# 不污染主干代码改变odoo行为

By 肖羊

要改变odoo行为，比如修复bug，最简单的方式是直接修改代码，但是这种方式会污染odoo主干代码，最直接的结果就是很难与社区版本同步更新，每次合并代码都需要做大量比较和测试。所以如果可以以开发组件的形式去打补丁，不管是修复bug还是功能增强，都比直接修改代码更靠谱，因为对主干代码改动越大，将来升级时我们的工作量就越大，甚至当改动大到一定程度时我们将不得不放弃社区版本而维护自己的版本。当这一天到来的时候，odoo开源带给我们的好处将不复存在。

我们要修改odoo的行为不外乎有两种情况，第一种是修复odoo模块的bug，第二种是扩展odoo模块的功能。尤其是修复bug，在修复bug时遇到最头痛的问题就是污染主干导致版本更新困难。我自己总结了如下几种场景下可能会需要改变odoo行为：

1. **某个模块对象的某个方法有bug**
2. **改变odoo web行为或者修复odoo js bug（website和管理页面）**
3. 基于1和2，由于python和js都是弱类型语言，有可能方法调用中的返回类型与预期的不符合，运行时会出现错误提示，出现这种问题时如果是自己的代码，可能加入一两行判断就可以了，但是想要不修改odoo官方的代码就需要另辟蹊径。
4. **改变某个url链接的行为**，这种情况主要是在website中出现的很多，因为website完全使用页面跳转，每个页面就是一个url地址。而后端的管理页面绝大多数都是使用ajax请求，相对来说url比较少，也比较固定。

针对上面这几种情况，我总结了如下几种方法：

1. 继承模块对象：在python对象中使用\_inherit字段，并且不定义”\_name”或者”\_name”字段与原来的类相同，这样的话你定义的对象就会覆盖对象池registry中的对象，并且继承原有对象的所有特性，需要修复哪个方法的bug，只需要覆盖该方法就可以了。这种方式尤其适合修复某个非基础组件的bug，同样也适合对非基础组件功能进行扩展。因为这种组件多数情况下没有子类，只需要重写有bug的方法就可以实现。当组件有别的组件继承它的时候，情况会复杂很多，尤其是当其子类中有super调用的时候。这个和odoo的继承链以及对象的加载顺序有关，在此就不细讲了。

举个例子，website的ir\_http就是上述情况，它继承并覆盖了base中的ir\_http，并且它的\_dispatch方法中调用倒了父类的\_dispatch方法。假如说base.ir\_http的\_dispatch方法有bug或者我们想扩展它。当我们直接使用”\_inherit”字段来继承 base.ir\_http就有可能出问题，因为我们并不能确定究竟是我们自己的ir\_http先加载还是website.ir\_http先加载（和数据库中的顺序有关），有一个折衷的做法是我们的模块同时也依赖于website模块，直接改变website.ir\_http的行为（被依赖的模块会先加载）。但是这样带来的问题是：即使用户不想使用website模块，我们的模块也会给他强制安装上website。

1. 对于后台页面js，重写相应模块js对象的 prototype方法：后台页面的js代码默认是会全部加载的（前提是你在\_\_openerp\_\_.py中已经引入了相应的xml文件）。一般情况下，对于web展示，每一个组件都有一个（或者多个）js对象，由于这些对象是在页面初始化完成之后才创建，所以我们完全可以在对象创建之前改变它的结构或者重写它的方法。为什么要重写prototype呢？因为几乎所有的js模型对象都是使用new关键字创建的（基础对象是采用花括号的方式创建并定义，相当于是单例），新生成的对象会以prototype作为模板创建，所以必须重写prototype，并且重写的prototype必须在原对象定义之后，否则会被原对象的prototype所覆盖，达不到想要的效果。

当然实际使用的时候远没有上面写的这么简单，也不止这一种方法，这主要取决于我们对原有代码修改程度有多大，上面写的只是一种思路。

1. 对于url请求，继承相应的controller：这种情况是比较容易实现的，直接继承相应的controller，覆盖你想修改的方法和route（类名可以和父类不相同），记住，**两个都需要覆盖**。

具体例子可以参考website包下的controller Website，website就是继承了web包下的Home类并且它覆盖了index方法，并且该方法的route指向网站的根”/”。

1. 对于website页面js：其实改变url的行为基本上已经可以覆盖到本条，因为你完全可以自己重新定义这条url地址返回的页面，返回的内容。但是有时候我们可能只想改变原有页面很小的一部分内容，比如在原有的js依赖中增加一个自己的js文件。这样的话就可以利用第2条方法去局部修改js对象的行为。但是现在还没有找到一个比较完善的方法。等如果整理出比较完善可行的方法时我再更新这篇文档。