Odoo中关于权限设置：

Odoo中关于权限的设置离不开“组”，我们在设置相关用户的权限时，其主要还是提前分好组，然后在组中加入用户以获得相应的权限，除了系统本身已经设置好的组以外，我们也可以根据自己的需要设置不同的组。

1. 建立组的相关信息，注意事项：

1、在openerp中加入'security/purchase\_security\_inherit.xml'和

'security/ir.model.access.csv'

2、在purchase\_security\_inherit.xml中category\_id(提前定义)，model\_id(代表定义的对象的视图元数据)

3、在ir.model.acess.csv中写入各个组对象对应的读，写，创建和删除权限。其中id和name随便取；model\_id：id是我们所定义的对象的model\_id,在purchase\_security\_inherit.xml中去model\_id对应的ref值； “perm\_read","perm\_write","perm\_create","perm\_unlink" 增删改查权限了，1代表有权限。（也可在odoo的设置—组中找到相应的组来设置权限）

二、隐藏的常用方法

1、处理系统原有的字段：

1）直接隐藏：

<group name="list\_price" position="attributes">

<attribute name="invisible">True</attribute>

</group>

2）满足条件的隐藏：

 <xpath expr="//field[@name='list\_price']" position='attributes'>

<attribute name="attrs">{'invisible': [('type','=', ‘product’)]}</attribute>

</xpath>

3）通过组来隐藏：

<xpath expr="//field[@name= list\_price']" position="attributes">

<attribute name="groups">project\_groups. project\_financial\_staff </attribute>

</xpath>

注意：在写groups的时候，project\_groups为文件夹名，project\_financial\_staff为定义的组中的record\_id(对于自己定义的groups必须是“文件名.record\_id”的形式)

4）利用组和某些条件来隐藏：

<xpath expr="//field[@name='list\_price']" position='replace'>

<field name="list\_price" groups="Product\_Template\_Inherited.group\_product\_users"

attrs="{'invisible': [('product\_type','=','bulk')]}"/>

< field name="list\_price"groups="Product\_Template\_Inherited.group\_product\_managers"/>

</xpath>

注意：代码升级后，以admin用户登录时会看到有两个'list\_price字段，主要是因为两个组中都包含有admin用户，可以在设置—用户—组找到对应的组，删除其中的admin用户。

2、处理自己添加的字段

1）在视图上指定字段的访问权限（利用组和某些条件）：

<field name="purchase\_type " domain="[('type','=','product')]"

groups="Product\_Template\_Inherited.group\_product\_managers"/>

2）在对象上指定字段的访问权限：

class purchase\_inherit(models.Model):

\_inherit = 'purchase.order'

project = fields.Many2one('project.project',string=u'所属项目')

OA = fields.Char(string=u'OA',write=[‘project\_groups. project\_financial\_staff’],

read=[‘project\_groups. project\_financial\_staff’])

1. 利用odoo系统做相应的设置：
2. 组的设置：

进入设置—用户—组，我们就可以看到我们自己设置好的组（也可通过系统自己创建），可以查看相关组的访问权限，也可在系统中直接通过编辑来修改相应的访问权限。通过为用户添加项目，就可以为这个组添加不同的用户。

1. 用户的设置：

进入设置—用户—用户，就可以创建不同的用户，创建好用户后，只需要在“更多”中修改密码就可以设置好我们所需要的用户。

1. 菜单的设置：

进入设置—用户界面—菜单项目中，可对相应的菜单利用组来进行设置，可以实现不同的组进去看到不同的菜单项，从而实现菜单的隐藏。

注：在隐藏菜单时，可能会发现对于系统本身的菜单，隐藏母菜单，其下面的子菜单也会相应的隐藏起来（也可以给子菜单单独设置权限组）。而当我们在系统的母菜单下利用新的模型建立子菜单时，即使给母菜单或者该子菜单设置了权限组也不起作用，此时就需要我们在新建的子菜单下面权限里加入相应的系统权限（仿照系统原有的菜单权限加入系统本身具有的一些权限组）。

1. 工作流的设置：

工作流的设置一般在工作流的xml文件中设置（迁移过程中）。

<record id="purchase\_apply\_tra\_draft\_wait\_prove" model="workflow.transition">

<field name="act\_from" ref="purchase\_apply\_act\_draft"/>

<field name="act\_to" ref="purchase\_apply\_act\_wait\_prove"/>

<field name="signal">action\_toprove</field>

<field name="group\_id" ref="project\_groups.project\_project\_manager"/>

</record>

也可以设置不同groups的人进去看到不同的按钮，从而有不同的工作流权限：

<button string="submit" name="action\_toprove" states="draft"

groups="project\_groups.project\_project\_manager">

<button string="confirm" name="action\_prove" states="wait\_prove"

groups="project\_groups.project\_purchase\_staff"/>

<button string="reject" name="action\_reject" states="wait\_prove"

groups="project\_groups.project\_purchase\_staff"/>

1. 注意事项：

上面所说的利用组和某些条件来隐藏的代码，一般只适用于有两个组的情况下。而当我们设置了三个组，且三个组各自要满足不同的条件，就如材料组，外包组和人工组，要求每个组的员工选产品时分别只能分别看到材料类、外包类和人工类的产品。这种情况下，我们就不能通过简单的代码和系统的设置完成了，需要写入相应的方法来设置权限。