**自定义指令中使用ngRepeat在ngReapeat渲染完毕以后执行代码**

业务中有时需要在异步获取数据并用ng-repeat遍历渲染完页面后执行某个操作，angular本身并没有提供监听ng-repeat渲染完成的指令，所以需要自己动手写。有经验的同学都应该知道，在ng-repeat模板实例内部会暴露出一些特殊属性$index/$first/$middle/$last/$odd/$even，$index会随着每次遍历（从0开始）递增，当遍历到最后一个时，$last的值为true，so，通过判断$last的值来监听ng-repeat的执行状态，怎么在遍历过程中拿到$last的值：自定义指令

小实例，我只写了最重要的部分

[复制代码](javascript:void(0);)

//要循环的数据

$scope.data = [

{

str: 'a'

},

{

str: 'b'

},

{

str: 'c'

}

]

//自定义指令repeatFinish

app.directive('repeatFinish',function(){

return {

link: function(scope,element,attr){

console.log(scope.$index)

if(scope.$last == true){

console.log('ng-repeat执行完毕')

}

}

}

})

<div id="box">

<span ng-repeat="item in data" repeat-finish>{{item.str}}</span>

</div>

[复制代码](javascript:void(0);)

打开控制台，会打印出0，1，2，当$index = 2点时候，$last值为true，ng-repeat渲染完毕

so easy!

当然指令最好是能够复用，在这个指令内写具体的业务逻辑不利于复用，可以通过给指令指定一个处理函数renderFinish

<div id="box">

<span ng-repeat="item in data" repeat-finish＝"renderFinish()">{{item.str}}</span>

</div>

再通过指令的attr参数获取这个处理函数

[复制代码](javascript:void(0);)

app.directive('repeatFinish',function(){

return {

link: function(scope,element,attr){

console.log(scope.$index)

if(scope.$last == true){

console.log('ng-repeat执行完毕')

scope.$eval( attr.repeatFinish )

}

}

}

})

//controller里对应的处理函数

$scope.renderFinish = function(){

console.log('渲染完之后的操作')

}

[复制代码](javascript:void(0);)

attr获取到的属性只是一个字符串表达式，$scope.$eval方法是专门执行AngularJS表达式的，通过它处理函数得以执行，这样，指令用在不同的地方，可传递不同的处理函数。

有些业务比较复杂，可能ng-repeat渲染完成之后，需要执行多个操作并且这多个操作有多个前端完成，需要用到angular的事件，在repeatFinish指令的link函数内触发一个事件，各位前端同学监听该事件完成各自的操作

[复制代码](javascript:void(0);)

app.directive('repeatFinish',function(){

return {

link: function(scope,element,attr){

console.log(scope.$index)

if(scope.$last == true){

console.log('ng-repeat执行完毕')

//向父控制器传递事件

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

//向子控制器传递事件

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

}

}

}

})

//父控制器中监听事件

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

//父控制器执行操作

})

//子控制器中监听事件

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

//子控制器执行操作

})

[复制代码](javascript:void(0);)

如何在当前控制器下监听到该事件呢？angular没有向当前控制器传递事件的方法，可以先向父（子）控制器传递事件，父（子）控制器监听到事件后反过来向子（父）控制器传递事件。

补充：IE8可以直接在元素上用指令ng-if="$last && renderFinish()"，当然IE8+也可以这样用

一句话总结：指令是angular的核心功能之一，用好了事半功倍，监听ng-repeat执行状态仅仅是它功能的冰山一角吧