



Création d'un Module Odoo

Gestion des Plaintes Clients

(Complaint Management System)

Module : ERP
Framework : Odoo 17.0
Année Universitaire : 2024-2025



Adobe Stock #231452877

Réalisé par : Saida EL AJIMI

Encadré par : mohammed aitdaoud

Date : Décembre 2025

Table des Matières

1 Objectif du LAB	2
2 Prérequis	2
3 Redémarrage et Options Docker	2
3.1 Mise à jour automatique (-u)	2
3.2 Développement XML (-dev=xml)	2
3.3 Redémarrer le conteneur Odoo	2
3.4 Nettoyage de l'environnement	3
4 Guide de Mise en Œuvre Étape par Étape	3
5 Structure du Module	3
5.1 Fichier Manifest (<code>__manifest__.py</code>)	4
6 Modèle de Données	5
6.1 Classe <code>ComplaintComplaint</code>	5
6.2 Champs du Modèle	5
7 Vues et Interface	6
7.1 Vue Liste (Tree View)	6
7.2 Vue Formulaire (Form View)	6
7.3 Vue Recherche (Search View)	7
7.4 Menu et Actions	7
8 Sécurité et Droits d'Accès	7
9 Captures d'Écran de l'Application	8
9.1 Liste des Applications Odoo	8
9.2 Vue Liste des Plaintes	8
9.3 Formulaire de Plainte	9
9.4 Filtres et Recherche	9
10 Installation et Configuration	9
10.1 Déploiement avec Docker	9
10.2 Étapes d'Installation	10
11 Fonctionnalités Principales	11
11.1 Création de Plainte	11
11.2 Gestion du Cycle de Vie	11
11.3 Recherche et Filtrage	12
12 Conclusion	12
12.1 Améliorations Possibles	12
13 Références	12

1 Objectif du LAB

L'objectif de ce LAB est de créer un module Odoo complet pour la **gestion des plaintes clients**. Ce module permet de :

- Ajouter un nouveau menu "Complaints"
- Gérer des enregistrements simples (Sujet, Client, Date, Statut)
- Afficher les données en liste et en formulaire
- Gérer le cycle de vie des plaintes (Draft → Open → Resolved)

2 Prérequis

Environnement requis :

- **Odoo** : Version 17.0
- **Python** : Version 3.10+
- **PostgreSQL** : Base de données configurée
- **Docker** : Pour containeriser l'environnement Odoo

3 Redémarrage et Options Docker

Dans le cadre du développement, certaines modifications nécessitent le redémarrage du conteneur.

3.1 Mise à jour automatique (-u)

L'option -u permet de mettre à jour un module spécifique au démarrage d'Odoo.

```
1 # Dans le service odoo17
2 command: odoo -d odoo_db -u complaint_management --db_host=db
```

Attention : L'option -i base doit être supprimée après la création initiale de la base pour éviter de réinitialiser toutes les données.

3.2 Développement XML (-dev=xml)

L'option -dev=xml permet de prendre en compte les modifications XML sans redémarrer le conteneur, par simple rafraîchissement de la page.

3.3 Redémarrer le conteneur Odoo

```
1 docker compose restart odoo17
2 # OU
3 docker restart odoo_app
```

3.4 Nettoyage de l'environnement

- **docker compose down** : Arrête les conteneurs.
- **docker compose down -v** : Supprime conteneurs et volumes (données).

4 Guide de Mise en Œuvre Étape par Étape

1. **Vérification** : docker ps pour voir si les conteneurs tournent.
2. **Dossier module** : mkdir -p addons/complaint_management puis ls addons.
3. **Arborescence** : Création des dossiers models, views, security et fichiers __init__.py.
4. **Manifest** : Configuration du fichier __manifest__.py.
5. **Liaison** : Imports dans les fichiers __init__.py.
6. **Modèle** : Création de la classe Python dans models/complaint.py.
7. **Sécurité** : Configuration de ir.model.access.csv.
8. **Interface** : Création des vues et menus dans complaint_views.xml.
9. **Mode Développeur** : Activation via http://localhost:8069/web?debug=1.

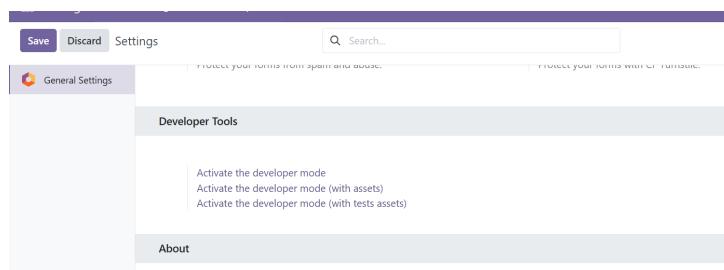


FIGURE 1 – Activation du mode développeur

10. **Installation** : Menu Applications > Mettre à jour > Installer.

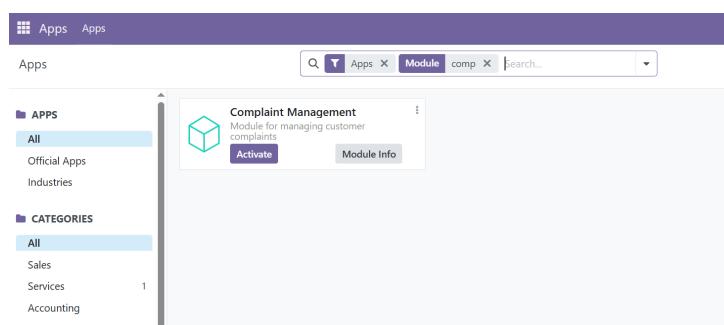


FIGURE 2 – Bouton d'activation du module

11. **Tests** : Création d'une nouvelle plainte et vérification des flux.

5 Structure du Module

Le module `complaint_management` suit l'architecture MVC d'Odoo et est organisé comme suit :

```

1 complaint_management/
2     __init__.py
3     __manifest__.py
4     models/
5         __init__.py
6         complaint.py
7     views/
8         complaint_views.xml
9     security/
10    ir.model.access.csv

```

Listing 1 – Arborescence du module

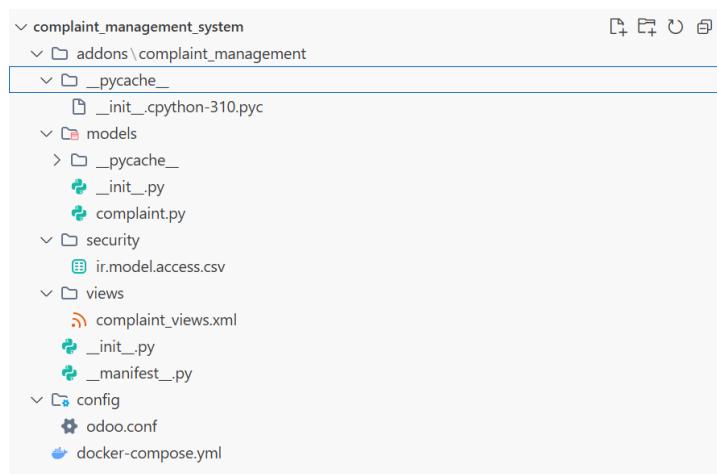


FIGURE 3 – Arborescence du module Odoo

5.1 Fichier Manifest (`__manifest__.py`)

Le fichier manifest définit les métadonnées du module :

```

1 {
2     "name": "Complaint Management",
3     "version": "1.0",
4     "summary": "Module for managing customer complaints",
5     "category": "Services",
6     "author": "Saida EL AJIMI",
7     "depends": ["base"],
8     "data": [
9         "security/ir.model.access.csv",
10        "views/complaint_views.xml",
11    ],
12    "installable": True,
13    "application": True,
14    "license": "LGPL-3",
15 }

```

Listing 2 – `__manifest__.py`

Points clés :

- **depends** : Le module dépend uniquement du module **base** d'Odoo
- **application** : Le module est défini comme une application autonome
- L'ordre de chargement des fichiers data est important (sécurité en premier)

6 Modèle de Données

6.1 Classe ComplaintComplaint

Le modèle principal du module est défini dans `complaint.py` :

```

1 from odoo import models, fields
2 class ComplaintComplaint(models.Model):
3     _name = "complaint.complaint"
4     _description = "Complaint Management"
5     name = fields.Char(string="Subject", required=True)
6     description = fields.Text(string="Description")
7     customer_name = fields.Char(string="Customer Name", required=
8         True)
9     category = fields.Selection([
10         ("service", "Service"),
11         ("product", "Product"),
12         ("other", "Other"),
13     ], string="Category", default="service")
14     state = fields.Selection([
15         ("draft", "Draft"),
16         ("open", "Open"),
17         ("resolved", "Resolved"),
18     ], string="Status", default="draft")
19     date_created = fields.Date(string="Date Created",
20                                default=fields.Date.today())

```

Listing 3 – `models/complaint.py`

6.2 Champs du Modèle

primary !20 Champ	Type	Description	Obligatoire
<code>name</code>	Char	Sujet de la plainte	
<code>description</code>	Text	Description détaillée	
<code>customer_name</code>	Char	Nom du client	
<code>category</code>	Selection	Catégorie (Service/Product/Other)	(default : service)
<code>state</code>	Selection	Statut (Draft/Open/Resolved)	(default : draft)
<code>date_created</code>	Date	Date de création	(default : today)

TABLE 1 – Description des champs du modèle Complaint

7 Vues et Interface

7.1 Vue Liste (Tree View)

La vue liste affiche toutes les plaintes avec des couleurs conditionnelles :

```

1 <tree decoration-info="state == 'draft'" 
2   decoration-warning="state == 'open'" 
3   decoration-success="state == 'resolved'">
4   <field name="name"/>
5   <field name="customer_name"/>
6   <field name="category"/>
7   <field name="date_created"/>
8   <field name="state" widget="badge"/>
9 </tree>
```

Listing 4 – Vue Tree avec décorations colorées

Décorations :

- **Bleu** pour les plaintes en "Draft"
- **Jaune** pour les plaintes "Open"
- **Vert** pour les plaintes "Resolved"

7.2 Vue Formulaire (Form View)

La vue formulaire permet de créer et modifier les plaintes :

```

1 <form>
2   <header>
3     <field name="state" widget="statusbar"
4       statusbar_visible="draft,open,resolved"/>
5   </header>
6   <sheet>
7     <div class="oe_title">
8       <h1>
9         <field name="name" placeholder="Complaint Subject"/>
10      </h1>
11    </div>
12    <group>
13      <group>
14        <field name="customer_name"/>
15        <field name="category"/>
16      </group>
17      <group>
18        <field name="date_created"/>
19      </group>
20    </group>
21    <notebook>
22      <page string="Description" name="description">
```

```

23         <field name="description"
24             placeholder="Describe the complaint..."/>
25     </page>
26 </notebook>
27 </sheet>
28 </form>

```

Listing 5 – Vue Form

7.3 Vue Recherche (Search View)

La vue de recherche offre des filtres et des regroupements :

```

1 <search>
2     <field name="name"/>
3     <field name="customer_name"/>
4     <field name="category"/>
5     <filter string="Draft" name="draft"
6         domain="[(('state', '=', 'draft'))]"/>
7     <filter string="Open" name="open"
8         domain="[(('state', '=', 'open'))]"/>
9     <filter string="Resolved" name="resolved"
10        domain="[(('state', '=', 'resolved'))]"/>
11     <group expand="0" string="Group By">
12         <filter string="Category" name="group_by_category"
13             context="{'group_by': 'category'}"/>
14         <filter string="Status" name="group_by_state"
15             context="{'group_by': 'state'}"/>
16     </group>
17 </search>

```

Listing 6 – Vue Search avec filtres

7.4 Menu et Actions

Le menu principal est défini avec :

- **Menu racine** : "Complaints"
- **Sous-menu** : "All Complaints" (déclenche l'action `action_complaint`)
- **Action Window** : Ouvre les vues tree et form du modèle `complaint.complaint`

8 Sécurité et Droits d'Accès

Le fichier `ir.model.access.csv` définit les droits d'accès :

```

1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,
2 perm_unlink
access_complaint_complaint,complaint.complaint,
model_complaint_complaint,base.group_user,1,1,1,1

```

Listing 7 – security/ir.model.access.csv

Configuration :

- **Groupe :** base.group_user (utilisateurs internes)
- **Permissions :** Lecture, écriture, création et suppression (1,1,1,1)

9 Captures d'Écran de l'Application

9.1 Liste des Applications Odoo

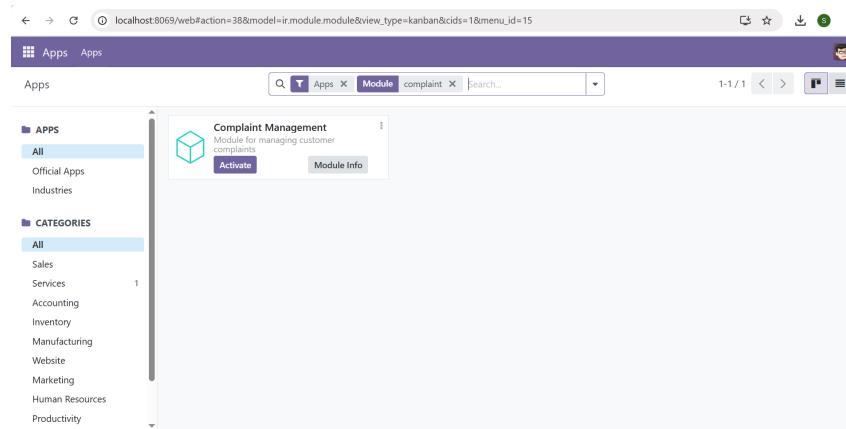


FIGURE 4 – Liste des applications montrant le module Complaint Management installé

Le module "Complaints" apparaît dans la liste des applications disponibles avec son icône et sa description.

9.2 Vue Liste des Plaintes

Complaints All Complaints				
New Complaints	Search...			
Attente excessive au restaurant	Abdelhakim Elajimi	Service	12/31/2025	Draft
Smartphone défectueux après 2 semaines	Ibrahim Elajimi	Product	12/23/2025	Draft
Colis endommagé lors de la livraison	Mohammed Elajimi	Service	12/31/2025	Draft
Chambre d'hôtel non conforme à la réservation	Abderrahim Elajimi	Other	12/31/2025	Draft

FIGURE 5 – Vue liste affichant toutes les plaintes avec leurs statuts

La vue liste montre :

- Le sujet de chaque plainte
- Le nom du client
- La catégorie
- La date de création
- Le statut avec badge coloré

9.3 Formulaire de Plainte

FIGURE 6 – Formulaire détaillé pour créer ou modifier une plainte

Le formulaire comprend :

- Une barre de statut en haut (Draft → Open → Resolved)
- Un champ titre pour le sujet
- Les informations du client
- La catégorie de la plainte
- La date de création
- Un onglet pour la description détaillée

9.4 Filtres et Recherche

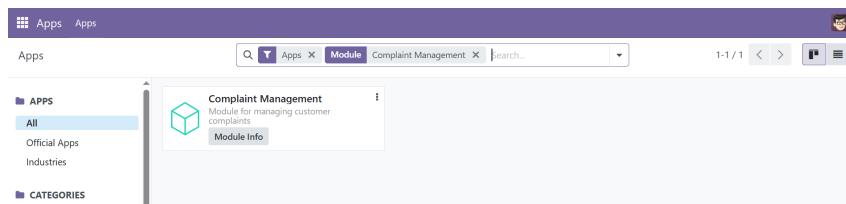


FIGURE 7 – Options de filtrage et de recherche des plaintes

La vue de recherche permet :

- De rechercher par sujet, nom du client ou catégorie
- De filtrer par statut (Draft, Open, Resolved)
- De regrouper par catégorie ou statut
- Une navigation rapide dans les données

10 Installation et Configuration

10.1 Déploiement avec Docker

Le projet utilise Docker Compose pour faciliter le déploiement :

```

1 version: '3.1'
2 services:
3   web:
4     image: odoo:17.0
5     depends_on:
6       - db
7     ports:
8       - "8069:8069"
9     volumes:
10      - ./addons:/mnt/extras-addons
11      - ./config:/etc/odoo
12     environment:
13       - HOST=db
14       - USER=odoo
15       - PASSWORD=odoo
16   db:
17     image: postgres:15
18     environment:
19       - POSTGRES_DB=postgres
20       - POSTGRES_PASSWORD=odoo
21       - POSTGRES_USER=odoo

```

Listing 8 – docker-compose.yml

10.2 Étapes d'Installation

1. Démarrer les conteneurs :

```

1 docker-compose up -d

```

2. Accéder à Odoo :

- URL : <http://localhost:8069>
- Créer une base de données

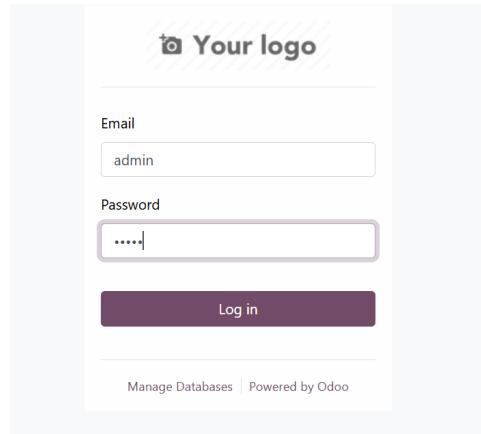


FIGURE 8 – Interface de connexion Odoo

3. Installer le module :

- Activer le mode développeur
- Aller dans Apps
- Mettre à jour la liste des applications
- Rechercher "Complaint Management"
- Cliquer sur "Install"

4. Utiliser le module :

- Le menu "Complaints" apparaît dans la barre de navigation
- Cliquer sur "All Complaints" pour gérer les plaintes

11 Fonctionnalités Principales

11.1 Crédation de Plainte

1. Cliquer sur "Complaints" → "All Complaints"

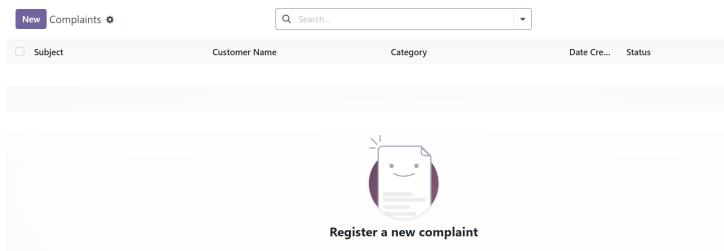


FIGURE 9 – Vue initiale avant l'ajout de plaintes

2. Cliquer sur "Create"
3. Remplir le formulaire :
 - Sujet (obligatoire)
 - Nom du client (obligatoire)
 - Catégorie
 - Description
4. Sauvegarder

11.2 Gestion du Cycle de Vie

Le statut de la plainte évolue selon le workflow :



- **Draft** : Nouvelle plainte créée
- **Open** : Plainte en cours de traitement
- **Resolved** : Plainte résolue

Le changement de statut se fait via la barre de statut dans le formulaire.

11.3 Recherche et Filtrage

Plusieurs options de filtrage sont disponibles :

- Par sujet
- Par nom de client
- Par catégorie (Service, Product, Other)
- Par statut (Draft, Open, Resolved)
- Regroupement par catégorie ou statut

12 Conclusion

Ce LAB a permis de créer un module Odoo complet et fonctionnel pour la gestion des plaintes clients. Le module respecte les bonnes pratiques Odoo :

Architecture MVC : Séparation claire entre modèles, vues et contrôleurs

Sécurité : Droits d'accès configurés via CSV

Interface utilisateur : Vues ergonomiques (liste, formulaire, recherche)

Workflow : Gestion du cycle de vie des plaintes

Extensibilité : Structure modulaire facilement extensible

12.1 Améliorations Possibles

Le module peut être enrichi avec :

- Ajout de champs supplémentaires (priorité, assigné à, etc.)
- Méthodes de workflow automatisées
- Notifications par email
- Rapports et statistiques
- Historique des modifications
- Pièces jointes
- Intégration avec d'autres modules Odoo
- Dashboard de suivi des plaintes
- Indicateurs de performance (KPI)

13 Références

- Documentation Odoo 17 : <https://www.odoo.com/documentation/17.0/>
- Odoo Developer Documentation : <https://www.odoo.com/documentation/17.0/developer.html>
- Code source du module : `complaint_management/`

Auteur : Saida EL AJIMI

Date : Décembre 2025

Version : 1.0