**$digest和$apply**

在Angular中，有$apply和$digest两个函数，我们刚才是通过$digest来让这个数据应用到界面上。但这个时候，也可以不用$digest，而是使用$apply，效果是一样的，那么，它们的差异是什么呢？

最直接的差异是，$apply可以带参数，它可以接受一个函数，然后在应用数据之后，调用这个函数。除此之外，还有别的区别吗？

在简单的数据模型中，这两者没有本质差别，但是当有层次结构的时候，就不一样了。考虑到有两层作用域，我们可以在父作用域上调用这两个函数，也可以在子作用域上调用，这个时候就能看到差别了。

对于$digest来说，在父作用域和子作用域上调用是有差别的，但是，对于$apply来说，这两者一样。我们来构造一个特殊的示例：

[复制代码](javascript:void(0);)

var app = angular.module("test", []);

app.directive("increasea", function() {

return function (scope, element, attr) {

element.on("click", function() {

scope.a++;

scope.$digest();

});

};

});

app.directive("increaseb", function() {

return function (scope, element, attr) {

element.on("click", function() {

scope.b++;

scope.$digest(); //这个换成$apply即可

});

};

});

app.controller("OuterCtrl", ["$scope", function($scope) {

$scope.a = 1;

$scope.$watch("a", function(newVal) {

console.log("a:" + newVal);

});

$scope.$on("test", function(evt) {

$scope.a++;

});

}]);

app.controller("InnerCtrl", ["$scope", function($scope) {

$scope.b = 2;

$scope.$watch("b", function(newVal) {

console.log("b:" + newVal);

$scope.$emit("test", newVal);

});

}]);

<div ng-app="test">

<div ng-controller="OuterCtrl">

<div ng-controller="InnerCtrl">

<button increaseb>increase b</button>

<span ng-bind="b"></span>

</div>

<button increasea>increase a</button>

<span ng-bind="a"></span>

</div>

</div>

[复制代码](javascript:void(0);)

这时候，我们就能看出差别了，在increase b按钮上点击，这时候，a跟b的值其实都已经变化了，但是界面上的a没有更新，直到点击一次increase a，这时候刚才对a的累加才会一次更新上来。怎么解决这个问题呢？只需在increaseb这个指令的实现中，把$digest换成$apply即可。

当调用$digest的时候，只触发当前作用域和它的子作用域上的监控，但是当调用$apply的时候，会触发作用域树上的所有监控。

因此，从性能上讲，如果能确定自己作的这个数据变更所造成的影响范围，应当尽量调用$digest，只有当无法精确知道数据变更造成的影响范围时，才去用$apply，很暴力地遍历整个作用域树，调用其中所有的监控。

从另外一个角度，我们也可以看到，为什么调用外部框架的时候，是推荐放在$apply中，因为只有这个地方才是对所有数据变更都应用的地方，如果用$digest，有可能临时丢失数据变更。

转自： https://github.com/xufei/blog/issues/10