**angularjs的事件 $broadcast and $emit and $on**

**Event 事件**

事件是解耦良器，angularjs提供了很方便的事件机构。 发送事件可以用

$scope.$emit('name', 'args');

或者是

$scope.$broadcast('name', 'args');

要了解两者的差别， 首先要了解angularjs controller的scope的定义。 这里就不叙述了， 简单来说 angularjs controller的scope就是一个像是js的基于prototye的对象继承。

看看下面这个例子.

首先我们定义几个嵌套的controller。 （木有嵌套， 基本上也不会用事件， 当然事件也可 有$rootscope发出， 但这个首先慎用， 其次也是下面的一个特例）

<div ng-controller="ParentCtrl">

<div ng-controller="SelfCtrl">

<a class="btn" ng-click="click()">click me</a>

<div ng-controller="ChildCtrl"></div>

</div>

<div ng-controller="BroCtrl"></div>

</div>

这里我们有四个controller, 层级关系如下：

ParentCtrl

-> SelfCtrl (\*)

-> ChildCtrl

-> BroCtrl

所有的事件都是由 SelfCtrl 发出去的.

**broadcast**

事件发送的方法：

$scope.$broadcast

值得注意的是发送的主语是 $scope， 因为所有的事件其实都是作用在scope上的。

broadcast 是从发送者向他的子scope发送时间的一个方法。 这里就是SelfCtrl发送， SelfCtrl 和 ChildCtrl 会接受到, *而ParentCtrl是不会收到的*

事件的接受只有一个方法

$scope.$on

例子如下

angular.module('TestApp', ['ng'])

.controller('ParentCtrl', function($scope) {

$scope.$on('to-child', function(e, d) {

console.log('关我毛事');

});

})

.controller('SelfCtrl', function($scope) {

$scope.click = function () {

$scope.$broadcast('to-child', 'haha');

}

})

.controller('ChildCtrl', function($scope){

$scope.$on('to-child', function(e, d) {

console.log('I\' the child, I got it', d);

});

})

.controller('BroCtrl', function($scope){

$scope.$on('to-child', function(e, d) {

console.log('关我毛事');

});

})

;

点击clickme后， 在console里是看不到“关我毛事“的， 原因嘛就是 “管他毛事啊”

**Emit**

了解了broadcast之后， emit可以用以此类推解释了。

angular.module('TestApp', ['ng'])

.controller('ParentCtrl', function($scope) {

$scope.$on('to-parent', function(e, d) {

console.log('we are the parent, I got it', d);

});

})

.controller('SelfCtrl', function($scope) {

$scope.click = function () {

$scope.$emit('to-parent', 'hehe');

}

})

.controller('ChildCtrl', function($scope){

$scope.$on('to-parent', function(e, d) {

console.log('关我毛事');

});

})

.controller('BroCtrl', function($scope){

$scope.$on('to-parent', function(e, d) {

console.log('关我毛事');

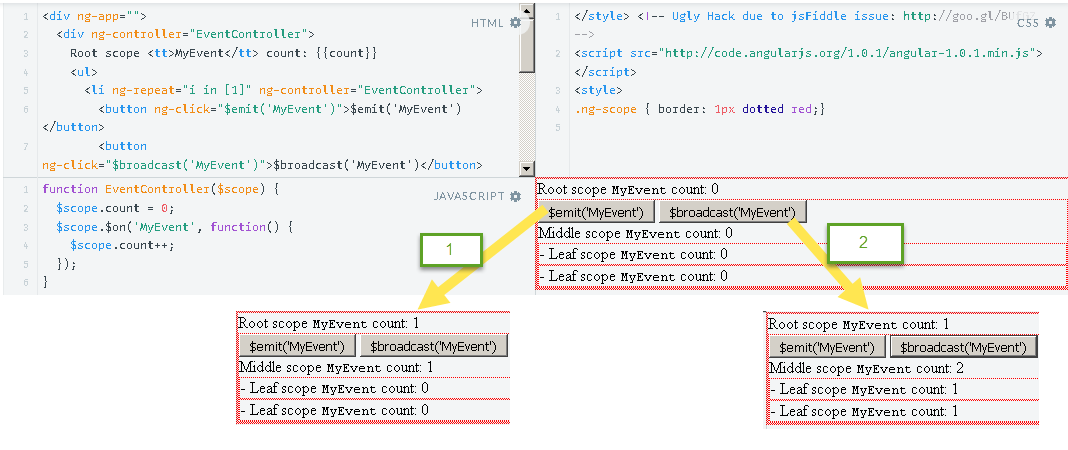
});

})

;

**Notes**

上面的例子可以看到， 事件和事件发生者的兄弟是没有关系的， 怎样都收不到。



|  |
| --- |
| <div ng-app="">  <div ng-controller="EventController">  Root scope <tt>MyEvent</tt> count: {{count}}  <ul>  <li ng-repeat="i in [1]" ng-controller="EventController">  <button ng-click="$emit('MyEvent')">$emit('MyEvent')</button>  <button ng-click="$broadcast('MyEvent')">$broadcast('MyEvent')</button>  <br>  Middle scope <tt>MyEvent</tt> count: {{count}}  <ul>  <li ng-repeat="item in [1, 2]" ng-controller="EventController">  - Leaf scope <tt>MyEvent</tt> count: {{count}}  </li>  </ul>  </li>  </ul>  </div>  </div> |
| function EventController($scope) {  $scope.count = 0;  $scope.$on('MyEvent', function() {  $scope.count++;  });  } |
| </style> <!-- Ugly Hack due to jsFiddle issue: http://goo.gl/BUfGZ -->  <script src="http://code.angularjs.org/1.0.1/angular-1.0.1.min.js"></script>  <style>  .ng-scope { border: 1px dotted red;} |