Create Custom Modul in Odoo with Python

Rochana P H

Typical case, modul development dalam Odoo adalah memodifikasi atau meng-extend modul yg sudah tersedia.

Yang akan kita coba,

- membuat modul baru dengan dependency sesedikit mungkin
- modul bukan hasil modifikasi atau extend dari modul yg sudah ada

Modul Basic Structure

```
__init__.py → python importable
__manifest__.py → descriptor file
```

model.py view.xml

Membuat aplikasi To-Do Task

Buku: Odoo 10 Development Essentials PACKT

Aktivasi mode developer Odoo

Masuk ke dev.mode:

Settings → Activate the developer mode (bagian kanan bawah) → Apps

Buat folder modul aplikasi dalam directory addons

Buat folder *todo_app*

Catatan:

Nama folder merupakan nama teknis (warna merah pada Odoo dev. mode)

Nama teknis → harus valid diidentifikasi Python

- Dimulai dengan huruf alfabet
- Hanya boleh memuat alfabet, angka, dan underscore

Sementara, buat file __init__.py kosong

1. Init file

Buat file, ___manifest___.py

2.Descriptor File

Descriptor file

Harus memuat sebuah Python dictionary dengan beberapa key descriptor yg memungkinkan. Minimal yg dibutuhkan ada key *name*.

```
'name': 'To-Do Application',
  'description': 'Manage your personal To-Do
  tasks.',
  'author': 'Daniel Reis',
  'depends': ['base'],
  'application': True,
}
```

Key depends valuenya list dari dependency modul2 yg dibutuhkan.

Kalau modul ini diinstall → yg di dalam depends juga diinstall.

Kalau modul yg dibuat tidak memerlukan modul lain, isikan value dengan base.

Cek list aplikasi

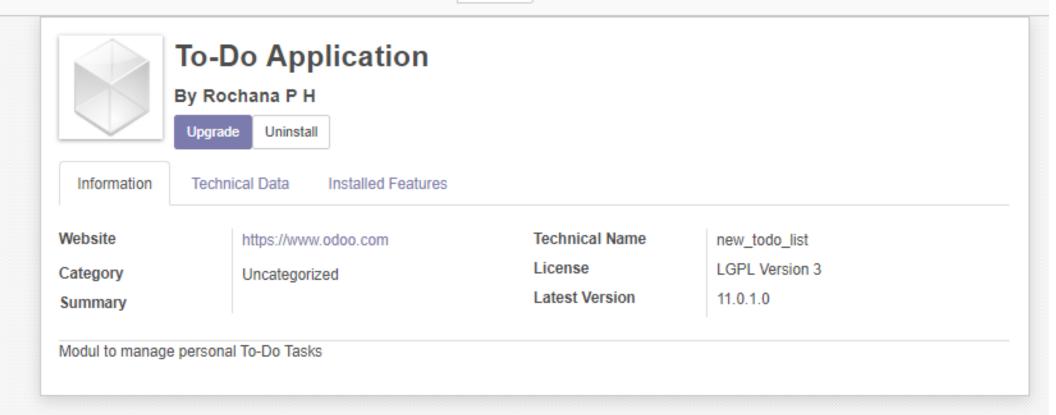
- Simpan file.
- Update list apps Odoo.

Apps → Update Apps List → Reload

- Cek apakah aplikasi berhasil dibuat, ada di daftar list apps
- Install.

Apps / To-Do Application

Action ▼



Diimplementasikan dengan Python class yang diambil dari template class Odoo

3.Model Layer

Model layer

Aplikasi yg akan kita buat:

- Aplikasi untuk mencatatkan to-do tasks.
- Akan dibuat single text field untuk deskripsi task
- Akan ada checkbox untuk menandai yg sudah selesai dilakukan
- Akan ditambahkan tombol clean untuk menghapus tugas2 yg sudah selesai

Data model

Buat file, todo_model.py

```
# -*- coding: utf-8 -*-
from odoo import models, fields

class TodoTask(models.Model):
    _name = 'todo.task'
    _description = 'To-do Task'
    name = fields.Char('Description', required=True)
    is_done = fields.Boolean('Done?')
    active = fields.Boolean('Active', default=True)
```

Data model

- **Baris 1** special marker, memberi info ke Python interpreter bahwa file tsb pakai format UTF-8 sehingga dapat menghandle non-ASCII characters.
- Baris 2 import statement, mengimport objek *models* dan *fields* dari Odoo core
- Baris 4 deklarasi model baru kita, yakni class yg diambil dari models. Model
- **Baris 5** atribut *_name* merupakan identifier yg akan digunakan Odoo utk merefer ke model ini (*todo.task*).
- **Baris** 6 atribut _description, tidak mandatory, tapi demi model record yg user-friendly untuk user messages.
- Baris 7:9 mendefinisikan field yg dibutuhkan model

Nah setelah file todo_model.py dibuat, file init harus dirubah agar dapat digunakan oleh modul. Maka harus diload di __init__.py, ubah menjadi

```
1 from . import todo_model
2
```

Upgrade

- Stop-start server Odoo (jika diperlukan)
- Masuk Odoo dev mode
- Update list apps Odoo
 - Apps → Update Apps List → Reload
- Upgrade To-Do Application

Cek technical menu

Sampai di sini, belum ada menu di dashboard yg bisa kita gunakan terkait todo apps, karena kita belum membuat viewnya. Untuk memastikan apa modelnya sudah terbuat, cek di:

- Setting → Database Structure (bagian kiri, di bawah technical) → Models
- Search todo.task → klik
- Cek apakah atribut2 dlm class TodoTask sudah ada apa belum. Tampilan di todo.task harusnya spt ini:

Print ▼ Action ▼

lodel Description lodel ransient Model	To-do Task todo.task	todo.task			Base Object new_todo_list	
lail Thread						
Fields Acces	s Rights Notes Field Label	Views Field Type	Required	Readonly	Indexed	Туре
_last_update	Last Modified on	datetime		€		Base Field
active	Active	boolean				Base Field
create_date	Created on	datetime				Base Field
create_uid	Created by	many2one				Base Field
display_name	Display Name	char		€		Base Field
id	ID	integer		€		Base Field
is_done	Done?	boolean				Base Field
name	Description	char	€			Base Field
write_date	Last Updated on	datetime				Base Field
write_uid	Last Updated by	many2one				Base Field

Atribut/fields lain bawaan Odoo

Di bawah ini adalah reserved fields Odoo, otomatis ditambahkan setiap bikin model baru

- id is a unique numeric identifier for each record in the model.
- create_date and create_uid specify when the record was created and who created it respectively.
- write_date and write_uid confirm when the record was last modified and who modified it respectively.
- __last_update is a helper that is not actually stored in the database. It is used for concurrency checks.

Mendeskripsikan user interface dengan XML.

4. View Layer

View layer

Buat sub-directory *views*, file XML diletakkan di directory tsb.

Jadi pathnya → todo_app/views

Menambahkan Menu

Di poin 3 kita sudah membuat model utk menyimpan data. Selanjutnya buat interface berupa menu pada dashboard Odoo.

Buat file, todo_menu.xml

```
<?xml version='1.0'?>
     <odoo>
         <!-- Action to open To-do Task list -->
         <act_window id="action_todo_task"
             name="To-do Task"
             res_model="todo.task"
             view mode="tree,form"/>
         <!-- Menu item to open To-do Task list -->
         <menuitem id="menu_todo_task"</pre>
10
             name="Todos"
             action="action_todo_task"/>
     </odoo>
```

Yang di atas tadi adalah Odoo data file, bisa dilihat ada 2 item utk ditambahkan ke dashboard Odoo:

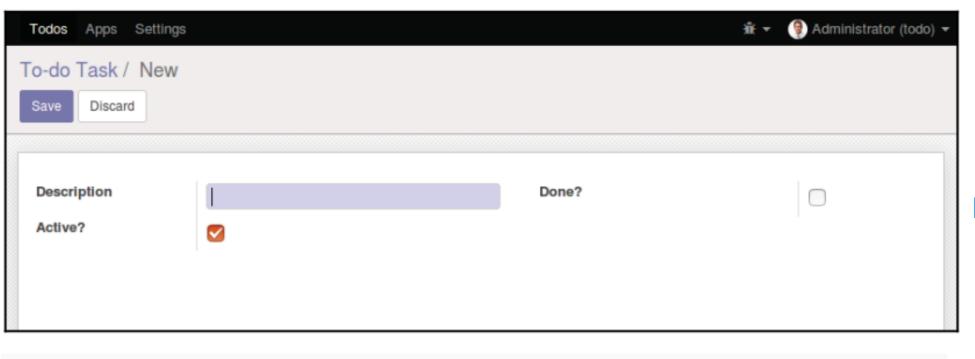
- <act_window> mendefinisikan action yg akan memanggil model todo.task yg kita buat sebelumnya (di file todo model.py).
 - dengan tree view dan form view respectively.
- <menuitem> mendefinisikan menu (yg di tab atas Odoo) dan memanggil action todo task yg sdh dibuat di atasnya tag ini.

Modul kita belum tahu adanya file data XML yg barusan kita buat.

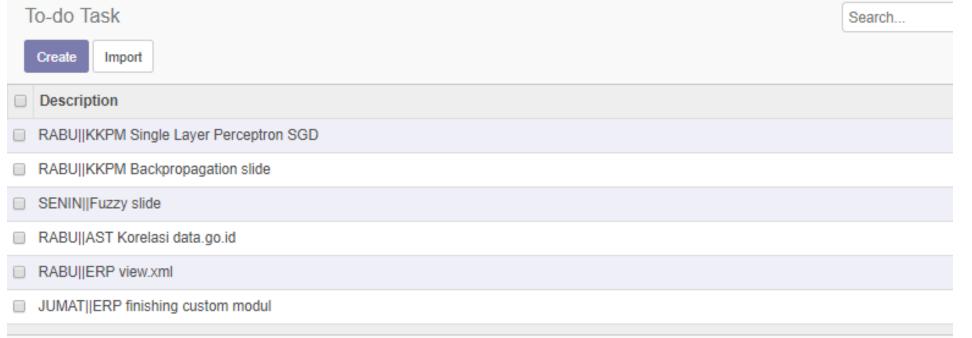
Maka, tambahkan key "data" pada file __manifest__.py

```
'name': 'To-Do Application',
'version': '1.0',
'description': """Modul to manage personal To-Do Tasks""",
'author': 'Rochana P H',
'depends': ['base'],
'data': ['views/todo_menu.xml'],
'installable': True,
'application': True,
'autoinstall': False
```

Lalu, upgrade modul.



Form view



Tree/list view

Tugas terakhir

- Mencari tutorial custom modul Odoo

 implement (boleh desain sendiri juga)
- Tiap kelompok tidak boleh sama
- Jumat 18 Mei, presentasi judul-ide
- Jumat 25 Mei, laporan dikumpulkan saat responsi
- Poin plus

 kalua bisa extend/mengkombinasikan modul yg sudah ada