记得第一次听说AngularJS这项很赞的Web的前端技术，那时还是2014年，年中时我们我的一个大牛兄弟当时去面试时，被问到了是否熟悉该技术，当时他了解和使用的技术比较多。我们询问他面试情况时，他给我们一个菜菜科普了该技术，印象比较深的是该技术一种前端MVC，完全可以将我们在VS后台()所做的大部分工作完成，当时就觉得很新奇，但由于自身技术基础比较薄弱，没有太多时间和积累去学习新的技术，因而搁置了。在2016新年初始，正好有一些富于时间，正好可以开学习这个被称为就是两个大括号的前端框架（当前已经非常成熟，国内大部分公司的部分项目均已使用），补补我薄弱无比的前端技术，当目前为止，写JS代码仍然是非常的抓瞎。

AngularJS诞生于2009，被用在很多我们熟知的Google应用，例如Gmail、Maps，它主要致力于快捷的构建AJAX应用，在示例库在Github的地址为：<https://github.com/shyamseshadri/angularjs-book>。

其最基本的几个概念如下所示：

1. 客户端模板：在我们过去使用的多页应用程序中，我们见html和数据装配混合起来生成页面后发送到浏览器，而单页面的AJAX应用则是将html模板和数据都直接发送给浏览器，由客户端装配。
2. MVC，概念在所有的Web应用中基本上都使用到。
3. 数据绑定，支持双向定，其实也就是观察者模式的实现，非常的方便。
4. 依赖注入，通过$scope, $location等内置对象，使得我们只需关心我们需要什么，而不关心起依赖，遵循了迪米特法则（最少知识原则，Law\_of\_Demeter）。
5. 指令，框架提供了很多指令，对html和Dom进行扩展，例如ng-controller指定控制器视图的某一部分，ng-model用于将输入数据绑定到模型属性。

一个简单例子如下，主要注意的是，很多地方的入门demo会省略ng-app后面的参数，Angular的Controller形式，以及相关模块的绑定等，浏览器肯能会报错，初学需要小心。此外，VS关于AngularJS的智能感知插件的下载和使用也是一个常见问题。

|  |
| --- |
| <!DOCTYPE html>  <html ng-app="myApp" xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <title>Shopping Cart</title>  </head>  <body ng-controller="CartController">  <h1>Your Order</h1>  <div ng-repeat="item in items">  <span>{{$index + 1}}</span>  <span>{{item.title}}</span>  <input **ng-model**="item.quantity" />  <span **ng-bind**=” item.price | currency”></span>  <span>{{item.price \* item.quantity | currency}}</span>  <button **ng-click**="remove($index)">Remove</button>  </div>  <script src="../Scripts/angular.js"></script>  <script>  var app = angular.module('myApp', []);  app.controller('CartController', function($scope) {  $scope.items = [  { title: 'Paint pots', quantity: 8, price: 3.95 },  { title: 'Polka dots', quantity: 17, price: 12.95 },  { title: 'Pebbles', quantity: 5, price: 6.95 }  ];  $scope.remove = function (index) {  $scope.items.splice(index, 1);  }  });  </script>  </body>  </html> |

在上例中，我们可以看到通过ng-app声明边界，即告诉框架哪一部分受其管理，以上为div元素；{{greeting.text}}的双括号插值语法，以及相同功能的ng-bind，推荐后者；ng-app命名空间的使用，控制angular框架的有效范围，这样可以很好的与遗留程序兼容；ng-repeat迭代数据；ng-model绑定数据，这是个双向绑定，View中的修改会影响到model，之后会有表单输入的例子再次强化这个概念；ng-click绑定单击事件处理函数。

大体来说，Angular程序一次请求的流程：用户请求应用起始页；浏览器向服务器发起http连接，加载index.html模板页面；Angular被加载到页面中，等待页面加载完成，然后查找ng-app指令，用于定义模板边界；之后Augular遍历模板，查找指令和绑定关系，触发注册监听器、执行DOM操作、获取服务器初始化数据；最后连接服务器请求其他数据(Ajax)。这种模板和数据完全分离的方式，非常适合浏览器缓存数据，提升应用性能。

* 表单输入

在框架中使用表单元素非常简单，可以通过ng-model将表单元素绑定到模型属性上，达到双向绑定的目的，这部分和.NET中的数据绑定效果一致；在表单提交时，ng-submit会自动阻止浏览器默认的POST操作；$watch可以监视Model中具体的属性和字段，而ng-change主要用来检视表单元素；ng-show和ng-hide用于显隐元素，在菜单场景下应用广泛

|  |
| --- |
| <body ng-app="myApp">  <form ng-controller="SomeController" ng-submit="requestFunding()">  <input type="checkbox" ng-model="youCheckIt" />  <br />  <input ng-change="computeNeeded()" ng-model="funding.startEstimate" />  Recommendation:{{funding.needed}}  <br />  <button>Fund my startup</button>  <br />  <button ng-click="reset()">Reset</button>  </form>  <script>  var app = angular.module('myApp', []);  app.controller('SomeController', function ($scope) {  var funding = {};  funding.startEstimate = 0;  funding.needed = 0;  $scope.funding = funding;  var computeNeeded = function () {  $scope.funding.needed = $scope.funding.startEstimate \* 10;  };  $scope.$watch('funding.startEstimate', computeNeeded);  $scope.requestFunding = function () { window.alert("Sorry, please get more customer first."); };  $scope.reset = function () { $scope.funding.startEstimate = 0; };  });  </script>  </body> |

Tip:

相信大家接触非侵入式javascript概念已经很久了吧，但通过以上例子，我们会发现angularJS的使用并没有这样做，而是将html模板和相关控制代码混写了，这难道是说该框架并不合理。其实不然，之前提取非侵入式的概念也是因为当时前端开发的痛点：不同浏览器对js的支持不同，运行方式也不同；事件处理器都引用全局命名空间的函数，在集成时存在命名冲突；事件监听器绑定数据结构和行为，难以维护。但对于JS来说，它通过框架自身解决兼容性问题，通过命名空间解决集成的问题，最后一点也是最重要的一点，所有的事件处理函数并不引用任何的DOM元素和事件。

参考资料：

1. 布拉德, 格林. 用AngularJS开发下一代Web应用[M]. 北京:电子工业出版社, 2013.