

Ingenierie SI ERP/Lims-BISD

Projet : Développement d'un modules Odoo

Annassiri Fatima Zahra & Jemlaoui Ikrame

MBISD2_ 2021

Introduction

Dans le cadre de l'évaluation de module Ingénierie SI ERP, on va développer un module qui 'est répond a un besoin spécifié en utilisent l'open ERP Odoo.

Odoo est l'un des ERPs les plus connus et les plus utilisés. Il est un progiciel open source de gestion intégré qui comprend de très nombreux modules pour répondre au besoin de gestion des entreprises (ERP).

Cahier de charge :

1. Problématique :

Pour gérer l'activité globale des patients au sein de l'hôpital, chaque service utilise indépendamment des applications informatiques et outils personnels. Ceci génère une difficulté d'échange de données entre les différents départements de l'hôpital. N'oublions pas l'hétérogénéité et l'existence des surcharges au niveau des rendez-vous. Ce qui ajoute des complexités de plus à la vie des patients qui souffrent déjà, un souci de plus !

2. Objectif et fonctionnalité :

L'objectif principal de notre projet consiste à automatiser un ensemble des processus des patient au sein d'hôpital "Gestion des patient", tout en travaillant sur une base de données unique et homogène afin de réduire les redondants et simplifier le processus des registres.

Notre mission donc consiste donc à développer et testé un module Odoo, répond aux besoins.

Solution & Test:

1. Architecture technique de l'OpenERP

Architecture Client /Serveur

Open ERP est basé sur une architecture client/serveur. Le serveur et le client communiquent via le protocole XML-RPC. C'est un simple protocole qui permet au client de faire des appels aux Procédures.

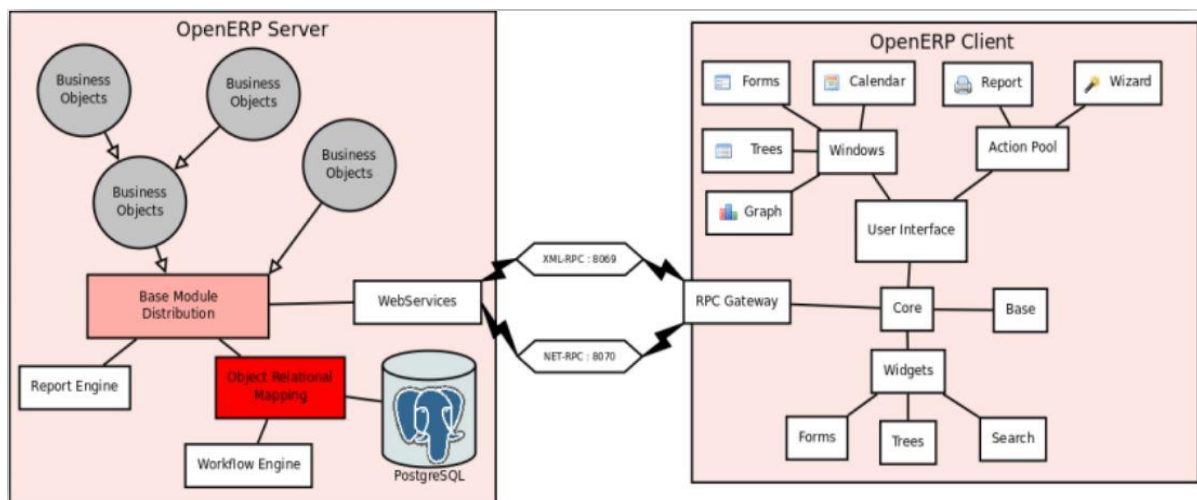
Une fois la fonction est appelée, ses arguments et ses résultats sont envoyés par le Protocole http, eux-mêmes sont encodés par le langage XML.

OpenERP est couplé à une base de données PostgreSQL. De plus, il est compatible au pack Open Office, et aussi avec des outils de reporting (ReportLab) pour produire des rapports en PDF ou en HTML. La logique d'openERP est entièrement du côté serveur. La tâche du client se résume à demander les données (formulaire ou listes) au serveur et de les renvoyer.

Avec cette approche, presque tout le développement est fait du côté serveur. Ce qui rend OPENERP plus simple au développement et à la maintenance. L'opération client est très simple. Quand un utilisateur exécute une action (sauvegarder un formulaire, ouvrir un menu, imprimer, ...) il envoie cette action au serveur. Le serveur envoie alors la nouvelle action pour s'exécuter côté client.

Il y a trois types d'actions :

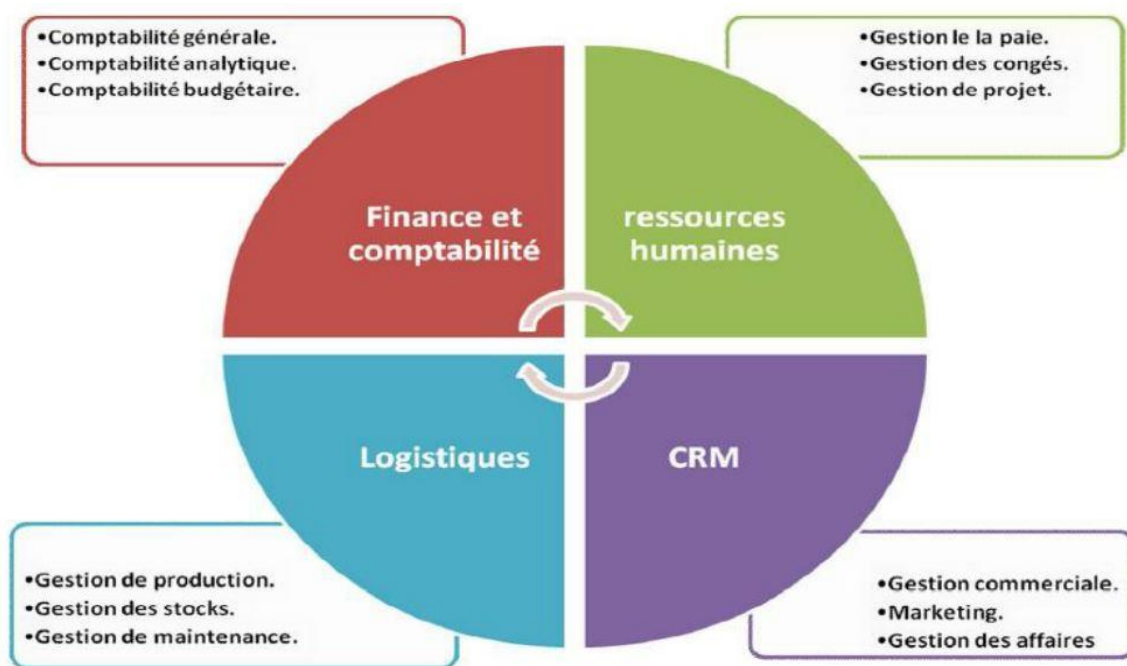
- Ouvrir une fenêtre (formulaire, listes)
- Imprimer un document.
- Exécuter un wizard.



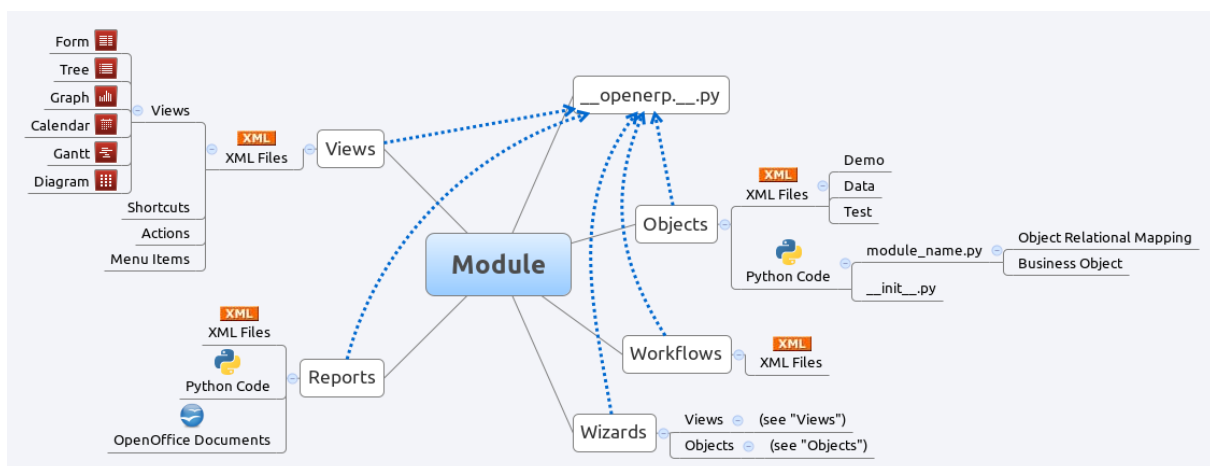
Architecture modulaire d'OpenERP / ODOO

Un module OpenERP est la définition, dans le « Framework » OpenERP, d'une gestion informatisée d'un domaine. Cette architecture n'est pas propre à open ERP. Elle est en fait partagée par tous les ERP. Il s'agit de la faculté de construire des applications informatiques de manière modulaire (modules indépendants entre eux) tout en partageant une base de données unique.

Ceci apporte une importance significative puisque les données sont maintenant standardisées et partagées. Ce qui élimine les saisies multiples et évite l'ambiguïté des données de même nature. L'architecture modulaire d'open ERP lui permet de couvrir plusieurs domaines illustrés dans la figure ci-dessous :



La figure ci – dessus illustre la Structure d'un module Odoo :



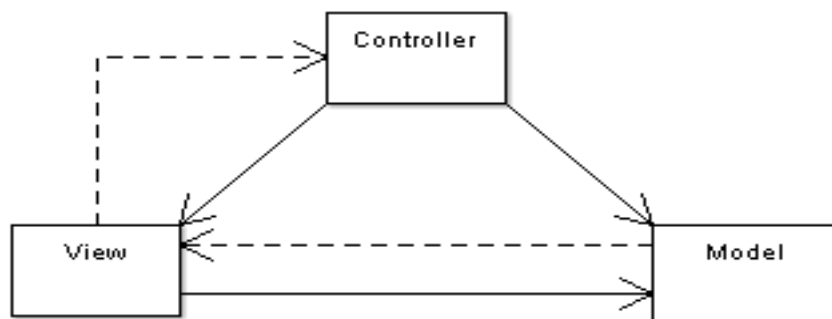
Modèle vue contrôleur :

Un (MVC) est une architecture de modèles utilisée en génie logiciel. Dans des applications complexes qui présentent des lots de données aux utilisateurs, on souhaite souvent séparer les données (modèle) et l'interface utilisateur (vue), de sorte que les changements à l'interface utilisateur n'affectent pas le traitement des données, et que les données peuvent être réorganisées sans changer l'interface utilisateur.

Le MVC résout ce genre de problème en découplant l'accès des données et la logique des applications de la présentation des données et de l'interaction utilisateur, en introduisant un composant intermédiaire : « le contrôleur ».

Dans open ERP, on peut appliquer cette sémantique de Model View Controller avec :

- Model : les modèles sont les objets déclarés dans OpenERP. Ils sont également des tables PostgreSQL.
- View : les vues sont définies en fichiers XML dans OpenERP.
- Controller : le contrôleur est les classes Python qui contrôle OpenERP.



2. Technologie mise en œuvre :

Ce paragraphe décrit les différentes technologies adoptées et utilisées pour la réalisation de ce projet, à commencer par le système d'exploitation Windows, tout en passant par le PGI OpenERP Odoo, le système de gestion de bases de données PostgreSQL, les langages nécessaires pour le développement, à savoir le langage Python et XML.

Odoo :

Odoo, anciennement OpenERP et Tiny ERP, est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble.

Il est utilisé par plus de deux millions d'utilisateurs pour gérer leurs entreprises à travers le monde. Odoo est le système ERP (Enterprise Resource Planning) open-source le plus populaire.



PostgreSQL:

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.



Python:

Python est un langage de programmation multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix en passant par Linux et Mac OS, avec Java ou encore .NET.

Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. Il est également apprécié par les pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation plus aisée aux concepts de base de la programmation.

Python est un langage :

Conçu pour produire du code de qualité, portable et facile à intégrer : grâce à sa syntaxe claire, cohérente et concise, Python permet aux développeurs de produire du code de qualité, lisible et maintenable.

Fourni dès le départ avec des modules de tests, Python est un langage agile. Le terme agile est originellement issu de la méthodologie de programmation agile, très proche de la programmation itérative. Cette méthodologie, qui réduit les risques liés à la conception de logiciels, introduit entre autres des principes de tests continus du code.

De haut niveau, orienté objet et totalement libre : même si elle n'est pas imposée, Python permet la programmation orientée objet. Tous les mécanismes objet essentiels sont implémentés et toutes les données manipulées sont des instances de classes, comme pour les langages SmallTalk ou Ruby. Dynamique : dans la plupart des implémentations, le code source n'est pas compilé contrairement à des langages comme C ou Pascal, mais exécuté à la volée. On parle alors de langage interprété. Ce mode de fonctionnement rend la programmation beaucoup plus souple puisqu'il est possible de changer un programme en cours d'exécution.



Xml:

Contrairement à HTML, qui est considéré comme un langage défini et figé (avec un nombre de balises limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte (Qui n'a jamais désiré une balise qui n'existait pas ?).

La force de XML réside dans sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir.

En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement à HTML). Ainsi, XML permet de séparer le contenu de la présentation, ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations.

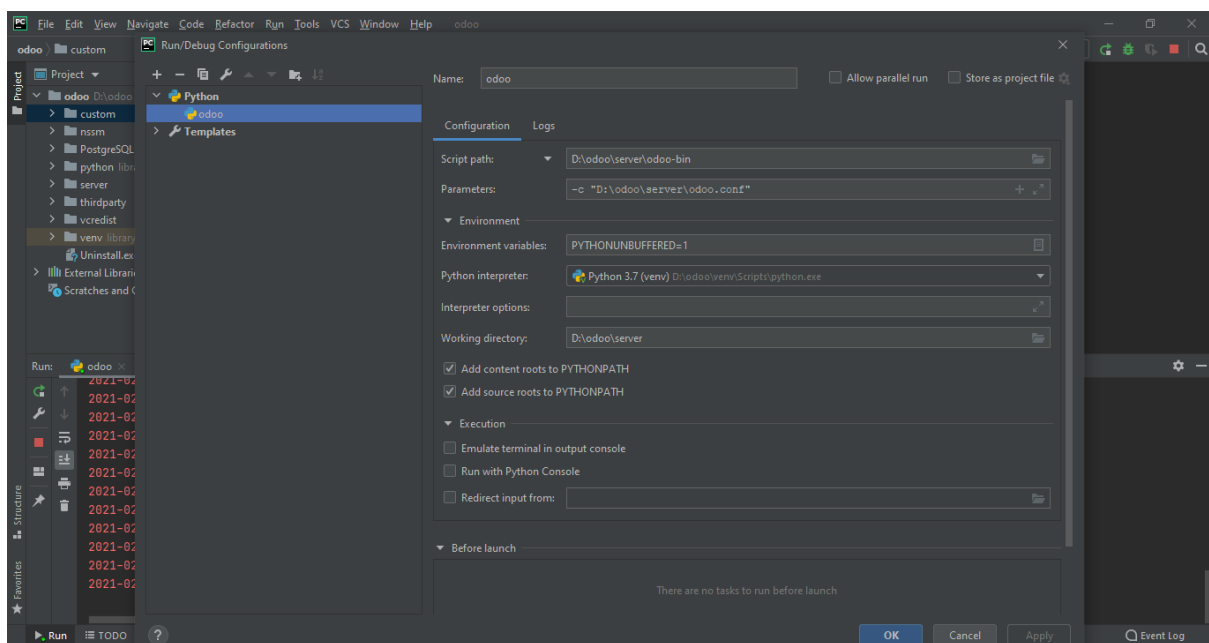
XML a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) dès 1996. Depuis le 10 février 1998, les spécifications XML 1.0 ont été reconnues comme recommandations par le W3C, ce qui en fait un langage reconnu. XML est un sous ensemble de SGML (Standard Generalized Markup Language), défini par le standard ISO8879 en 1986, utilisé dans le milieu de la Gestion Electronique Documentaire (GED).

XML reprend la majeure partie des fonctionnalités de SGML, il s'agit donc d'une simplification de SGML afin de le rendre utilisable sur le web , XML fait partie du code des modules composants OpenERP, les vues par lesquelles sont représentés les différents objets sont écrites en XML, ainsi nous y trouvons la description détaillée de l'affichage des arbres, formulaires, menus et autres actions.

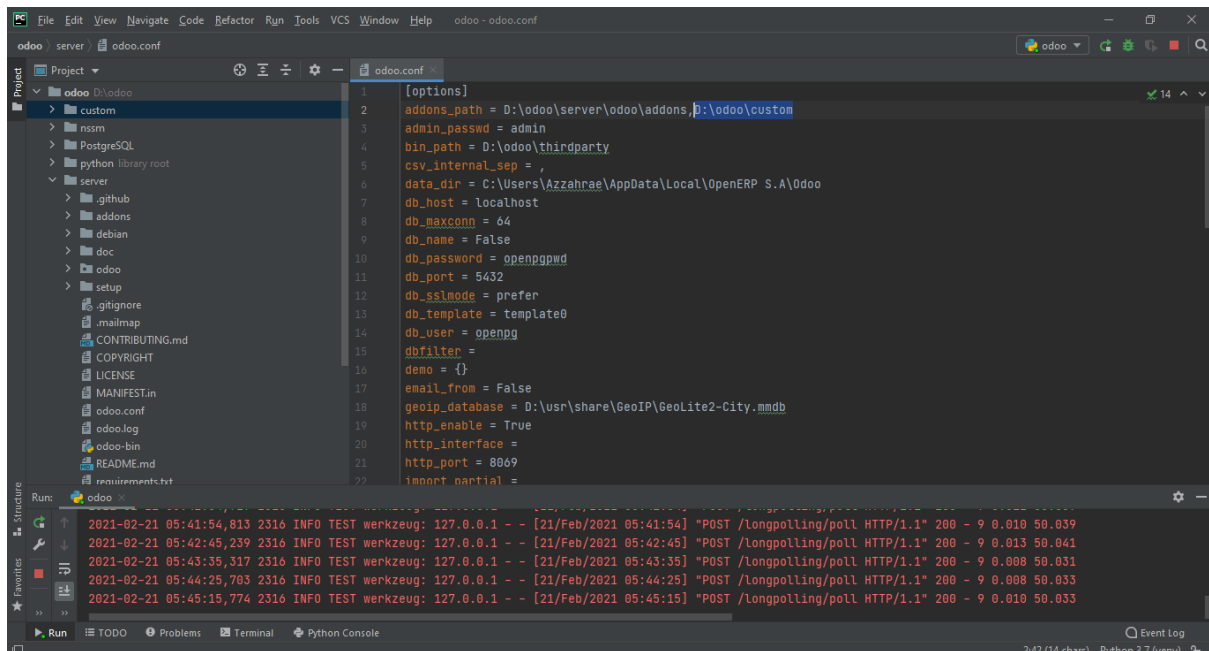


3. Manipulation des modules

Configuration Odoo avec pycharm :



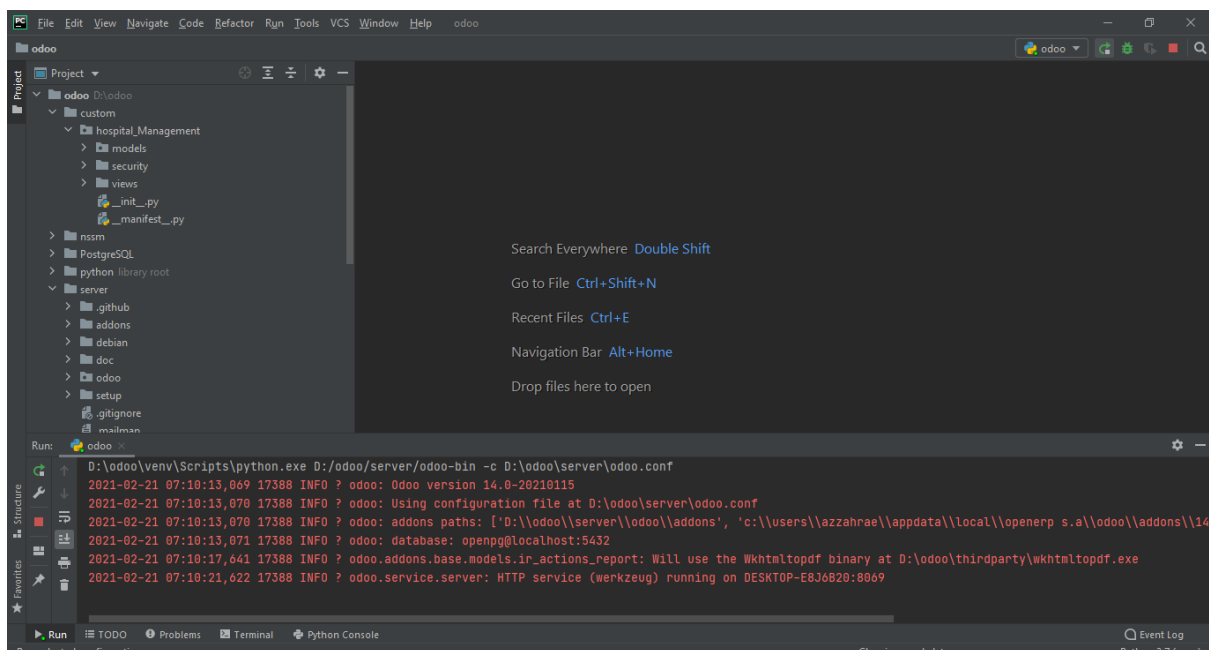
Création d'un nouveau dossier pour les nouveaux module a développer et ajouter son path au fichier .conf , ensuite run le projet



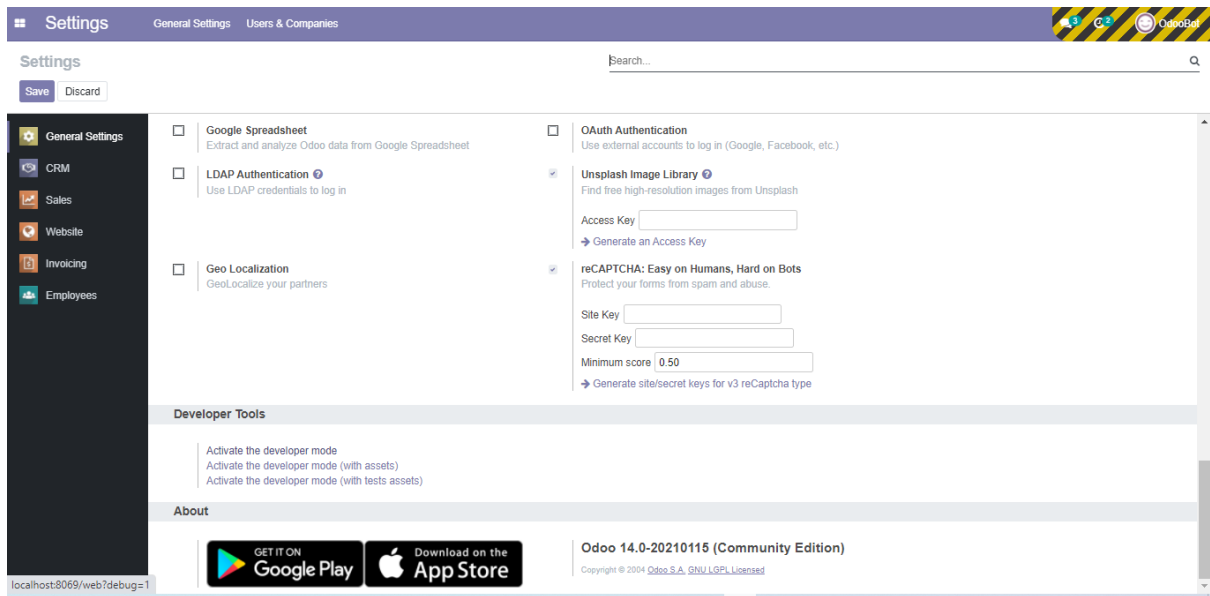
Crée un nouveau module à l'aide du command :

"D:\odoo\python\python.exe" "D:\odoo\server\odoo-bin" scaffold hospital "D:\odoo\custom\"

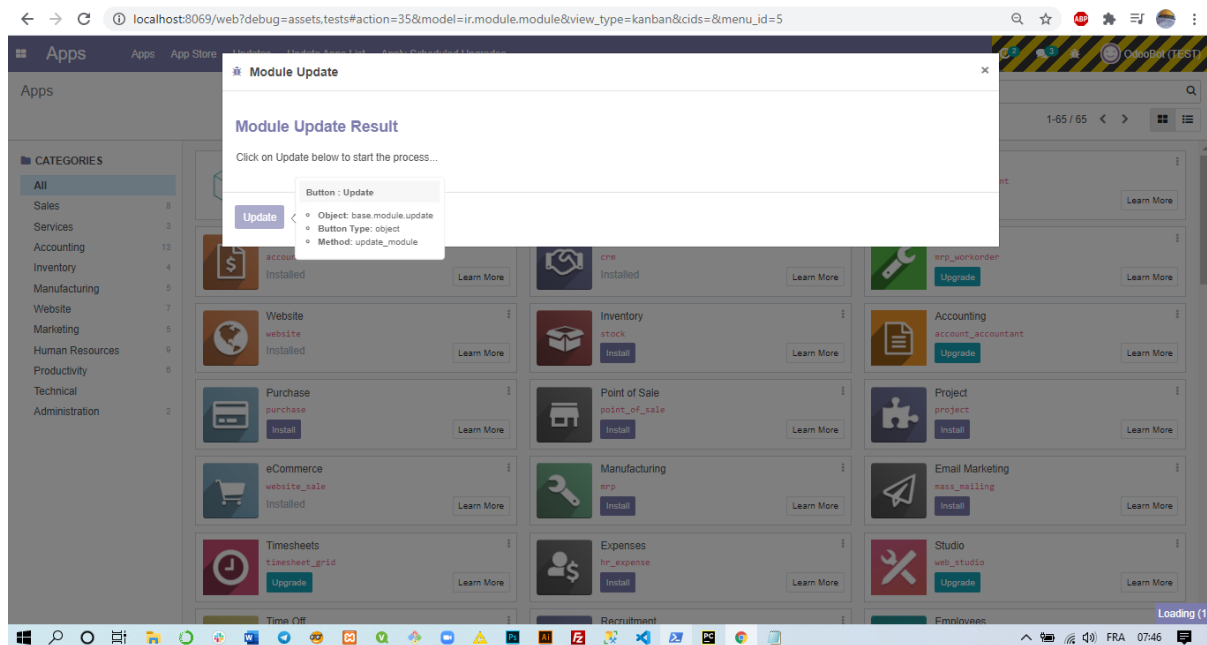
Le dossier de nouveaux module est bien créer, ensuite en relance le projet Odoo pour restart les service :

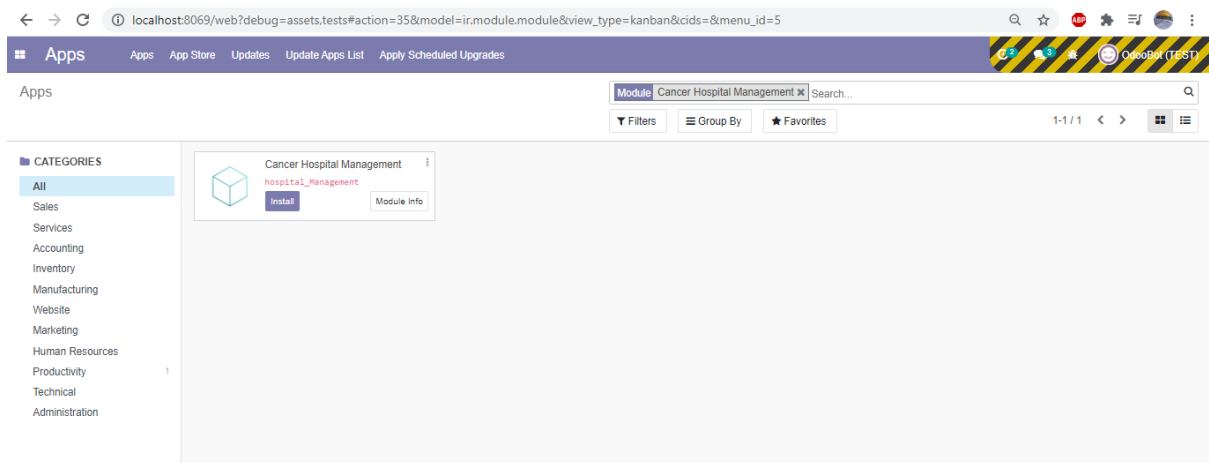


Le http service et lancé avec succès sur le port 8069, dans le browser on tape : <http://localhost:8069/web>, ensuite setting et en switch en mode développeur :



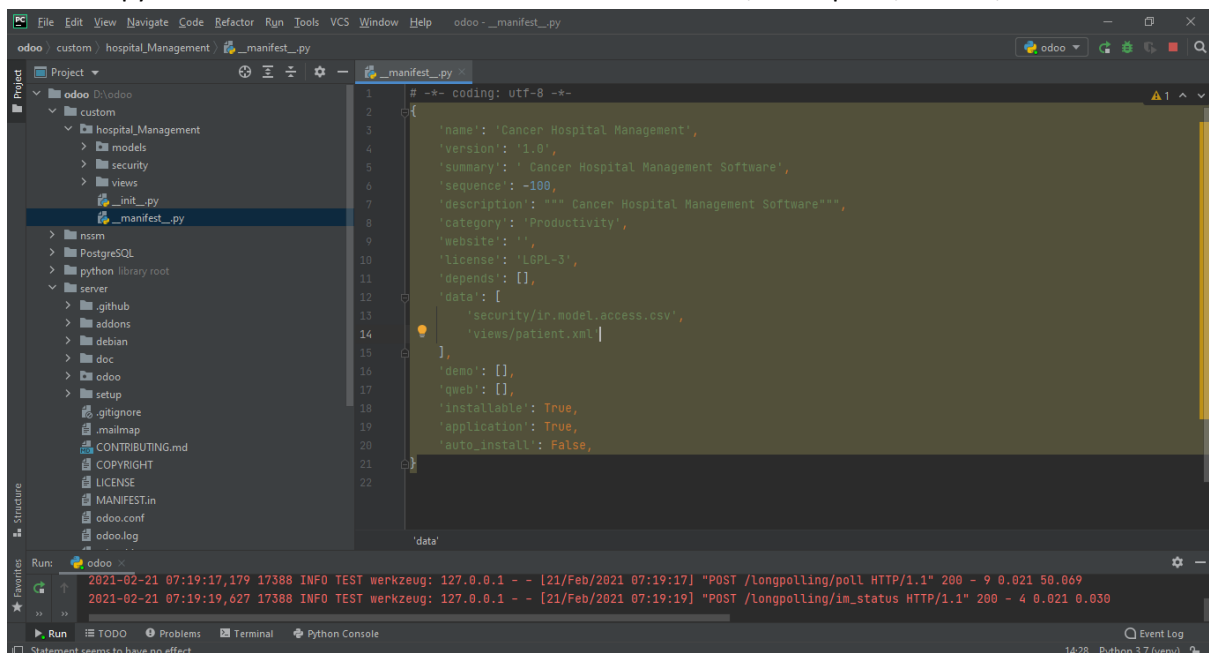
Ensuite en switch en mode debugge, super user on fait une update a la liste des modules et au menu :



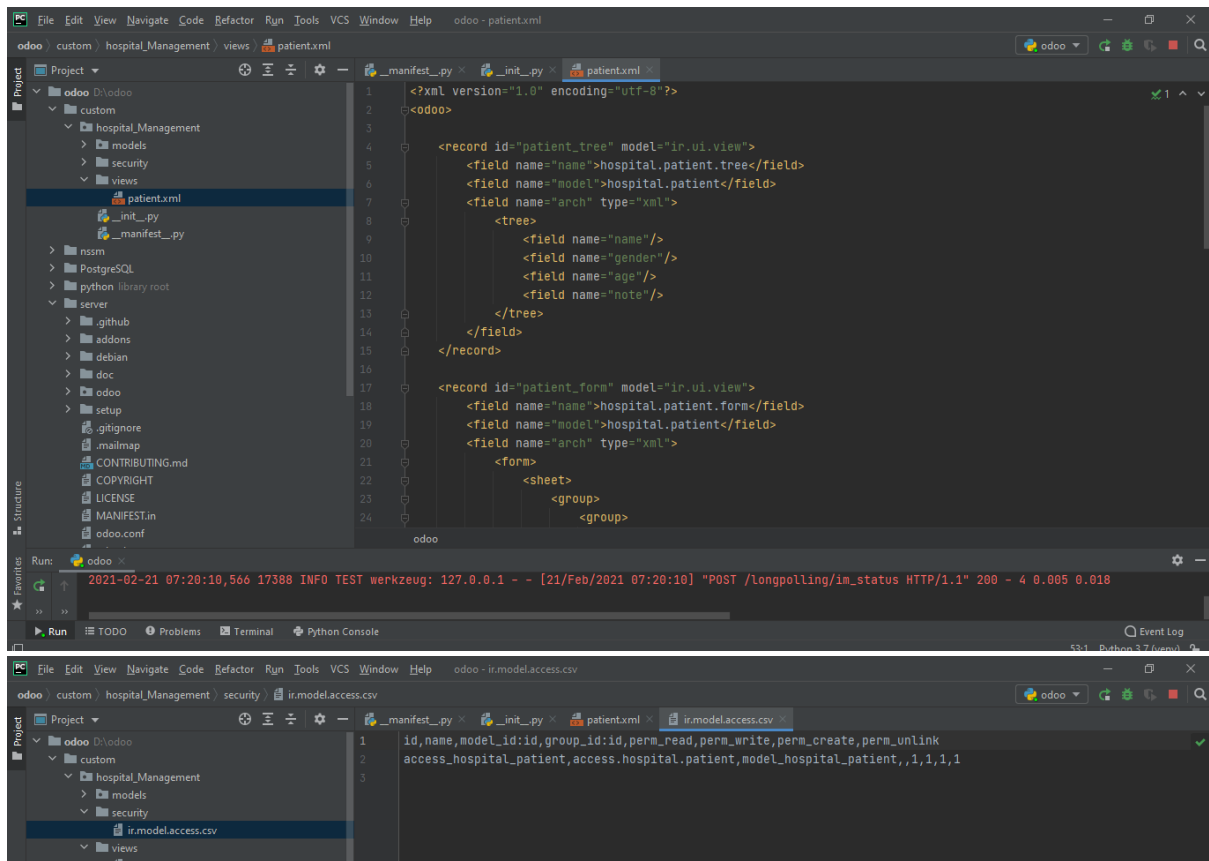


Le codage des fichiers de module :

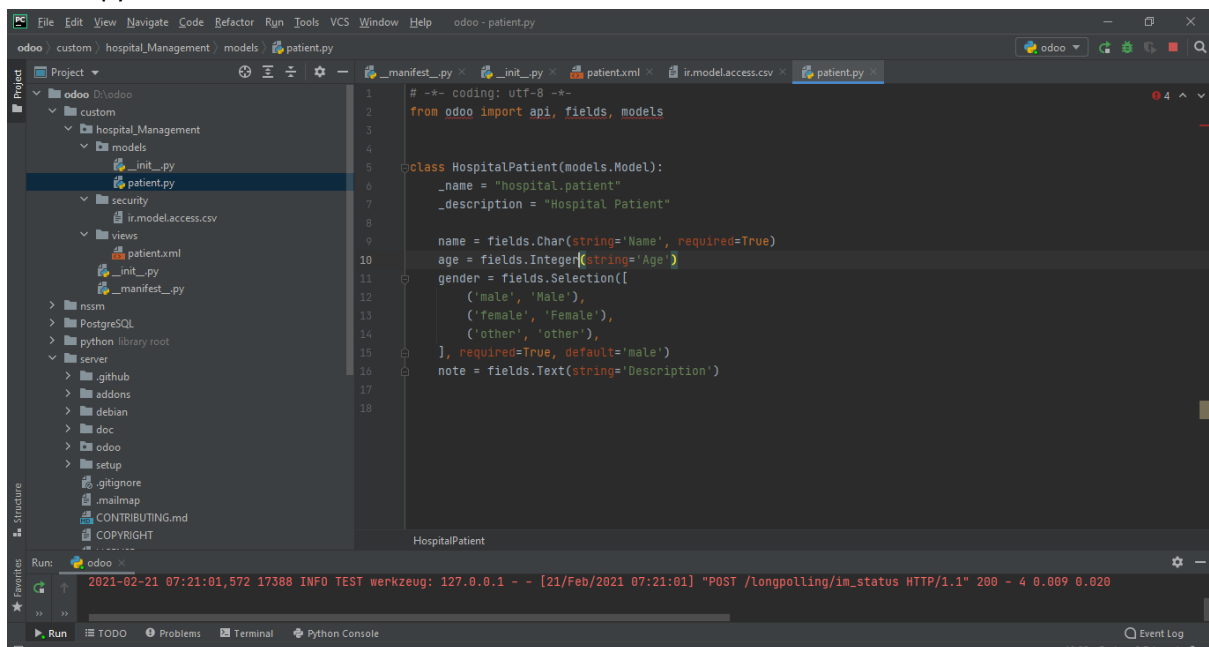
Manifest.py : contient tous les informations de module Name, description, version, data ex ..



View -> patient.xml contient les champs de base de données ainsi ls information de menu , il assure la relation client server via un web service



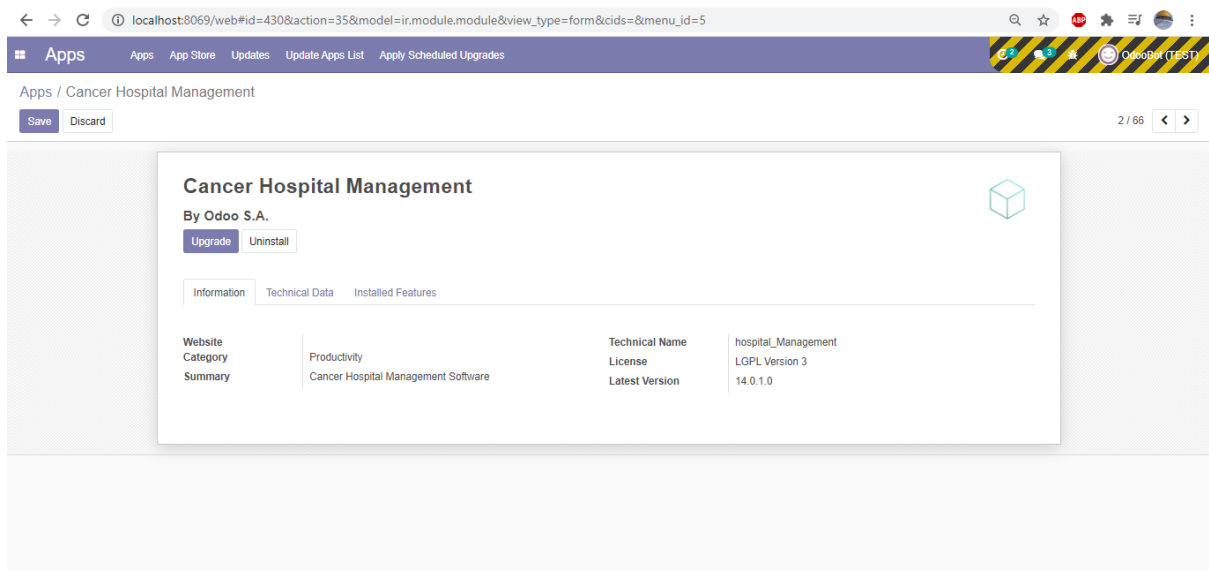
Patient.py contient le model et les classe du BD



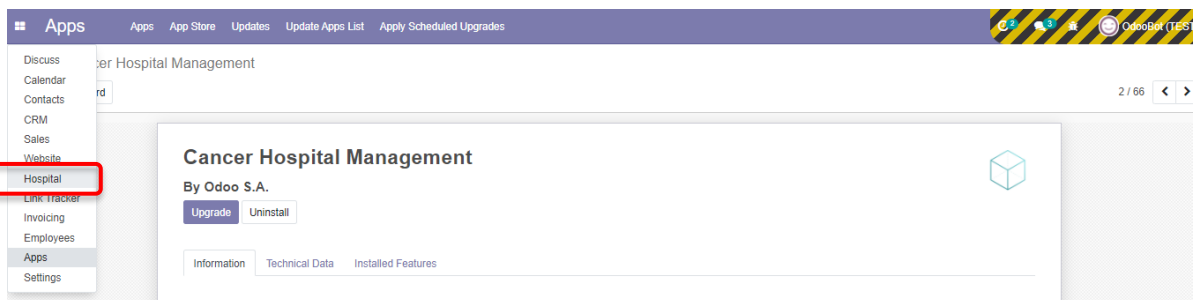
L'installation de module et le test :

```
1 # -*- coding: utf-8 -*-
2 from odoo import api, fields, models
3
4
5 class HospitalPatient(models.Model):
6     _name = "hospital.patient"
7     _description = "Hospital Patient"
8
9
10     name = fields.Char(string='Name', required=True)
11     age = fields.Integer(string='Age')
12     gender = fields.Selection([
13         ('male', 'Male'),
14         ('female', 'Female'),
15     ],
```

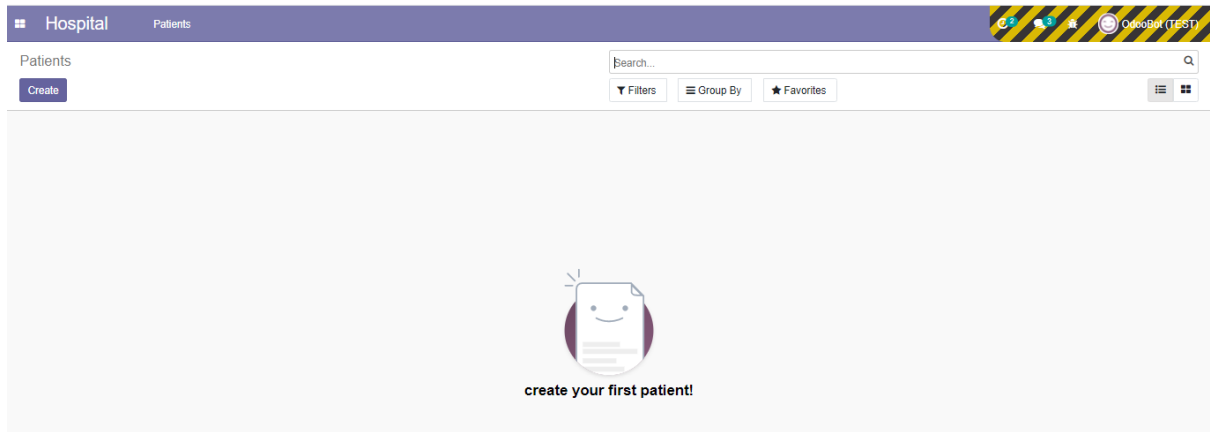
L'installation est donc !, pour plus d'information sur le module :



Le module maintenant et appartient au menu, cliquent sur Hospital dans le menu :



La liste est encore vide, cliquant sur create pour créer notre premier patient :



Voilà le test est done ! Avec succès :

