

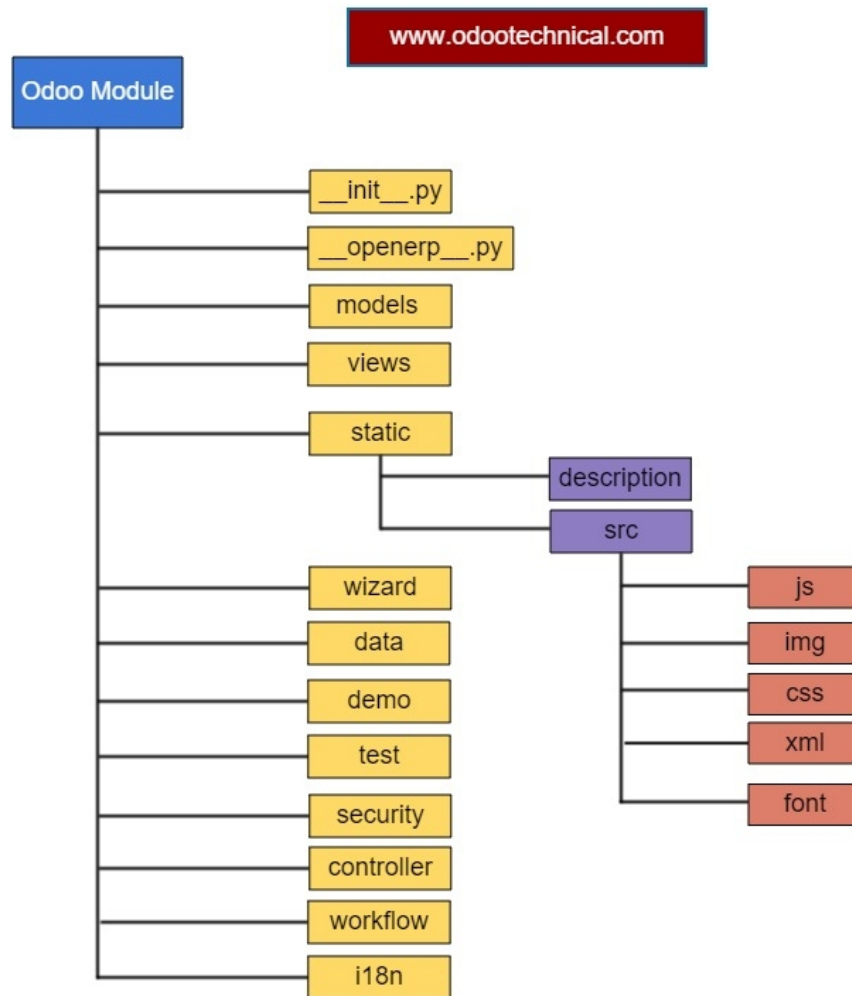
Day 2 Contents

- 1- Odoo directory structure
- 2- Odoo Module structure
- 3- MVC Architecture
- 4- Create new module
- 5- Override existing module
- 6- Integrate external modules

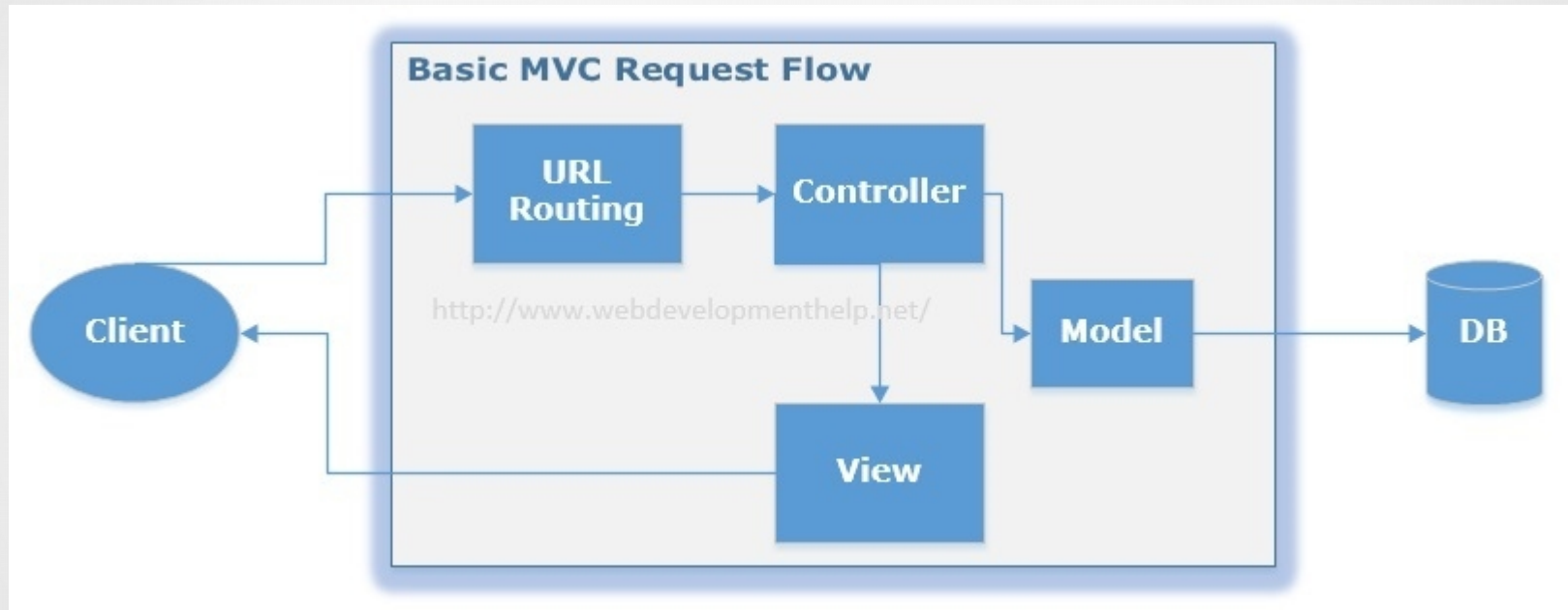
Odoo Directory Structure

- addons # includes odoo modules
- debian
- doc
- odoo # includes odoo 10 files
- setup
- odoo-bin # entry point to start odoo server
- COPYRIGHT
- LICENSE
- MANIFEST.in
- Makefile
- requirements.txt
- setup.cfg
- setup.py

Odoo Module Structure



MVC architecture



Build a new Odoo module

- 1- Odoo modules can either add brand new business logic to an Odoo system, or alter and extend existing business logic
- 2- Everything in Odoo thus starts and ends with modules.

Build module files

- 1- `odoo-bin scaffold <module name> <where to put it>`
- 2- `mkdir extra_addons`
- 3- `./odoo-bin scaffold openacademy extra_addons`

Model

- 1- A key component of Odoo is the ORM layer. This layer avoids having to write most SQL by hand and provides extensibility and security services2.
- 2- Business objects are declared as Python classes extending Model which integrates them into the automated persistence system.
- 3- Models can be configured by setting a number of attributes at their definition. The most important attribute is `_name` which is required and defines the name for the model in the Odoo system.

Model (fields)

- 1- simple (ex boolean,date,char)
- 2- relational (ex many2one)
- 3- Reserved (id,create_date,create_uid,write_date,write_uid)

Common Attributes(string,required,help,index)

Exercise



Exercise

Define a model

Define a new data model *Course* in the *openacademy* module. A course has a title and a description. Courses must have a title.

Edit the file `openacademy/models/models.py` to include a *Course* class.

openacademy/models.py

```
from odoo import models, fields, api

class Course(models.Model):
    _name = 'openacademy.course'

    name = fields.Char(string="Title", required=True)
    description = fields.Text()
```

Data files

- Module data is declared via data files, XML files with <record> elements. Each <record> element creates or updates a database record.

Data files

```
<odoo>
  <data>
    <record model="{model name}" id="{record identifier}">
      <field name="{a field name}">{a value}</field>
    </record>
  </data>
</odoo>
```

Exercise



Exercise

Define demonstration data

Create demonstration data filling the *Courses* model with a few demonstration courses.

Edit the file `openacademy/demo/demo.xml` to include some data.

`openacademy/demo.xml`

```
<odoo>
  <data>
    <record model="openacademy.course" id="course0">
      <field name="name">Course 0</field>
      <field name="description">Course 0's description

Can have multiple lines
    </field>
    </record>
    <record model="openacademy.course" id="course1">
      <field name="name">Course 1</field>
      <!-- no description for this one -->
    </record>
    <record model="openacademy.course" id="course2">
      <field name="name">Course 2</field>
      <field name="description">Course 2's description</field>
    </record>
  </data>
</odoo>
```


Actions and Menus

- 1- Actions and menus are regular records in database, usually declared through data files. Actions can be triggered in three ways:
- 2- by clicking on menu items (linked to specific actions)
- 3- by clicking on buttons in views (if these are connected to actions) as contextual actions on object
- 4- The action must be declared before its corresponding menu in the XML file.
- 5- Data files are executed sequentially, the action's id must be present in the database before the menu can be created.

Exercise



Exercise

Define new menu entries

Define new menu entries to access courses under the OpenAcademy menu entry. A user should be able to

- display a list of all the courses
 - create/modify courses
1. Create `openacademy/views/openacademy.xml` with an action and the menus triggering the action
 2. Add it to the `data` list of `openacademy/__manifest__.py`

`openacademy/__manifest__.py`

```
'data': [  
    # 'security/ir.model.access.csv',  
    'templates.xml',  
    'views/openacademy.xml',  
],  
# only loaded in demonstration mode  
'demo': [  

```

Exercise

```
openacademy/views/openacademy.xml
<?xml version="1.0" encoding="UTF-8"?>
<odoo>
  <data>
    <!-- window action -->
    <!--
      The following tag is an action definition for a "window action",
      that is an action opening a view or a set of views
    -->
    <record model="ir.actions.act_window" id="course_list_action">
      <field name="name">Courses</field>
      <field name="res_model">openacademy.course</field>
      <field name="view_type">form</field>
      <field name="view_mode">tree,form</field>
      <field name="help" type="html">
        <p class="oe_view_nocontent_create">Create the first course</p>
      </field>
    </record>
    <!-- top level menu: no parent -->
    <menuitem id="main_openacademy_menu" name="Open Academy"/>
    <!-- A first level in the left side menu is needed
      before using action= attribute -->
    <menuitem id="openacademy_menu" name="Open Academy"
      parent="main_openacademy_menu"/>
    <!-- the following menuitem should appear *after*
      its parent openacademy_menu and *after* its
      action course_list_action -->
    <menuitem id="courses_menu" name="Courses" parent="openacademy_menu"
      action="course_list_action"/>
    <!-- Full id location:
      action="openacademy.course_list_action"
      It is not required when it is the same module -->
  </data>
</odoo>
```

Basic views

- 1- Generic view declaration
- 2- Tree views
- 3- Form views
- 4- Search views

Exercise

openacademy/views/openacademy.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<odoo>
```

```
  <data>
```

```
    <record model="ir.ui.view" id="course_form_view">
```

```
      <field name="name">course.form</field>
```

```
      <field name="model">openacademy.course</field>
```

```
      <field name="arch" type="xml">
```

```
        <form string="Course Form">
```

```
          <sheet>
```

```
            <group>
```

```
              <field name="name"/>
```

```
              <field name="description"/>
```

```
            </group>
```

```
          </sheet>
```

```
        </form>
```

```
      </field>
```

```
    </record>
```

```
    <record model="ir.ui.view" id="course_search_view">
```

```
      <field name="name">course.search</field>
```

```
      <field name="model">openacademy.course</field>
```

```
      <field name="arch" type="xml">
```

```
        <search>
```

```
          <field name="name"/>
```

```
          <field name="description"/>
```

```
        </search>
```

```
      </field>
```

```
    </record>
```

```
    <!-- window action -->
```

```
    <!--
```

The following tag is an action definition for a "window action",

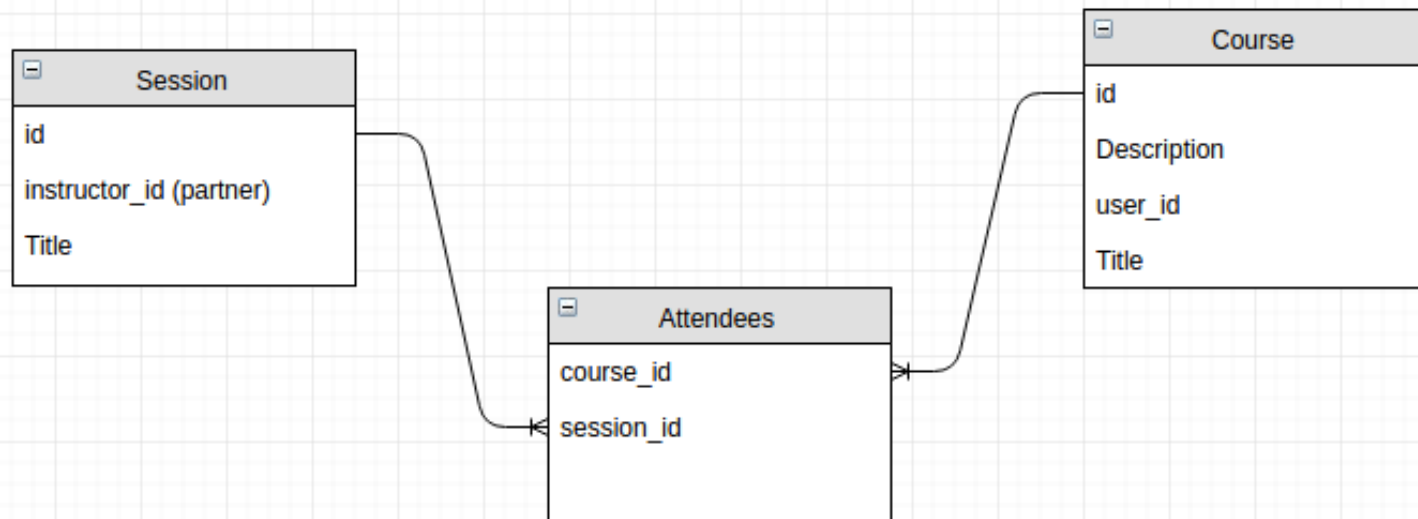


Let's update our module

Relations between models

- 1- Many to one
- 2- One to many
- 3- Many to many

Exercise



Exercise

<!-- replace openacademy.xml -->

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<odoo>
```

```
  <data>
```

```
    <!-- window action -->
```

```
    <!--
```

The following tag is an action definition for a "window action",
that is an action opening a view or a set of views

```
  -->
```

```
  <record model="ir.actions.act_window" id="course_list_action">
```

```
    <field name="name">Courses</field>
```

```
    <field name="res_model">openacademy.course</field>
```

```
    <field name="view_type">form</field>
```

```
    <field name="view_mode">tree,form</field>
```

```
    <field name="help" type="html">
```

```
      <p class="oe_view_nocontent_create">Create the first course
```

```
    </p>
```

```
  </field>
```

```
</record>
```

```
<record model="ir.ui.view" id="course_tree_view">
```

```
  <field name="name">course.tree</field>
```

```
  <field name="model">openacademy.course</field>
```

```
  <field name="arch" type="xml">
```

```
    <tree string="Course Tree">
```

```
      <field name="name"/>
```

```
      <field name="responsible_id"/>
```

```
    </tree>
```

```
  </field>
```

```
</record>
```

Exercise

```
<record model="ir.ui.view" id="course_form_view">
  <field name="name">course.form</field>
  <field name="model">openacademy.course</field>
  <field name="arch" type="xml">
    <form string="Course Form">
      <sheet>
        <group>
          <field name="name"/>
        </group>
        <notebook>
          <page string="Description">
            <field name="description"/>
          </page>
          <page string="Sessions">
            <field name="session_ids">
              <tree string="Registered sessions">
                <field name="name"/>
                <field name="instructor_id"/>
              </tree>
            </field>
          </page>
        </notebook>
      </sheet>
    </form>
  </field>
</record>
```

Exercise

```
<record model="ir.ui.view" id="course_search_view">  
  <field name="name">course.search</field>  
  <field name="model">openacademy.course</field>  
  <field name="arch" type="xml">  
    <search>  
      <field name="name"/>  
      <field name="description"/>  
    </search>  
  </field>  
</record>
```


Exercise

```
<record model="ir.ui.view" id="session_form_view">
  <field name="name">session.form</field>
  <field name="model">openacademy.session</field>
  <field name="arch" type="xml">
    <form string="Session Form">
      <sheet>
        <group>
          <field name="name"/>
          <field name="start_date"/>
          <field name="duration"/>
          <field name="seats"/>
        </group>
        <field name="attendee_ids"/>
      </sheet>
    </form>
  </field>
</record>
<record model="ir.actions.act_window" id="session_list_action">
  <field name="name">Sessions</field>
  <field name="res_model">openacademy.session</field>
  <field name="view_type">form</field>
  <field name="view_mode">tree,form</field>
</record>
```


Exercise

```
<!-- top level menu: no parent -->
  <menuitem id="main_openacademy_menu" name="Open Academy"/>

  <menuitem id="openacademy_menu" name="Open Academy"
    parent="main_openacademy_menu"/>

  <menuitem id="courses_menu" name="Courses" parent="openacademy_menu"
    action="course_list_action"/>

  <menuitem id="session_menu" name="Sessions"
    parent="openacademy_menu"
    action="session_list_action"/>

</data>
</odoo>
```

Exercise

#Replace models.py

```
from odoo import models, fields, api
```

```
class Course(models.Model):
    _name = 'openacademy.course'

    name = fields.Char(string="Title", required=True)
    description = fields.Text()
    responsible_id = fields.Many2one('res.users',
        ondelete='set null', string="Responsible", index=True)
    session_ids = fields.One2many(
        'openacademy.session', 'course_id', string="Sessions")
```

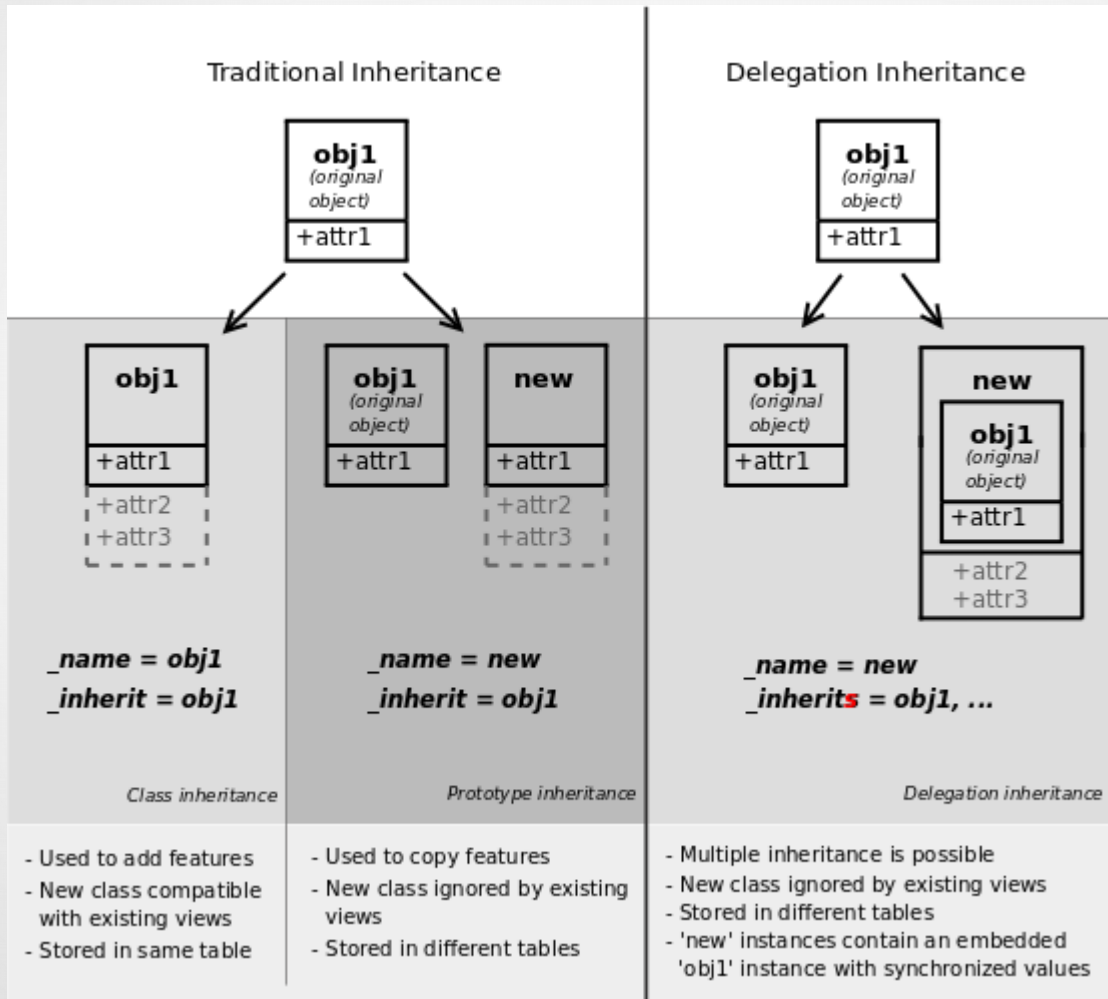
```
class Session(models.Model):
    _name = 'openacademy.session'
    name = fields.Char(string="Title", required=True)
    start_date = fields.Date()
    duration = fields.Float(digits=(6, 2), help="Duration in days")
    seats = fields.Integer(string="Number of seats")

    instructor_id = fields.Many2one('res.partner', string="Instructor")
    course_id = fields.Many2one('openacademy.course',
        ondelete='cascade', string="Course", required=True)
    attendee_ids = fields.Many2many('res.partner', string="Attendees")
```



Let's update our module

Model Inheritance



View Inheritance

```
<!-- improved idea categories list -->
<record id="idea_category_list2" model="ir.ui.view">
  <field name="name">id.category.list2</field>
  <field name="model">idea.category</field>
  <field name="inherit_id" ref="id_category_list"/>
  <field name="arch" type="xml">
    <!-- find field description and add the field
         idea_ids after it -->
    <xpath expr="//field[@name='description']" position="after">
      <field name="idea_ids" string="Number of ideas"/>
    </xpath>
  </field>
</record>
```

Exercise

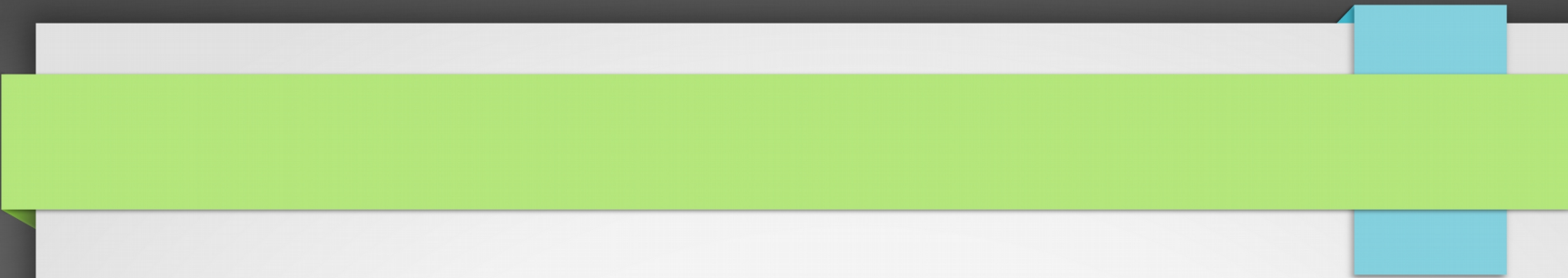
- 1- Using model inheritance, modify the existing Partner model to add an instructor boolean field, and a many2many field that corresponds to the session-partner relation
- 2- Using view inheritance, display this fields in the partner form view

Exercise

```
#openacademy/__init__.py
```

```
from . import controllers
from . import models
from models import partner
```

```
#openacademy/__manifest__.py
# 'security/ir.model.access.csv',
'templates.xml',
'views/openacademy.xml',
'views/partner.xml',
],
# only loaded in demonstration mode
'demo': [
```



```
#openacademy/partner.py
```

```
from odoo import fields, models
class Partner(models.Model):
    _inherit = 'res.partner'
    # Add a new column to the res.partner model, by
    # default partners are not
    # instructors
    instructor = fields.Boolean("Instructor",
    default=False)
    session_ids =
    fields.Many2many('openacademy.session',
        string="Attended Sessions", readonly=True)
```



```
<?xml version="1.0" encoding="UTF-8"?>
<odoo>
  <data>
    <!-- Add instructor field to existing view -->
    <record model="ir.ui.view" id="partner_instructor_form_view">
      <field name="name">partner.instructor</field>
      <field name="model">res.partner</field>
      <field name="inherit_id" ref="base.view_partner_form"/>
      <field name="arch" type="xml">
        <notebook position="inside">
          <page string="Sessions">
            <group>
              <field name="instructor"/>
              <field name="session_ids"/>
            </group>
          </page>
        </notebook>
      </field>
    </record>
    <record model="ir.actions.act_window" id="contact_list_action">
      <field name="name">Contacts</field>
      <field name="res_model">res.partner</field>
      <field name="view_mode">tree,form</field>
    </record>
    <menuitem id="configuration_menu" name="Configuration"
      parent="main_openacademy_menu"/>
    <menuitem id="contact_menu" name="Contacts"
      parent="configuration_menu"
      action="contact_list_action"/>
  </data>
</odoo>
```



Let's update our module

Domain

A domain parameter can be added to relational fields to limit valid records for the relation when trying to select records in the client interface.

```
instructor_id = fields.Many2one('res.partner', string="Instructor", domain=[('instructor', '=', True)])
```

Additional Features

- 1- Computed fields
- 2- Onchange
- 3- Workflows
- 4- I18n
- 5- Reporting
- 6- Web services

[illegible]