrés par :
    - contact,
    - statut,
    - type de message,
    - période.
- 

## 9. Paramétrage fonctionnel recommandé

### – Étape 1 : Configuration de base

- Créer un enregistrement dans **WhatsApp** ► **Configuration**.
- Renseigner :
  - Phone Number ID,
  - Access Token,
  - Facebook App Secret,
  - Verify Token,
  - Webhook URL.
- Tester avec **Vérifier les paramètres**.

### – Étape 2 : Webhook Meta

- Dans Meta Business Suite :
  - configurer l'URL webhook Odoo,
  - définir le même **Verify Token**,
  - activer les champs **messages** et **message\_status**.

### – Étape 3 : Templates

- Créer et faire approuver les templates dans Meta.
- Lancer **Synchroniser les templates** depuis la config WhatsApp.

### – Étape 4 : Scénarios et actions

- Définir les **actions de boutons** pour les IDs standards utilisés (validation commande, paiement, etc.).
- Créer les **scénarios interactifs** correspondant aux parcours clients souhaités.

- **Étape 5 : Tests bout en bout**
    - Créer une commande test et vérifier :
      - la réception du message de création,
      - le bon fonctionnement des boutons,
      - la création de la facture et l'envoi WhatsApp.
- 

## 10. Résumé fonctionnel

- **Avant-vente & vente**
  - Validation de commande par WhatsApp.
  - Envoi des détails de commande.
  - Menus et scénarios interactifs d'accompagnement client.
- **Facturation & recouvrement**
  - Envoi automatique des factures en PDF.
  - Notifications de changements de montant résiduel.
  - Rappels automatiques d'impayés.
- **Relation client**
  - Historique complet des échanges par client.
  - Réponse directe aux messages entrants.
  - Centralisation des interactions dans Odoo.