



MANUEL ODOO POUR DÉVELOPPEURS AMBITIEUX

**Une méthode complète pour maîtriser
Odoo et en faire une compétence rentable**

"J'aurais aimé avoir un document pareil quand j'ai commencé sur Odoo."

Par ABDOU NASSER

v1.0 – août 2025

© Tous droits réservés



1. Introduction — Pourquoi ce manuel ?

Je suis Abdou Nasser, développeur et intégrateur Odoo depuis 2016. J'ai travaillé sur une cinquantaine de projets Odoo dans plusieurs secteurs (retail, santé, éducation, manufacturing ...).

À l'époque, j'avais commencé sous Odoo 8 et il était très difficile de trouver de bonnes ressources sur internet. Il y avait certes quelques blogs et Stack Overflow qui m'aidaient, mais jusque-là c'était compliqué.

Je ne savais pas par où commencer, je n'avais aucune idée sur la façon de m'y prendre pour mettre sur pied une solution Odoo optimale sans casser les fonctionnalités existantes. Dès fois, je développais des fonctionnalités qui pourtant existaient déjà dans la version standard.

Lorsque je commençais, j'avais besoin d'un mentor ou d'un document qui m'accompagnerait dans le vaste écosystème d'Odoo:

- j'avais besoin qu'on m'explique comment fonctionne l'ORM,
- j'avais besoin qu'on m'explique comment Odoo interagissait avec le Javascript pour générer des pages interactives,
- j'avais besoin qu'on attire mon attention sur les erreurs courantes qui peuvent nuire à la performance d'Odoo,

mais hélas, personne ! Je ne pouvais compter que sur moi-même, les blogs que je lisais et Stack Overflow.

Face à toutes les difficultés que j'ai rencontrées, **j'ai décidé de rédiger ce manuel dans le but d'aider ceux qui se lancent dans Odoo et de leur rendre la tâche facile.**

Il ne s'agit ni d'une documentation officielle, encore moins d'un tutoriel.

Il s'agit d'un manuel qui résume ce que j'ai appris au cours des différents projets sur lesquels je suis intervenu.

En parcourant ce manuel, vous serez en mesure de plonger dans l'écosystème Odoo aussi bien du côté technique que fonctionnel et d'être apte à mieux gérer vos projets clients.

Ce manuel s'adresse :

- aux développeurs Python qui veulent élargir leurs compétences
- aux CTO et team lead qui veulent maîtriser l'écosystème Odoo
- aux Freelances et indépendants qui veulent vendre Odoo

2. Comprendre l'écosystème Odoo

Un marché en forte croissance

Odoo est un ERP open source qui intègre toutes les composantes fonctionnelles d'une entreprise : gestion de production, gestion commerciale, logistique, ressources humaines, comptabilité, contrôle de gestion, etc.

À l'aide de ce système unifié, les utilisateurs de différents métiers travaillent dans un environnement applicatif identique qui repose sur une base de données unique. Ce modèle permet d'assurer l'intégrité des données, la non-redondance de l'information, ainsi que la réduction des temps de traitement.

Odoo se positionne aujourd'hui comme un acteur majeur dans le marché des ERP, avec une croissance rapide et une part de marché en expansion auprès des PME grâce à la possibilité de l'intégrer à diverses solutions (e-commerce, banques....), et sa capacité à offrir une gestion complète et personnalisée des opérations d'une entreprise.

En plus, **les tendances actuelles favorisent les solutions ERP modulaires, ouvertes, et abordables, et Odoo excelle dans ce domaine** grâce à sa flexibilité et son modèle open source.

Son écosystème est de plus en plus grandissant avec une large gamme de partenaires officiels, consultants et ESN qui offrent **Odoo comme solution alternative à Sage, SAP et MS Dynamics**, etc.,

Un écosystème large

La modularité de cet ERP permet d'offrir tout ce dont on a besoin pour développer des applications fonctionnelles. Déjà, la version standard offre toutes les bases pour un processus métier fluide du projet sur lequel on travaille.

Si on implémente un e-commerce, le module e-commerce ([website_sale](#)) existe déjà, il suffit juste de l'installer.



En effet, il existe une [marketplace Odoo](#) qui regroupe toutes les applications dont vous aurez besoin ou presque, certaines sont payantes et d'autres gratuites. D'autre part, il existe une association, [Odoo Community Association](#) (OCA) qui développe et maintient des centaines d'applications prêtes à l'emploi. Il suffit juste de les installer et de les utiliser. N'oubliez surtout pas d'apporter vos contributions aux [dépôts de OCA](#) 😊.

Odoo étant un logiciel open source, les développeurs des 4 coins du monde y contribuent et toutes les modifications peuvent être suivies sur le [runbot](#), une plateforme de déploiement continue qui permet de suivre et tester les nouvelles fonctionnalités ou les améliorations apportées au logiciel.

Bien qu'étant open source, l'ERP possède également une version payante, [Odoo Enterprise](#). Les applications de cette version sont développées par la société Odoo SA et seuls les clients et partenaires d'Odoo SA ont accès au code source.

Enfin, Odoo possède aussi son propre cloud, [Odoo.sh](#), une plateforme de CI/CD qui permet de tester et déployer des projets à grande échelle.

La documentation et la méthodologie

Pour ce qui est de la documentation officielle, Odoo propose aujourd'hui une large gamme de ressources :

- [Odoo Learn](#), la plateforme e-learning pour apprendre les bases et au-delà
- La [documentation officielle](#) et les [tutoriels](#) pour aller plus en profondeur
- La [méthodologie d'implémentation Odoo](#) pour mieux gérer les projets clients

Vous venez d'avoir là toutes les ressources pour vous lancer dans Odoo de manière autonome.

Si vous avez besoin de passer par la pratique et prendre en main Odoo rapidement, veuillez juste continuer avec la lecture active du document : par lecture active, je sous-entends lire et pratiquer.

3. Préparer un environnement de dev

Lancer Odoo avec Docker

Le moyen le plus simple de lancer Odoo sur son PC est de le faire à travers Docker qui va télécharger l'image officielle d'Odoo et préparer tout l'environnement. Pour cela, vous n'avez que 2 choses à faire :

Créer un dossier pour votre projet avec la structure suivante :

- **Votre_Projet/**
 - **addons/**
 - **addons-custom/**
 - **addons-external/**
 - **Dockerfile**
 - **docker-compose.yml**
 - **odoo.conf**

Explications :

- **Votre_Projet** est le dossier qui va contenir tout le code et les configurations
- **addons** est le dossier qui va contenir les modules
- **addons-custom** est le dossier situé dans addons qui va contenir vos propres modules et personnalisations techniques
- **addons-external** est le dossier situé dans addons qui va contenir tous les modules communautaires trouvés sur l'appstore d'Odoo
- **odoo.conf** est le fichier de configuration d'Odoo dans lequel on renseigne toutes les config utiles. [Vous pouvez trouver l'aperçu ici.](#)
- **Dockerfile** c'est le fichier docker qui va vous permettre de télécharger Odoo. Son contenu se présente comme suit:

```
FROM odoo:18.0
```

```
USER root
```

```
RUN apt-get update && apt install -y python3-pip python3-debugpy
```

```
COPY ./addons /mnt
```

```
ENV ODOO_RUNNING_ENV=dev
```

```
USER odoo
```



- **docker-compose.yml** est le fichier qui permet de configurer tous les services dont Odoo a besoin dont voici le contenu :

services:

odoo:

platform: linux/amd64

container_name: NOM_DE_VOTRE_CONTAINER_ODOO

build:

context: .

dockerfile: Dockerfile

depends_on:

- db

ports:

- "8069:8069"

tty: true

stdin_open: true

restart: unless-stopped

volumes:

- odoo-data:/var/lib/odoo

- ./odoo.conf:/etc/odoo/odoo.conf

- ./addons:/mnt

db:

container_name: NOM_DU_CONTAINER_DE_BD

image: postgres:13

environment:

- POSTGRES_DB=postgres
- POSTGRES_PASSWORD=odoo
- POSTGRES_USER=odoo
- PGDATA=/var/lib/postgresql/data/pgdata

ports:

- "5432:5432"

restart: unless-stopped

volumes:

- postgres-data:/var/lib/postgresql/data/pgdata

volumes:

odoo-data:

postgres-data:

💡 Pour recevoir d'autres astuces pratiques et poser tes questions directement, rejoins notre canal WhatsApp Odoo 👉

<https://whatsapp.com/channel/0029Vb6hrqBCBtxKZcMfQj3U>

Les extensions utiles

Pour un développement aisé, voici les extensions que vous pouvez utiliser

Sur le navigateur :

- [Odoo Debug](#) pour activer et désactiver le mode développeur en un clic
- [OdooTerminal](#) pour accéder aux modèles et méthodes de l'ORM depuis le navigateur
- [Owl devtools](#) pour accéder aux composants javascript dans le Chrome DevTools

Les extensions VS Code :

- [XML Tools](#) pour le confort dans les fichiers XML
- [Odoo Language Server](#) pour la prise en compte des spécificités d'Odoo lors du développement

4. Architecture d'Odoo — Ce qu'il faut vraiment comprendre

Le modèle MVC

Odoo est un ERP et un framework architecturé sous un modèle [3-tiers](#) client-serveurs :

1. le serveur de base de données [PostgreSQL](#)
2. le serveur d'application Odoo basé sur [Werkzeug](#)
3. le client web basé sur JavaScript, utilisant [OWL](#) depuis la version 14

La structure des modules

Chaque module dépend d'un ou plusieurs autres modules et présente une structure qui contient tous les éléments nécessaires à son fonctionnement. Par exemple, si vous allez dans le module [point_of_sale](#), vous allez constater que ce dernier contient plus ou moins la structure suivante :

- le dossier **data** où toutes les données statiques sont conservées (menus, crons jobs, données à exporter, etc.)
- le dossier **models** qui contient tous les modèles de la base de donnée basées sur l'ORM d'Odoo
- le dossier **reports** à partir duquel tous les rapports PDF et autres documents à imprimer sont chargés
- le dossier **security** pour la conservation de toutes les règles de sécurité et autres droits d'accès
- le dossier **views** pour les différentes vues écrites en XML
- le dossier **wizard** pour toutes les fenêtres modales
- le dossier **static** pour les images et autres codes JavaScript et CSS
- le fichier **README.md** pour la description du projet
- les fichiers **__init__** et **__manifest.py__** obligatoires pour le chargement et l'installation du module



Vous pouvez consulter [cet article publié sur notre site](#) pour en savoir plus notamment sur les attributs et méthodes de l'ORM utilisé au niveau des modèles.

Les relations et dépendances entre modules

Les modules interagissent entre eux à travers les champs relationnels. Ces champs permettent de créer des liens entre différents modèles, facilitant la gestion de relations entre les informations de divers modules.

Par exemple, le champ **pos_payment_method_ids** dans le module point de vente définit une relation One2Many avec le modèle **account.journal** et cela a pour conséquence de faire en sorte que le module **point_of_sale** dépende du module **account**.

Ainsi, on se rend compte que les modules peuvent être dépendants les uns des autres, ce qui signifie que l'installation ou la mise à jour d'un module peut affecter d'autres.

Certains modules dépendent d'autres pour fonctionner et nécessitent leur installation préalable. C'est le cas de **point_of_sale** qui dépend de **account**.

L'installation d'un module peut entraîner l'installation automatique de modules dépendants. En installant **point_of_sale**, on installe plusieurs autres modules comme **stock**, **barcode**, **account**, etc.

Les modules peuvent interagir pour offrir des fonctionnalités intégrées. Par exemple, le module **CRM** interagit avec le module **Contacts** pour gérer les relations clients.

5. Créer son premier module — étape par étape

Cette section vous guide pas à pas dans la création de votre premier module Odoo sur la base d'un module existant. En effet, dans le but d'éviter de mettre en place des fonctionnalités qui existent déjà, il est important d'étendre les modules Odoo existants pour ajouter vos fonctionnalités spécifiques.

Dans cet exemple, **l'objectif est d'ajouter un système de notation par étoiles (1 à 5) au formulaire de contact pour évaluer la qualité de la relation client.**

Réalisez chaque étape par vous-même et ne passez pas à l'étape suivante tant que vous n'avez pas compris l'étape en cours.

Étape 1: préparation et analyse

- Installer Odoo avec Docker comme expliqué à la section 3 de ce manuel
- Lancez Odoo, créez votre base de donnée dans l'interface qui s'affiche et installez le module contacts
- Créez un contact et examinez la structure du formulaire pour identifier où vous souhaitez placer vos champs (notation & commentaires)

Étape 2: structure du module

- Dans votre dossier addons-custom, créez un nouveau répertoire avec la structure appropriée comme expliqué à la section 4 de ce manuel.
- Choisissez un nom de module en anglais, sans espaces, qui reflète sa fonction, par exemple custom_contact_rating
- Définissez le contenu du fichier `__manifest__` (name, depends, etc.) . N'hésitez pas à regarder le manifest d'un module Odoo pour vous en inspirer
- Créez les fichiers `__init__.py` pour que Python reconnaisse vos dossiers comme des packages.

Étape 3: Modèle de données

- Créez votre classe Python qui étend le modèle `res.partner` (`__inherit="res.partner"`)
- Définissez un champ Selection avec 5 options (1 à 5 étoiles) avec des libellés explicites (`fields.Selection`)
- Créez un champ Text pour permettre de noter des explications sur la notation (`fields.Text`)
- Ajoutez une méthode qui se déclenche lors du changement de notation pour afficher des messages contextuels. Par exemple, si un client a une notation de 1, afficher le message "ce client nécessite un suivi particulier" (`@api.onchange`)

Étape 4: Interface utilisateur

- Modifiez la vue existante (`base.view_partner_form`) pour intégrer vos champs de notation à un endroit spécifique du formulaire principal.
- Rendez la notation invisible pour les contacts de type entreprise (invisible si `is_company = True`).
- Modifiez la vue liste (`base.view_partner_tree`) pour afficher la notation dans une colonne optionnelle.

Étape 5: Installation et test

- Mettez à jour la liste des applications dans Odoo (menu Applications)
- Recherchez et installez votre module.
- Testez la fonctionnalité en vérifiant que la notation apparaît bien, que les messages contextuels fonctionnent
- Vérifiez que la colonne notation s'affiche dans la vue liste des contacts.

🤔 Qu'avez-vous appris et constaté ?

👉 La vidéo explicative est disponible ici : <https://youtu.be/XksYHEWZF5E>

👉 Et le code source disponible ici : https://github.com/nasares/odoo-apps/tree/17.0/custom_contact_rating

6. Cadrer un projet client Odoo

Analyse des besoins métiers

Avant de penser à mettre en place Odoo, **il est impératif de comprendre exactement les procédures et le métier de l'entreprise** et comment les gens travaillent au quotidien.

Parfois, le besoin d'un client n'est pas d'avoir un logiciel, mais d'**apporter une solution à un seul problème qui est lié soit au stock, soit à la comptabilité.**

Identification des problèmes réels

Si un client dit qu'il veut "digitaliser son entreprise" ou "Améliorer la gestion", ça reste du flou et dans ce cadre-là, aucun problème réel n'est identifié.

Les problèmes réels doivent être par exemple :

- **On passe trop de temps sur les devis**
- **On veut voir le stock disponible en temps réel**
- **On subit trop de vols de la part de nos employés**
- **On veut éviter les erreurs de saisie**
- **etc.**



Surtout, le piège à éviter c'est de lister plusieurs problèmes. Il faut **se concentrer sur 3 à 5 problèmes critiques** uniquement.

Analyse des processus existants

Il est **impératif de documenter comment l'entreprise travaille aujourd'hui, étape par étape**. Si les workflows ne sont pas connus, ça sera un échec total parce qu'un ERP doit s'adapter à l'entreprise et non l'inverse.

Par exemple, une entreprise peut expliquer son processus de vente ainsi :

- Nos commerciaux font la prospection sur le terrain
- Ils notent toutes les informations sur le papier
- Ils rentrent au bureau et enregistrent tout sur Excel
- Après, ils préparent le devis sur Word
- etc.

À ce niveau, quand l'entreprise explique comment elle travaille, il faut éclaircir certains points :

- **À quel niveau est-ce-que vous perdez le plus de temps ?**
- **Où faites-vous le plus d'erreurs ?**
- **Quelles sont les informations qui reviennent chaque fois ?**
- **Qui fait quoi et quand ?**

Sur la base de toutes ces réponses, vous pouvez rédiger un cahier des charges assez simple et pratique qui va permettre de planifier et facturer le client.

💡 Si tu veux accéder à des stratégies d'implémentation Odoo plus poussées (méthodologie projet, déploiement, retour d'expérience), rejoins aussi le canal WhatsApp Odoo 🙌 <https://whatsapp.com/channel/0029Vb6hrqBCBtxKZcMfQj3U>

7. Comment je me suis positionné comme consultant Odoo ?

Pourquoi choisir "consultant Odoo" plutôt que "développeur Python" ?



Quand j'étais encore étudiant, la perspective du chômage après mes études me préoccupait. Il fallait que je trouve un moyen de me démarquer sur le marché du travail et d'éviter absolument le chômage. Lors d'un stage, j'ai découvert Odoo, et cela a changé ma trajectoire professionnelle.

Au lieu de me présenter comme un simple "développeur Python" parmi tant d'autres, je me présentais fièrement comme développeur Odoo. Cet ERP était en forte demande, mais le nombre de personnes qualifiées restait limité au Cameroun. En me positionnant comme consultant Odoo, les employeurs voyaient en moi le spécialiste qu'ils recherchaient.

Odoo connaît aujourd'hui une croissance exceptionnelle dans les PME qui recherchent des solutions ERP abordables et flexibles, ce qui fait d'Odoo le choix idéal. En vous spécialisant dans cette technologie, vous vous positionnez sur un marché porteur avec plus de demandes.

Les meilleurs canaux pour lancer sa carrière dans Odoo

Le bouche-à-oreille : jusqu'à aujourd'hui, le bouche-à-oreille reste mon levier d'acquisition le plus efficace. Chaque projet réussi génère naturellement de nouvelles opportunités.

Mon CV et portfolio : c'est grâce à mon CV enrichi de réalisations concrètes et mesurables que j'ai décroché un emploi à temps plein en Allemagne, tout en étant encore au Cameroun.

Mes blogs : j'ai acquis des clients et des emplois grâce à mes blogs dans lesquels je partageais mes connaissances techniques sur Odoo. En créant du contenu, j'ai établi ma crédibilité et intéressé ceux qui recherchaient mon expertise. Parmi tous ces sites, seul peef.dev est actif aujourd'hui.

Construire sa crédibilité

Contenu technique de qualité : j'ai investi du temps dans la création de contenus techniques qui démontrent mon expertise

Travail de qualité irréprochable : je fais tout mon possible pour offrir un travail qui ne doit pas compromettre la qualité, même sous la pression des délais.



Communication claire et professionnelle : Au-delà des compétences techniques, la communication reste cruciale. Au départ, c'était compliqué, mais j'ai appris à maintenir une communication transparente partout où je travaille.

L'importance de contribuer à l'écosystème open source

Je ne suis pas un contributeur très actif de l'open source, mais je sais qu'elle est déterminante pour la crédibilité. En participant activement au développement de modules communautaires, vous pouvez :

- **Démontrer votre expertise technique** auprès de la communauté
- **Construire un réseau professionnel solide** avec d'autres experts Odoo
- **Acquérir une visibilité** qui génère des opportunités business inattendues
- **Apprendre des meilleures** en collaborant avec des développeurs expérimentés

En vous positionnant comme consultant Odoo spécialisé plutôt que comme développeur Python généraliste, vous pouvez transformer votre carrière professionnelle et créer des opportunités auprès de vous.

8. Conclusion

Nous voici arrivée à la fin de ce manuel que j'aurais aimé avoir quand je me suis lancé sur Odoo et j'espère qu'il vous sera d'une grande utilité.

Pour aller plus loin et vraiment progresser sur Odoo, j'ai créé un canal WhatsApp privé.

Tu y trouveras :

- ✓ Des guides PDF exclusifs (installation, configuration, dev avancé)
- ✓ Un environnement Docker prêt à l'emploi pour lancer Odoo en un clic
- ✓ Des modules Odoo complets pour t'inspirer
- ✓ Des stratégies d'implémentation basées sur +10 projets réels
- ✓ Un accompagnement direct avec moi et l'accès à d'autres développeurs Odoo



👉 Rejoins le canal WhatsApp maintenant :

<https://whatsapp.com/channel/0029Vb6hrqBCBtxKZcMfQj3U>

© ABDOU NASSER – Tous droits réservés

Ce document est protégé par le droit d'auteur. Toute reproduction, distribution ou diffusion non autorisée est interdite.

Vous pouvez partager le lien d'accès, mais pas le contenu sans mon accord écrit.

*Peef est en collaboration avec la société Proxima Technologies Cameroon SAS
située à Douala - Cameroun NIU: M122117229052A - RCCM: RC/DLN/2021/B/2884
Email : hello@peef.dev — Site web : www.peef.dev*