

Les vues de base

Les vues :

- définissent comment **afficher** les enregistrements ;
- représentent un **mode de visualisation particulier** (liste d'enregistrements, un graphique synthétique, etc.) ;
- peuvent être **appelées** de manière **génériques** par leur type (ex Liste de partenaires) ou par leur **identifiant** (id).

Pour les vues génériques, celle qui a le type requis et la plus basse priorité est appelée (la vue de qui a la plus basse priorité de chaque type est la vue par défaut du type en question).

L'utilisation du mécanisme **d'héritage** va permettre de **modifier** la déclaration d'une vue (ajout ou suppression de contenu).

Déclaration d'un vue générique

Une vue est déclarée comme un **enregistrement** du modèle **ir.ui.view**. Le **type** de vue est déterminé implicitement par l'élément **root** de **l'arborescence** des champs :

```
<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>
```

Attention :

La **description** d'une vue est faite en **XML**.

Le champ **arch** doit être déclaré comme **type="xml"** pour que le fichier soit parsé (interprété) correctement.

Les vues Liste

Les **vues liste** (Tree views) affichent les données sous forme tabulaire.

L'élément root est **<tree>**. La forme la plus simple d'une vue liste consiste à afficher tous les champs de la table (une colonne par champ) :

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

Les vue formulaire

Les vues **formulaire** permettent de **créer** et **d'éditer** de simples **enregistrements**.

L'élément root est **<form**

>. Les formulaires sont composés d'éléments complexes (group et notebook) et interactifs (boutons and champs) :

```
<form string="Idea form">
  <group colspan="4">
    <group colspan="2" col="2">
      <separator string="General stuff" colspan="2"/>
      <field name="name"/>
      <field name="inventor_id"/>
    </group>
  </group>
```

```

<group colspan="2" col="2">
  <separator string="Dates" colspan="2"/>
  <field name="active"/>
  <field name="invent_date" readonly="1"/>
</group>

<notebook colspan="4">
  <page string="Description">
    <field name="description" nolabel="1"/>
  </page>
</notebook>

<field name="state"/>
</group>
</form>

```

Exercice : Créer sa propre vue formulaire pour afficher un cours :

Les informations à afficher sont le nom et la description du cours.

- modifiez le fichier **openacademy.xml** situé dans le sous-dossier **views** :

```

<?xml version="1.0" encoding="UTF-8"?>
<openerp>
  <data>
    <record model="ir.ui.view" id="cours_form_view">
      <field name="name">cours.form</field>
      <field name="model">openacademy.cours</field>
      <field name="arch" type="xml">
        <form string="Formulaire Cours">

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

        </form>
      </field>
    </record>

    <!-- window action -->
    <!--
      The following tag is an action definition for a "window action",

```

Rappel : pour **visualiser** vos modifications, **mettez à jour** le module openAcademy.

Exercice d'utilisation de la balise notebook :

Dans la vue formulaire, mettez le champ description sous un onglet ce qui facilitera l'ajout ultérieur d'autres onglets pour des informations supplémentaires

- Modifiez le formulaire Cours **openacademy.xml** de la manière suivante :

```

<sheet>
  <group>
    <field name="name"/>
  </group>
  <notebook>
    <page string="Description">
      <field name="description"/>
    </page>
    <page string="About">
      This is an example of notebooks

```

```

        </page>
    </notebook>
</sheet>
</form>
</field>

```

Vous pouvez aussi utiliser du **code HTML** pour plus de souplesse dans la création de vos formulaires.

```

<form string="Idea Form">
  <header>
    <button string="Confirm" type="object" name="action_confirm"
      states="draft" class="oe_highlight" />
    <button string="Mark as done" type="object" name="action_done"
      states="confirmed" class="oe_highlight"/>
    <button string="Reset to draft" type="object" name="action_draft"
      states="confirmed,done" />
    <field name="state" widget="statusbar"/>
  </header>
  <sheet>
    <div class="oe_title">
      <label for="name" class="oe_edit_only" string="Idea Name" />
      <h1><field name="name" /></h1>
    </div>
    <separator string="General" colspan="2" />
    <group colspan="2" col="2">
      <field name="description" placeholder="Idea description..." />
    </group>
  </sheet>
</form>

```

Vue recherche

Les vues **recherche** personnalisent le champ de recherche associé à la vue liste.

L'élément root est `<search>` et il définit quel champs sont utilisé pour la recherche :

```

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

```

S'il vous ne créez pas de vue **recherche** pour un de vos modèles, Odoo **génère** une vue de recherche mais **uniquement** pour le champ **name**.

Exercice : créer une vue recherche pour les cours, vue basée sur les champs titre et description :

- Modifiez le fichier **openacademy.xml**

```

    </field>
  </record>

  <record model="ir.ui.view" id="cours_search_view">
    <field name="name">course.search</field>
    <field name="model">openacademy.cours</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",

```

Retour au sommaire de création d'un module

[Développer un module dans OpenERP 7](#)

From:

[/ - Les cours du BTS SIO](#)

Permanent link:

[/doku.php/dev/openerp/creermodule/basicview](#)

Last update: **2015/12/10 11:23**

