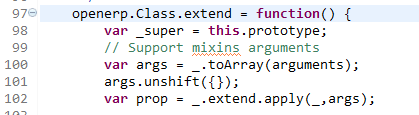
# odoo的js继承机制解读

前文说过，odoo前端js中openerp这个对象是所有对象的根，其他的对象都是由openerp这个对象继承而来。下面我们解读下他的js机制。

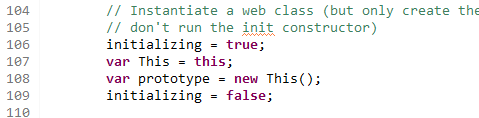
odoo实现继承的代码在openerpframework.js第97行-188行这段代码中定义。

**注意：**继承这部分代码中有两个Class，一个是openerp.Class，一个是extend中定义的Class，openerp.Class的作用是在extend中调用时生成一个空对象实例，然后把参数中定义的方法填充到这个对象中，并且这个对象会赋给extend中定义的Class的prototype。

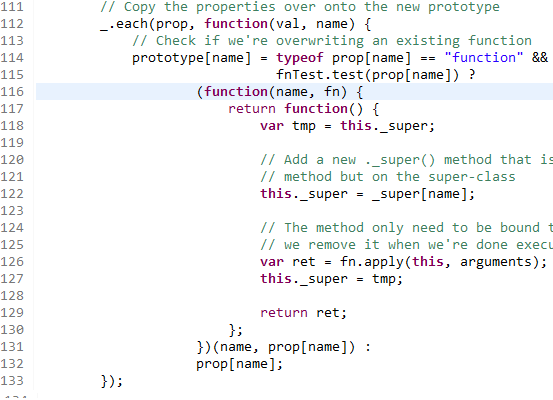
opener.Class.extend方法会返回一个Class的方法定义，函数开始这几行是处理参数，因为extend()方法允许传递多个对象作为参数，所以需要先把这些对象里面定义的方法重组成一个Array。



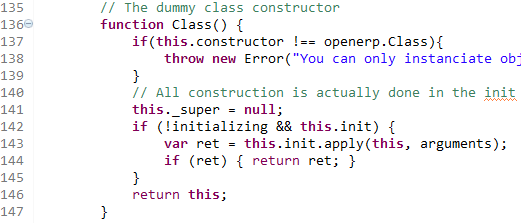
接下来这部分代码的作用是new一个新的openerp.Class对象，这个空对象会被填充参数中定义的方法。



遍历参数中的每个方法定义，添加到刚生成的prototype对象中去：



这个Class是作为extend方法的返回值，调用extend方法时返回的对象就是这个Class，

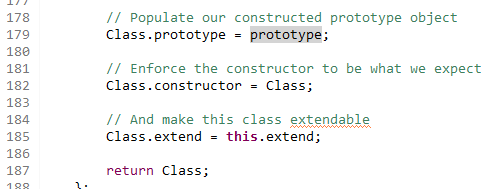


Class.include的作用是把参数中定义的函数添加到Class中去，可以实现对该对象的扩展。



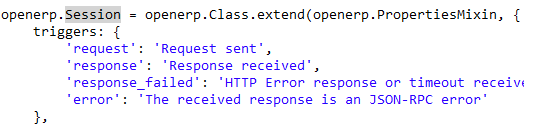
把刚才new出来的openerp.Class作为extend方法内定义的Class的prototype，也就是说在使用new生成对象实例时，新生成的对象实例就具备了参数中定义的所有方法。

把openerp.Class的extend方法赋给Class的extend，使新定义的Class也可以实现继承。



下面看一个具体的继承例子：

1、首先调用extend方法，此时方法返回一个Class的方法定义，也就是说Session对象是一个方法定义，并非一个对象实例，在该方法定义中，openerp.PropertiesMinxin，triggers这些参数中定义的对象会被添加到Class的prototype中去。



2、new一个新的Session对象实例

C:\Users\xiaoyang123\Desktop\QQ截图20140808155440.png