# Odoo-新API的介绍与应用

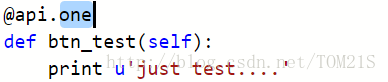
2017年02月09日

[one](https://blog.csdn.net/TOM21S/article/details/54959719" \l "one)

* [multi](https://blog.csdn.net/TOM21S/article/details/54959719" \l "multi)
* [model](https://blog.csdn.net/TOM21S/article/details/54959719" \l "model)
* [contrains](https://blog.csdn.net/TOM21S/article/details/54959719" \l "contrains)
* [depends](https://blog.csdn.net/TOM21S/article/details/54959719" \l "depends)
* [onchange](https://blog.csdn.net/TOM21S/article/details/54959719" \l "onchange)
* [returns](https://blog.csdn.net/TOM21S/article/details/54959719" \l "returns)

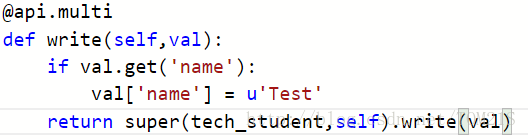
odoo 新api的实现是借助于python装饰器。新API的装饰器主要有以下几种：   
model,multi,one,constrains,depends,onchange,returns   
还有一些像   
cr,cr\_context,cr\_uid,cr\_uid\_context,cr\_uid\_id,cr\_uid\_id\_contet,cr\_uid\_ids,cr\_uid\_ids\_context这些主要用于一些特定的方法。

# **one**

one的用法主要用于self为单一集合的情况，与之相对应的是multi。被one装饰的方法会返回一个list（某些误人子弟的教程说one不返回值的，实际是返回了[None]），对于v7版本中类似于 def funct(self,cr,uid,ids,context)的方法可以改用api.one装饰器这种V8写法。   
例子： 

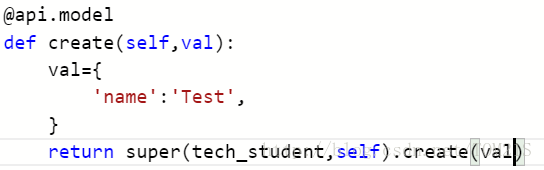
# **multi**

multi则要求self是多个记录的合集。



# **model**

model要求self是一个记录集，env中并不包含ids参数，即适用于v7中类似于 def funct(self,cr,uid,args,context)的方法。

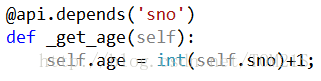


# **contrains**

已经在上篇文章中介绍过，略过不讲。

# **depends**

depends 主要用于compute方法，v8当中已经取消了function字段，对于任何fields都可以通过添加compute属性动态赋值。depends就是用来标该方法依赖于哪些字段的装饰。   
例如我们让学员你的年龄等于学员的编号+1（非常扯淡的逻辑，只做演示）：



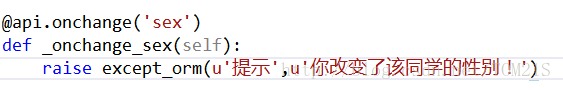


事实上，不一定非要使用depends装饰，去掉depends装饰后，以上的代码仍然work，只不过运行的时机有了改变，这是为什么呢？请各位同学看官自行体会。

插曲：我们知道7.0中function字段是默认不存储的，需要使用store参数进行存储。v8当中所有带有compute参数的字段默认都不会存储，store参数也变成了boolean类型，不再提供store的触发函数。这里使用depends的字段就相当于v7当中的trigger，当依赖的字段发生改变时，会触发这里的函数对数据库进行更新。但如果depends依赖的字段也没有存储，那么依旧不会触发。   
对于存储的目的是用来搜索的童鞋来说，我们提供另外一种实现方式，将在后续的文章中提到，敬请期待。

# **onchange**

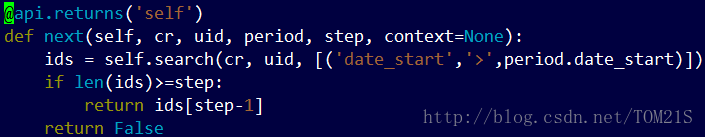
onchange的使用方法非常简单，就是当字段发生改变时，触发绑定的函数。

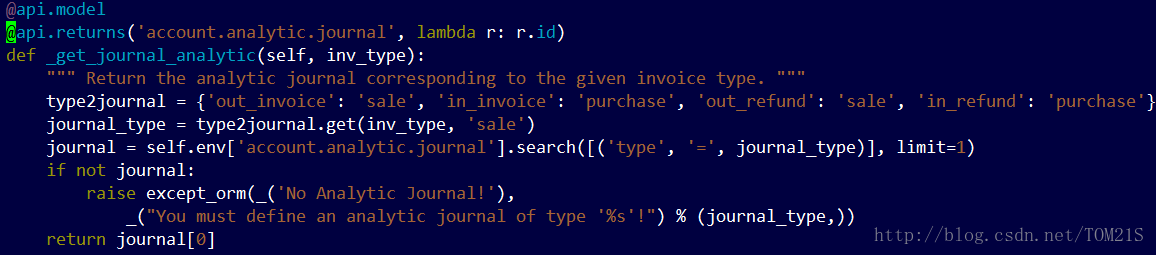




# **returns**

returns的用法主要是用来指定返回值的格式，它接受三个参数，第一个为返回值的model，第二个为向下兼容的method，第三个为向上兼容的method   
第一个参数如果是对象本身，以’self’,如果是其他对象，则写其他对象名如：@api.returns(‘ir.ui.view’)   
例子：





注意看第二个例子，如果调用该方法的是一个v7版本的方法，返回值将进行向下兼容的转化，即通过第二个参数的lamda表达式返回该记录的id。

本文是对网上资料进行了整理引用了以下网站的内容：   
<http://www.cnblogs.com/kfx2007/p/6093994.html>