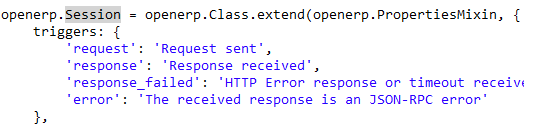
上周发过一篇《odoo的js继承机制解读》，简单介绍了odoo如何实现继承，由于那时候对odoo的js源码了解不够深，这篇文章将对openerp继承代码进行逐行解读，希望对大家有所帮助。

第一部分 例子

在开始之前我们先看一个继承的具体例子

1、首先调用extend方法，此时方法返回一个Class的方法定义，这个Class是extend方法内部定义的一个内部方法。



从上面这段调用可以看出，extend方法需要多个对象作为参数（如openerp.PropertiesMixin），按照我们对继承的理解和猜测，Session应该包含这多个对象里面定义的所有变量和方法。同时openerp.Session应该等于openerp.Class.extend()的返回值，那么**这个返回值是什么类型的？**

2、new一个新的Session对象实例

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

第二步很简单，就是一个简单的new，这说明继承的动作是在extend()方法里完成的，那么extend方法返回的对象里应该就已经包含了父类的所有方法和变量，**这又是怎么做到的呢？**

到现在为止，我们已经有两个疑问了：

extend()返回的是什么类型？

extend()如何做到把父类的属性添加到返回值中？

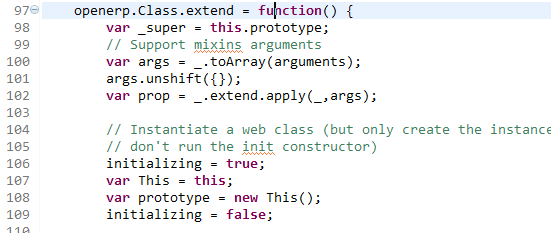
带着疑问，我们将通过逐行阅读源码的方式揭开真相。

第二部分 源码

在代码90行定义了一个空方法，该方法什么都不会做，但是以后所有的扩展都扩展自Class。

C:\Users\xiaoyang123\Desktop\QQ截图20140815163035.png

下面是实现继承代码部分



98行定义了一个\_super变量，

100行定义了变量args，需要注意的是\_.toArray(arguments)中的下划线(\_)是开源js库underscore的一个变量，有兴趣的童鞋可以阅读下它的源码，这个开源库提供了很多工具方法，并且为早期浏览器提供了很多很多兼容方法，如ECMAScript 5中提供的新方法foreach。回到正题，args其实就是参数列表。

101行在参数数组之前插入一个新的空对象{}

102行是重点，它调用了underscore的extend方法，该方法的作用是把从第二个参数到最后一个参数里的每一个方法复制到第一个参数中去，并且把第一个参数作为返回值。这也就是为什么在上一行中插入一个空对象，也就是说prop其实就是上一行中插入的空对象，只不过这个对象在返回的时候已经有了很多很多方法了。现在已经有点继承的赶脚了。