周小结

Angular部分

基于MVC，MVC不是26个基础设计模式中的一个，而是其中3个设计模式的综合

Controller使用过程中的注意点

1.不要试图去复用一个Controller,一个控制器一般只负责一小块试图

2.不要再Controller中操作DOM（封装在directive中），这不是控制器的职责

3.不要再~里面做数据格式化，ng有很好用的表单控件

4.不要再~里面做数据过滤操作，ng有$filter服务

5.一般来说，~是不会相互调用的，控制器之间的交互会通过事件进行

ng的MVC是借助于$scope实现的

$scope 作用域

$rootscope 根作用域

$scope是拥有事件方式 拥有事件捕获和事件冒泡机制

ng-click用来绑定事件 内部为以下俩种方式

$emit('事件名')事件将向上传播 ，$broadcast('事件名')事件将向下传播

$scope详细

是一个POJO(Plain Old JS Object)

提供了一些工具方法$wactch()/$apply()

是表达式的执行环境（或者叫作用域）

$scope是一个树形结构，与DOM标签平行

子$scope对象会继承父$scope上的属性和方法

每一个ng应用都只有一个根$scope对象（一般位于ng-app上）

$scope可以传播事件，类似DOM事件，

~不仅是MVC基础，也是实现双向数据绑定的基础

angular,element($0).scope()进行调试 可以获得当前元素上的$scope

路由，模块，依赖注入

ngRoute进行视图路由

ng模块是由模型视图控制器 等等组成在一起实现某个功能

$routeProvider.when

#是阻止向服务器进行请求页面

双向数据绑定

ng-bind:<span ng-bind="greeting.text"></span> 在index页面用ng-bind其他子页面可以用{{}}来取值

ng-click:可以直接调用在当前$scope和父$scope所设置的方法，对于数据绑定用ng-model与对应的Controller进行数据绑定作用域也还是$scope

ng-class 可以在class里面进行表达式绑定

ng-show/ng-hide显示隐藏

ng-show="user.isshow" ng-hide与ng-show相反

ngAnimate

路由

为什么要路由，Ajax请求不会留下History记录

用户无法直接通过URL进入应该用中的制定页面（保存数千、链接分享给朋友？）

SEO无法实现，即搜索引擎优化

config.(fnc($routeProvider){

$~.when()

.when()

.otherwiser()

})

NG自带的路由 无法进行深层次路由

所谓的深层次路由 即ng的无法进行嵌套路由 即外层能够 但是内层不能定义ng-view

有人就实现了ui-router

前端路由基本原理

哈希# HTML5中新的historyAPI

路由的核心是给应用定义“状态”

ui-router可以进行嵌套路由，在要进行嵌套的地方用'main(ng-view定义的名字)@index(父层嵌套)'

当 一个state要更换前面的嵌套的时候在view 中直接对main@index进行重置.

除此之外view还有controller属性

ui-router文档:[https://github.com/angular-ui/ui-router/wiki/Quick-Reference#stateprovider-1](https://github.com/angular-ui/ui-router/wiki/Quick-Reference" \l "stateprovider-1)

指令

最简单的hello指令

html段

<hello></hello> <div hello></div> <div class="hello"></div>

<!-- directive:hello -->

<div></div>

js段

(function($) {

var hello = $.angular.module('hello', []);

hello.directive('hello', function() {

// Runs during compile

return {

restrict: 'AEMC',//匹配模式，属性，元素，注释，class名

replace: true,

transclude:true

template: '<div>Hi everyone</div><div ng-transclude></div>',

//在后面的ng-transclude里面将保存我们原本div的内容,指令互相嵌套时需要

};

});

})(window);

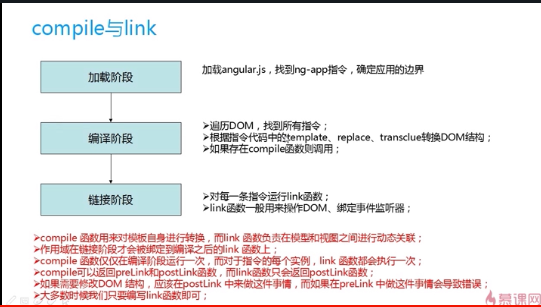
模板

template/templateUrl/templateCache（当模板在其他指令也要用到的时候，让HTML缓存模板）

缓存方法$templateCache.put("hello.html", "<div>Hello everyone!!</div>");

使用方法，其实类似于OC的字典，JAVA的MAP template: $templateCache.get("hello.html"),

在directive内还有compile和link



利用link进行自定的指令的时间绑定 和DOM相关操作

见ngexam/hello相关页面

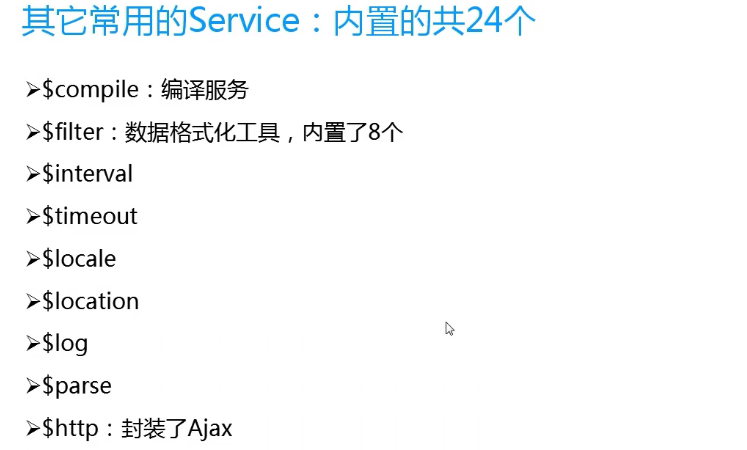
link方法中有3个参数scope,element,attr，scope就是对应controller下的scope,element就是生成的该元素的element，就是directive生成的指令,attr是该指令里面指定的属性.scope.$apply(attr.howtoload);可以调用写在属性里面的方法

指令间的交互

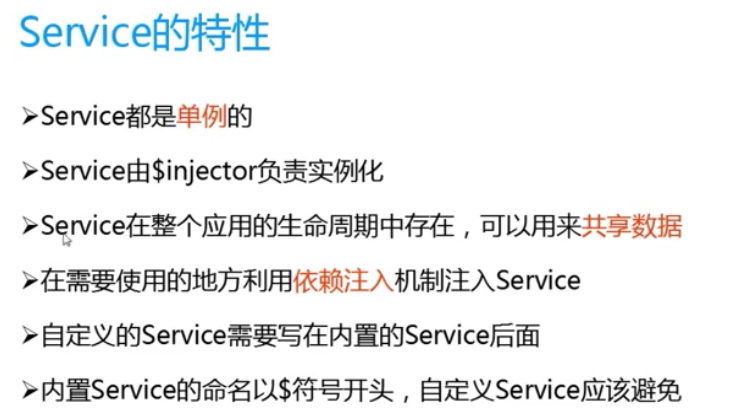
在directive中的require:'^superman'子指令依赖父指令

内置表单指令





Service



controller执行顺序

当我们在路由指定了对应的controller后，一旦跳到对应的 controller

系统会自动去执行一遍我们所需要的controller初始化我们要用的所有数据

ng-repeat="item in items"/"(x,y) in items"

x是单个项目的key,v是对应key的值

看到所有城市，行程天数，出行事件

Ionic部分

tab标签

<ion-tabs class="tabs-icon-top tabs-color-active-positive">

<!-- Dashboard Tab -->

<ion-tab title="Status" icon-off="ion-ios-pulse" icon-on="ion-ios-pulse-strong" href="#/tab/dash">

<ion-nav-view name="tab-dash"></ion-nav-view>

</ion-tab>

<!-- Chats Tab -->

<ion-tab title="Chats" icon-off="ion-ios-chatboxes-outline" icon-on="ion-ios-chatboxes" ui-sref="tab.chats">

<ion-nav-view name="tab-chats"></ion-nav-view>

</ion-tab>

<!-- Account Tab -->

<ion-tab title="Account" icon-off="ion-ios-gear-outline" icon-on="ion-ios-gear" href="#/tab/account">

<ion-nav-view name="tab-account"></ion-nav-view>

</ion-tab>

</ion-tabs>

title:显示该tab页面的描述，icon-off/icon-on是用来表示icon被选中和没被选中时候的图标样式

href/ng-sref用来表示页面的跳转sref是路由 直接找app.js中对应的路由 会解析成href样式

<ion-nav-bar ></ion-nav-bar> 顶部栏

<ion-tabs></ion-tabs>选项卡

选项卡内置的css

tabs-color-active-positive//选中活动时 对应选项卡的颜色改变

tabs-color-颜色//代表为选中的时候的颜色

tabs-top//将tabs提到上面

tabs-striped//选中选项的下划线

选项卡图标和文字的排版

tabs-icon-only//只有图片

tabs-icon-top//图标在上

tabs-icon-left//图标在左

滑动指令

<ion-view view-title="">

<ion-content class="padding">

<ion-slide-box does-continue="true" delegate-handle="image-viewer1" auto-play="true" show-pager="true" slide-interval="600">

<ion-slide ng-repeat="slide in banner" style="width:100%;" on-slide-changed="slideHasChanged($index)">

<div class="box">

<img ng-src="{{slide}}" style="height:100%;width:100%;position: relative;">

</div>

</ion-slide>

</ion-slide-box>

</ion-content>

</ion-view>

在对应的controller里面当banner数据绑定完成后 记得用注入的$ionicSlideBoxDelegate的update(0方法更新一下。 不然无法渲染出对应的样式和效果

在更新前$ionicSlideBoxDelegate.$getByHandle("slideboximgs").loop(true);调用， 使轮轮播收尾相连无限循环。

does-continue设置为true配合上面的JS

auto-play自动播放 show-pager显示小圆点 slider-interval切换时间

列表

<ion-list><ion-item></ion-item></ion-list>

滚动

<ion-tabs></ion-tabs>单纯的给予一个滚动指令

<ion-infinite-scroll></ion-infinite-scroll>滚动到特定位置时触发on-infinite事件

可以用ng-if阻止无限滚动 icon默认ion-loading-c

对于弹出窗口

在ctroller里面注入$ionicModal实现