onnet

2021
VIE

Odoo Views, Menus and Actions

www.on.net.vn
www.arrowhitech.com

onnet
BY AHT TECH JSC

Table of contents

- Overview
- Generic Structure
- View Types
- View Inheritance
- Menus and Actions

onnet
BY AHT TECH JSC

Overview

What is view?

- Views are what define how records should be displayed to end-users
- They are specified in XML which means that they can be edited independently from the models that they represent
- There exist various types of views: form, kanban, tree, calendar, etc...
- Each view is an instance of ir.ui.view model
- View are defined in *.xml files, normally stored in /views folder and should be declared in __manifest__.py file

Generic Structure

Fields

- name (Char): use as as a mnemonic/description of the view
- model (Char): the model linked to the view
- priority (Integer): the lowest priority is the default
- groups_id (Many2many -> res_users.Groups)
- arch (Text): description of the view layout

```
<record model="ir.ui.view" id="view_id">
    <field name="name">view.name</field>
    <field name="model">object_name</field>
    <field name="priority" eval="16"/>
    <field name="arch" type="xml">
        <!-- view content: <form>, <tree>, <graph>, ... -->
    </field>
</record>
```

Generic Structure

Attributes

- create
- edit (form, list, gantt)
- delete: disable/enable record deletion through the **Action** menu dropdown
- duplicate
- decoration-{@name} (list, gantt)
- sample: populate the view with a set of sample records if none are found for the current model.
- string

```
<record id="view_partner_tree" model="ir.ui.view">
      <field name="name">res.partner.tree</field>
      <field name="model">res.partner</field>
      <field eval="8" name="priority"/>
      <field name="arch" type="xml">
        <tree string="Contacts" sample="1" multi_edit="1">
          <field name="display_name" string="Name"/>
          <field name="active" invisible="1"/>
            ...
        </tree>
      </field>
    </record>
```

View Types

- Activity: displays the activities linked to the records
- Calendar: displays records as events in a daily, weekly, monthly or yearly calendar
- Cohort: is used to display and understand the way some data changes over a period of time.
- Dashboard: displays aggregate data, can embed sub views
- Form: displays the data from a single record
- Gantt: displays Gantt charts (for scheduling)
- Graph: visualizes aggregations over a number of records or record groups
- Grid: new in Odoo 15.0, in progress
- Kanban: a kanban board visualisation, it displays records as “cards”
- List: displays records in a tabular form
- Map: displays records on a map and the routes between them
- Pivot: visualizes aggregations as a pivot table
- Qweb: standard QWeb Templates inside a view’s arch
- Search: filter other view’s content

Form View

```
<form string="Product">
    <header>
        <button .../>
    </header>
    <sheet>
        ...
        <notebook>
            <page string="General Information" name="general_information">
                ...
              </page>
            <page string="Sales" attrs="{'invisible':[('sale_ok','=',False)]}" name="sales" invisible="1">
                ...
              </page>
            <page string="Purchase" name="purchase" attrs="{'invisible': [('purchase_ok','=',False)]}" invisible="1">
                ...
            </page>
            ...
        </notebook>
    </sheet>
    <div class="oe_chatter">
        <field name="message_follower_ids"/>
        <field name="activity_ids"/>
        <field name="message_ids"/>
    </div>
</form>
```

List View

```
<tree string="Idea list">
   <field name="name"/>
   <field name="inventor_id"/>
</tree>
```

Search View

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

View Inheritance

View inheritance allows altering views declared elsewhere (adding or removing content)

Inheritance fields:

- inherit_id (Many2one): the current view's parent view
- mode (Selection): extension / primary

Inheritance specs:

- Inheritance specs are comprised of an element locator, to match the inherited element in the parent view
- There are three types of element locators for matching a target element:
  - xpath
  - field
  - any other element
- The inheritance spec may have an optional position attribute specifying how the matched node should be altered:
  - inside
  - replace
  - after
  - before
  - attributes
  - move

Menus & Actions

The **actions** define the behavior of the system in response to the actions of the users ; login of a new user, double-click on an invoice, click on the action button, ...

There are different types of simple actions:

- Window
- Report
- Wizard
- ...

Actions can be triggered in three ways:

- by clicking on menu items (linked to specific actions)
- by clicking on buttons in views (if these are connected to actions)
- as contextual actions on object

Menus & Actions

The action must be declared before its corresponding menu in the XML file.

Data files are executed sequentially, the action’s id must be present in the database before the menu can be created.

```
<record model="ir.actions.act_window" id="action_list_ideas">
   <field name="name">Ideas</field>
   <field name="res_model">idea.idea</field>
   <field name="view_mode">tree,form</field>
</record>
<menuitem id="menu_ideas" parent="menu_root" name="Ideas"
sequence="10"
         action="action_list_ideas"/>
```

Practicals

Create a module
Create an empty module Open Academy, and install it in Odoo

Create 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.

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

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

Q&A