

Documentation Fonctionnelle — Intégration Sweego Mail

Sommaire

1. Présentation du module	2
2. Prérequis	3
2.1. Côté Sweego	3
2.2. Côté Odoo	3
3. Installation	3
4. Configuration	4
4.1. Paramètres généraux Sweego	4
4.2. Domaine alias par société	4
4.3. Serveur d'envoi (SMTP Sweego)	5
4.3.1. Champs spécifiques Sweego	5
4.4. Configuration des webhooks dans Sweego	5
4.4.1. Webhook Sortant (Outbound)	5
4.4.2. Webhook Entrant (Inbound)	6
5. Fonctionnement	6
5.1. Envoi d'un email	6
5.2. Résolution du domaine Reply-To (multi-société)	6
5.3. Suppression différée des emails	7
5.4. Gestion des bounces et emails en erreur	7
5.5. Réception des réponses (Inbound)	8
5.6. Pièces jointes inbound	9
5.7. Détection des boucles email	9
5.8. Vérification des signatures webhook	9
5.9. Architecture de traitement outbound	9
6. Tâche planifiée — Synchronisation des logs	10
6.1. Rôle	10
6.2. Comportement	10
6.3. Paramétrage	10
7. Mécanisme du token Reply-To	10
7.1. Format	10
7.2. Décodage	11
8. Champs techniques ajoutés	11
8.1. Sur <code>ir.mail_server</code>	11
8.2. Sur <code>mail.mail</code>	11
8.2.1. Méthodes sur <code>mail.mail</code>	11

8.3. Sur <code>mail.message</code>	12
8.4. Dans <code>ir.config_parameter</code>	12
9. Endpoints webhook	12
10. Dépannage	12
10.1. Les webhooks sont rejetés (401)	12
10.2. Les réponses n'apparaissent pas dans le chatter	12
10.3. Les pièces jointes inbound n'apparaissent pas	13
10.4. Les emails ne sont pas supprimés après livraison	13
10.5. L'enveloppe reste grise après un bounce	13
10.6. La tâche planifiée ne traite aucun email	13
10.7. Le serveur refuse de se créer	13
11. Glossaire	13

Module Odoo 19 — `mangono_sweego_mail`

Version	1.3
Date	Juin 2026
Auteur	Mangono
Compatibilité	Odoo 19

1. Présentation du module

Le module `mangono_sweego_mail` est une extension Odoo 19 qui intègre [Sweego](#), plateforme française d'envoi de messages transactionnels, au cœur du système de messagerie d'Odoo.

Il apporte quatre fonctionnalités principales :

- Envoi des emails Odoo via l'infrastructure SMTP de Sweego
- Traitement des réponses email de vos clients directement dans le chatter des documents Odoo (commandes, factures, etc.)
- Suppression fiable des emails de la file d'attente uniquement après confirmation de livraison par Sweego, via webhook ou tâche planifiée
- Signalement des emails en erreur (bounce) avec mise à jour du statut dans le chatter (enveloppe rouge)

Sweego est une plateforme d'envoi email française offrant un relais SMTP performant, des statistiques de délivrabilité et des webhooks pour le suivi des envois.

2. Prérequis

2.1. Côté Sweego

- Un compte Sweego actif (<https://app.sweego.io>)
- Un domaine d'envoi vérifié et configuré dans Sweego
- Un domaine inbound configuré avec les MX pointant vers inbound.sweego.co
- Les identifiants SMTP fournis par Sweego
- Les secrets de signature webhook générés depuis le tableau de bord Sweego
- La clé API Sweego (section **API** du tableau de bord)
- L'UUID client Sweego (visible dans les paramètres du tableau de bord)

2.2. Côté Odoo

- Odoo 19 avec le module `mail` installé
- Le module `mangono_sweego_base` installé (fournit la clé API et l'UUID client partagés)
- Un domaine alias configuré sur chaque société (**Paramètres** → **Technique** → **Email** → **Domaines alias**)
- Accès administrateur à Odoo
- L'URL publique de votre instance Odoo (accessible par Sweego pour les webhooks)

3. Installation

Copiez les dossiers `mangono_sweego_base` et `mangono_sweego_mail` dans votre répertoire d'addons Odoo, puis installez `mangono_sweego_mail` depuis le menu **Paramètres** → **Activer le mode développeur** → **Applications**. `mangono_sweego_base` sera installé automatiquement en tant que dépendance.

Structure du module

```
mangono_sweego_mail/  
├── models/  
│   ├── ir_mail_server.py    ← Serveur sortant, logique token, cron  
│   ├── mail_mail.py        ← Suppression différée, gestion bounces  
│   └── mail_message.py     ← Champ sweego_message_id  
├── controllers/  
│   └── webhook.py          ← Endpoints outbound + inbound  
├── data/  
│   └── ir_cron.xml          ← Tâche planifiée de synchronisation  
├── views/  
│   └── ir_mail_server.xml   ← Formulaire étendu  
└── i18n/  
    └── fr.po                ← Traductions françaises
```

```

mangono_sweego_base/          ← Module de base partagé (dépendance)
├── models/
│   ├── res_config_settings.py ← Clé API et UUID client
│   └── views/
│       └── res_config_settings_views.xml
└── i18n/
    └── fr.po

```

4. Configuration

4.1. Paramètres généraux Sweego

Les paramètres de connexion à l'API Sweego sont gérés par le module `mangono_sweego_base` et partagés entre tous les modules Sweego installés (`mangono_sweego_mail`, `mangono_sweego_sms`).

Accédez à **Paramètres** → **Sweego** pour renseigner ces paramètres.

Champ	Description
Clé API Sweego	Clé API disponible dans votre tableau de bord Sweego (section API). Utilisée pour les appels à l'API REST : téléchargement des pièces jointes inbound et synchronisation des logs de livraison.
UUID Client Sweego	UUID client visible dans votre tableau de bord Sweego. Requis pour construire les URLs d'appel à l'API REST (téléchargement des pièces jointes inbound).



Ces paramètres sont stockés dans `ir.config_parameter` sous les clés `mangono_sweego.api_key` et `mangono_sweego.client_uuid`. Ils sont globaux à l'instance Odoo et partagés entre tous les modules Sweego installés.

4.2. Domaine alias par société

Le domaine inbound utilisé pour le routage des réponses email est résolu automatiquement depuis le **domaine alias** configuré sur chaque société Odoo. Il n'est plus nécessaire de le saisir manuellement sur le serveur d'envoi.

Pour configurer le domaine alias : **Paramètres** → **Technique** → **Email** → **Domaines alias** → **Créer**.



En configuration multi-société, chaque société peut avoir son propre domaine alias. Le module détermine automatiquement le bon domaine lors de l'envoi d'un email en lisant `company_id.alias_domain_id` sur l'enregistrement source (commande, facture, etc.).

4.3. Serveur d'envoi (SMTP Sweego)

Accédez à **Paramètres** → **Technique** → **Email** → **Serveurs d'envoi**, puis créez un nouveau serveur.

Champ	Valeur / Description
Nom	Sweego (ou tout autre nom descriptif)
Authentification	Mangono Sweego SMTP — sélectionner cette option dans la liste déroulante
Serveur SMTP	smtp.sweego.io — à saisir manuellement
Port	2525 (rempli automatiquement à la sélection de l'authentification)
Chiffrement	STARTTLS (rempli automatiquement)
Nom d'utilisateur	Votre identifiant SMTP Sweego (obligatoire)
Mot de passe	Votre mot de passe SMTP Sweego



En sélectionnant l'authentification **Mangono Sweego SMTP**, le port (2525) et le chiffrement (STARTTLS) sont remplis automatiquement. L'hôte SMTP (**smtp.sweego.io**) est remis à blanc et doit être saisi manuellement.

4.3.1. Champs spécifiques Sweego

Champ	Description
Secret webhook sortant Sweego	Secret HMAC fourni par Sweego pour vérifier la signature des webhooks outbound. Copier tel quel depuis le tableau de bord Sweego (encodé en base64). Sert également à signer les tokens Reply-To.
Secret webhook entrant Sweego	Secret HMAC fourni par Sweego pour vérifier la signature des webhooks inbound. Distinct du secret sortant.
Suppression différée	Si coché, les emails ne sont supprimés qu'après confirmation de livraison par Sweego (webhook ou tâche planifiée).

4.4. Configuration des webhooks dans Sweego

Connectez-vous à votre tableau de bord Sweego et configurez les deux webhooks suivants.

4.4.1. Webhook Sortant (Outbound)

Paramètre	Valeur
URL	https://votre-odoo.com/sweego/webhook/outbound
Événements	email_sent , delivered , soft-bounce , hard-bounce (au minimum)

Paramètre	Valeur
Secret	Copier ici et reporter dans Secret webhook sortant Sweego dans Odoo

4.4.2. Webhook Entrant (Inbound)

Paramètre	Valeur
URL	https://votre-odoo.com/sweego/webhook/inbound
Événements	<code>email_inbound</code>
Secret	Copier ici et reporter dans Secret webhook entrant Sweego dans Odoo



Les secrets webhook sont encodés en base64 par Sweego. Copiez-les tels quels dans les champs Odoo — le module se charge du décodage avant le calcul HMAC-SHA256. Les secrets outbound et inbound sont distincts et indépendants.

5. Fonctionnement

5.1. Envoi d'un email

Lorsqu'un utilisateur envoie un email depuis Odoo (devis, facture, bon de commande, message depuis le chatter...), le module intercepte l'envoi et injecte automatiquement deux éléments dans l'email :

- Un en-tête `X-Message-Id` contenant l'identifiant interne Odoo du message. Cet identifiant est renvoyé par Sweego dans ses webhooks et logs, et permet de retrouver le message correspondant.
- Un `Reply-To` personnalisé au format `reply+TOKEN@domaine-alias`, où le token encode de façon sécurisée le modèle Odoo et l'identifiant de l'enregistrement (ex : `sale.order#23`). Le domaine est résolu depuis `company_id.alias_domain_id` de l'enregistrement source, avec fallback sur la société courante.



Sweego peut ré-encoder l'en-tête `X-Message-Id` en RFC 2047 (ex : `=?utf-8?q?...?=?`) lorsque sa longueur dépasse 78 caractères — ce qui arrive systématiquement avec des domaines alias basés sur un UUID. Le module décode automatiquement cet encodage lors de la réception des webhooks et des logs de synchronisation.

5.2. Résolution du domaine Reply-To (multi-société)

Le domaine inbound utilisé dans l'adresse Reply-To est déterminé dynamiquement à l'envoi :

1. Le module lit `record.company_id.alias_domain_id.name` sur l'enregistrement source
2. Si l'enregistrement n'a pas de `company_id`, la société courante (`self.env.company`) est utilisée en

fallback

3. Si aucun domaine alias n'est configuré sur la société, un warning est loggé et le Reply-To reste celui d'Odoo par défaut — les réponses ne seront pas routées dans le chatter



En configuration multi-société, les emails de chaque société utilisent automatiquement le bon domaine inbound, sans aucune configuration manuelle sur le serveur d'envoi.

5.3. Suppression différée des emails

Par défaut dans Odoo, un email est supprimé de la table `mail.mail` dès qu'il est remis au serveur SMTP. Avec Sweego, le module retarde cette suppression jusqu'à confirmation de livraison ou signalement d'erreur.

La confirmation peut arriver par deux chemins complémentaires :

- **Webhook outbound** : Sweego notifie Odoo en temps réel lors de la livraison ou du bounce
- **Tâche planifiée** : un cron interroge périodiquement l'API Sweego pour récupérer les livraisons manquées

Table 1. Cycle de vie d'un email

Étape	État	Description
1	Envoi SMTP	L'email est transmis à Sweego via SMTP. <code>mail.notification.notification_status</code> passe à <code>sent</code> .
2	Attente confirmation	<code>sweego_pending_delete = True</code> . L'email reste en base, la suppression est suspendue.
3	Webhook <code>email_sent</code>	Sweego confirme la prise en charge — log informatif uniquement.
4a	Webhook <code>delivered</code>	Sweego notifie la livraison en temps réel. L'enregistrement <code>mail.mail</code> est supprimé.
4b	Cron synchronisation	Si le webhook <code>delivered</code> n'a pas été reçu, le cron confirme les livraisons manquées → suppression.
4c	Bounce (webhook ou cron)	<code>mail.mail.state = exception</code> . <code>mail.notification.notification_status = exception</code> → enveloppe rouge dans le chatter.

5.4. Gestion des bounces et emails en erreur

Lorsqu'un email ne peut pas être livré, Sweego signale l'erreur soit via le webhook outbound (événements `soft-bounce` / `hard-bounce`) soit via l'API de logs (statut `undelivered`).

Le module traduit ces événements en états Odoo :

Événement Sweego	failure_type Odoo	Cause
soft-bounce	mail_smtp	Échec temporaire (boîte pleine, serveur indisponible...)
hard_bounce	mail_email_invalid	Rejet permanent (adresse invalide)
undelivered (bounce hard)	mail_email_invalid	Idem via l'API de logs
undelivered (bounce soft)	mail_smtp	Idem via l'API de logs
undelivered (blacklist)	mail_bl	Adresse présente dans <code>x-emails-blacklisted</code>
undelivered (autre)	unknown	Raison non classifiable

Lors d'un bounce, la méthode `sweego_mark_undelivered(failure_type, failure_reason)` est appelée. Elle :

1. Met à jour `mail.mail : state = exception, failure_type, failure_reason, sweego_pending_delete = False`
2. Met à jour `mail.notification : notification_status = exception` — ce que `_postprocess_sent_message` ne peut pas faire rétroactivement car il filtre les notifications déjà à `sent`
3. Appelle `_notify_message_notification_update()` pour rafraîchir l'interface en temps réel

5.5. Réception des réponses (Inbound)

Lorsqu'un client répond à un email envoyé par Odoo :

1. La réponse arrive sur le domaine alias configuré dans Sweego (ex : `reply+TOKEN@test.mangono.cloud`)
2. Sweego transmet la réponse au webhook inbound d'Odoo via une requête POST JSON
3. Le module décode le TOKEN contenu dans l'adresse Reply-To pour identifier le modèle Odoo et l'enregistrement concerné
4. Un message RFC2822 est reconstruit depuis le payload Sweego (corps, pièces jointes téléchargées via l'API Sweego)
5. Le message est injecté dans le chatter via `mail.thread.message_process`
6. La réponse apparaît directement dans le chatter de l'enregistrement Odoo



Les adresses `reply+TOKEN@domaine` ne sont jamais exposées à Odoo comme destinataires réels. Le module les filtre automatiquement lors de la reconstruction du message RFC2822. Si après filtrage il ne reste aucune adresse To, l'adresse de l'expéditeur est utilisée en fallback.

5.6. Pièces jointes inbound

Lorsqu'un client joint un fichier à sa réponse, le module télécharge automatiquement la pièce jointe depuis l'API REST Sweego avant de la transmettre à Odoo. La clé API et l'UUID client configurés dans **Paramètres** → **Sweego** sont requis pour cette fonctionnalité.

5.7. Détection des boucles email

Le module détecte automatiquement les emails en boucle via deux signaux :

- Présence de l'en-tête `x-zone-loop`
- Présence de l'en-tête `x-mailer: Sweego`

Lorsqu'une boucle est détectée, le webhook répond `200 OK` mais ignore le message.

5.8. Vérification des signatures webhook

Tous les webhooks reçus de Sweego sont vérifiés via le protocole [Standard Webhooks](#) (HMAC-SHA256). La signature est calculée sur la chaîne `{Webhook-Id}.{Webhook-Timestamp}.{body JSON brut}`. Si invalide, la requête est rejetée avec HTTP `401`.

5.9. Architecture de traitement outbound

Le traitement des événements Sweego est géré à deux niveaux :

Webhook outbound (`/sweego/webhook/outbound`) — traitement en temps réel par événement :

<code>event_type</code>	Action	Notes
<code>email_sent</code>	Log informatif	
<code>delivered</code>	<code>sweego_confirm_and_delete()</code> sur <code>mail.mail</code> + mise à jour <code>sweego_message_id</code>	
<code>soft-bounce</code>	<code>sweego_mark_undelivered(mail_smtp, ...)</code>	Enveloppe rouge dans le chatter
<code>hard_bounce</code>	<code>sweego_mark_undelivered(mail_email_invalid, ...)</code>	Enveloppe rouge dans le chatter
<code>list_unsub</code>	Log informatif	
<code>complaint</code>	Log warning	
<code>email_clicked</code>	Log informatif	
<code>email_opened</code>	Log informatif	

Cron de synchronisation — traitement via l'API de logs, unifié par `_sweego_process_log_entry` :

status	Handler	Action
email_sent	_sweego_on_email_sent	Log informatif
delivered	_sweego_on_delivered	sweego_confirm_and_delete() + mise à jour sweego_message_id
undelivered	_sweego_on_undelivered	sweego_mark_undelivered(...) → enveloppe rouge dans le chatter

6. Tâche planifiée — Synchronisation des logs

6.1. Rôle

Filet de sécurité pour les livraisons et bounces dont le webhook outbound n'a pas été reçu. Interroge périodiquement l'API Sweego `/logs/` et traite les `mail.mail` en attente.

6.2. Comportement

1. Recherche les `mail.mail` avec `sweego_pending_delete = True` et `sweego_confirmed = False`
2. Si aucun email en attente, termine immédiatement sans appel API
3. Récupère tous les domaines alias actifs via `mail.alias.domain` (support multi-société)
4. Interroge l'API Sweego, pagine les résultats par tranches de 500 jusqu'à épuisement
5. Filtre côté client sur les `Message-Id` des emails en attente
6. Délègue chaque entrée à `_sweego_process_log_entry`



Le filtre par domaines alias garantit qu'en cas de compte Sweego partagé entre plusieurs instances Odoo, chaque instance ne traite que ses propres emails.

6.3. Paramétrage

Créée automatiquement à l'installation, s'exécute toutes les 30 minutes par défaut. Pour modifier : Paramètres → Technique → Automatisation → Actions planifiées → Sweego : Synchronisation des logs de livraison.

7. Mécanisme du token Reply-To

7.1. Format

```
reply+{payload_hex}.{signature}@domaine-alias
```

Exemple :

```
reply+32332d73616c652e6f72646572.cb8e0be9@test.mangono.cloud
```

Décodé :

payload_hex → 32332d73616c652e6f72646572 → 23-sale.order

signature → cb8e0be9 (8 premiers caractères HMAC-SHA256)

- Le payload `{id}-{model}` est encodé en hexadécimal (résistant à la normalisation minuscules par les serveurs mail)
- La signature HMAC utilise le **secret webhook sortant** (`sweego_outbound_webhook_secret`) configuré sur le serveur d'envoi

7.2. Décodage

À la réception, le module extrait le token, vérifie la signature HMAC avec le secret sortant et décode le payload. Si la signature est invalide, le message est ignoré.

8. Champs techniques ajoutés

8.1. Sur `ir.mail_server`

Champ	Description
<code>sweego_outbound_webhook_secret</code>	Secret HMAC webhooks outbound (base64, accès admin). Utilisé aussi pour signer les tokens Reply-To.
<code>sweego_inbound_webhook_secret</code>	Secret HMAC webhooks inbound (base64, accès admin)
<code>sweego_delete_on_sent</code>	Activer la suppression différée (Boolean, défaut : <code>False</code>)



Le champ `sweego_inbound_domain` a été supprimé en v1.2. Le domaine inbound est désormais résolu depuis `company.alias_domain_id` (envoi) et `mail.alias.domain` (cron).

8.2. Sur `mail.mail`

Champ	Description
<code>sweego_pending_delete</code>	<code>True</code> après envoi — suppression suspendue en attente de confirmation Sweego
<code>sweego_confirmed</code>	<code>True</code> après confirmation de livraison — autorise la suppression définitive

8.2.1. Méthodes sur `mail.mail`

Méthode	Description
<code>sweego_confirm_and_delete()</code>	Appelée par le webhook <code>delivered</code> ou le cron. Marque l'email comme confirmé et le supprime.
<code>sweego_mark_undelivered(type, msg)</code>	Appelée lors d'un bounce. Met à jour <code>mail.mail</code> et <code>mail.notification</code> pour afficher l'enveloppe rouge dans le chatter.

8.3. Sur `mail.message`

Champ	Description
<code>sweego_message_id</code>	<code>transaction_id</code> Sweego enregistré lors du <code>delivered</code> (indexé)

8.4. Dans `ir.config_parameter`

Clé	Description
<code>mangono_sweego.api_key</code>	Clé API Sweego (géré par <code>mangono_sweego_base</code>)
<code>mangono_sweego.client_uuid</code>	UUID client Sweego (géré par <code>mangono_sweego_base</code>)

9. Endpoints webhook

Endpoint	Rôle
<code>POST /sweego/webhook/outbound</code>	Événements outbound : livraison, bounces, ouvertures, clics...
<code>POST /sweego/webhook/inbound</code>	Emails entrants (réponses clients)

10. Dépannage

10.1. Les webhooks sont rejetés (401)

- Secret webhook mal copié — copier tel quel (base64) depuis le tableau de bord Sweego
- Vérifier que outbound et inbound sont dans leurs champs respectifs
- Vérifier que l'URL webhook est accessible publiquement

10.2. Les réponses n'apparaissent pas dans le chatter

- Vérifier que le MX du domaine alias pointe vers `inbound.sweego.co`
- Vérifier la configuration du webhook inbound dans Sweego
- Consulter les logs : `invalid inbound token signature`
- Vérifier qu'un domaine alias est configuré sur la société (**Paramètres** → **Technique** → **Email** → **Domaines alias**)

- Consulter les logs : `no alias domain configured` — aucun domaine résolu pour la société

10.3. Les pièces jointes inbound n'apparaissent pas

- Vérifier la Clé API Sweego et l'UUID Client Sweego dans Paramètres → Sweego
- Consulter les logs : `failed to download attachment`

10.4. Les emails ne sont pas supprimés après livraison

- Vérifier que le webhook outbound est configuré avec l'événement `delivered`
- Vérifier que **Suppression différée** est activée sur le serveur
- Le cron prend le relais sous 30 minutes si le webhook n'est pas reçu
- Consulter les logs : `mail.mail X confirmed by Sweego → deleting`

10.5. L'enveloppe reste grise après un bounce

- Vérifier que le webhook outbound est configuré avec les événements `soft-bounce` et `hard_bounce`
- Si le webhook n'est pas disponible, le cron traite les statuts `undelivered` sous 30 minutes
- Consulter les logs : `mail.mail X marked undelivered by Sweego`
- L'enveloppe rouge n'est visible que pour les emails avec **Suppression différée** activée (sinon `mail.mail` est déjà supprimé au moment du bounce)

10.6. La tâche planifiée ne traite aucun email

- Vérifier la Clé API Sweego dans Paramètres → Sweego
- Vérifier qu'au moins un domaine alias est configuré sur une société
- Vérifier que la tâche est active dans Paramètres → Technique → Automatisation → Actions planifiées

10.7. Le serveur refuse de se créer

- Le champ Nom d'utilisateur est obligatoire avec l'authentification Mangono Sweego SMTP

11. Glossaire

Terme	Définition
SMTP	Simple Mail Transfer Protocol — protocole standard d'envoi d'emails
Webhook	Notification HTTP envoyée par Sweego vers Odoo pour signaler un événement

Terme	Définition
HMAC-SHA256	Algorithme de signature cryptographique utilisé pour authentifier les webhooks
Token Reply-To	Jeton encodé dans l'adresse Reply-To permettant de router les réponses vers le bon objet Odoo
Domaine alias	Domaine email configuré sur la société dans Odoo (<code>mail.alias.domain</code>), utilisé pour la réception des réponses (MX → Sweego)
RFC 2047	Standard d'encodage des en-têtes email non-ASCII (ex : <code>=?utf-8?q?...?=</code>), appliqué par certains MTA lors du repli de lignes longues
<code>mail.mail</code>	Table Odoo contenant les emails en attente d'envoi ou récemment envoyés
<code>mail.notification</code>	Table Odoo liant un message à un destinataire, portant le statut de notification (<code>sent, exception...</code>) affiché dans le chatter
<code>mail.message</code>	Table Odoo contenant les messages du chatter (emails, notes internes...)
chatter	Interface de messagerie intégrée aux fiches Odoo
Standard Webhooks	Protocole ouvert de signature des webhooks basé sur HMAC-SHA256
API REST Sweego	Interface HTTP Sweego pour le téléchargement des pièces jointes et la synchronisation des logs
<code>mangono_sweego_base</code>	Module Odoo partagé gérant la configuration de connexion Sweego (clé API, UUID client)