

Stock Simple + Inventaire

Module Odoo 17 (simple_stock_inventory)

Auteur : *Anas Slimani / Badr Ezziyati* | Encadrant : *Nidal Lamghari*

École Nationale des Sciences Appliquées de Khouribga

28 décembre 2025



Version : 17.0.1.0.0 **Catégorie :** Inventory **Licence :** LGPL-3

Plan

- 1 Introduction
- 2 Présentation d'Odoo
- 3 Analyse des besoins
- 4 Architecture du module
- 5 Modèles de données
- 6 Vues et interfaces
- 7 Sécurité
- 8 Workflow des mouvements
- 9 Démonstration
- 10 Extraits de code (Python/XML)
- 11 Installation
- 12 Conclusion

Contexte & objectifs

Contexte

Odoo est une plateforme ERP modulaire. Dans plusieurs organisations, la gestion de stock complète peut être trop riche pour un besoin simple (produits, emplacements, mouvements, inventaire).

Objectifs du module

- Proposer une **gestion de stock simplifiée et maîtrisée**
- Couvrir le cycle : **produits → mouvements → inventaire**
- Garantir la **tracerabilité** via le chatter (mail.thread)
- Appliquer une **sécurité claire** (Utilisateur / Manager)

Qu'est-ce qu'Odoo ?

- ERP open-source (apps : CRM, Vente, Achat, Stock, Comptabilité, RH...)
- **Architecture modulaire** : un module = modèles + vues + sécurité + données
- Basé sur :
 - Python (ORM)
 - XML (views / actions / menus)
 - PostgreSQL (données)

Pourquoi un module personnalisé ?

Adapter l'outil au besoin métier, réduire la complexité et gagner en productivité.

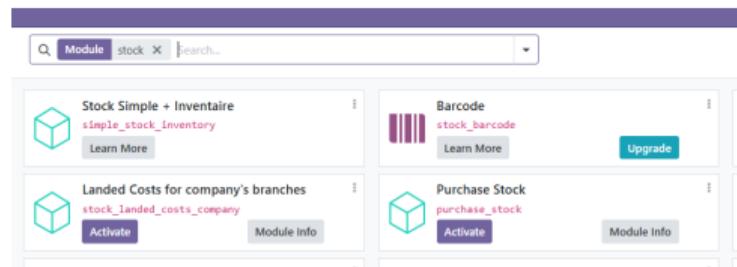


Figure – Module visible dans la liste des Apps

Problématique & besoins fonctionnels

Problématique

Mettre en place une gestion de stock **simple**, centrée sur l'essentiel, tout en conservant : **contrôle, workflow, inventaire, traçabilité**.

Besoins fonctionnels

- ① Gestion des produits (prix de revient / vente + stock calculé)
- ② Gestion des emplacements (Interne / Fournisseur / Client / Inventaire)
- ③ Mouvements de stock avec workflow (Brouillon → Validé → Annulé)
- ④ Vérification stock avant sortie
- ⑤ Inventaire physique avec calcul d'écart
- ⑥ Génération automatique des mouvements d'ajustement

Architecture technique (vue d'ensemble)

Composants

- **Modèles Python** : ssi.product, ssi.location, ssi.quant, ssi.move, ssi.inventory, ssi.inventory.line
- **Vues XML** : tree/form + menus + actions
- **Sécurité** : 2 groupes + access rules
- **Suivi** : mail.thread + mail.activity.mixin

Règles importantes

- Séquences automatiques (références)
- Contrainte SQL : unicité (produit, emplacement) dans les quants
- Validations métier (ex : stock insuffisant)

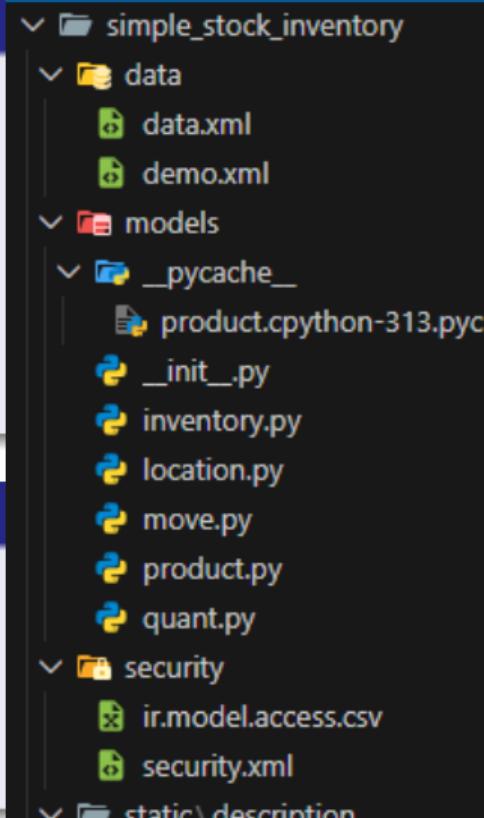
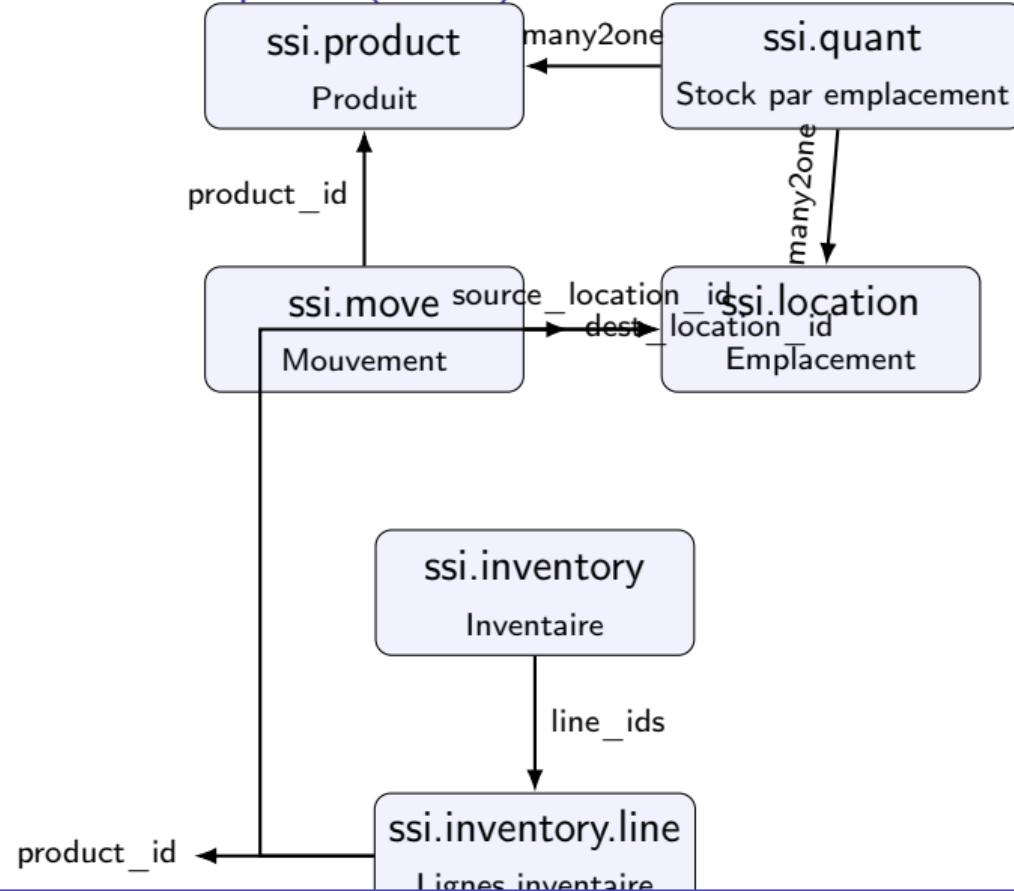


Diagramme de classes simplifié (TikZ)



Modèles : vue synthèse

Modèle	Rôle principal
ssi.product	Définition produit + prix + stock calculé
ssi.location	Emplacements (interne, fournisseur, client, inventaire)
ssi.quant	Quantité disponible par (produit, emplacement)
ssi.move	Entrée / sortie / transfert / ajustement + workflow
ssi.inventory	Session d'inventaire (préparation, validation)
ssi.inventory.line	Quantité théorique vs réelle, calcul d'écart

ssi.product — Produit

Champs clés

Champ	Description
name (Char)	Nom du produit
default_code (Char)	Référence interne (séquence)
cost_price (Float)	Prix de revient
sale_price (Float)	Prix de vente
qty_available (Float, computed)	Stock total calculé depuis ssi.quant
active (Boolean)	Actif / Archivé

Suivi

mail.thread + mail.activity.mixin : historique, messages, activités.

ssi.location — Emplacement

Types d'emplacements

- **Interne** : stock réel de l'entreprise
- **Fournisseur** : origine d'entrée
- **Client** : destination de sortie
- **Inventaire** : emplacement technique pour ajustement

Champs

Champ	Description
name (Char)	Libellé emplacement
usage (Selection)	internal / supplier / customer / inventory
active (Boolean)	Actif / Archivé

ssi.quant — Quantité par produit/emplacement

Objectif

Centraliser le stock disponible par couple (**product_id, location_id**).

Champs

Champ	Description
product_id (Many2one)	Produit concerné
location_id (Many2one)	Emplacement concerné
quantity (Float)	Quantité disponible

Contrainte SQL

Unicité : **un seul quant** par (produit, emplacement).

ssi.move — Mouvement de stock

Workflow

Brouillon → Validé → Annulé

Champs principaux

Champ	Description
name (Char)	Référence (séquence)
move_type (Selection)	entrée / sortie / transfert / ajustement
product_id (Many2one)	Produit
quantity (Float)	Quantité
source_location_id (Many2one)	Source
dest_location_id (Many2one)	Destination
state (Selection)	draft / done / cancel

Règle métier

Avant validation d'une **sortie**, vérification du stock disponible (sinon blocage).

ssi.inventory & ssi.inventory.line — Inventaire

Principe

Ajout automatique de tous les produits, comparaison **théorique vs réelle**, calcul de l'écart, puis génération des mouvements d'ajustement.

Modèle	Champs clés
ssi.inventory	name, date, location_id, state, line_ids
ssi.inventory.line	product_id, theoretical_qty, counted_qty, difference_qty

Automatisation

Lors de la validation : création automatique des mouvements (ajustement) vers/depuis l'emplacement **Inventaire**.

Menu principal (Stock Simple)

The screenshot shows the main interface of the Stock Simple application. At the top is a purple header bar with the title "Stock Simple" and five navigation links: "Produits", "Emplacements", "Mouvements", and "Inventaires". Below the header is a white content area. On the left side of this area is a purple button labeled "New" and "Produits" with a gear icon. To the right is a search bar with a magnifying glass icon. A horizontal grey bar runs across the middle of the content area.

- Accès direct aux : Produits, Emplacements, Mouvements, Inventaires

Produits : liste et formulaire

The screenshot shows a software application for managing a simple stock and inventory system. On the left, there is a sidebar with a tree view of categories: 'Nom' (Name), 'Laptop HP ProBook' (selected), 'Smartphone Logitech', 'Clavier mécanique', 'Écran Dell 27 pouces', 'Kéeper avec Bluetooth', 'Câble USB-C 2m', and 'Souris'. The main area displays a table of products with columns: Référence (Reference), Code-barres (Barcode), Stock total (Total Stock), Prix de revient (Purchase Price), and Prix de vente (Sale Price). The table contains the following data:

	Référence	Code-barres	Stock total	Prix de revient	Prix de vente
Laptop HP ProBook	LAPTOP-001	5001234123457	25.00	450.00	650.00
Smartphone Logitech	PHONE-001	5001234123458	150.00	150.00	250.00
Clavier mécanique	KB10-001	5001234123459	80.00	100.00	150.00
Écran Dell 27 pouces	MON-001	5001234123460	40.00	100.00	140.00
Kéeper avec Bluetooth	KEYBOARD-001	5001234123461	80.00	250.00	350.00
Câble USB-C 2m	CABLE-001	5001234123462	500.00	5.00	12.00
Souris			0.00	100.00	120.00

To the right of the table, there is a detailed product form for 'Laptop HP ProBook' (selected in the sidebar). The form includes fields for Référence (5001234123457), Code-barres (5001234123457), Stock total (0.00), and Prix de vente (0.00). Below this, there are tabs for 'Stock par emplacement' (Stock by location) and 'Mouvements' (Movements), with 'Emplacement' selected. A button 'Add a line' is also present.

Figure – Liste des produits

Figure – Formulaire produit (prix + stock)

Emplacements : liste et formulaire

Stock Simple			Products	Emplacements	Movements	Inventaire
New Emplacements Search...						
<input type="checkbox"/> Non						
<input type="checkbox"/>	Usage	Intérieur		<input type="checkbox"/>	Actif	
<input type="checkbox"/>	Fournisseur	Fournisseur		<input type="checkbox"/>		
<input type="checkbox"/>	Clients	Clients		<input type="checkbox"/>		
<input type="checkbox"/>	Ajoutement Inventaire	Ajoutement Inventaire		<input type="checkbox"/>		
<input type="checkbox"/>	Entrepôt A	Intérieur		<input type="checkbox"/>		
<input type="checkbox"/>	Entrepôt B	Intérieur		<input type="checkbox"/>		

Figure – Liste des emplacements

Stock Simple			Products	Emplacements	Movements	Inventaire
New Emplacements Send message Log note Activities						
<input type="checkbox"/> Non						
<input type="checkbox"/>	Usage	Intérieur		<input type="checkbox"/>	Actif	
<input type="checkbox"/>	Fournisseur	Fournisseur		<input type="checkbox"/>		
<input type="checkbox"/>	Clients	Clients		<input type="checkbox"/>		
<input type="checkbox"/>	Ajoutement Inventaire	Ajoutement Inventaire		<input type="checkbox"/>		
<input type="checkbox"/>	Entrepôt A	Intérieur		<input type="checkbox"/>		
<input type="checkbox"/>	Entrepôt B	Intérieur		<input type="checkbox"/>		

Figure – Formulaire emplacement (type usage)

Groupes & droits d'accès

Groupes

- **Stock Simple / Utilisateur** : accès lecture/écriture aux opérations courantes
- **Stock Simple / Manager** : droits étendus (validation, configuration, suppression si autorisée)

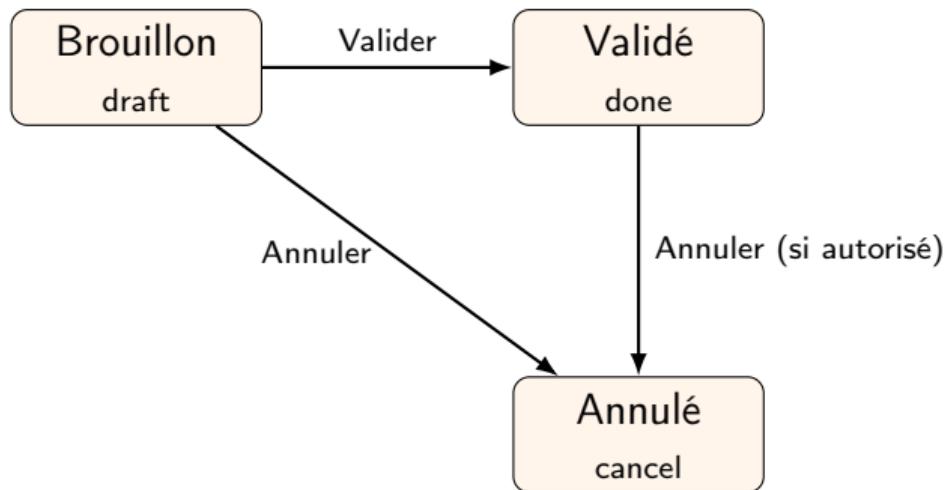
Mécanisme

- **ir.model.access.csv** : droits CRUD par modèle
- (Optionnel) **record rules** : filtrage par société / périmètre

The screenshot shows the Odoo Groups settings page. A new group named 'Stock Simple / Manager' has been created. The group details are as follows:

Application	Stock Simple					
Name	Manager					
Share Group ?	<input type="checkbox"/>					
Users	Inherited	Menus	Views	Access Rights	Record Rules	Notes
Name	Login			Language	Latest authentication	Status
Mitchell Admin	anasslimani039@gmail.com			English (US)	12/28/2025 17:37:08	Confirmed

Diagramme d'états : Mouvement (TikZ)



Contrôle avant validation

Pour une **sortie**, si stock insuffisant : message d'erreur et état conservé en **brouillon**.

Scénario 1 : création & suivi (chatter)

Étapes

- ① Créer un produit
- ② Vérifier le stock calculé
- ③ Observer l'historique (messages / tracking)

Valeur ajoutée

Traçabilité des modifications et communication interne via le chatter.

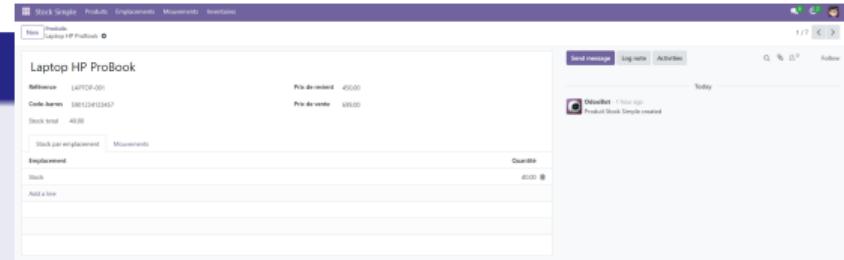


Figure – Exemple de chatter

Scénario 2 : mouvements (brouillon → validé)

New Movements						
Reference	Date	Product	Emplacement source	Emplacement destination	Quantity	Type
MOVE-DEMO-001	12/08/2025 11:32:56	Laptop HP ProBook	Fournisseur	Stock	10,00	Entree
MOVE-DEMO-002	12/08/2025 11:32:56	Souris sans fil Logitech	Stock	Bureau A	30,00	Transfer
MOVE-DEMO-003	12/08/2025 11:32:56	Caisse encaisse	Stock	Caisses	5,00	Sortie

Figure – Liste des mouvements

New Movement MOVE-DEMO-001

MOVE-DEMO-001

Date: 12/08/2025 17:32:58
Product: Laptop HP ProBook
Quantity: 10,00

Notes:
Réception fournisseur - Laptops

Emplacement source: Fournisseur
Emplacement destination: Stock
Type: Entrée

Validé Annuler Envoyer message Log note Activités

Today OdooBot 1 hour ago Movement Stock Simple created

Figure – Form mouvement (brouillon)

Scénario 2 (suite) : validation du mouvement

New Mouvements
MOVE-DEMO-001

Annuler Brouillon Validé Annulé Send me

MOVE-DEMO-001

Date	12/28/2025 17:32:58	Emplacement source	Fournisseurs
Produit	Laptop HP ProBook	Emplacement destination	Stock
Quantité	10.00	Type	Entrée

Notes

Réception fournisseur - Laptops

Mit Od Mo

Figure – Mouvement validé (stock mis à jour)

Scénario 3 : inventaire (préparation & lignes)

The screenshot shows a list of inventories. There is one entry visible:

Référence	Date	Emplacement	Etat
INV-00003	13/08/2025 18:45:47	Stock	Vrai

Figure – Liste des inventaires

The screenshot shows an inventory draft for reference INV-00006. It contains one line item:

Produit	Quantité théorique	Quantité comptée	Écart
Souris sans fil Logitech	0.00	0.00	0.00

Below the table, there is a link "Add a line".

Figure – Inventaire brouillon + lignes

Scénario 3 (suite) : validation inventaire & ajustements

Brouillon

Validé

INV-00003

Date 12/28/2025 18:41:47

Emplacement Stock

Lignes

Produit	Quantité théorique	Quantité comptée	Écart
Laptop HP ProBook	0.00	5.00	5.00

Figure – Inventaire validé (écart traités)

Extrait Python : validation d'une sortie (exemple)

Idée

Empêcher la validation si la quantité demandée dépasse le stock disponible sur l'emplacement source.

Listing 1 – Exemple simplifié (ssi.move)

```
1 def action_validate(self):
2     for move in self:
3         if move.move_type == 'out':
4             available = move.product_id.qty_available
5             if move.quantity > available:
6                 msg = "Stock insuffisant: disponible=%s, demande
7                     =%s"
8                 raise ValidationError(msg % (available, move.
9                                     quantity))
10            move.state = 'done'
11            move._apply_quant_update()
```

Extrait XML : vue formulaire (exemple)

Listing 2 – Exemple simplifié d'une vue produit

```
1 <odoo>
2   <record id="ssi_product_form_view" model="ir.ui.view">
3     <field name="name">ssi.product.form</field>
4     <field name="model">ssi.product</field>
5     <field name="arch" type="xml">
6       <form string="Produit">
7         <sheet>
8           <group>
9             <field name="name"/>
10            <field name="default_code"/>
11          </group>
12          <group>
13            <field name="cost_price"/>
14            <field name="sale_price"/>
15            <field name="qty_available" readonly="1"/>
16          </group>
17        </sheet>
18      </form>
19    </field>
20  </record>
21 </odoo>
```

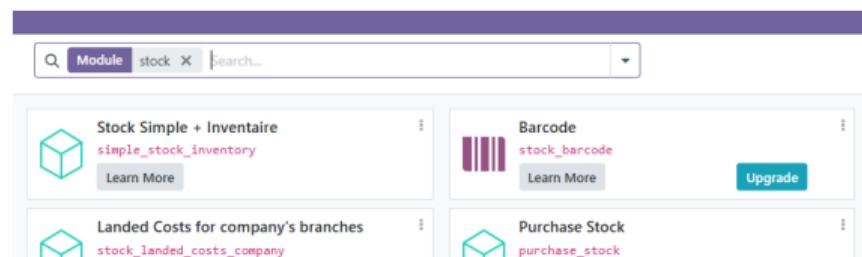
Installation du module

Prérequis

- Odoo 17 installé (environnement dev)
- Accès administrateur pour installer / mettre à jour Apps

Étapes

- ① Copier simple_stock_inventory/ dans le dossier addons
- ② Redémarrer Odoo
- ③ Mode développeur → **Update Apps List**
- ④ Apps → rechercher **Stock Simple** → Installer



Bilan & perspectives

Bilan

- Module fonctionnel couvrant : Produits, Emplacements, Mouvements, Inventaires
- Workflow + contrôles métier (stock insuffisant)
- Traçabilité via mail.thread (chatter)
- Sécurité structurée (Utilisateur / Manager)

Perspectives

- Ajout d'un reporting (mouvements par période / valorisation)
- Support multi-entrepôts / multi-sociétés
- Import/export avancé + QR/Barcode (optionnel)