## Backbone的 listenTo 和 on

## 使用情景区别

listenTo用于监听自身意外的对象

on用于监听自身

## listenTo和on中的回调函数里的this的区别

[复制代码](javascript:void(0);)

listener.listenTo(object, 'eventName', function(){

//此处的this指向listener

})

object.on('eventName', function(){

//此处的this指向object

})

object.on('eventName', function(){

//此处的this指向context

}, context)

[复制代码](javascript:void(0);)

## 触发顺序

当存在同名事件时,会按定义顺序执行,on和listenTo定义的事件会处在同一队列中

若监听了all事件,则触发任意事件都将会在执行完相应回调后执行all事件的回调,  all事件的监听会传递一个参数,该参数为事件名字

## demo

[复制代码](javascript:void(0);)

1 var a = {type: "I'm listening"}, b = {type: "I'm listened"};

2 \_.extend(a, Backbone.Events);

3 \_.extend(b, Backbone.Events);

4 b.on('onNoContext', function(){

5 console.log('trigger onNoContext');

6 console.log(this.type);

7 });

8 b.on('onContext', function(){

9 console.log('trigger onContext');

10 console.log(this.type);

11 }, a);

12 a.listenTo(b, 'listened', function(){

13 console.log('trigger listened');

14 console.log(this.type);

15 });

16 b.on('all', function(){

17 console.log('trigger on all');

18 });

19 a.listenTo(b,'all', function(){

20 console.log('trigger listen all');

21 });

22

23 console.log('###b.trigger("all")###');

24 b.trigger('all');

25 console.log('###b.trigger("onNoContext")###');

26 b.trigger('onNoContext');

27 console.log('###b.trigger("onContext")###');

28 b.trigger('onContext');

29 console.log('###b.trigger("listened")###');

30 b.trigger('listened');

[复制代码](javascript:void(0);)

运行结果:

[复制代码](javascript:void(0);)

1 ###b.trigger("all")###

2 trigger on all

3 trigger listen all

4 trigger on all

5 trigger listen all

6 ###b.trigger("onNoContext")###

7 trigger onNoContext

8 I'm listened

9 trigger on all

10 trigger listen all

11 ###b.trigger("onContext")###

12 trigger onContext

13 I'm listening

14 trigger on all

15 trigger listen all

16 ###b.trigger("listened")###

17 trigger listened

18 I'm listening

19 trigger on all

20 trigger listen all

[复制代码](javascript:void(0);)