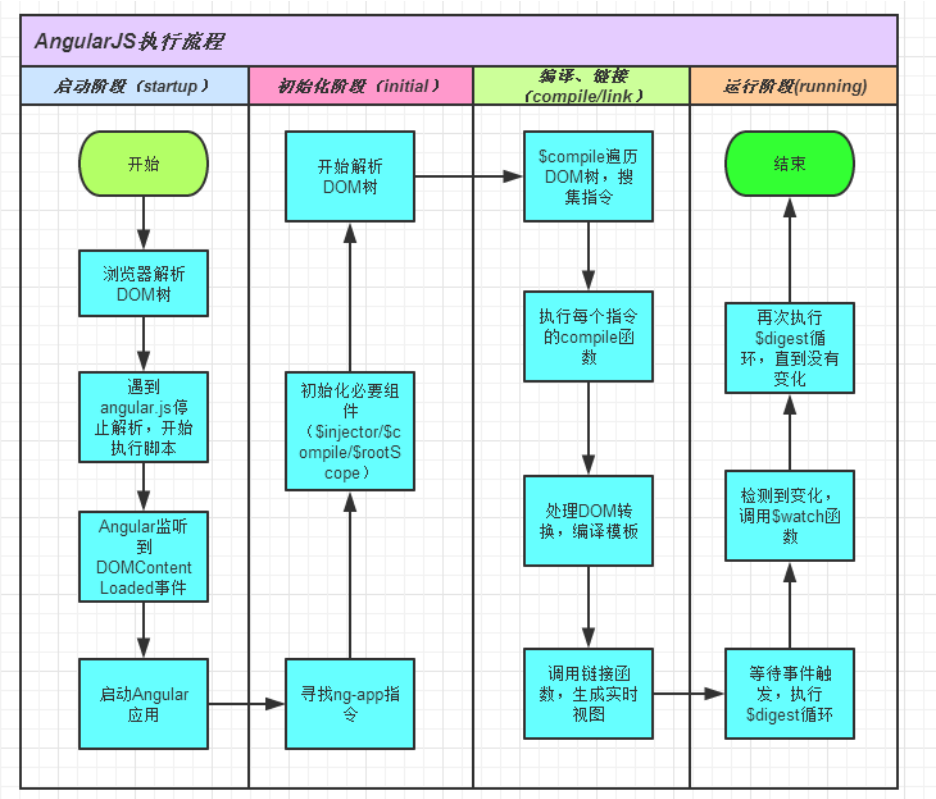
Angular1.x入门与精通 第3章：Angular1.x启动分析

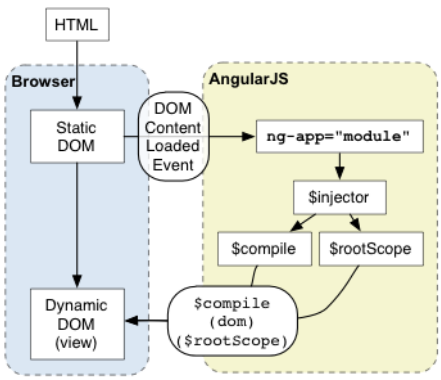
> 讲师：徐礼文 2016/7/18 21:27:53

AngularJS的源码在整体上，与其它很多库和框架一样，是一个自执行函数，其整体结构简化如下：

```
(function(window, document, undefined) {
// define variables and functions
// and do some operations

if (window.angular.bootstrap) {
console.log('WARNING: Tried to load angular more than once.');
return;
}

bindJQuery();

publishExternalAPI(angular);

jqLite(document).ready(function() {
angularInit(document, bootstrap);
});
})(window, document);
```



自动初始化



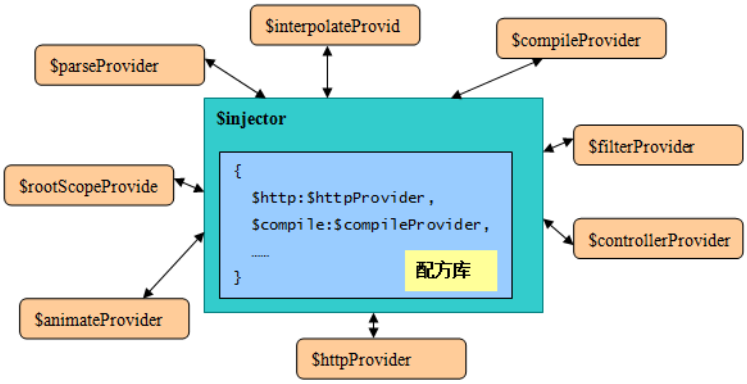
AngularJS API之$injector ---- 依赖注入

Dependency Injection (DI,依赖注入)是一种软件设计模式,用于处理如何让程序获得其依赖(对象的)引用。

注入器是AngularJS框架实现和应用开发的关键，这是一个DI/IoC容器的实现。

AngularJS将功能分成了不同类型的组件分别实现，这些组件有一个统称 - 供给者/provider， 下图中列出了AngularJS几个常用的内置服务：

AngularJS的组件之间不可以互相直接调用，一个组件必须通过注入器才 可以调用另一个组件。这样的好处是组件之间相互解耦，对象的整个生命周期的管理 甩给了注入器。



> 在AngularJS中也有依赖注入的概念，像spring中的依赖注入，但是又有所不同。Spring中使用构造注入或者设值注入的方式，还需要做一些额外的操作，但是angular中只需要在需要的地方声明一下即可，类似模块的引用，因此十分方便。
>
> 参考：[angular api doc] (http://docs.angularjs.cn/api/auto/service/$injector)

推断式注入

> 这种注入方式，需要在保证参数名称与服务名称相同。如果代码要经过压缩等操作，就会导致注入失败。

```
 app.controller("myCtrl1", function($scope,hello1,hello2){
   $scope.hello = function(){
    hello1.hello();
    hello2.hello();
  }
});
```

标记式注入

> 这种注入方式，需要设置一个依赖数组，数组内是依赖的服务名字，在函数参数中，可以随意设置参数名称，但是必须保证顺序的一致性。

```
 var myCtrl2 = function($scope,hello1,hello2){
 $scope.hello = function(){
    hello1.hello();
    hello2.hello();
  }
}
myCtrl2.$injector = ['hello1','hello2'];
app.controller("myCtrl2", myCtrl2);
```

内联式注入

> 这种注入方式直接传入两个参数，一个是名字，另一个是一个数组。这个数组的最后一个参数是真正的方法体，其他的都是依赖的目标，但是要保证与方法体的参数顺序一致（与标记注入一样）。

```
app.controller("myCtrl3",['$scope','hello1','hello2',function($scope,hello1,hello2){
    $scope.hello = function(){
       hello1.hello();
       hello2.hello();
  }
}]);
```

$injector常用的方法

在angular中，可以通过angular.injector()获得注入器。

```
var $injector = angular.injector();
```

通过$injector.get('serviceName')获得依赖的服务名字

```
$injector.get('$scope')
```

通过$injector.annotate('xxx')获得xxx的所有依赖项 $injector.annotate(xxx)