

angularjs DI(Dependency Injection)

在前面已经介绍了关于angularjs，以及扩展了一些jQuery ui的一些组件为angularjs的directive。在这里应[进□007](#)在[上篇](#)留言我们来看看在angularjs中的DI特性。

DI：依赖注入，是一种软件设计模式，应DIP依赖倒置原则，描述组件之间高层组件不应该依赖于底层组件。依赖倒置是指实现和接口倒置，采用自顶向下的方式关注所需的底层组件接口，而不是其实现。其应用框架则为IOC，在.net中有很多我们熟悉的IOC框架，如Unity，Castle windsor，Ninject，Autofact等等，其常常分为构造注入，函数注入，属性注。同时在IOC和Service Locator(服务查找)，如果你想更多的了解IOC和DI请参见martin fowler的[Inversion of Control Containers and the Dependency Injection pattern](#)。

回到angularjs：在框架中为我们提供了angular.injector(modules)DI注入注射器。但是在我们使用注入的时候常常是不需要关心具体的如何注入。我们只需要按照其规则书写我们的angularjs代码就会很容易的得到angularjs的DI特性，DI方式有三种：

1：推断式注入：在angularjs中我们可以在我们需要注入的地方按照名称注入，这里要求参数名称必须和注入服务实例名称相同，一种名称约定。angularjs会提取参数名称查找相应DI实例注入。

例如：



```
1var myModule = angular.module('myModule', []);
2
3 myModule.factory('$alert', function($window) {
4
5 return {
6     alert: function(text) {
7         $window.alert(text);
8     }
9 };
10 });
11
12 var myController = function($scope, $alert) {
13     $scope.message = function(msg) {
14         console.log(msg);
15         $alert.alert(msg);
16     };
17 };
18 myModule.controller("myController", myController);
```



在上面实例我利用已知的window服务新建了一个alert的服务.并利用注入到我们的controller使用.这里采用的都是约定注入(根据参数名称注入).

jsfiddle在线演示<http://jsfiddle.net/whitewolf/zyA5B/7/>

2:标记注入:在angularjs中我们可以利用\$inject标注DI注入,这里需要注入服务名称的顺序和构造参数名对应.这里可以解决以上约定的死板性.

将上例代码改变为如下：

代码如下：



```
1var myModule = angular.module('myModule', []);
2
3 myModule.factory('$alert', ['$window', function($window) {
4
5 return {
6     alert: function(text) {
7         $window.alert(text);
8     }
9 };
10 });
11
12 var myController = function($scope, $alert) {
13     $scope.message = function(msg) {
14         console.log(msg);
15         $alert.alert(msg);
16     };
17 };
18 myController.$inject = ['$scope', '$alert'];
19 myModule.controller("myController", myController);
```



jsfiddle在线演示<http://jsfiddle.net/whitewolf/zyA5B/8/>

3：内联注入：对于directives，factory，filter等特殊指令使用\$inject标注注入使用不是那么友好，angularjs特别增加了内联注入。如上面的\$alert服务



```

1 myModule.factory('$alert', ['$window', function($window) {
2
3 return {
4     alert: function(text) {
5         $window.alert(text);
6     }
7 }]);

```



在angularjs中我们可以在controller中实用DI特性，同时一系列的工厂方法如directives, services, filters同样可以实用内联注入得到DI特性。

1：在controller中形如：



```

1 var MyController = function(dep1, dep2) {
2
3 ...
4
5 }
6
7 MyController.$inject = ['dep1', 'dep2'];
8
9
10
11 MyController.prototype.aMethod = function() {
12
13 ...
14
15 }

```



2：工厂方法注入形如：



```

1 angular.module('myModule', []).
2
3 config(['depProvider', function(depProvider){
4
5 ...
6
7 }]).
8
9 factory('serviceId', ['depService', function(depService) {
10
11 ...
12
13 }]).
14
15 directive('directiveName', ['depService', function(depService) {
16
17 ...
18
19 }]).
20
21 filter('filterName', ['depService', function(depService) {
22
23 ...
24
25 }]).
26
27 run(['depService', function(depService) {
28
29 ...
30
31 }]);

```

