参考：https://segmentfault.com/a/1190000014760312

https://www.cnblogs.com/ygj0930/p/10826105.html

https://segmentfault.com/a/1190000018501650

odoo的权限管理

1、权限组和数据表访问权限

odoo 使用 **res.groups** 对权限进行分组，并使用 **ir.module.category** 进行权限组非分类。

使用.csv进行具体数据模型的CRUD权限控制，一般使用 **security** 目录下的 **ir.model.access.csv**。

权限组即是一条record，配置完成后，会在群组菜单中出现，然后修改记录，添加用户，以及菜单、视图等访问权限（csv中配置的权限组）即可完成权限配

置。

新增的xml和csv需要添加到 __manifest__.py 文件的data数组中，先添加xml，再添加csv

2、用户权限组

（1）权限组分类

使用 **ir.module.category** 对全部进行分组

```
<record id="into_apply_menu_management" model="ir.module.category">
 <field name="name">申请菜单权限组</field>
 <field name="description">申请菜单权限组</field>
</record>
```

（2）权限组

每个权限组就是一笔数据，使用 **record** 创建

可以将多个权限组放在到一个分类中

```
<record model="res.groups" id="group_into_apply_menu_r">
 <field name="name">管线入廊申请-获取</field>
 <field name="category_id" ref="ccm.into_apply_menu_management"/>
 <field name="implied_ids" eval="[(4, ref('base.group_user'))]"/>
 <field name="comment">管线入廊申请-获取</field>
</record>
```

- 这里的 **id** 模块内唯一，而且，需要用在csv文件中；
- 使用 **category_id** 进行分类，**ref** 需要加上模块名；
- **implied_ids** 表示继承组，就是说，也有用继承的组的权限；

eval用法

```
(0, 0, {values}) 根据values的值新建一条记录
(1, ID, {values}) 更新id=ID的记录，（写入values的值）
(2, ID) 删除id=ID这条记录，（调用unlink方法，删除数据及整个主从数据链接关系）
(3, ID) 切断主从数据的链接关系但是不删除这个记录
(4, ID) 为id=ID的数据添加主从链接关系
(5) 删除所有的从数据的链接关系，也就是向所有的从数据调用(3, ID)
(6, 0, {[IDs]}) 用IDs中的记录替换原来的记录（相当于先执行(5)在循环执行(4, ID)）
```

3、数据模型的访问权限控制

一般使用 **security** 目录下的 **ir.model.access.csv** 文件配置。

文件示例如下：

```
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create
erm_unlink
2 access_ccm_into_apply_group_into_apply_menu_r,读申请,model_ccm_i
to_apply,ccm.group_into_apply_menu_r,1,0,0,0
```

示例的模块名为 **ccm**，数据模型名为 **ccm.into_apply**

各个列的内容格式如下：

id

该条访问权限的id，理论上随意，模块中唯一即可，但一般会以：**access_模型名_用户组** 来表示，这里面出现的所有点号，均使用下划线替代

但是也不是必须要按照通用模式来写

上例中的id为 **access_ccm_into_apply_group_into_apply_menu_r**

- **access** 固定；
- **ccm_into_apply** 是模型名，点号换成了下划线；
- **group_into_apply_menu_r** 是第四列的 **group_id**；

name

该条访问权限的名称，随意，可重复

model_id:id

该条访问权限控制的数据模型，标准写法是：**模块名.model_模型名**

- 如果该模型就在当前模块，则模块名可以省略；
- 模型名中间的点号要换成下划线；
- 如果写错，则update后会出现找不到模型的报错；

上例中为：**model_ccm_into_apply**，也可以写成 **ccm.model_ccm_into_apply**

- **ccm** 是模块名，因为在当前模块，则可以省略；
- **model** 固定；
- **ccm_into_apply** 为模型名，点号换成了下划线；

group_id:id

权限组的id，标准写法是：**模块名.权限组id**

- 模块名不能省略，即使是当前模块；

- 权限组即是xml中使用 **res.groups** 配置的权限组；

上例中为：**ccm.group_into_apply_menu_r**

- **ccm** 是模块名，不能忽略；
- **group_into_apply_menu_r** 是用户权限组的id，在xml中找到；

perm_read/perm_write/perm_create/perm_unlink

分别表示读取/修改/创建/删除的权限，1表示有权限，0表示无权限

4、字段权限控制

加载 **视图的field** 或者 **模型的fields**上，可以控制查看字段的权限，只有在权限groups内的用户才能看到这些字段

也可以加在 **button** 上，保证按钮针对权限groups的权限

示例：

```
<button name="to_submit" type="object"
 string="提交" attrs="{'invisible':[('state','!=', 1)]}"
 groups="ccm.group_into_apply_menu_r,ccm.group_into_apply_menu_w"/>
```

```
<field name="account_id" groups="ccm.group_into_apply_menu_r"/>
```

```
desc = fields.Text(string="描述", groups="ccm.group_into_apply_menu_r")
```

- 使用 **groups** 添加权限组的id即可，多个id使用逗号隔开；
- 格式为：**模块名.权限组id**；
- 加在视图中，则只最特定视图有效；加在数据模型中，则对使用到的所有地方有效；