

# Création d'un module Odoo

M. AIT DAOUID

*BOUTHAYNA BEKKOUCH*

Compte rendu

# Introduction

Ce travail pratique a pour objectif de concevoir et développer un module Odoo permettant la **gestion des clubs étudiants**. Le module a été réalisé dans le cadre d'un apprentissage de la plateforme Odoo, en respectant l'architecture standard d'un module (models, views, security, manifest).

L'application développée permet de gérer les clubs, les étudiants membres, les responsables de clubs ainsi que les événements organisés par chaque club.

## Environnement de travail

Le développement du module a été réalisé dans un environnement **Docker**, afin de faciliter le déploiement et l'isolation de l'application.

### 1. Technologies utilisées

- **Odoo 17**
- **Python** pour la définition des modèles
- **XML** pour les vues
- **Docker**
- **Git bash**

### 2. Architecture Docker

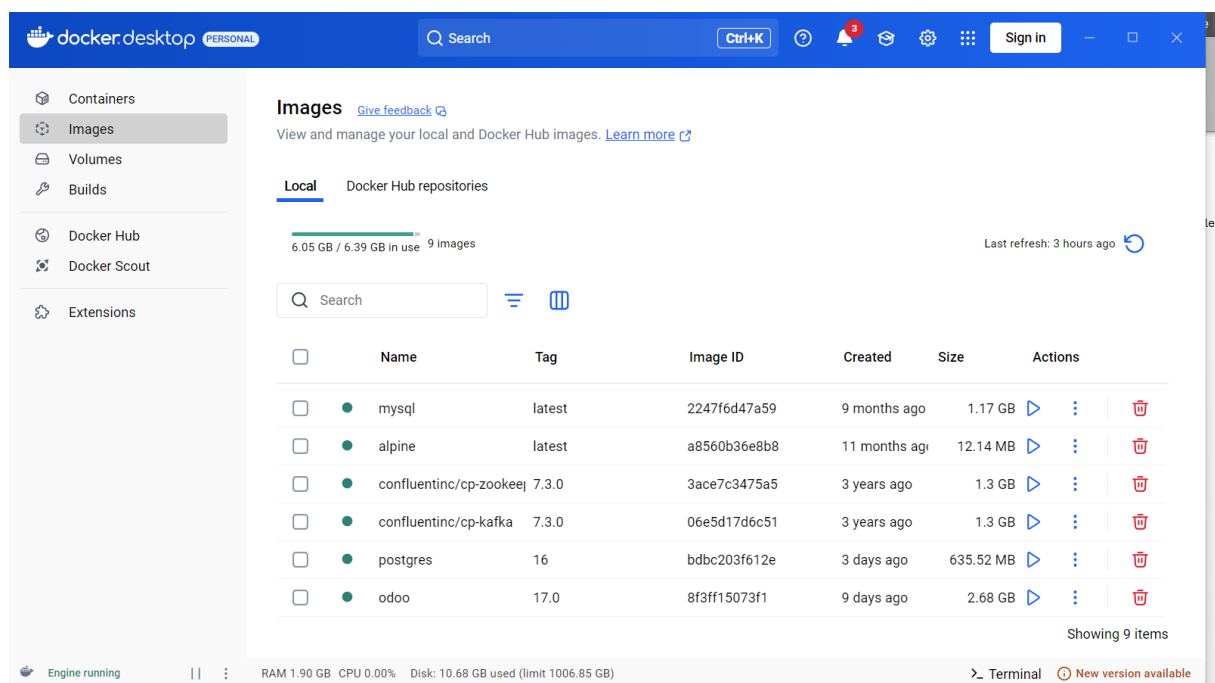


Figure 1 : conteneur « Odoo » pour l'application sur Docker Desktop

# Structure du module Odoo

Le module développé se nomme : **gestion\_clubs\_etudiants**.

Il respecte la structure standard d'un module Odoo :

- models : définition des modèles métiers
- views : interfaces utilisateur (formulaire et listes)
- security : règles d'accès aux données
- \_\_manifest\_\_.py : description du module

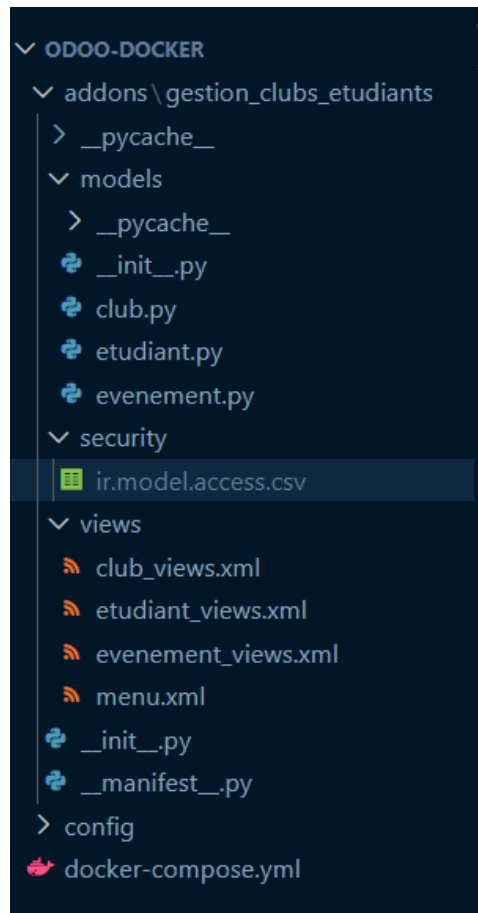


Figure 2 : arborescence du module *gestion\_clubs\_etudiants*

# Modélisation des données

## 1. Gestion des clubs

Le modèle **Club** permet de stocker les informations relatives aux clubs étudiants.

Champs principaux :

- Nom du club
- Description
- Date de création
- Responsable du club (étudiant)
- Nombre d'étudiants (champ calculé)

```
addons > gestion_clubs_etudiants > models > club.py > Club > _compute_nb_etudiants
1  from odoo import models, fields, api # type: ignore
2
3  class Club(models.Model):
4      _name = "gestion.club"
5      _description = "Club Étudiant"
6
7      name = fields.Char(string="Nom du club", required=True)
8      description = fields.Text(string="Description")
9      date_creation = fields.Date(string="Date de création")
10     responsable_id = fields.Many2one("gestion.etudiant", string="Responsable du club")
11     nb_etudiants = fields.Integer(string="Nombre d'étudiants", compute="_compute_nb_etudiants")
12
13     @api.depends('name')
14     def _compute_nb_etudiants(self):
15         for club in self:
16             club.nb_etudiants = self.env['gestion.etudiant'].search_count([
17                 ('club_id', '=', club.id)
18             ])
```

Figure 3 : code du model Club

## 2. Gestion des étudiants

Le modèle **Étudiant** permet de gérer les membres des clubs.

Champs principaux :

- Nom complet
- Email
- Club
- Date d'adhésion

Chaque étudiant est lié à un club via une relation **Many2one**.

```

addons > gestion_clubs_etudiants > models > etudiant.py > Etudiant > date_adhesion
1  from odoo import models, fields # type: ignore
2
3  class Etudiant(models.Model):
4      _name = "gestion.etudiant"
5      _description = "Étudiant"
6
7      name = fields.Char(string="Nom complet", required=True)
8      email = fields.Char(string="Email")
9      club_id = fields.Many2one("gestion.club", string="Club")
10     date_adhesion = fields.Date(string="Date d'adhésion")

```

Figure 4 : code du model étudiant

### 3. Gestion des événements

Un club peut organiser plusieurs événements.

Le modèle **Événement** contient :

- Nom de l'événement
- Date
- Lieu
- Club organisateur
- Description

```

addons > gestion_clubs_etudiants > models > evenement.py > ...
1  from odoo import models, fields # type: ignore
2
3  class Evenement(models.Model):
4      _name = "gestion.evenement"
5      _description = "Événement du club"
6
7      name = fields.Char(string="Nom de l'événement", required=True)
8      date = fields.Date(string="Date")
9      lieu = fields.Char(string="Lieu")
10     club_id = fields.Many2one("gestion.club", string="Club", required=True)
11     description = fields.Text(string="Description")
12

```

Figure 5 : code du model événement

# Interfaces utilisateur

Les interfaces ont été développées à l'aide de vues **Tree** (liste) et **Form** (formulaire).

Un menu principal **Clubs Étudiants** permet d'accéder aux sous-menus :

- Clubs
- Étudiants
- Événements

```
ddons > gestion_clubs_etudiants > views > club_views.xml
1  <odoo>
2  ... <record id="view_club_tree" model="ir.ui.view">
3  ... <field name="name">club.tree</field>
4  ... <field name="model">gestion.club</field>
5  ... <field name="arch" type="xml">
6  ... <tree>
7  ... <field name="name"/>
8  ... <field name="date_creation"/>
9  ... <field name="responsable_id"/>
10 ... <field name="nb_etudiants"/>
11 ...
12 ... </tree>
13 ... </field>
14 ... </record>
15
16 ... <record id="view_club_form" model="ir.ui.view">
17 ... <field name="name">club.form</field>
18 ... <field name="model">gestion.club</field>
19 ... <field name="arch" type="xml">
20 ... <form>
21 ... <sheet>
22 ... <group>
23 ... <field name="name"/>
24 ... <field name="date_creation"/>
25 ... <field name="responsable_id"/>
26 ... <field name="description"/>
27 ... <field name="nb_etudiants" readonly="1"/>
28 ... </group>
29 ... </sheet>
30 ... </form>
31 ... </field>
32 ... </record>
33 </odoo>
34
```

Figure 6 : code du view club

Clubs Étudiants

Clubs

Étudiants

Événements

New

Clubs

New

Nom du club

Date de création

Responsable du club

Description

Nombre d'étudiants

0

Figure 7 : formulaire de création d'un club

```

addons > gestion_clubs_etudiants > views > etudiant_views.xml
1  <odoo>
2  ... <record id="view_etudiant_tree" model="ir.ui.view">
3  ... <field name="name">etudiant.tree</field>
4  ... <field name="model">gestion.etudiant</field>
5  ... <field name="arch" type="xml">
6  ... <tree>
7  ... <field name="name"/>
8  ... <field name="email"/>
9  ... <field name="club_id"/>
10 ... <field name="date_adhesion"/>
11 ... </tree>
12 ... </field>
13 ... </record>
14
15
16 ... <record id="view_etudiant_form" model="ir.ui.view">
17 ... <field name="name">etudiant.form</field>
18 ... <field name="model">gestion.etudiant</field>
19 ... <field name="arch" type="xml">
20 ... <form>
21 ... <sheet>
22 ... <group>
23 ... <field name="name"/>
24 ... <field name="email"/>
25 ... <field name="club_id"/>
26 ... <field name="date_adhesion"/>
27 ... </group>
28 ... </sheet>
29 ... </form>
30 ... </field>
31 ... </record>
32 </odoo>
33

```

Figure 8 : code du view étudiant

Clubs Étudiants

Clubs

Étudiants

Événements

New

Étudiants

New

Nom complet

Email

Club

Date d'adhésion

Figure 9 : formulaire de création d'un étudiant

```

addons > gestion_clubs_etudiants > views > evenement_views.xml
1  <odoo>
2  ....<record id="view_evenement_tree" model="ir.ui.view">
3  ....<field name="name">evenement.tree</field>
4  ....<field name="model">gestion.evenement</field>
5  ....<field name="arch" type="xml">
6  ....<tree>
7  ....<field name="name"/>
8  ....<field name="date"/>
9  ....<field name="club_id"/>
10 ....<field name="lieu"/>
11 ....</tree>
12 ....</field>
13 ....</record>
14
15 ....<record id="view_evenement_form" model="ir.ui.view">
16 ....<field name="name">evenement.form</field>
17 ....<field name="model">gestion.evenement</field>
18 ....<field name="arch" type="xml">
19 ....<form>
20 ....<sheet>
21 ....<group>
22 ....<field name="name"/>
23 ....<field name="club_id"/>
24 ....<field name="date"/>
25 ....<field name="lieu"/>
26 ....<field name="description"/>
27 ....</group>
28 ....</sheet>
29 ....</form>
30 ....</field>
31 ....</record>
32 </odoo>
33

```

Figure 10 : code du view événement



Clubs Étudiants Clubs Étudiants Événements

New Événements New

Nom de l'événement \_\_\_\_\_ Date

Lieu \_\_\_\_\_ Club

Description

Figure 11 : formulaire de création des événements

Clubs Étudiants Clubs Étudiants Événements

New Clubs

1-2 / 2

<input type="checkbox"/> Nom du club	Date de création	Responsable du club	Nombre d'étudiants
<input type="checkbox"/> Ambition jeunes EMSI	01/04/2016		1
<input type="checkbox"/> fun and games	01/07/2019	Bouthayna Bekkouch	0

Figure 12 : menu principal Clubs Étudiants

## Fichier `__manifest__.py`

Le fichier `__manifest__.py` est le descripteur du module. Il définit les métadonnées, les dépendances, et liste les fichiers de données à charger (sécurité, menus, vues) dans l'ordre approprié. Ce fichier permet à Odoo d'installer et de configurer le module automatiquement.

```

addons > gestion_clubs_etudiants > __manifest__.py
1  {
2      "name": "Gestion des clubs étudiants",
3      "version": "1.0",
4      "summary": "Gestion des clubs et des étudiants",
5      "author": "Bouthayna",
6      "category": "Education",
7      "depends": ["base"],
8      "data": [
9          "security/ir.model.access.csv",
10         "views/menu.xml",
11         "views/club_views.xml",
12         "views/etudiant_views.xml",
13     ],
14     "application": True,
15 }
16

```

Figure 13 fichier `__manifest__.py`

# Sécurité et droits d'accès

Afin de garantir l'accès aux données, un fichier `ir.model.access.csv` a été mis en place. Il permet aux utilisateurs d'effectuer les opérations CRUD (Create, Read, Update, Delete) sur les modèles du module.

```
addons > gestion_clubs_etudiants > security > ir.model.access.csv > data
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_club,access_club,model_gestion_club,,1,1,1,1
3 access_etudiant,access_etudiant,model_gestion_etudiant,,1,1,1,1
4 access_gestion_evenement,access_gestion_evenement,model_gestion_evenement,,1,1,1,1
```

Figure 14 : fichier `ir.model.access.csv`

## Conclusion

Ce projet a permis de mettre en pratique les concepts fondamentaux du développement sous Odoo, notamment la création de modules, la gestion des modèles, des vues et de la sécurité.

Le module développé répond aux besoins de gestion des clubs étudiants et peut être enrichi ultérieurement par l'ajout de rôles, de statistiques avancées ou de workflows.