

Projet

Gestion de librairies : esi-lecture

Consignes

- $\,\rhd\,$ Ce projet peut être réalisé individuellement ou par équipe de 2
- ▷ Chaque équipe possède un dépôt sur le serveur git de l'ESI créé par l'enseignant
- ▶ La date de la remise finale est affichée sur PoEsi.

Première partie : Le module esi_lecture

Le but du projet est de développer un module de vente de livres. La première partie du projet consiste en la création du module et en l'implémentation des fonctionnalités suivantes :

- ▷ consulter/ajouter/supprimer/modifier la liste des livres
- > consulter les auteurs des livres et les classer par nombre de livres enregistrés
- ▷ permettre de liker un livre

1.1 Premier Business Object : Un livre

Créez le Business Object central de votre module : le livre.

Description d'un livre

- $\, \triangleright \, \, \mathrm{un \,\, nom \,\, unique} \,$
- ▷ une description contenant le résumé du livre, la quatrième de couverture,...
- \triangleright une image de la couverture
- ⊳ une date de publication, cette date est antérieure à la date du jour
- ▷ le nombre de pages du livre (strictement supérieur à 0)
- ▷ une liste des auteurs

Les auteurs sont des partenaires au sein du module, ils ne peuvent pas se connecter au module pour l'instant.

Les vues associées aux livres

Créez des menus permettant d'accéder à la vue liste et à la vue formulaire des livres. Ensuite modifiez les vues génériques fournies par ODOO :



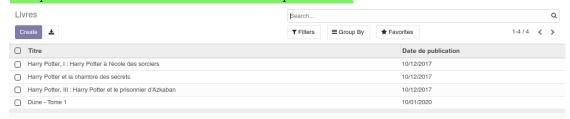
vue formulaire

doit regrouper les éléments pour améliorer l'affichage



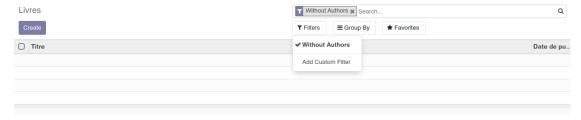
vue liste

doit présenter les colonnes nom et date de publication



filtre de recherche

doit permettre de filtrer sur les livres sans auteurs

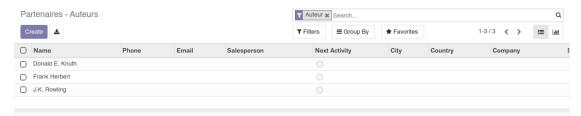


1.3

1.3 Les vues associées aux auteurs

Créez des menus permettant d'accéder à la vue liste et à la vue formulaire des auteurs.

Lorsque vous accédez à la vue liste, modifiez le contexte de l'action pour qu'un filtre soit automatiquement activé : seul les auteurs doivent apparaître.



1.4

Les données chargées

Lors de l'installation de votre module, nous vous demandons de remplir les données relatives à plusieurs livres et plusieurs auteurs ¹.

1. Si vous manquez d'inspiration, les données utilisées dans ce document sont disponibles sur PoEsi.

1.5 Gestion des likes

Un utilisateur doit pouvoir ajouter/enlever un like sur un livre. Une indication doit permettre à l'utilisateur de savoir si il a déjà aimé le livre affiché. Par exemple votre interface peut s'inspirer de la notion de Smart Button:



1.6 Utilisation de service web

Afin de vous familiarisez avec l'utilisation de XML-RPC, écrivez un script python qui :

- > se connecte à votre instance d'ODOO via XML-RPC;
- > tant que l'utilisateur le souhaite recherche un livre sur base de son nom (titre).

Les détails de l'implémentation de ces appels sont disponibles dans la documentation ². Un exemple de ces appels concernant le module OpenAcademy est également disponible dans les slides du cours théorique (slide 26 de la séance 5).

Pensez à afficher des messages d'erreurs pertinents en cas d'erreur.

2 Seconde partie : Intégration aux modules existants

Vous allez transformer votre module pour lui permettre de gérer les stocks et les ventes de livre. Pour ce faire vous aurez besoin de plusieurs modules existants.

Modules à intégrer

- > Point de vente : permet de gérer les ventes réalisées dans un magasin
- > Stock : permet de gérer les stocks et les approvisionnements

Commencez par ajouter à votre module la dépendance au module point_of_sale et mettez à jour votre module. Vérifiez les dépendances du module point_of_sale pour connaître les différents modules réellement installés lors de la mise à jour.

^{2.} https://www.odoo.com/documentation/14.0/fr/developer/api/external_api.html

2.1 La notion de produit

Le module **product** est automatiquement installé via l'installation de la gestion des stocks. Ce module permet de gérer les produits.

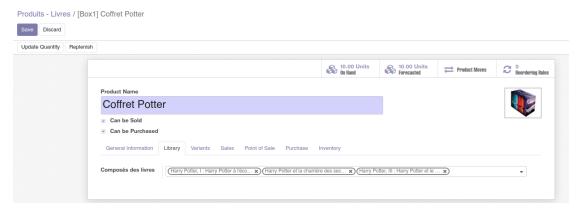
Chaque produit est une instance de product.template. Vous pouvez en consulter le code via odoo-src/odoo-14.0/addons/product/models/product_template.py

Dans le module esi-lecture nous souhaitons vendre des livres de deux façons :

- □ une vente simple : un livre est vendu à la pièce ;
- > une vente en collection : plusieurs livres sont réunis pour former une collection.
 Exemple : les 3 premiers Harry Potter.

Pour créer un produit à vendre qui correspond à ce besoin, étendez le modèle product.template et liez une instance de product.template à une liste de livres.

N'oubliez pas de créer au sein de votre module des menus permettant d'accéder à la vue liste et à la vue formulaire de product.template et étendez la vue formulaire de product.template pour permettre d'ajouter des livres.



Créez ensuite un premier produit en lui associant un ou plusieurs livres.



2.2 Gestion du stock

Pour l'instant le stock de livres est vide, il faut passer des commandes pour l'alimenter. Créez un nouveau partenaire intitulé : Presses Universitaires de Bruxelles³. Ce partenaire va représenter notre fournisseur de livres.

Alimentation du stock de livre

- ▷ Consultez un produit et notez la quantité disponible en stock
- > Consultez la liste des livraisons et créez une livraison pour votre produit
- > Choisissez un fournisseur parmi les partenaires disponibles
- > Validez la livraison
- ▷ Consultez le produit et notez la quantité disponible en stock
- 3. N'hésitez pas à l'ajouter à vos données de chargement



Alimentation initiale du stock

Dans le module stock, consultez dans le dossier data le fichier stock_demo.xml alimentant la base de données du module. Prenez attention aux éléments suivants :

stock.inventory

stock.inventory.line

_action_start

```
continuous contin
```

action_validate

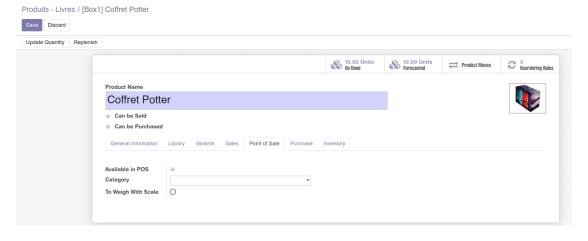
```
cfunction model="stock.inventory" name="action_validate">
cfunction eval="[[('state','=','confirm'),('id', '=',
ref('stock_inventory_0'))]]" model="stock.inventory" name="search"/>
c/function>
```

En s'inspirant du fichier stock_demo.xml essayez de mettre à jour les données de chargement de votre module pour que le stock de livres soit rempli après l'installation du module.



2.4 Point de vente

Pour que votre produit soit disponible dans les points de vente, vous devez l'activer via l'onglet Point de vente de la vue formulaire des produits.



Modifiez vos données de chargement pour que vos produits soient disponibles dans les points de ventes. En effet il manque un champs qui correspond au booléen Avalaible in POS du formulaire des produits. Consultez l'extension du modèle product.template du module point de vente pour voir si un des champs de cette extension correspond.

Vente d'un livre

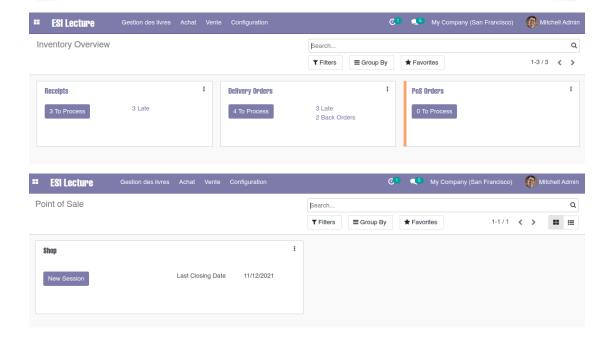
- > ouvrez une session de point de vente
- > sélectionnez un produit
- > sélectionnez un client
- réglez la vente de ce produit
- > fermez la session
- vérifiez que le stock de votre produit à diminué

2.5 Mise à jour des menus

Afin qu'un utilisateur puisse accéder à l'ensemble des opérations du livres à partir de votre module, ajoutez à vos menus :

- un lien vers l'action stock_picking_type_action du module stock
- > un lien vers l'action action_pos_config_kanban du module point de vente

Pouvez-vous trouver ce que réalise ces actions en analysant le code des deux modules point de vente et stock?



3 Troisième partie : Interface Django

Créez le projet Django esi-lecture-client qui va permettre aux utilisateurs d'interagir avec le module ODOO esi-lecture via une interface indépendante d'ODOO.

Ce projet est constitué de deux applications :

- > une application de configuration de la connexion ODOO
- > une application de notations des livres

3.1 Application de configuration de la connexion ODOO

Cette application permet:

- d'enregistrer les paramètres nécessaires à un appel à l'API ODOO (l'url, le nom d'utilisateur,...);
- de tester si la connexion fonctionne avec les paramètres enregistrés.

3.2 Application de recherche de livres

Cette application permet:

- > de rechercher un livre en se basant sur son titre via le module esi-lecture d'ODOO;
- d'ajouter un like sur un livre.

Lors d'une recherche les résultats sont triés par nombre de like, puis par nom (titre).

4 Défense du projet

L'examen consiste en une présentation orale individuelle de votre projet.

Pour cette présentation vous devrez préparer un support qui permet :

- ▷ d'expliquer la répartition des tâches utilisée;
- ▷ d'expliquer l'architecture du projet;
- > de mettre en évidence les points les plus importants de votre code.

La présentation de votre support doit durer 15 minutes et sera suivie de 15 minutes de réponses aux questions de votre enseignant.