

模块化：module

var myModule = angular.module("HelloAngular", []);

myModule.controller("helloAngular", ['$scope',

function HelloAngular($scope) {

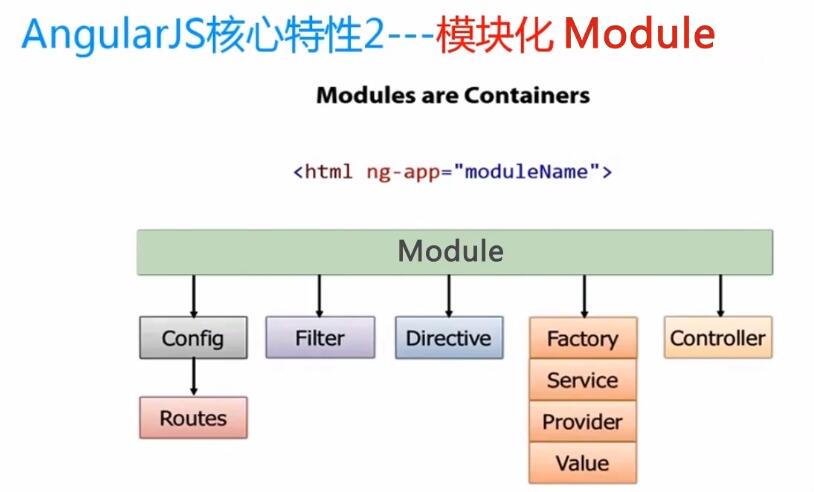
$scope.greeting = {

text: 'Hello'

};

}

]);



指令：template 通过自定义的标签设置HTML内容，

单页只能有一个ng-app=’mymodule’ //启动ng-app

var myModule = angular.module("MyModule", []);

myModule.directive("hello", function() {

return {

restrict: 'E',

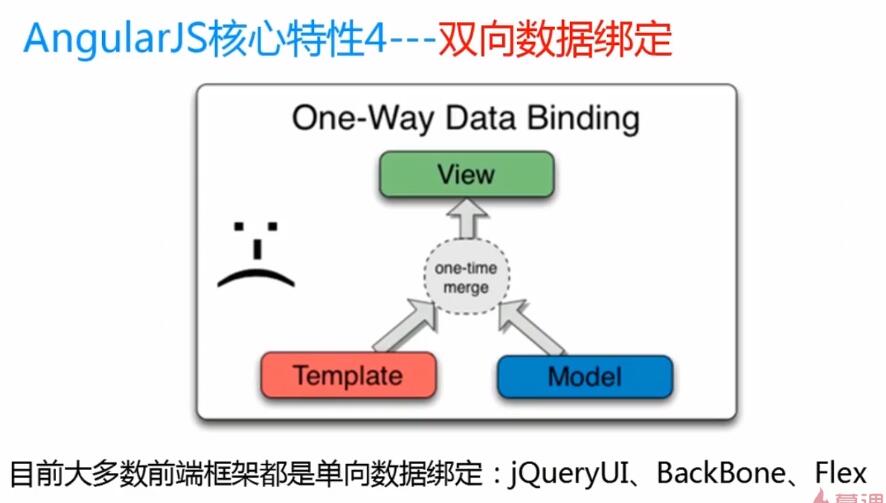
template: '<div>Hi everyone!</div>',

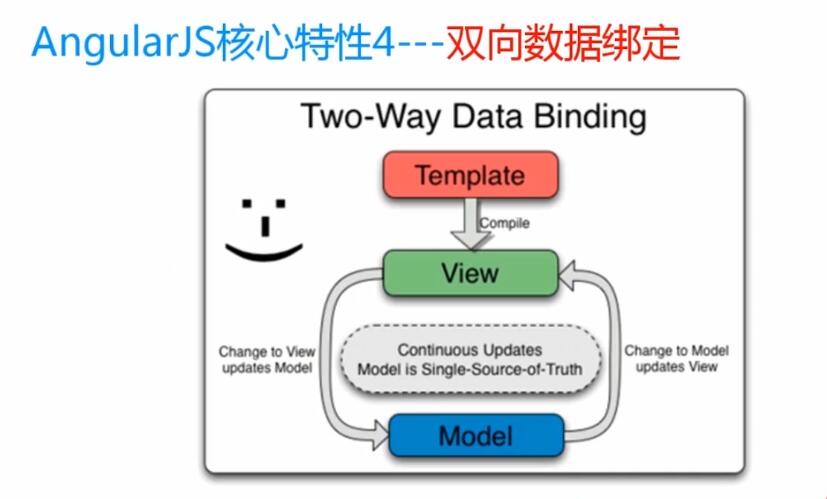
replace: true

}

});



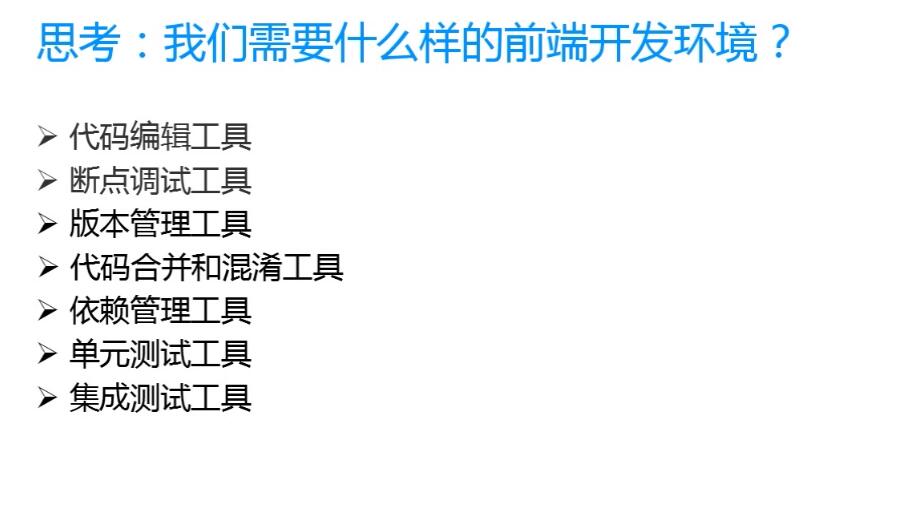




{{ }} ------双括号用来取值

调试工具：npm

npm run protractor ------可以用来自动化的测试，可以打开我们本机安装的浏览器，自动模拟鼠标键盘操作等



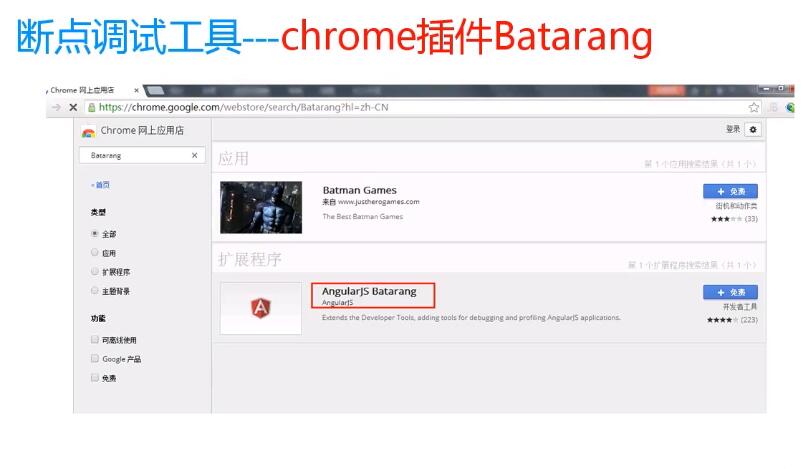


node.js是作为基础的存在

sublime：插件实现自动化

webstorm：编辑器

batarang 针对angularjs，断点调试工具或者原始的firebug



1.git：版本管理工具

2.git 小乌龟：会出现一些可视化管理

3.nodejs下的npm:可以安装用到的插件 ---npm install grunt命令会在当前的目录下安装这个插件

4.grunt：代码合并和混淆工具，需要配合插件安装使用，通过npm命令安装即可

5.bower:自动安装需要的文件，比如jQuery，还会自动维护互相之间的依赖关系，如果删掉了什么东西会提醒你

bower install jquery 执行此命令就会安装jQuery

bower install bootstrap 安装bootstrap



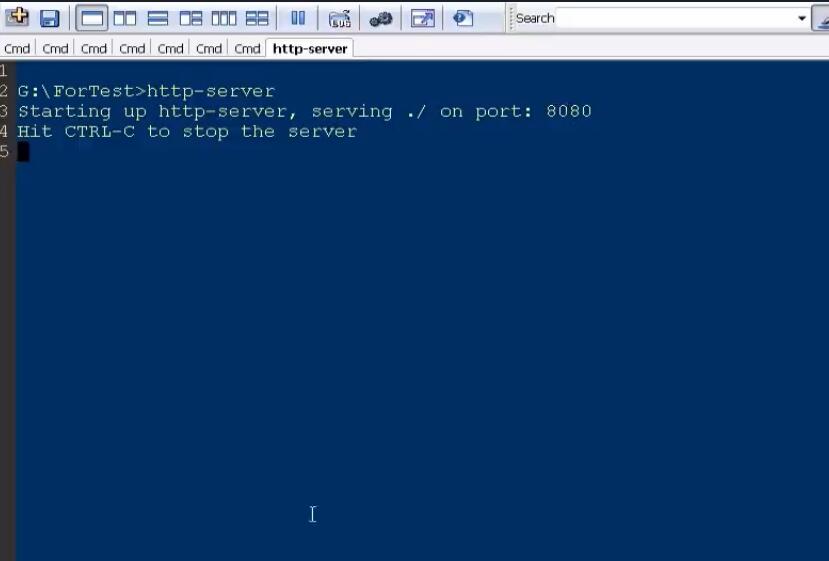


6.基于node.js的server：可以把目录里的文件变成可以在http访问的文件

在需要变成http地址的目录里执行http-sever

用来模拟测试数据





1. 单元测试神器runner--karma

配合jasmine

npm 命令安装

karma编写测试文件 然后cmd 里karma start

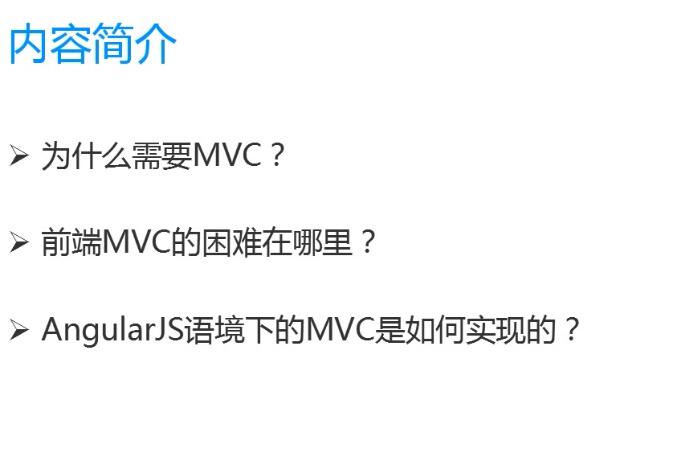




1. 专门为angular指定的，不能用于其他框架





1.为什么需要MVC？

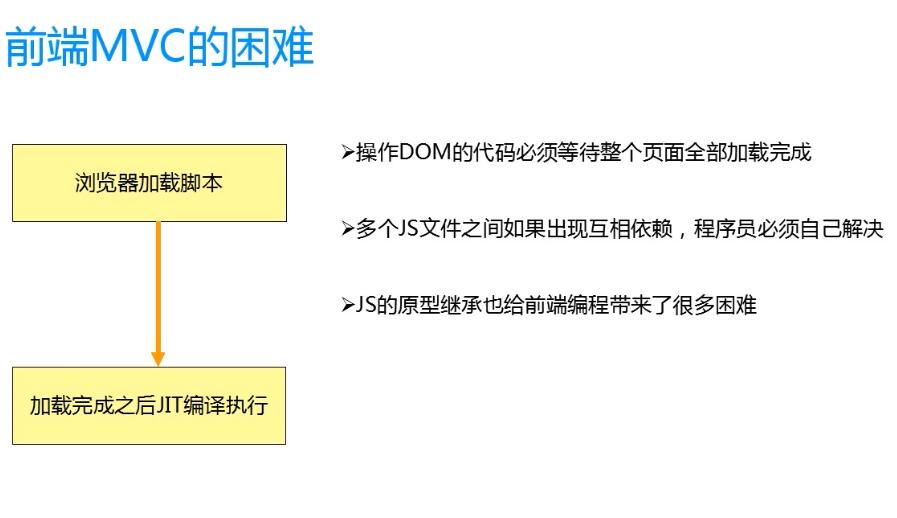
代码规模越来越大，切分职责是大势所趋；

为了复用：很多逻辑是一模一样的；

为了后期维护的方便：修改一块功能不影响其他功能；

MVC 只是手段，终极目标是模块化和复用

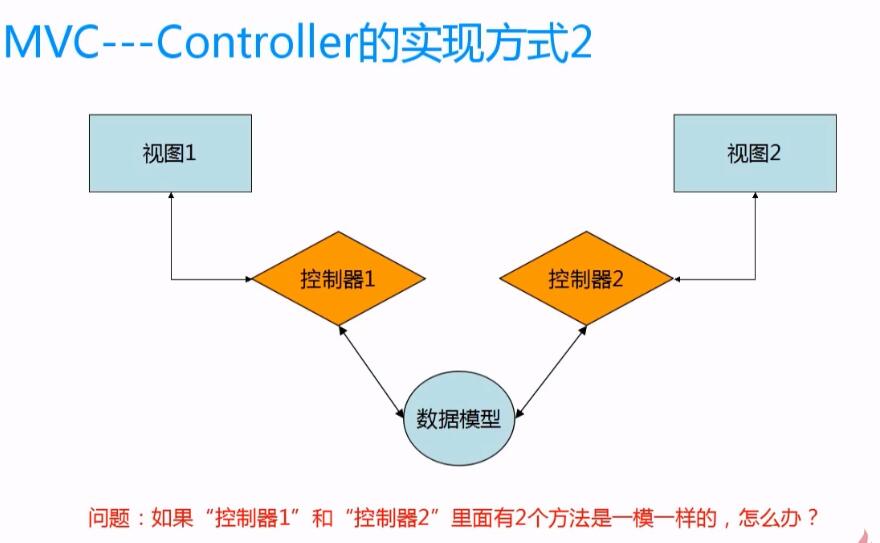
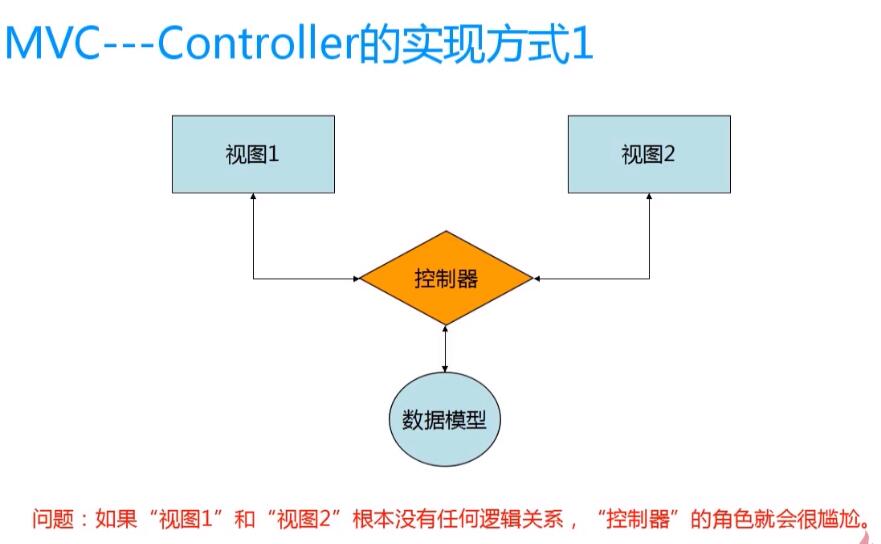
2.前端MVC的困难

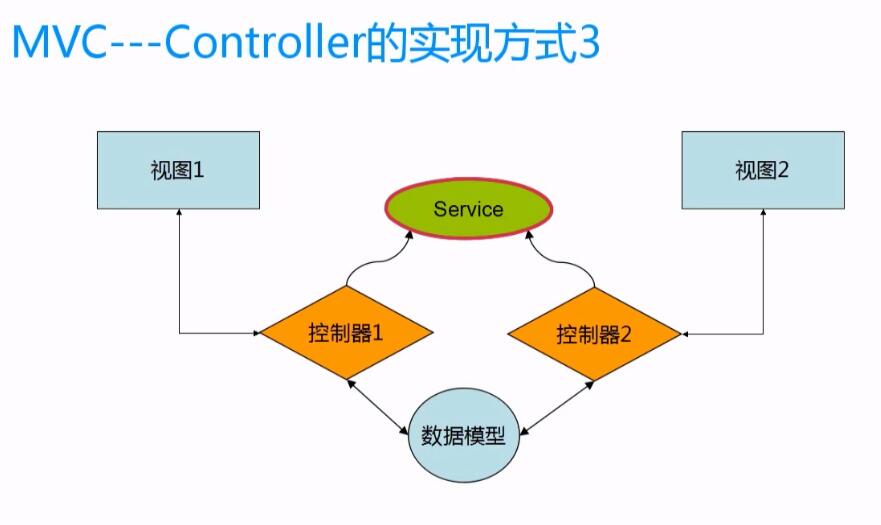


3.是怎么实现MVC的？

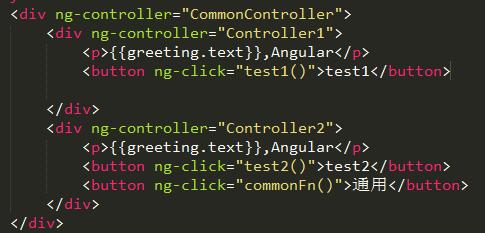
angularjs的controller







将公共的部分写成server一个服务，不要试图继承，如下就是错误的





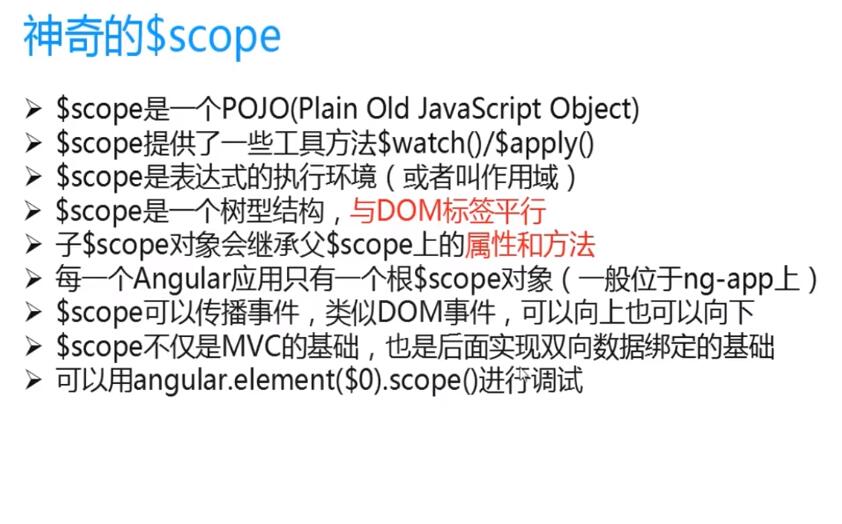
model是实现



视图view

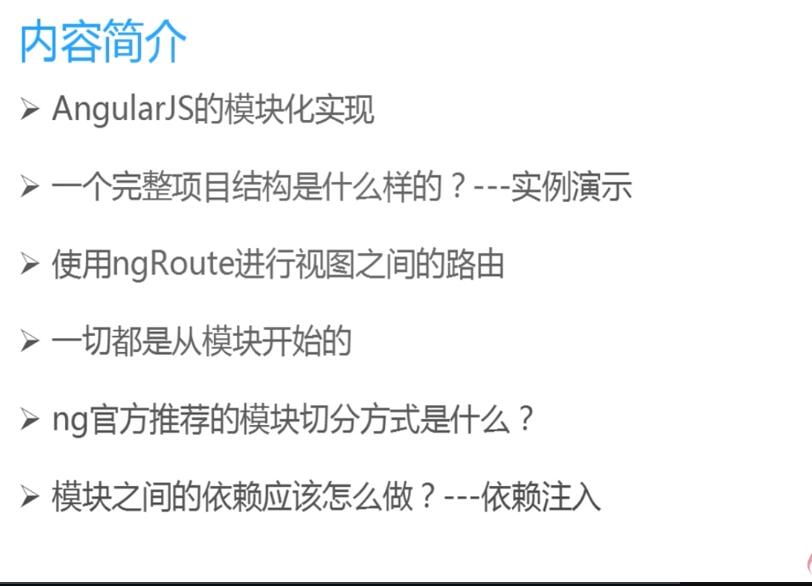
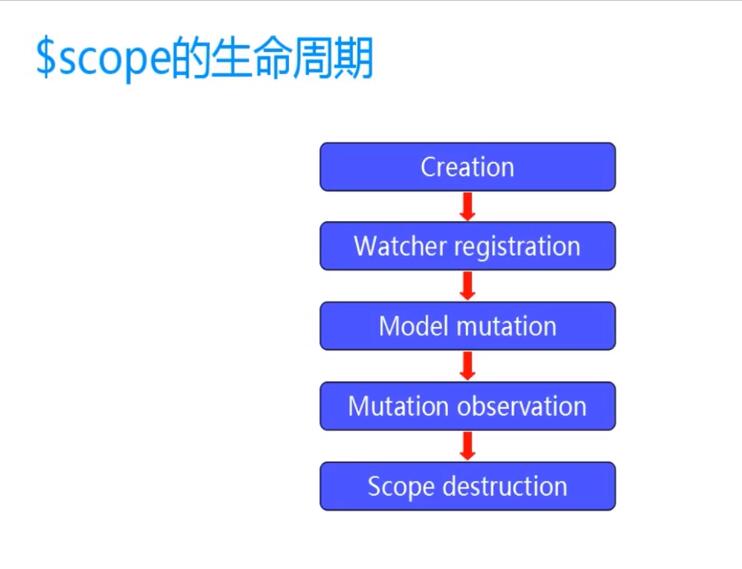


angularjs里借助都是借助于$scope作用域实现的

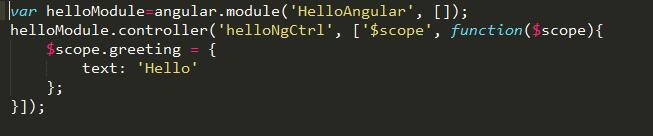


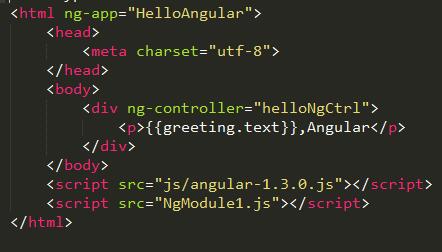
调试插件inspect angular scope

右击就会看到详细的scope信息

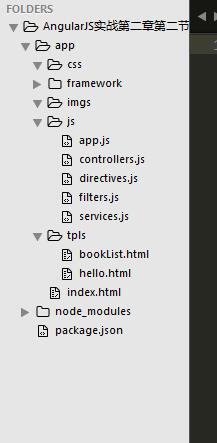


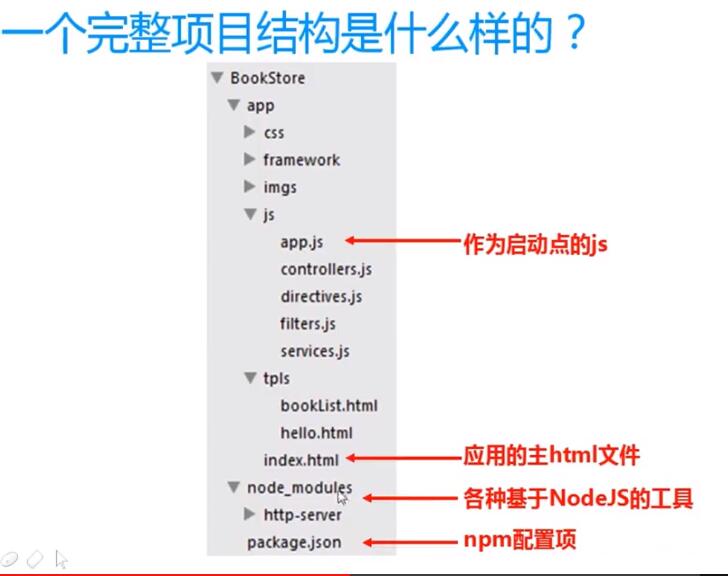
实现模块化

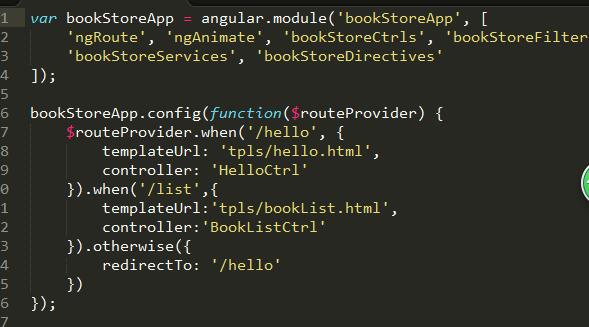




真实项目目录结构



路由：根据URL的地址不同使用不同的模板

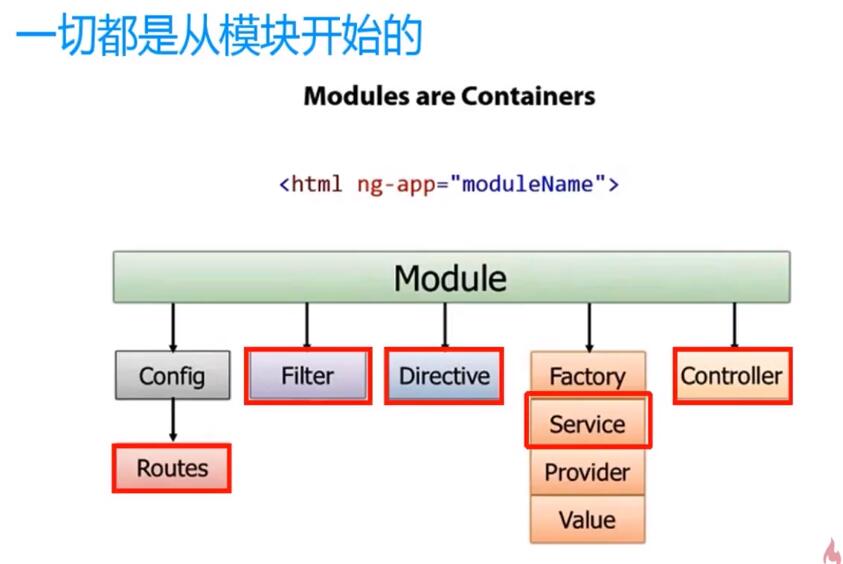
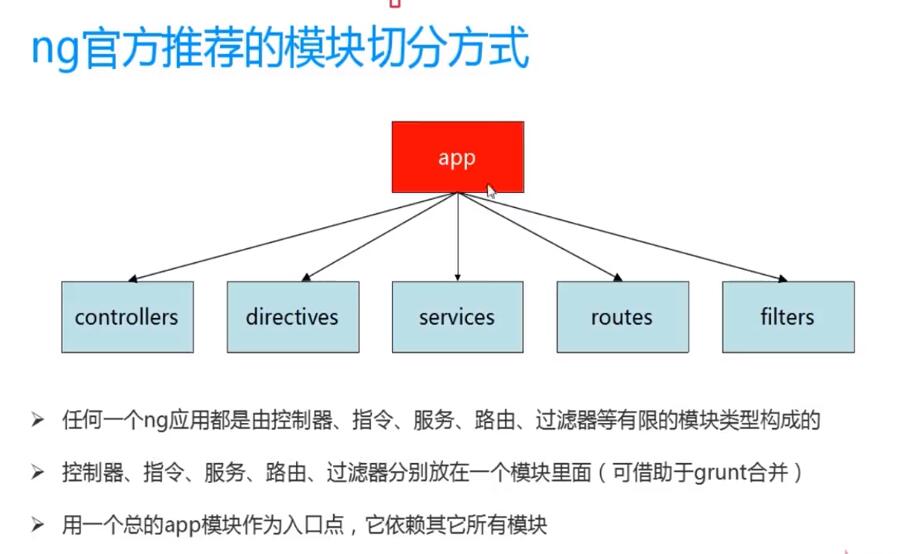


var bookStoreApp = angular.module('bookStoreApp', [

'ngRoute', 'ngAnimate', 'bookStoreCtrls', 'bookStoreFilters',

'bookStoreServices', 'bookStoreDirectives'

]); //第二个参数是依赖哪些模块



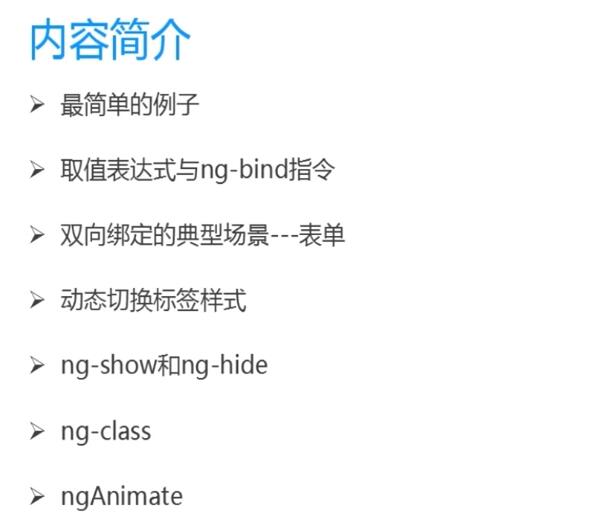
依赖注入



需要依赖哪个模块就添加到参数后就好

ng开头的是angular提供的





{{}}绑定时，网络不行时会出现{{}},用ng-bind=’greeting.text’

在第一个页面时使用ng-bind=’’,在其他的页面可以使用{{}}绑定

<div ng-controller="CSSCtrl">

<p class="text-{{color}}">测试CSS样式</p>

<button class="btn btn-default" ng-click="setGreen()">绿色</button>

</div>

<div ng-controller='HeaderController'>

<div ng-class='{error: isError, warning: isWarning}'>{{messageText}}</div>

<button ng-click='showError()'>Simulate Error</button>

<button ng-click='showWarning()'>Simulate Warning</button>

</div>

<div ng-controller='DeathrayMenuController'>

<button ng-click='toggleMenu()'>Toggle Menu</button>

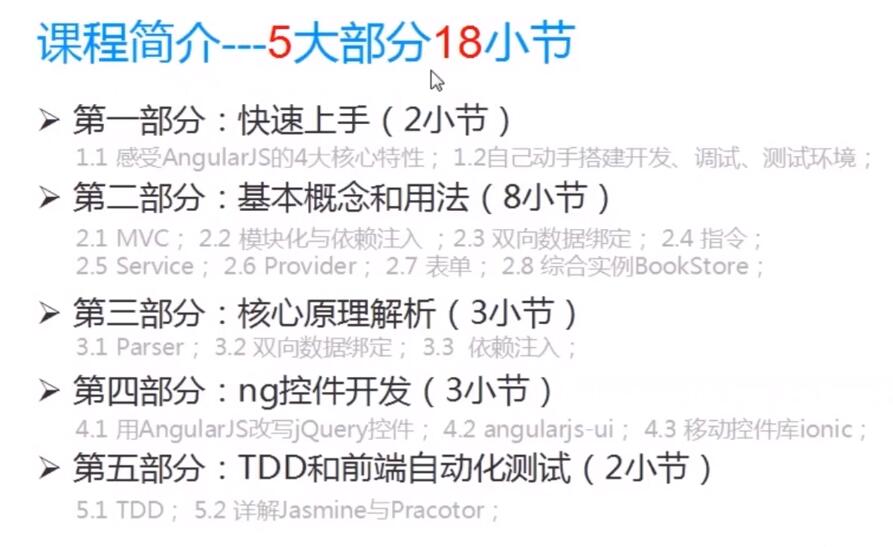
<ul ng-show='menuState.show'>

<li ng-click='stun()'>Stun</li>

<li ng-click='disintegrate()'>Disintegrate</li>

<li ng-click='erase()'>Erase from history</li>

<div/>



为什么用前端路由

