

dm_vincent的专栏

记录生活和学习的点滴

目录视图

摘要视图

RSS 订阅

个人资料



dm_vincent



访问： 309879次

积分： 4358

等级： **BLOG > 5**

排名： 第3127名

原创： 96篇 转载： 1篇

译文： 41篇 评论： 100条

文章搜索

文章分类

- Java 8 (14)
- Algorithm (7)
- Java (36)
- Maven (2)
- TestNG (4)
- AngularJS (3)
- jQuery (1)
- JavaScript (44)
- Java Performance (9)
- Lambda (13)
- Spring (1)
- Hibernate Search (6)
- Concurrency (9)
- Elasticsearch (40)
- Search (40)
- JSON (1)
- Xuggler (0)
- PHP (3)
- LAMP (3)

CSDN博乐 举荐之美 博主线下趴：程序人生，不止一面 【面向专家】极客头条使用体验征文 [张传波]活用UML—打造软件设计高手

理解Angular中的\$apply()以及\$digest()

分类： AngularJS JavaScript

2014-08-20 13:49

8822人阅读

评论(1)

收藏

举报

angularjs

javascript

angular

\$apply()和\$digest()在AngularJS中是两个核心概念，但是有时候它们又让人困惑。而为了工作方式，首先需要了解\$apply()和\$digest()是如何工作的。这篇文章旨在解释\$apply()和\$digest()及在日常的编码中如何应用它们。

探索\$apply()和\$digest()

AngularJS提供了一个非常酷的特性叫做双向数据绑定(Two-way Data Binding)，这个特性改变了代码编写方式。数据绑定意味着当View中有任何数据发生了变化，那么这个变化也会反映到数据上，也即意味着scope模型会自动地更新。类似地，当scope模型发生变化时，view会更新到最新的值。那么AngularJS是如何做到这一点的呢？当你写下表达式如{{ aModel }}时，AngularJS会在你在scope模型上设置一个watcher，它用来在数据发生变化时更新view。这里我们会看到在AngularJS中设置的watcher是一样的：

```
[javascript] view plain copy print ?
01. $scope.$watch('aModel', function(newValue, oldValue) {
02.     //update the DOM with newValue
03. });
```

传入到\$watch()中的第二个参数是一个回调函数，该函数在aModel的值发生变化的时候会被调用。当aModel发生变化的时候，这个回调函数会被调用来更新view这一点不难理解，但是有一个重要的问题！AngularJS是如何知道什么时候要调用这个回调函数呢？换句话说，AngularJS是如何知道scope发生了变化，才调用了对应的回调函数呢？它会周期性的运行一个函数来检查scope模型是否发生了变化吗？好吧，这就是\$digest循环的用武之地了。

在\$digest循环中，watchers会被触发。当一个watcher被触发时，AngularJS会检测scope模型是否发生了变化那么关联到该watcher的回调函数就会被调用。那么，下一个问题就是\$digest循环是如何开始的？

在调用了\$scope.\$digest()后，\$digest循环就开始了。假设你正在一个ng-click指令对应的handler中修改了scope中的一条数据，此时AngularJS会自动地通过调用\$digest()来触发一轮\$digest循环。循环开始后，它会触发每个watcher。这些watchers会检查scope中的当前model值是否和上次检查到的model值不同。如果不同，那么对应的回调函数会被执行。调用该函数的结果，就会更新scope中的内容(译注：诸如{{ aModel }})会被更新。除了ng-click指令，还有一些其它的built-in指令如ng-model, \$timeout等)和自动触发一次\$digest循环。

其它 (1)

文章存档

- 2015年07月 (4)
- 2015年04月 (1)
- 2015年03月 (3)
- 2015年02月 (1)
- 2015年01月 (7)

展开

阅读排行

- 并查集(Union-Find)算法 (39340)
- 拓扑排序的原理及其实现 (27337)
- 并查集(Union-Find) 应用 (9192)
- 理解Angular中的\$apply (8784)
- 求解强连通分量算法之一 (5024)
- [Elasticsearch] 过滤查询 (4174)
- TestNG源代码分析 — 依 (3755)
- [Elasticsearch] 多字段搜 (3384)
- [Elasticsearch] 全文搜索 (3319)
- [Elasticsearch] 聚合的测 (3082)

评论排行

- 并查集(Union-Find)算法 (37)
- 拓扑排序的原理及其实现 (20)
- 并查集(Union-Find) 应用 (18)
- 求解强连通分量算法之一 (4)
- Effective JavaScript Item (2)
- Effective JavaScript Item (2)
- Effective JavaScript Item (2)
- [PHP]利用openssl_ranc (2)
- Effective JavaScript Item (2)
- [Elasticsearch] 部分匹配 (2)

推荐文章

- * Android HandlerThread 源码分析
- * storm是如何保证at least once语义的
- * 小胖学PHP总结: PHP的循环语句
- * Spring基于注解@AspectJ的AOP
- * HTML5梦幻之旅: 仿Qt示例Drag and Drop Robot (换装机器人)
- * 学习笔记: Twitter核心数据类库团队的Hadoop优化经验

最新评论

- 拓扑排序的原理及其实现 qq_29635923: 谢谢
- 求解强连通分量算法之一—Kosara dm_vincent: @lucker1991:感谢回复。请参考这里: https://github.com/destiny1...
- 求解强连通分量算法之一—Kosara lucker1991: 您里边的Digraph是什么啊, 没有定义啊, 难不成还需要什么jar包么?
- [PHP]利用openssl_random_ps自由de单车: 你好, 在PHP官方文档中, openssl_random_pseudo_bytes(的原型是stri...
- [PHP]利用openssl_random_ps自由de单车: 你好, 在PHP官方文档中,

目前为止还不错!但是,有一个小问题。在上面的例子中,AngularJS并不直接调用\$digest调用\$scope.\$apply(),后者会调用\$rootScope.\$digest()。因此,一轮\$digest循环在\$rootScope访问到所有的children scope中的watchers。

现在,假设你将ng-click指令关联到了一个button上,并传入了一个function名到ng-click击时,AngularJS会将此function包装到一个wrapping function中,然后传入到\$scope.\$apply你的function会正常被执行,修改models(如果需要的话),此时一轮\$digest循环也会被触发你也会更新。

Note: \$scope.\$apply()会自动地调用\$rootScope.\$digest()。\$apply()方法有两种形式。第一个function作为参数,执行该function并且触发一轮\$digest循环。第二种会不接受任何一轮\$digest循环。我们马上会看到为什么第一种形式更好。

什么时候手动调用\$apply()方法?

如果AngularJS总是将我们的代码wrap到一个function中并传入\$apply(),以此来开始一轮\$digest循环,那么什么时候才需要我们手动地调用\$apply()方法呢?实际上,AngularJS对此有着非常明确的责任:负责发生于AngularJS上下文环境中的变更会做出自动地响应(即,在\$apply()方法中发生更改)。AngularJS的built-in指令就是这样做的,所以任何的model变更都会被反映到view中。如果你在AngularJS上下文之外的任何地方修改了model,那么你就需要通过手动调用\$apply()来告诉AngularJS,你修改了一些models,希望AngularJS帮你触发watchers来做出正

比如,如果你使用了JavaScript中的setTimeout()来更新一个scope model,那么AngularJS更改了什么。这种情况下,调用\$apply()就是你的责任了,通过调用它来触发一轮\$digest循环。如果你有一个指令用来设置一个DOM事件listener并且在该listener中修改了一些models,那么你就需要通过手动调用\$apply()来确保变更会被正确的反映到view中。

让我们来看一个例子。加入你有一个页面,一旦该页面加载完毕了,你希望在两秒钟后更新你的实现可能是下面这个样子的:

HTML:

```
[html] view plain copy print ?
01. <body ng-app="myApp">
02.   <div ng-controller="MessageController">
03.     Delayed Message: {{message}}
04.   </div>
05. </body>
```

JavaScript:

```
[javascript] view plain copy print ?
01. /* What happens without an $apply() */
02.
03. angular.module('myApp', []).controller('MessageController', function($scope) {
04.
05.   $scope.getMessage = function() {
06.     setTimeout(function() {
07.       $scope.message = 'Fetched after 3 seconds';
08.       console.log('message:'+$scope.message);
09.     }, 2000);
10.   }
11.
12.   $scope.getMessage();
13.
14. });
```

通过运行这个例子,你会看到过了两秒钟之后,控制台确实会显示出已经更新的model。有更新。原因也许你已经知道了,就是我们忘了调用\$apply()方法。因此,我们需要修

openssl_random_pseudo_bytes(
的原型是stri...

[Elasticsearch] 多字段搜索 (二) ·
jfljdsa: 写的很好啊!

[Elasticsearch] 过滤查询以及聚
dm_vincent: @lwp978555542:
感谢回复,也有用户反映了这个问题:
https://github.com/...

[Elasticsearch] 过滤查询以及聚
致若亦: 有没有像SQL里
面的HAVING语句来过滤聚合结果
(例如根据color聚合avg price,
要求a...

[Effective Java Distilled] Item 2
Mr_feng: mark,很好的文章, 将
构造者模式、链式赋值、以及
TC/JAVABEAN/构造器之间的区
别讲的很明白...

理解Angular中的\$apply()以及\$d
wuyangcherish: 刚开始学。受教
了。

下所示:

```
[javascript] view plain copy print ?  
01.  /* What happens with $apply */  
02.  angular.module('myApp', []).controller('MessageController', function($scope) {  
03.  
04.      $scope.getMessage = function() {  
05.          setTimeout(function() {  
06.              $scope.$apply(function() {  
07.                  //wrapped this within $apply  
08.                  $scope.message = 'Fetched after 3 seconds';  
09.                  console.log('message:' + $scope.message);  
10.              });  
11.          }, 2000);  
12.      }  
13.  
14.      $scope.getMessage();  
15.  
16.  });
```

如果你运行了上面的例子, 你会看到view在两秒钟之后也会更新。唯一的变化是我们在被wrapped到了\$scope.\$apply()中, 它会自动触发\$rootScope.\$digest(), 从而让watch更新view。

Note:顺便提一下, 你应该使用\$timeout service来代替setTimeout(), 因为前者会帮你调需要手动地调用它。

而且, 注意在以上的代码中你也可以在修改了model之后手动调用没有参数的\$apply()

```
[javascript] view plain copy print ?  
01.  $scope.getMessage = function() {  
02.      setTimeout(function() {  
03.          $scope.message = 'Fetched after two seconds';  
04.          console.log('message:' + $scope.message);  
05.          $scope.$apply(); //this triggers a $digest  
06.      }, 2000);  
07.  };
```

以上的代码使用了\$apply()的第二种形式, 也就是没有参数的形式。需要记住的是你总一个function作为参数的\$apply()方法。这是因为当你传入一个function到\$apply()中的时被包装到一个try...catch块中, 所以一旦有异常发生, 该异常会被\$exceptionHandler ser

\$digest循环会运行多少次?

当一个\$digest循环运行时, watchers会被执行来检查scope中的models是否发生了变化。那么相应的listener函数就会被执行。这涉及到一个重要的问题。如果listener函数本身model呢? AngularJS会怎么处理这种情况?

答案是\$digest循环不会只运行一次。在当前的一次循环结束后, 它会再执行一次循环来检查是否有models发生了变化。这就是脏检查(Dirty Checking), 它用来处理在listener函数被执起的model变化。因此, \$digest循环会持续运行直到model不再发生变化, 或者\$digest循环次。因此, 尽可能地不要在listener函数中修改model。

Note: \$digest循环最少也会运行两次, 即使在listener函数中并没有改变任何model。正如它会多运行一次来确保models没有变化。

结语

我希望这篇文章解释清楚了\$apply和\$digest。需要记住的最重要的是AngularJS是否能模

修改。如果它不能检测到，那么你就需要手动地调用\$apply()。

原文地址

<http://www.sitepoint.com/understanding-angulars-apply-digest/>

上一篇 Effective JavaScript Basics Item 1-6

下一篇 Effective JavaScript String Encoding Item 7

顶

4

踩

0

猜你在找

getClientRects 和 getBoundingClientRect 的用
jQuery 中的事件冒泡和阻止默认行为
express解析http请求
spring+websocket整合
压缩版JS的调试利器JavaScript Source Map基本介

准备好了么？跳吧！

更多职位尽在 **CSDN JOB**

服务器运维工程师（若干，男女不限）	我要跳槽	web前端设计师（若干，男女不限）	我要跳槽
深圳软石文化有限公司	3-6K/月	深圳软石文化有限公司	3-6K/月
.net软件工程师（若干，男女不限）	我要跳槽	系统架构师	我要跳槽
深圳软石文化有限公司	3-6K/月	深圳软石文化有限公司	5-8K/月

查看评论

1楼 [wuyangcherish](#) 2015-04-12 22:18发表



刚开始学。受教了。

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide
Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase

[Pure](#) [Solr](#) [Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 