eBook Gratuit

APPRENEZ odoo-8

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#odoo-8

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: odoo-8

It is an unofficial and free odoo-8 ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official odoo-8.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec odoo-8

Remarques

Cette section fournit une vue d'ensemble de ce qu'est odoo-8 et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans odoo-8, et établir un lien avec les sujets connexes. La documentation pour odoo-8 étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

Numéro de version	Communauté	Entreprise	Licence	Date de sortie
8.0	Oui	Non	GNU AGPL	2014-09-18
9.0	Oui	Oui	GNU AGPL V3	2015-10-01

Examples

Installer

Odoo peut être installé de trois manières différentes:

1. Installateurs intégrés (plus faciles, moins flexibles)
2. Installation de la source (prend parfois du temps à configurer, très flexible)
3. Une image officielle de docker de docker.com

Les paquets officiels avec toutes les exigences de dépendance pertinentes sont disponibles sur odoo.com .

les fenêtres

Téléchargez et exécutez le programme d' installation .

Remarque: sous Windows 8, vous pouvez voir un avertissement intitulé "Windows a protégé votre PC". Cliquez sur Plus d'infos, puis exécutez-le quand même. Acceptez l'invite UAC et suivez les différentes étapes d'installation. Odoo sera automatiquement démarré à la fin de l'installation.

Configuration:

Le fichier de configuration se trouve à l'adresse% PROGRAMFILES% \ Odoo 8.0-id \ server \ openerp-server.conf. (id est votre nom d'utilisateur système)

Le fichier de configuration peut être modifié pour se connecter à un serveur PostgreSQL distant, modifier les emplacements des fichiers ou définir un filtre de base de données. Pour recharger le fichier de configuration, redémarrez le service Odoo via Services server odoo server.

Linux

Distributions basées sur Debian

Pour installer Odoo 8.0 sur une distribution basée sur Debian, exécutez les commandes suivantes en tant que root:

```
# wget -O - https://nightly.odoo.com/odoo.key | apt-key add -
# echo "deb http://nightly.odoo.com/8.0/nightly/deb/ ./" >> /etc/apt/sources.list
# apt-get update && apt-get install odoo
```

Cela installera automatiquement toutes les dépendances, installera Odoo comme démon et le lancera automatiquement.

Notez que

pour imprimer des rapports PDF, vous devez installer wkhtmltopdf vous-même: la version de wkhtmltopdf disponible dans les référentiels debian ne prend pas en charge les en-têtes et les pieds de page de sorte qu'elle ne puisse pas être installée automatiquement. La version recommandée est 0.12.1 et est disponible sur la page de téléchargement de wkhtmltopdf, dans la section archive. Comme il n'y a pas de version officielle pour Debian Jessie, vous pouvez trouver le paquet sur http://nightly.odoo.com/extra/ . ou vous pouvez le télécharger et l'installer depuis la page de téléchargement de wkhtmltopdf comme ceci

```
# wget https://bitbucket.org/wkhtmltopdf/wkhtmltopdf/downloads/{path to correct distro and
system architecture}
# sudo dpkg -i {.deb package}
# sudo cp /usr/local/bin/wkhtmlto* /usr/bin/
```

Le fichier de configuration se trouve dans /etc/odoo/openerp-server.conf

Lorsque le fichier de configuration est édité, Odoo doit être redémarré en utilisant service:

$ sudo service odoo restart Redémarrage de odoo: ok

Distributions basées sur RPM

Avec les distributions basées sur RHEL (RHEL, CentOS, Scientific Linux), EPEL doit être ajouté aux référentiels de la distribution pour que toutes les dépendances d'Odoo soient disponibles. Pour CentOS:

```
$ sudo yum install -y epel-release
```

Pour d'autres distributions basées sur RHEL, voir la documentation EPEL.

Voici les étapes d'installation.

```
$ sudo yum install -y postgresql-server
$ sudo postgresql-setup initdb
$ sudo systemctl enable postgresql
$ sudo systemctl start postgresql
$ sudo yum-config-manager --add-repo=https://nightly.odoo.com/8.0/nightly/rpm/odoo.repo
$ sudo yum install -y odoo
$ sudo systemctl enable odoo
$ sudo systemctl start odoo
```

Notez que

Pour imprimer des rapports PDF, vous devez installer wkhtmltopdf vous-même: la version de wkhtmltopdf disponible dans les référentiels Fedora / CentOS ne prend pas en charge les en-têtes et les pieds de page de sorte qu'elle ne puisse pas être installée automatiquement. Utilisez la version disponible sur la page de téléchargement de wkhtmltopdf. Configuration, similaire à debian, il peut être installé avec

```
wget https://bitbucket.org/wkhtmltopdf/wkhtmltopdf/downloads/{path to correct distro and
system architecture}
sudo rpm -i  {.rpm package}
sudo cp /usr/local/bin/wkhtmlto* /usr/bin/
```

Le fichier de configuration se trouve dans /etc/odoo/openerp-server.conf

Lorsque le fichier de configuration est édité, Odoo doit être redémarré via Systemd:

```
$ sudo systemctl restart odoo
```

Installation de la source

Le fichier zip d'Odoo peut être téléchargé à l' adresse https://nightly.odoo.com/8.0/nightly/src/odoo_8.0.latest.zip , le fichier zip doit alors être décompressé pour utiliser son contenu.

Git permet une mise à jour plus simple et une commutation plus facile entre les différentes versions d'Odoo. Il simplifie également la gestion des correctifs et des contributions non modulaires. L'inconvénient principal de git est qu'il est beaucoup plus grand qu'une archive car il contient toute l'histoire du projet Odoo.

Le dépôt git est `https://github.com/odoo/odoo.git`.

Ensuite, vous pouvez cloner le référentiel avec

```
$ git clone https://github.com/odoo/odoo.git
```

Installation de dépendances

L'installation source nécessite l'installation manuelle des dépendances:

Python 2.7. sous Linux et OS X, inclus par défaut

sous Windows, utilisez le programme d’installation officiel de Python 2.7.9.

si Python est déjà installé, assurez-vous qu'il est 2.7.9, les versions précédentes sont moins pratiques et les versions 3.x ne sont pas compatibles avec Odoo

configurer PostgreSQL

Après l'installation, vous devrez créer un utilisateur postgres: par défaut, le seul utilisateur est postgres, et Odoo interdit de se connecter en tant que postgres.

sous Linux, utilisez le package de votre distribution, puis créez un utilisateur postgres nommé comme votre nom d'utilisateur:

```
$ sudo su - postgres -c "createuser -s $USER"
```

Comme la connexion au rôle est la même que celle utilisée pour les connexions Unix, les sockets Unix peuvent être utilisés sans mot de passe. sous OS X, postgres.app est le moyen le plus simple de démarrer, puis de créer un utilisateur postgres sous Linux

sous Windows, utilisez PostgreSQL pour Windows puis ajoutez le répertoire bin de PostgreSQL (par défaut: C: \ Program Files \ PostgreSQL \ 9.4 \ bin) à votre PATH

créer un utilisateur postgres avec un mot de passe à l'aide de l'interface d'administration pg admin: ouvrez pgAdminIII, double-cliquez sur le serveur pour créer une connexion, sélectionnez Edition Object Nouvel objet Role Nouveau rôle de connexion, entrez le nom d'utilisateur dans le champ Nom du rôle Ouvrez l'onglet Définition et entrez le mot de passe (par exemple, odoo), puis cliquez sur OK.

L'utilisateur et le mot de passe doivent être transmis à Odoo à l'aide des options -w et -r ou du fichier de configuration.

Les dépendances Python sont répertoriées dans le fichier requirements.txt.

Sous Linux, les dépendances python peuvent être installées avec le gestionnaire de paquets du système ou avec pip.

Pour les bibliothèques utilisant du code natif (Pillow, lxml, greenlet, gevent, psycopg2, ldap), il peut être nécessaire d'installer des outils de développement et des dépendances natives avant que pip puisse installer les dépendances. Celles-ci sont disponibles dans les packages -dev ou -devel pour Python, Postgres, libxml2, libxslt, libevent, libsasl2 et libldap2. Ensuite, les dépendances Python peuvent elles-mêmes être installées:

```
$ pip install -r requirements.txt
```

Sous OS X, vous devrez installer les outils de ligne de commande (xcode-select --install), puis télécharger et installer un gestionnaire de paquets de votre choix (homebrew, macports) pour installer des dépendances autres que Python. pip peut alors être utilisé pour installer les dépendances Python comme sous Linux:

```
$ pip install -r requirements.txt
```

sous Windows, vous devez installer certaines des dépendances manuellement, modifier le fichier requirements.txt, puis exécuter pip pour installer les versions restantes.

```
Install psycopg using the installer here http://www.stickpeople.com/projects/python/win-psycopg/
```

Ensuite, éditez le fichier requirements.txt: supprimez psycopg2 comme vous l'avez déjà. Supprimez les options facultatives python-ldap, gevent et psutil car elles nécessitent une compilation. Ajoutez pypiwin32 car il est nécessaire sous Windows.

Ensuite, utilisez pip pour installer les dépendances à l'aide de la commande suivante à partir d'une invite cmd.exe (remplacez \ YourOdooPath par le chemin réel où vous avez téléchargé Odoo):

```
C:\> cd \YourOdooPath
C:\YourOdooPath> C:\Python27\Scripts\pip.exe install -r requirements.txt
```

Moins de CSS via nodejs

sous Linux, utilisez le gestionnaire de paquets de votre distribution pour installer nodejs et npm.

Notez que

Dans Debian Wheezy et Ubuntu 13.10 et avant que vous ayez besoin d'installer nodejs manuellement:

```
$ wget -qO- https://deb.nodesource.com/setup | bash -
$ apt-get install -y nodejs
```

Dans les versions suivantes de Debian (> jessie) et ubuntu (> 14.04), vous devrez peut-être ajouter un lien symbolique car les paquets npm appellent le nœud mais debian appelle le nœud binaire

```
$ apt-get install -y npm
$ sudo ln -s /usr/bin/nodejs /usr/bin/node

Once npm is installed, use it to install less and less-plugin-clean-css:

$ sudo npm install -g less less-plugin-clean-css

on OS X, install nodejs via your preferred package manager (homebrew, macports) then install less and less-plugin-clean-css:

$ sudo npm install -g less less-plugin-clean-css
```

sur Windows, installez `nodejs` , redémarrez (pour mettre à jour le PATH) et installez de `less less-plugin-clean-css` :

```
C:\> npm install -g less less-plugin-clean-css
```

Courir Odoo

Une fois toutes les dépendances configurées, Odoo peut être lancé en exécutant odoo.py.

La configuration peut être fournie via des arguments de ligne de commande ou via un fichier de configuration.

Les configurations nécessaires communes sont:

```
PostgreSQL host, port, user and password.
```

Odoo n'a pas de valeurs par défaut au-delà des valeurs par défaut de psycopg2: se connecte via un socket UNIX sur le port 5432 avec l'utilisateur actuel et sans mot de passe. Par défaut, cela devrait fonctionner sous Linux et OS X, mais cela ne fonctionnera pas sous Windows car il ne supporte pas les sockets UNIX. Chemin d'accès des addons personnalisés au-delà des valeurs par défaut, pour charger vos propres modules

Sous Windows, une façon typique d'exécuter odoo serait:

```
C:\YourOdooPath> python odoo.py -w odoo -r odoo --addons-path=addons,../mymodules --db-
filter=mydb$
```

Où odoo, odoo sont le login et le mot de passe postgresql, ../mymodules un répertoire avec des addons supplémentaires et mydb le db par défaut à servir sur localhost: 8069

Sous les systèmes * nix, une façon typique d’exécuter odoo serait:

```
$ ./odoo.py --addons-path=addons,../mymodules --db-filter=mydb$Packaged installers
```

Qu'est ce que Odoo?

Odoo (anciennement OpenERP et auparavant, TinyERP) est une suite d'applications de gestion d'entreprise à cœur ouvert. Ciblant les entreprises de toutes tailles, la suite d'applications couvre tous les besoins de l'entreprise, du site Web / commerce électronique à la fabrication, en passant par l'inventaire et la comptabilité, tous parfaitement intégrés. C'est la première fois qu'un éditeur de logiciels parvient à atteindre une couverture aussi fonctionnelle. Odoo est le logiciel d'entreprise le plus installé au monde. Odoo est utilisé par plus de 2 000 000 d'utilisateurs dans le monde, des très petites entreprises (1 utilisateur) aux très grandes entreprises (300 000 utilisateurs).

Le code source du framework OpenObject et des principaux modules ERP (Enterprise Resource Planning) est organisé par Odoo SA, basé en Belgique. De plus, une communauté mondiale active et un réseau de 500 partenaires officiels fournissent une programmation, un support et d’autres services personnalisés. Les principaux composants Odoo sont le framework OpenObject, environ 30 modules de base (également appelés modules officiels) et plus de 3000 modules de communauté.

Odoo a été utilisé comme composante des cours universitaires. Une étude sur l'apprentissage par

l'expérience a suggéré qu'OpenERP constitue une alternative appropriée aux systèmes propriétaires pour compléter l'enseignement.

Plusieurs livres ont été écrits sur Odoo, certains couvrant des domaines spécifiques tels que la comptabilité ou le développement.

Odoo a reçu des prix, notamment Trends Gazelle et BOSSIE Awards trois années de suite.

Il utilise les scripts Python et PostgreSQL comme base de données. Son édition communautaire est complétée par une édition Enterprise à USD 240 / - par utilisateur et par an et par une édition en ligne prise en charge par le commerce. Le référentiel de développement se trouve sur GitHub.

En 2013, l'association à but non lucratif Odoo Community Association a été créée pour assurer la promotion et la maintenance des versions et des modules de la communauté Odoo afin de compléter le travail d'Odoo SA. Cette organisation regroupe plus de 150 membres individuels et organisations.

Lire Démarrer avec odoo-8 en ligne: https://riptutorial.com/fr/odoo-8/topic/2151/demarrer-avec-odoo-8

Chapitre 2: Ajouter des fichiers CSS et Javascript au module Odoo

Syntaxe

- Note à propos de la syntaxe XML: Comme l'enregistrement est réalisé à l'intérieur d'un fichier XML, vous ne pouvez pas laisser de tag non fermé comme vous le feriez dans un HTML simple, comme: <link rel = 'stylesheet' href = "..." **>** , fermez le balise de lien à la place, comme:
    - <link rel = 'stylesheet' href = "..." **/** >

Paramètres

Valeurs possibles du paramètre *inherit_id*	sens
web.assets_backend	Utilisé dans les pages internes uniquement, NON inclus dans un site Web public.
website.assets_frontend	Utilisé sur un site Web public uniquement (via le module " *website* ").
Web.assets_common	Utilisé à la fois sur le site Web public et les pages internes.

Remarques

Si vous n'êtes pas sûr de l'option qui vous convient, essayez la première option (backend) car elle est utilisée dans la plupart des cas et presque dans tous les cas si vous n'avez pas installé le module "website". Odoo différencie les ressources "backend" et "frontend" car le site Web public fourni par le module "website" utilise un style et un code JS différents de ceux des pages internes destinées aux tâches ERP. "backend" est associé à des pages internes pour ERP (le sens de "frontend" et "backend" sont spécifiques à Odoo ici, mais ils sont tous deux "frontend" au sens plus général).

Vous pouvez non seulement choisir et utiliser l'une des options, mais également utiliser une combinaison de ces options (deux d'entre elles ou toutes) dans le même module. Facturer un backend, un frontend et un code JS / CSS commun dans des fichiers séparés pour mieux adhérer à DRY et disposer du code approprié sur le site Web public et dans les pages internes.

N'oubliez pas d'ajouter "web" (lorsque vous utilisez l' *option 1*) ou "website" (lorsque vous utilisez l' *option 2*) à la liste des dépendances dans le manifeste `__openerp__.py` .

Examples

Stocker les fichiers CSS et JS correctement dans le module Odoo

Les fichiers CSS et JS doivent se trouver sous le répertoire 'static' dans le répertoire racine du module (le reste de l'arborescence des sous-répertoires sous 'static' est une convention facultative):

- static / src / css / **votre_fichier.css**
- static / src / js / **votre_fichier.js**

Ajoutez ensuite des liens vers ces fichiers en supprimant l'une des trois méthodes répertoriées dans les exemples suivants.

Option 1: [BACKEND] Ajoutez des fichiers CSS et Javascript à utiliser dans les pages internes

La méthode Odoo v8.0 consiste à ajouter l'enregistrement correspondant dans le fichier XML:

- Ajoutez le fichier XML au manifeste (c.-à- `__openerp__.py` . Le fichier `__openerp__.py` .):

  ...
  'data': ['your_file.xml'],
  ...

- Ajoutez ensuite l'enregistrement suivant dans `'your_file.xml'` :

```
    <openerp>
        <data>
            <template id="assets_backend" name="your_module_name assets"
inherit_id="web.assets_backend">
                <xpath expr="." position="inside">
                    <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css"/>
                    <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
                </xpath>
            </template>
        ....
        ....
        </data>
     </openerp>
```

Option 2: [FRONTEND] Ajoutez des fichiers CSS et Javascript à utiliser sur un site Web public

Remarque: vous devez utiliser cette méthode si vous avez installé un module "site Web" et que vous avez un site Web public disponible.

- Ajoutez l'enregistrement suivant dans `'your_file.xml'` :

```
    <openerp>
        <data>

            <template id="assets_frontend" name="your_module_name assets"
inherit_id="website.assets_frontend">
                <xpath expr="link[last()]" position="after">
                    <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css"/>
                </xpath>
                <xpath expr="script[last()]" position="after">
                    <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
                </xpath>
            </template>

        </data>
     </openerp>
```

Option 3: [COMMON] Ajoutez des fichiers CSS et Javascript à utiliser dans toutes les pages (backend & frontend)

- Ajoutez l'enregistrement suivant dans `'your_file.xml'` :

```
    <openerp>
        <data>

            <template id="assets_common" name="your_module_name assets"
inherit_id="web.assets_common">
                <xpath expr="." position="inside">
                    <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css"/>
                    <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
                </xpath>
            </template>

        </data>
     </openerp>
```

Lire Ajouter des fichiers CSS et Javascript au module Odoo en ligne:
https://riptutorial.com/fr/odoo-8/topic/3401/ajouter-des-fichiers-css-et-javascript-au-module-odoo

Chapitre 3: Champs utilisés dans Odoo 8

Introduction

Ceci est la section où vous pouvez trouver les détails sur les champs qui sont utilisés dans Odoo 8

Paramètres

Paramètres	La description
string = "Nom"	Etiquette facultative du champ
compute = "_ compute_name_custom"	Transformer les champs en champs calculés
store = True	Si calculé, il stockera le résultat
select = True	Index de force sur le terrain
en lecture seule = vrai	Field sera en lecture seule dans les vues
inverse = "_ nom_écriture"	Sur déclenchement déclencheur
requis = vrai	Champ obligatoire
traduire = vrai	Permettre la traduction
help = 'blabla'	Aide texte d'info-bulle
comodel_name = "model.name"	Nom du modèle associé
inverse_name = "field_name"	colonne relationnelle du modèle opposé
relation = 'many2many_table_name'	nom de table relationnelle pour many2many
columns1 = 'left_column_name'	nom de la colonne de gauche de la table relationnelle
column2 = 'right_column_name'	nom de colonne de droite de table relationnelle

Remarques

Odoo et ORM: Odoo utilise la technique ORM (Object Relational Mapping) pour interagir avec la base de données. ORM aidera à créer une base de données d'objets virtuels pouvant être utilisée à partir du Python. Dans la technique ORM, chaque modèle est représenté par une classe qui

sous-classe Models.model.

Models.model est la super classe principale pour les modèles Odoo persistants de base de données. Les modèles Odoo sont créés en héritant de cette classe.

Exemple:

```
class Employee(Models.model):
    _name = 'module.employee'

    #Rest of the code goes here
```

Ici, _name est un attribut structurel qui indique au système le nom de la table de base de données à créer.

Chaque modèle possède un certain nombre de variables de classe, chacune représentant un champ de base de données dans le modèle. Chaque champ est représenté par une instance de la classe openerp.fields.Field. Les champs dans Odoo sont listés ci-dessous.

1 champ booléen

```
ex: flag = fields.Boolean()
```

2 champs de char

```
ex: flag = fields.Char()
```

3 texte

```
ex: flag = fields.Text()
```

4 html

```
ex: flag = fields.Html()
```

5 Entier

```
ex: flag = fields.Integer()
```

6 flotteur

```
ex: flag = fields.Float()
```

7 date

```
ex: flag = fields.Date()
```

8 date-heure

```
ex: flag = fields.Datetime()
```

9 sélection

```
ex: flag = fields.Selection()
```

10 Many2one

```
ex: flag = fields.Many2one()
```

11 One2many

```
ex: flag = fields.One2many()
```

12 Many2many

```
ex: flag = fields.Many2many()
```

Examples

Exemples de champs d'Odoo 8

Odoo utilise la technique ORM (Object Relational Mapping) pour interagir avec la base de données. ORM aidera à créer une base de données d'objets virtuels pouvant être utilisée à partir du Python. Dans la technique ORM, chaque modèle est représenté par une classe qui sous-classe Models.model. Models.model est la super classe principale pour les modèles Odoo persistants de base de données. Les modèles Odoo sont créés en héritant de cette classe

```
name = fields.Char(string='New Value')

flag = fields.Boolean(string='Flag',default=False)

amount = fields.Float(string='Amount',digits=(32, 32))

code = fields.Selection(string='Code',selection=[('a', 'A'),('b','B')])

customer = fields.Many2one(comodel_name='res.users')

sale_order_line = fields.One2many(comodel_name='res.users', inverse_name='rel_id')

tags = fields.Many2many(comodel_name='res.users',
                        relation='table_name',
                        column1='col_name',
                        column2='other_col_name')
```

Lire Champs utilisés dans Odoo 8 en ligne: https://riptutorial.com/fr/odoo-8/topic/8152/champs-utilises-dans-odoo-8

Chapitre 4: Comment activer le mode développeur OpenERP

Remarques

Mode développeur

Le mode développeur Odoo vous permet d'apporter des modifications substantielles à la base de données Odoo, telles que l'ajout de champs à vos documents et vues. Vous modifiez les vues par défaut de vos actions et pouvez même créer des formulaires dynamiques basés sur d'autres champs dans vos modèles.

Avantage

Bien qu'Odoo soit un puissant cadre d'application, le cycle de développement peut être brutal pour tester les modifications apportées à votre application. En utilisant le mode développeur, vous pouvez tester des expressions et résoudre de nombreux problèmes fonctionnels sans avoir à redémarrer le serveur encore et encore pour tester des modifications simples.

De plus, l'outil de développement Odoo est idéal pour examiner l'architecture des formulaires et des vues afin de voir comment les champs sont liés aux modules, à leurs domaines, contextes et autres attributs. Dans cette vidéo, nous explorons exactement comment utiliser ces outils pour modifier et créer des applications Odoo.

Limites

Bien qu'il soit très tentant d'utiliser le mode développeur pour apporter beaucoup de modifications à votre application, il existe des inconvénients. Selon ce que vous modifiez et modifiez, vous pouvez perdre ces modifications avec les futures mises à jour des modules ou lorsque vous installez des applications supplémentaires dans Odoo. Cela est particulièrement vrai pour les changements de vues.

> Pour activer le mode développeur, il suffit de noter
>
> pour la version v7
>
> & debug =
>
> avant # signe, il suffit de l'ajouter.
>
> http: // localhost: 8069 /? db = test_db & debug = #
>
> pour la version> v7
>
> http: // localhost: 8069 / web? debug =

Vous ne verrez peut-être pas le menu **À propos d'Odoo** car il pourrait y avoir **un module de**

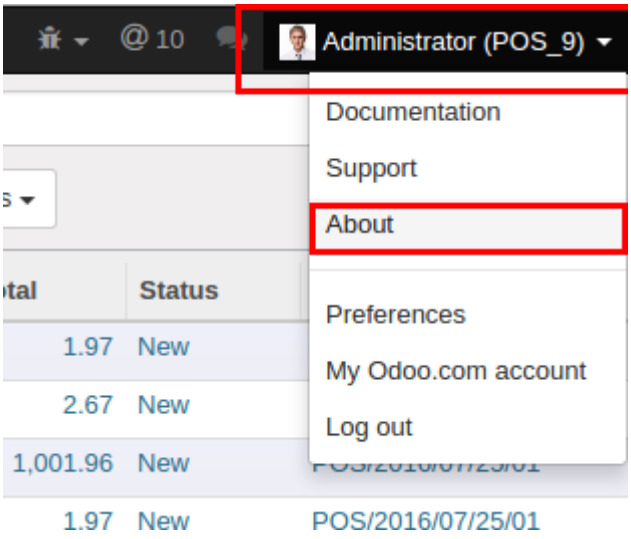
débridage Odoo installé.

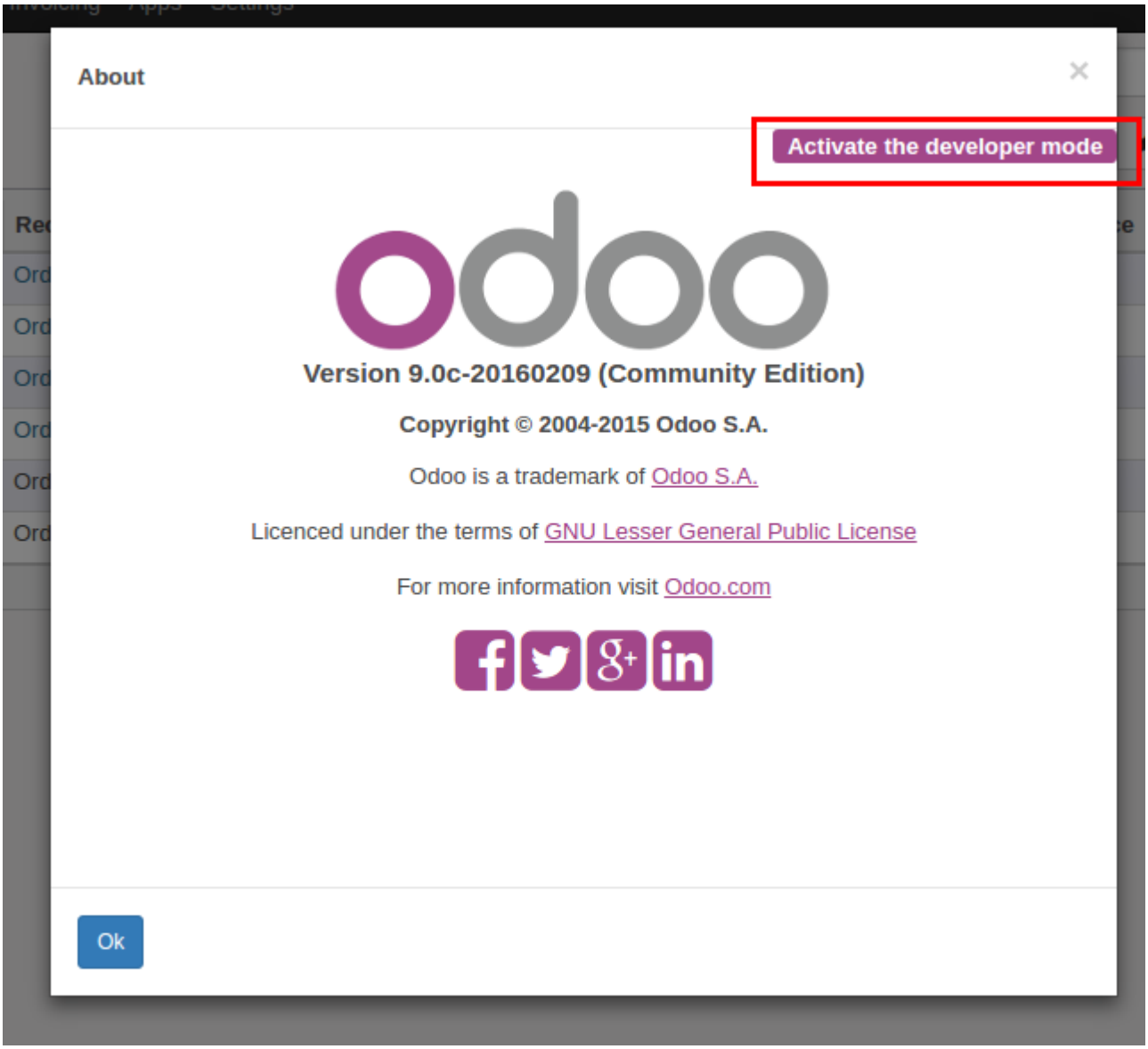
Examples

Activer le mode développeur

Pour activer le mode développeur:

1. Connectez-vous au front end ODOO
2. Cliquez sur le nom d'utilisateur dans la partie supérieure droite
3. Sélectionnez 'À propos de'
4. Cliquez sur «Activer le mode développeur» dans la fenêtre contextuelle.





Activation du mode développeur dans Odoo 8

Lorsque vous vous connectez à l'application Odoo, vous trouverez une option pour voir qui est la personne actuellement connectée dans le coin supérieur droit. Ces informations utilisateur ont un bouton déroulant. Cliquez sur le menu déroulant, vous trouverez alors une liste. Dans cette liste, sélectionnez l'option Odoo.com. En cliquant dessus, vous ouvrez une fenêtre contextuelle **À propos de** . Dans cette fenêtre, en haut à droite, vous trouverez une option comme le **mode développeur Acivate** . En cliquant sur ce lien, vous rechargez la page Web.

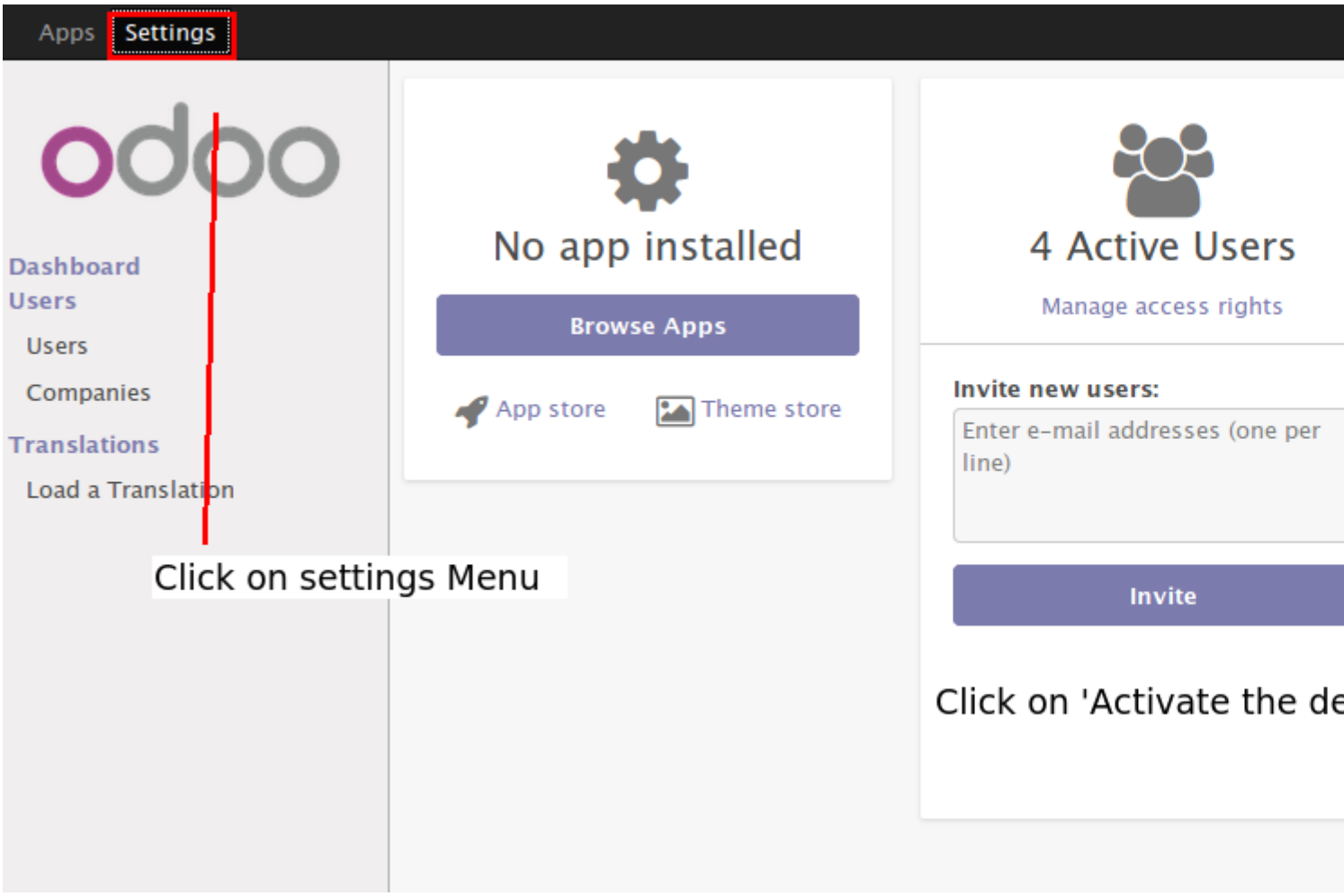
Après le rechargement, il sera en mode développeur. Ensuite, le lien changera en quelque chose comme *http: // localhost: 8069 / web? Debug = # id = 23 & view_type = form & model = res.partner*

Activer le mode développeur dans Odoo 10

Activer le mode développeur:

1. Connectez-vous à l'application odoo.
2. Après la connexion, l'utilisateur peut voir plusieurs menus odoo. Cliquez sur le menu de

réglage.



1. Cliquez sur "Activer le mode développeur" situé dans le coin inférieur droit de la page des paramètres.
2. Le mode développeur est maintenant activé.

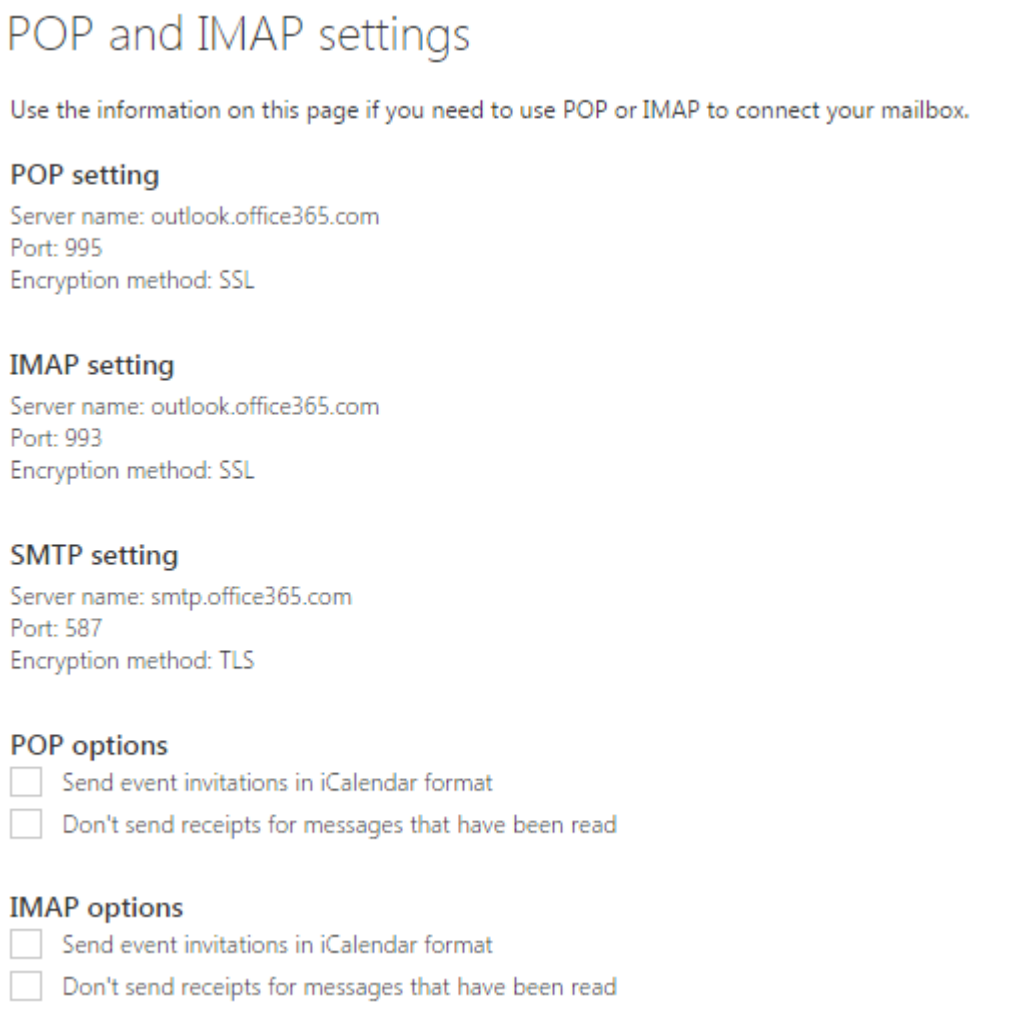
Lire Comment activer le mode développeur OpenERP en ligne: https://riptutorial.com/fr/odoo-8/topic/3311/comment-activer-le-mode-developpeur-openerp

Chapitre 5: Configurer le courrier électronique - Office 365 dans Odoo

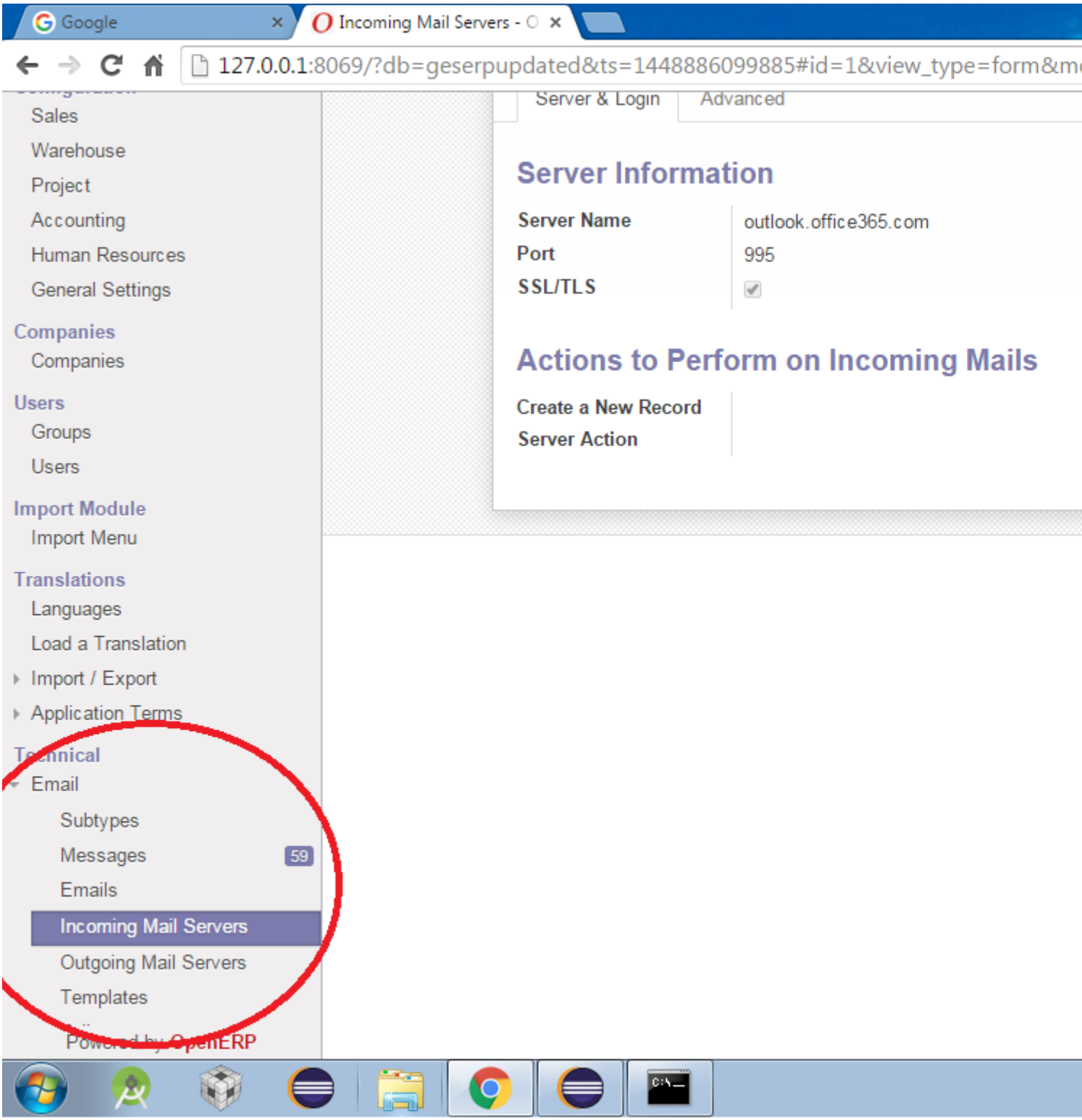
Examples

Configurer le courrier électronique

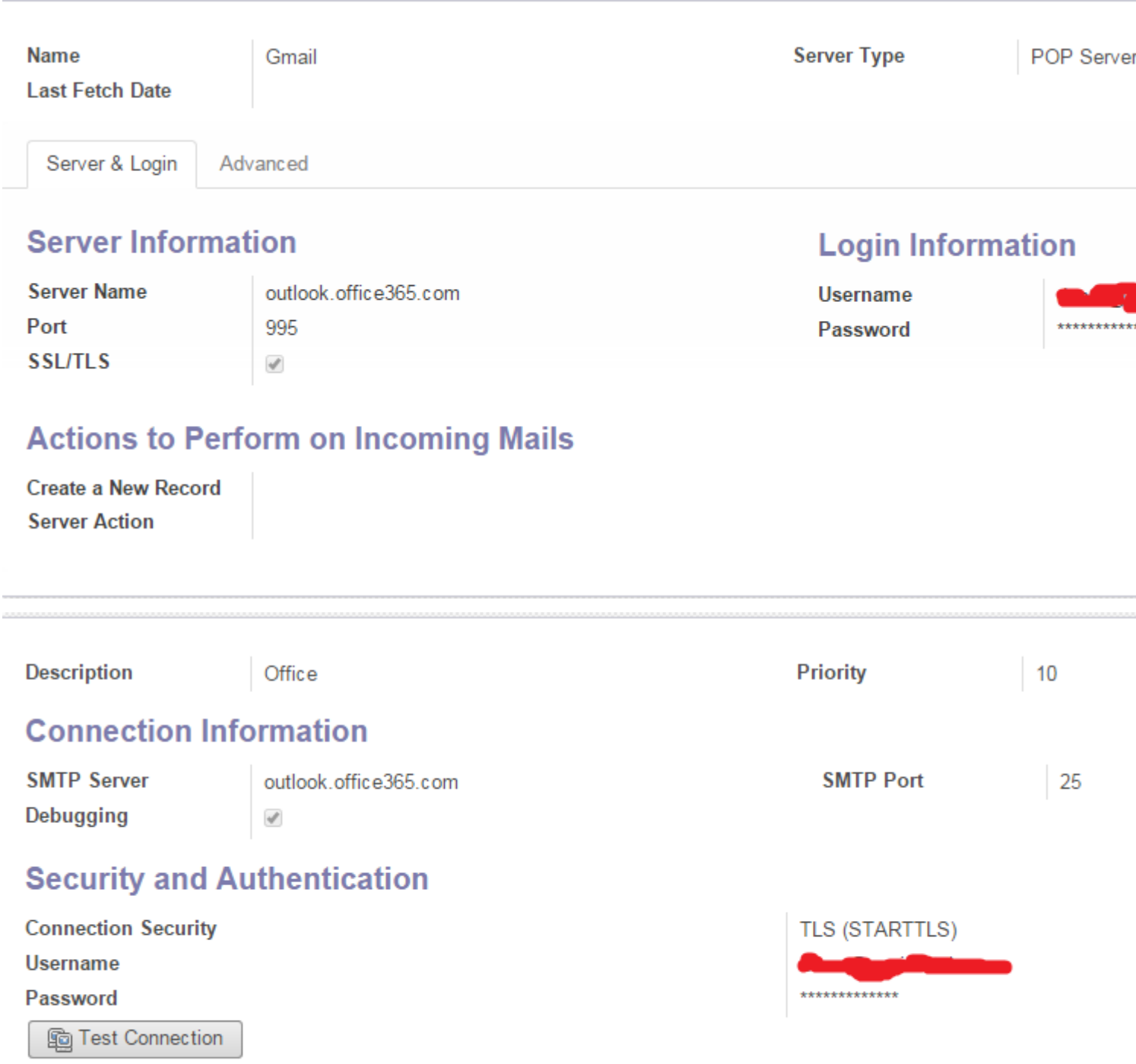
- Vérifiez initialement vos paramètres de messagerie



- **Dans Odoo, allez dans Paramètres -> Email.**



- **Entrez les valeurs de champ dans les options "Serveurs de messagerie entrants" et "Serveurs de messagerie sortants".**



Lire Configurer le courrier électronique - Office 365 dans Odoo en ligne: https://riptutorial.com/fr/odoo-8/topic/6648/configurer-le-courrier-electronique---office-365-dans-odoo

Chapitre 6: Créer des fonctions automatisées pour le modèle

Introduction

Nous avons souvent besoin d'exécuter du code automatiquement pendant l'installation du module. Cela a de nombreuses raisons, par exemple, configurer les paramètres du module `Sale` pour répondre aux exigences de notre projet.

Dans cette rubrique, vous apprendrez comment faire fonctionner automatiquement les fonctions sur l'installation des modules.

Examples

Tout d'abord, vous devez créer un fichier XML pour l'appel de la fonction make

```
<?xml version="1.0"?>
<openerp>
    <data noupdate="1">
        <function model="*model_name*" name="_configure_sales"/>
    </data>
</openerp>
```

Ce fichier xml simple `_configure_sales` fonction *_configure_sales du* modèle *model_name* .

REMARQUE: ce fichier xml doit figurer en haut du tableau de `data` , car Odoo traite les fichiers XML de haut en bas.

Fichier Python correspondant

```
class *model_name*(models.Model):
    _name = *model_name*

    @api.model
    def _configure_sales(self):
        # Do the configuration here
```

Chaque fois que le module sera installé, cette fonction s'exécutera.

Remarque: Si vous supprimez `noupdate` de xml, la fonction s'exécutera également lors de la mise à niveau.

Lire Créer des fonctions automatisées pour le modèle en ligne: https://riptutorial.com/fr/odoo-8/topic/10633/creer-des-fonctions-automatisees-pour-le-modele

Chapitre 7: Quelles sont les méthodes et les détails de l'ORM?

Remarques

Méthode de création: Créer un nouvel enregistrement avec la valeur spécifiée. Prend un certain nombre de valeurs de champs et renvoie un jeu d'enregistrements contenant l'enregistrement créé

```
def create(self,vals):
    return super(class_name, self).create(vals)
```

Méthode d'écriture: met à jour les enregistrements avec des identifiants donnés avec les valeurs de champ données. Prend un certain nombre de valeurs de champ, les écrit dans tous les enregistrements de son jeu d'enregistrements. Ne retourne rien

```
def write(self,vals):
    return super(class_name, self).write(vals)
```

Méthode de recherche: recherche des enregistrements en fonction d'un domaine de recherche. Prend un domaine de recherche et renvoie un jeu d'enregistrements correspondants. Peut renvoyer un sous-ensemble d'enregistrements correspondants (paramètres de décalage et de limite) et être commandé (paramètre de commande)

```
self.search([('customer','=',True)])
self.env['res.partner'].search(['partner','=',True])
```

Méthode Browse: récupère les enregistrements en tant qu'objets permettant d'utiliser la notation par points pour parcourir les champs et les relations. Prend un identifiant de base de données ou une liste d'identifiants et renvoie un jeu utile lorsque les identifiants sont obtenus en dehors d'Odoo ou lors de l'appel de méthodes dans l'ancienne API.

```
self.browse([7,8,9])
self.env['res.partner'].browse([7,8,9])
```

Méthodes Exists: Renvoie un nouveau jeu d'enregistrements contenant uniquement les enregistrements existant dans la base de données. Peut être utilisé pour vérifier si un enregistrement (par exemple obtenu en externe) existe toujours.

```
records = records.exists()
```

Méthode ref: méthode d' environnement renvoyant l'enregistrement correspondant à un identifiant externe fourni

```
self.env.ref('base.group_public')
```

Méthode Ensure_one: vérifie que le jeu d'enregistrements est un singleton (ne contient qu'un seul enregistrement), déclenche une erreur sinon

```
records.ensure_one()
```

Examples

Différents types de méthodes ORM

1. créer()
2. écrire()
3. chercher()
4. Feuilleter()
5. existe ()
6. ref ()
7. sure_one ()

Lire Quelles sont les méthodes et les détails de l'ORM? en ligne: https://riptutorial.com/fr/odoo-8/topic/6150/quelles-sont-les-methodes-et-les-details-de-l-orm-

Chapitre 8: RPC utilisant l'API Odoo v8 (fonction Call Python à partir de JavaScript)

Remarques

Si vous envisagez d'ajouter de nouvelles méthodes en Python pour les utiliser dans RPC à partir de JavaScript, envisagez les options suivantes pour les décorateurs de méthodes: si vous devez gérer les identifiants / jeux d'enregistrements, puis la définition de méthode python, choisissez décorateur:

- @ api.multi - pour obtenir un jeu d' *enregistrements* dans votre méthode
- @ api.one - pour obtenir *browse_records* un à un dans votre méthode in example ci-dessus examples @ api.multi est utilisé, mais @ api.one peut également être utilisé pour traiter les identifiants, selon les besoins (il est cependant fortement recommandé d'utiliser @ api.multi au lieu de @ api.one pour des raisons de performances).

Ou si c'est une fonction simple qui n'a pas à gérer les enregistrements / identifiants, alors pour la méthode python, choisissez décorateur:

- @ api.model - Permet d'être poli avec l'ancienne API.
- @ api.multi - Encore une fois, vous pouvez aussi l'utiliser ici, passez simplement `[ ]` (tableau vide) en premier argument dans javascript ...

Références: documentation Odoo RPC , décorateurs de méthodes Odoo 8 API

Examples

Un exemple de modèle Odoo pour appeler des méthodes depuis

```
class my_model(models.Model):
    _name = "my.model"

    name = fields.Char('Name')

    @api.multi
    def foo_manipulate_records_1(self):
        """ function returns list of tuples (id,name) """
        return [(i.id,i.name) for i in self]

    @api.multi
    def foo_manipulate_records_2(self, arg1, arg2)
        #here you can take advantage of "self" recordset and same time use aditional arguments
"arg1", "arg2"
        pass

    @api.model
    def bar_no_deal_with_ids(self, arg1, arg2):
        """ concatenate arg1 and arg2 """
```

```
        return unicode(arg1) + unicode(arg2)
```

Odoo RPC exemples

Les exemples ci-dessous montrent comment appeler la fonction Python depuis JavaScript dans Odoo 8. Dans les exemples, nous appelons les méthodes de *my_model* décrites plus haut sur cette page.
Nous supposons que dans les exemples suivants, la variable "list_of_ids" contient la liste (tableau) des identifiants des enregistrements existants du modèle "my.model".

- Appel de la méthode **foo_manipulate_records_1** décoré avec @ **api.multi** :

```
new instance.web.Model("my.model")
    .call( "foo_manipulate_records_1", [list_of_ids])
        .then(function (result) {
            // do something with result
    });
```

- Appel de la méthode **foo_manipulate_records_2** décoré avec @ **api.multi** :

```
new instance.web.Model("my.model")
    .call( "foo_manipulate_records_2", [list_of_ids, arg1, arg2])
        .then(function (result) {
            // do something with result
    });
```

- Appel de la méthode **bar_no_deal_with_ids** décoré avec @ **api.model** :

```
new instance.web.Model("my.model")
    .call( "bar_no_deal_with_ids", [arg1, arg2])
        .then(function (result) {
            // do something with result
    });
```

Aussi, si cela a du sens selon l'implémentation, alors vous pouvez appeler la fonction décorée avec @ api.multi même si vous n'avez pas à gérer les identifiants (passez simplement un tableau vide à la place des identifiants, comme premier élément de la liste d'arguments):

```
new instance.web.Model("my.model")
    .call( "foo_manipulate_records_2", [[], arg1, arg2])
        .then(function (result) {
            // do something with result
    });
```

Cette façon de faire peut être utile dans certains cas, car la fonction non décorée dans la version 8.0 api est considérée comme @ api.multi (comme @ api.multi est un décorateur par défaut)

Si vous utilisez deux paramètres pour l'appel RPC utilisés dans les exemples ci-dessus (le nom de la fonction et la liste d'arguments), vous pouvez utiliser le **troisième paramètre** - un

dictionnaire d'arguments de mots clés . Il est fortement recommandé de contourner un contexte (dans certains cas, cela peut même être nécessaire), car cela peut changer le comportement de la procédure distante (localisation, etc.). Voir ci-dessous l'exemple avec un argument de contexte dans l'appel RPC (même chose pour tous les exemples ci-dessus)

```
var self = this;
new instance.web.Model("my.model")
    .call("foo_manipulate_records_2", [[], arg1, arg2], {'context':self.session.user_context})
        .then(function (result) {
            // do something with result
    });
```

Bien sûr, vous pouvez également utiliser un contexte personnalisé, si nécessaire, plutôt que de contourner l'existant comme dans cet exemple.

Lire RPC utilisant l'API Odoo v8 (fonction Call Python à partir de JavaScript) en ligne: https://riptutorial.com/fr/odoo-8/topic/6613/rpc-utilisant-l-api-odoo-v8--fonction-call-python-a-partir-de-javascript-

Chapitre 9: Widgets personnalisés pour les champs

Remarques

- assurez-vous d' ajouter correctement le fichier javascript à votre module
- N'oubliez pas d'ajouter 'web' comme dépendance dans __openerp__.py:

```
'depends': ['web',....]
```

Examples

Widget personnalisé pour les champs numériques à utiliser dans TreeView

Le widget d'exemple ci-dessous montre comment mettre en forme certaines cellules d'une colonne TreeView de manière conditionnelle, en fonction de la valeur du champ dans la cellule particulière. Si la valeur du champ est négative, elle sera affichée en rouge et le symbole moins sera masqué, sinon elle sera affichée en couleur normale.

Un widget doit être écrit en JavaScript, utilisons `custom_widget_name` comme nom pour un nouveau widget et `your_module_name` est un nom technique de votre module (identique au nom du répertoire racine de votre module)

Installez le fichier javascript statique (par exemple static / src / js / *custom_widget* .js) avec un widget personnalisé:

```
 openerp.your_module_name = function (instance) {

    instance.web.list.columns.add('field.custom_widget_name',
'instance.your_module_name.custom_widget_name');

    instance.your_module_name.custom_widget_name = instance.web.list.Column.extend({
        _format: function (row_data, options) {
            res = this._super.apply(this, arguments);
            var amount = parseFloat(res);
            if (amount < 0){
                return "<font color='#ff0000'>"+(-amount)+"</font>";
            }
            return res
        }
    });
    //
    //here you can add more widgets if you need, as above...
    //
};
```

L'exemple de widget ci-dessus peut être utilisé dans une vue de liste pour le champ de type float et il applique des règles personnalisées comme suit:

- Nombres négatifs:
  - Sont affichés en rouge.
  - Le symbole moins (un caractère '-') est "caché".
- Pour les nombres positifs, la disposition par défaut est utilisée.

Cet exemple de widget peut être appliqué à un champ dans une vue arborescente d'Odoo. Vous pouvez utiliser un widget comme celui-ci pour une colonne dont vous avez besoin d'appliquer les règles personnalisées à:

```
. . .
<tree >
    . . .
    <field name="some_field" widget="my_widget" />
    . . .
</tree>
. . .
```

Lire Widgets personnalisés pour les champs en ligne: https://riptutorial.com/fr/odoo-8/topic/6198/widgets-personnalises-pour-les-champs

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec odoo-8	4444, Community, danidee, T.V.
2	Ajouter des fichiers CSS et Javascript au module Odoo	George Daramouskas, T.V.
3	Champs utilisés dans Odoo 8	AKHIL MATHEW
4	Comment activer le mode développeur OpenERP	AKHIL MATHEW, Emipro Technologies Pvt. Ltd., Gopakumar N G, Mehedi Hasan
5	Configurer le courrier électronique - Office 365 dans Odoo	Don Chakkappan
6	Créer des fonctions automatisées pour le modèle	Dachi Darchiashvili
7	Quelles sont les méthodes et les détails de l'ORM?	AKHIL MATHEW, Mani
8	RPC utilisant l'API Odoo v8 (fonction Call Python à partir de JavaScript)	T.V.
9	Widgets personnalisés pour les champs	T.V.