Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2022 – 2023
ERPG5
JLC - SDR

ERPG5 – Laboratoire de Développement collaboratif

Projet

Gestion immobilière : le module realtor

Consignes

- ▷ 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 remise finale est attendue sur le dépôt pour le **vendredi 23 décembre à midi**.

1 Description des besoins

Une société de vente d'appartements standardisés souhaite proposer une application de vente en ligne pour ses nouvelles constructions. Ces immeubles sont divisés en plusieurs appartements identiques mis en vente de façon indépendante. Cette société souhaite :

- ▷ encoder ces appartements dans un stock ;
- ▷ proposer la liste des appartements disponibles à des acheteurs potentiels ;
- ▷ permettre à ces acheteurs de faire une offre d'achat pour un appartement.

Vous devez développer une solution logiciel pour répondre à ces besoins. La gestion du stock sera gérée par Odoo. L'interaction des acheteurs avec le système se fera via une interface développée via Django. La connexion entre ces deux applications sera effectuée via des appels à l' API d'Odoo[1] .

2 Première partie : Le module realtor

Vous devez créer au sein d'Odoo le module `realtor`. Ce module permet de consulter/ajouter/supprimer/modifier des appartements.

2.1 Premier Business Object : Un appartement

Créez le Business Object central de votre module : `l'appartement`.

1. https://www.odoo.com/documentation/14.0/fr/developer/api/external_api.html

Version du document
21 novembre 2022

Description d'un appartement

- un nom **unique**
- une description textuelle de l'appartement
- une photo de l'appartement
- une date de disponibilité, **minimum 3 mois après la création** de l'appartement dans le module
- le prix attendu (**strictement supérieur à 0**)
- la surface de l'appartement (**strictement supérieure à 0**)
- la surface de la terrasse (**supérieure ou égale à 0**)
- la surface totale, calculée à partir des surfaces précédentes
- l'acheteur ayant fait la meilleure offre
- le montant de la meilleure offre, ce montant doit au minimum être de 90% du prix attendu

Les acheteurs sont des partenaires au sein du module, ils ne peuvent pas se connecter à Odoo.

2.2 Les vues associées aux appartements

Créez des menus permettant d'accéder à la vue liste et à la vue formulaire des appartements. Conservez les vues génériques fournies par Odoo pour la suite du développement.

2.3 Les données chargées

Lors de l'installation de votre module, nous vous demandons de remplir les données relatives à plusieurs appartements, vendeurs et acheteurs.

Lors de l'installation du module, des appartements doivent être créés, donc, il faut faire un fichier de seeders.
Le fichier de seeder est en fait le demo.xml

2.4 Utilisation de service web

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

- demande à l'utilisateur son login et son mot de passe (ou sa clé API)
- se connecte à votre instance d'ODOO via XML-RPC
- tant que l'utilisateur le souhaite recherche un appartement sur base de son nom

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. https://www.odoo.com/documentation/14.0/fr/developer/api/external_api.html

3 Seconde partie : Intégration aux modules existants

Vous allez transformer votre module pour lui permettre de gérer les stocks.

Ajouter à votre module la dépendance au module `Stock` et mettez à jour votre module `realtor`. Vérifiez les dépendances du module `Stock` pour connaître les différents modules réellement installés lors de la mise à jour.

3.1 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

https://github.com/odoo/odoo/blob/14.0/addons/product/models/product_template.py

Pour créer un produit à vendre, créez une extension du modèle `product.template` dans le module `realtor` et liez une instance de `product.template` à une instance d'appartement. Testez votre implémentation en créant un premier produit et en lui associant un appartement.

3.2 Gestion du stock

Pour l'instant le stock d'appartements est vide, il faut passer des commandes pour l'alimenter. Créez un nouveau partenaire intitulé : `Immobilier ESI`[3]. Ce partenaire va représenter notre "fournisseur"[4] d'appartements. Alimenter le stock avec quelques appartements afin de vous familiariser avec ce module.

Alimentation du stock d'appartements

- ▷ 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

3.3 Alimentation initiale du stock

Dans le module `stock`, consultez dans le dossier `data` le fichier[5] `stock_demo.xml` alimentant la base de données du module. Prenez attention aux éléments suivants :

3. N'hésitez pas à l'ajouter à vos données de chargement
4. constructeur
5. https://github.com/odoo/odoo/blob/14.0/addons/stock/data/stock_demo.xml

stock.inventory

```xml
<record id="stock_inventory_0" model="stock.inventory">
  <field name="name">Starting Inventory</field>
</record>
```

stock.inventory.line

```xml
<record id="stock_inventory_line_1" model="stock.inventory.line">
  <field name="product_id" ref="product.product_product_24"/>
  <field name="product_uom_id" ref="uom.product_uom_unit"/>
  <field name="inventory_id" ref="stock_inventory_0"/>
  <field name="product_qty">16.0</field>
  <field name="location_id" model="stock.location"
↪ eval="obj().env.ref('stock.warehouse0').lot_stock_id.id"/>
</record>
```

4 morceaux de code sur lesquels il ne faut que faire attention. Ce ne sont pas l'entièreté du code

_action_start

```xml
<function model="stock.inventory" name="_action_start">
  <function eval="[[('state','=','draft'),('id', '=',
↪ ref('stock_inventory_0'))]]" model="stock.inventory" name="search"/>
</function>
```

action_validate

```xml
<function model="stock.inventory" name="action_validate">
  <function eval="[[('state','=','confirm'),('id', '=',
↪ ref('stock_inventory_0'))]]" model="stock.inventory" name="search"/>
</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 d'appartements soit rempli après l'installation du module.

3.4 Utilisation de service web

Suite à vos derniers développements, une instance d'appartement est liée à une instance de produit. Afin de vérifier que vous savez effectuer cette recherche via l'API, mettez à jour votre script python pour rechercher un produit sur base de l'identifiant de l'appartement. Lorsque l'utilisateur recherchera un appartement via le script, il verra les informations de l'appartement et la quantité d'appartements disponibles. Cette quantité provenant du Business Object produit.

Pourquoi rechercher un appartement via les produits ?
Si nous recherchons un appartement, on peut model.execute sur le model realtor.apartment et ne pas s'occuper du modele product

4 Troisième partie : Interface Django

Créez le projet Django `realtor-client` qui va permettre de fournir aux acheteurs une interface pour interagir avec le module Odoo `realtor`.

Ce projet est constitué de deux applications :

- une application de configuration de la connexion Odoo
- une application d'achat d'appartements

4.1 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.

4.2 Application d'achat d'appartements

Cette application permet :

- ▷ de consulter la liste des appartements mis en vente via le module `realtor` d'Odoo ;
- ▷ pour chaque appartement de connaître la quantité disponible dans le stock ;
- ▷ de proposer une offre d'achat.

Proposer une offre d'achat revient à proposer un prix supérieur au montant actuel de la meilleure offre. Si cette offre est supérieure à la meilleure offre enregistrée, il faut mettre à jour les données du module odoo `realtor` via l'API. Cette mise à jour doit contenir le montant de la nouvelle offre et l'acheteur qui a effectué cette offre. Si cet acheteur n'existe pas, il faut le créer via l'API odoo.

5 Défense du projet

Il faut donc faire un PowerPoint

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.