1. 集合中的initialize方法是在填入数据前执行的

2. 通过引用类型实现时间传递

注意\*\*\*\*\*\*\*// 通过对象引用传递事件

roomView.detachEvent = self.detachEvent;

即当前view引用父view的方法，触发当前view的方法时就会触发父view的方法了

如：

var self = this;

this.render();

// 遍历房间集合追加

\_.each(this.collection.models, function(M, i){

// roomModel 缺省的属性 bed, breakfast, net, price, discount,max取roomPlan集合中的第一个占位

if(M.get('plans').length){

var firstRoomPlan = M.get('plans')[0];

M.set({'bed': firstRoomPlan.bed,

'breakfast': firstRoomPlan.breakfast,

'net': firstRoomPlan.net,

'price': firstRoomPlan.price,

'discount': firstRoomPlan.discount,

'max': firstRoomPlan.max

});

}

var roomView = new RoomView({model: M, collection: new RoomPlanCollection(M.get('plans'))});

注意\*\*\*\*\*\*\*// 通过对象引用传递事件

roomView.detachEvent = self.detachEvent;

// 此参数暂写死 tpl 里面统一一个类

self.el.find('.J-hotelRoomContain').append(roomView.el);

3. 通过引用类型实现时间传递附加

通过触发子view的方法可以触发父view的方法，但是这个方法最后面一直是由父view调用的，所以

这个方法中的this永远指向于父view对象（就是当前父view对象）哈哈哈，通过这个我们可以拿到

父view对应的model,即实现的子view获取父view对应的model的所有属性；

联想，能不能通过这种方式从子view获取祖父view的model，通过实验其实

**3. this**. **el** = $(String(**this**.temp(**window**.***routePath***)(**this**.**model**.attributes)));

时events中的事件没有作用

但是改为**this**.**$el** = $(String(**this**.temp(**window**.***routePath***)(**this**.**model**.attributes)));

之后就好了

4. 为什么有时候用view.$el调用事件是好的，而用view.el不行；

原因是：el没有渲染到页面，而$el本来就是一个缓存的对象，所以在el没有加载完的时候他是可以调用方法的，这就是原因。解决办法就是\_.defer

5. **this**.**collection**.on(**'reset'**, **this**.render, **this**);如果在这个后面不加上下午问限制的话，则在render中的this指向的是collection而不是他的view

6.首屏渲染然后把一个ele作为某一个view对象的ele属性时总是不对，做的很烂。最后发现原来是这里出错了，因为我在实例化的时候指向了一个现成的ele，然后在他的initialize中又用了这一句：this.el = $(temple(this.model.attributes))

所以把实例化时候传进来的ele覆盖掉了。

解决办法很简单：如下

*/\*  
 避免重复渲染  
 此处做判断，区别el是在实例化的时候传进来，还是在此处生成  
 \*/***var** virtualEle = **this**.**el**;  
**if**(virtualEle.**nodeType** === 1 && virtualEle.**childNodes**.**length** === 0 && virtualEle.**tagName** === **'DIV'**){  
 **this**.**el** = $(String(scenicTicketTemple(**this**.**model**.attributes)));  
}