Model

数据的表现形式。可以创建、校验、销毁、保存到服务端。

当UI操作（比如点击按钮）导致model变更，model会“触发”一个“change”事件，使用这个Model展示的所有View都会接收到该事件并做出响应（重新渲染）。

View

数据展现形式，如表格、文本框等。

backbone语法 参数是非严格形式的json，key可以不带双引号

Backbone.Events 事件，可以混入任何Object，使该Object拥有绑定和触发自定义方法的能力。

on object.on(event,callback,[context])

示例：object.on(“alert”,function(msg){alert(msg);})

给object绑定alert事件。event参数可以使用“:”配置命名空间，如“change:select”，也可以同时绑定多个事件，用空格符分开，如“change:select change:author”。

context上下文，如 object.on(“change”,o.render,o)，o是另一个对象。

off object.off([event],[callback],[context]) 跟on相反，事件解绑

如果没有指定context，就解除所有event和callback相同的事件绑定；如果没有指定callback，就解除所有event相同的事件绑定；如果没有指定event，就解除所有事件绑定。

trigger 触发事件 object.trigger(event,[\*args])

Backbone.Model 模型，提供一系列基本方法。我们可以通过继承模型来获得这些能力。

非常重要：Model的属性，是通过get和set方法设置的，内部用一个hash来维护。model.get(attr)和model.attr是两码事。

extend Backbone.Model.extend(properties,[classProperties])

自动设置原型链，以便子类继承。

调用父类方法(super)：

var Note = Backbone.Model.extend({

set:function(attributes,options){

Backbone.Model.prototype.set.call(this,attributes,options);

…

}

});

initialize 相当于Constructor 构造方法，new一个实例的时候会调用该方法。

new Book({“name”:”三国演义”})，这里的参数是通过调用set方法设置的，但不会触发事件。

get get(attribute) 略

set set(attributes,[options])

get和set明显是一对。set的第一个参数和jQuery里的css方法一样，可同时设置多个参数。第二个参数用于控制某些行为，比如传入{silent:true}，则不会触发相应的change事件。另外，如果该model拥有validate方法，则会先执行该方法，如不通过则不执行set方法。

escape 和get类似，不过获取的attribute值经过了转义处理，以避免xss攻击。

has model.has(attr) 如果attr是一个非null并且非undefined的值，返回true，否则返回false

unset model.unset(attr,[options])

从属性hash表里删除掉attr属性，并根据options设置决定是否触发相应change事件。

clear model.clear([options])

删除model的所有属性，并根据options设置决定是否触发相应change事件。

id model.id

id是一个特别属性，可以是字符串、整数或者uuid。如果使用set方法设置了id，则会将该值覆盖到model.id。Model可以通过id从collection中获取，并且id默认用于生成Model的URL。

idAttribute model.idAttribute

对于某些