# odoo中self的使用

2018年04月19日

## 一：self是什么

目前新版的Odoo中使用到的self，是对 游标cr、用户ID、模型、上下文、记录集、缓存 的封装。

我们可以通过 self.XX 获取到这些封装的东西，比如：self.cr、self.uid。

## 二：通过直接设置属性来改变数据库中字段值

我们在查出某模型的记录后，可以通过 record.XX = value 来直接修改记录的字段内容。

同样，在重写模型的write方法中，也可以通过 self.XX = value 来指定新增记录中某字段的值。

这里需要注意两点：

1:修改查出来的记录字段值来改变数据库内容，是通过改变缓存中的值出发数据库写记录来达到的。

2:重写write方法时，在write方法中每调用一次 self.XX = value 语句，都会触发数据库写操作，因此一般采用:

for rec in self:

rec.XX = XX

的写法。

## 三：环境

1:操作缓存

环境储存了模型的缓存记录集，因此我们可以通过环境来获取、增加、修改、删除记录，而触发数据库更改，从而达到操作数据库的目的。

例如：新增一条记录

self.env['模型'].create(vals)

2:改变用户权限

我们可以通过self.sudo()获得超级权限，从而确保我们的操作能够进行。

self.env[‘model'].sudo().create(vals)

3：访问当前用户

self.env.user

4：获取XML的ID

self.env.ref('external id')

5：更新缓存，触发数据库操作

self.env.invalidate\_all()

## 四：self常用接口

1:普通查询：返回记录集，后续通过修改记录值来触发数据库修改

self.search(domain) //从当前模型的self中查询

self.env['model'].search(domain) // 获取某个model的环境，查询其中的记录集

2:只读查询：返回列表，只能提取内容，不能触发数据库修改

self.search\_read([],['要查询的字段'])

3:统计数量：返回符合条件的记录条数

self.search\_count(domain)

4:浏览：通过一系列id值，返回对应的记录集

self.browse([id]

5:删除

self.unlink(domain)

## 五：NEW ids

Odoo在创建一个新记录时，会使用models.ids虚拟一个记录id。

可以通过

if is instance(record.id,models.NewId)：

来判断。

## 六、数据库查询

import psycopg2

class XXXXX(models.Model):

........

@api.multi

def OOOO(self):

db = psycopg2.connect("dbname=test4 user=postgres")*#查找名叫 test4 的数据库 postgres 是数据库的超级用户名称*

vals = db.cursor()

vals.execute("SELECT sales,prices,sale\_date FROM run\_chart")*#执行sql语句查询数据*

tables = vals.fetchall()*#返回查询结果*

*#或者*

sql = "select \*from 表名"

self.env.cr.execute(sql) *#执行SQL语句*

dicts = self.env.cr.dictfetchall() *#获取SQL的查询结果*

转载来自：<http://www.cnblogs.com/zcy1103/p/8483646.html>