

MVC模式&angularJS入门

一、MVC设计模式(思想)

M : model 业务逻辑层

V : view 展示层

C : controller 控制层

view层请求到达controller控制层,由controller控制层于model层进行数据交互,完成业务处理。业务处理结果会返回给controller层,controller再返回给view进行页面展示。使用MVC的优点在于分层处理后,降低了代码之间的耦合度。

二、MWM

Model View ViewModel

双向绑定机制。

三、angularJS模块化

1、一个APP应用可以划分成N个模块(module),每个模块都有自己的功能。

2、通过模块化处理,最大的优势就是提高代码的复用率。

注意:

AngularJS2.x版本是基于ES6(ECMAScript 6.0)

当前主流的环境还是:angularjs 1.x 版本,主要用于web应用。

我们使用的是angularJS 1.5.8版本

步骤:

1、引入angularJS的文件库

2、在html标签中指定ng-app表明这是一个应用。

3、通过angular对象的module方法,创建一个模块,方法有两个参数:

◇模块的名称--> ng-app属性的值

◇需要依赖的其他模块名称

4、创建controller层

通过模块对象的controller方法,创建一个控制层,方法有两个参数:

◇控制层的名称,这个名称需要与ng-controller属性值相同

◇数组:前面的元素都是依赖的关系名称(字符串)

通常会包含一个\$scope作用域,这个作用域用于存储数据。

数组的最后一个元素是一个function函数。函数中通常可以完成数据的加载。

5、在HTML结构中,在ng-controller的标签中通过({属性})的方式进行数据展示。

例如:

```
<!DOCTYPE html>
<html ng-app="myModule">
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <script src="lib/angular/angular.min.js"></script>
</head>
<body>
  <div ng-controller="userController">
    <ul>
      <li ng-repeat="user in users">
```

```

        姓名:{{user.userName}},年龄:{{user.age}}
    </li>
</ul>
</div>
<div ng-controller="newsController">
    <ol>
        <li ng-repeat="news in newsList">
            {{news.title}}
        </li>
    </ol>
</div>
</body>
<script>
    var myModule = angular.module("myModule",[]);
    myModule.controller("userController",["$scope", function ($scope)
    {
        $scope.users = [{"userName":"葫芦大娃","age":7},{ "userName":"黑
猫警长","age":25}]];
    }]);
    myModule.controller("newsController",["$scope", function ($scope)
    {
        $scope.newsList = [{title:"新闻1"},{title:"新闻2"},{title:"新
闻3"}]];
    }]);
</script>
</html>

```

四、AngularJS的指令

angularJS本身提供了一些内置指令，比如：ng-app,ng-show,ng-if...等
 这些指令本身就是对HTML标签的属性拓展，可以认为是html标签的自定义属性。
 除了内置指令以外，还可以通过directive方法实现自定义属性。
 通过模块对象.directive("指令名称", function () {

```

        return {

    };
});

```

directive方法有2个参数，第一个是指令名称，第二个是指令对应的动作。
 在function中，返回的是一个json对象。

restrict : "" 表示调用该指令的方法

E : element
 A : attribute
 M : marker
 C : class

例如：restrict : "EA" 表示可以通过元素名和属性来调用这个指令。

template : "" 表示替换的模板内容

replace : boolean (true表示替换原有的标签, false表示不替换)

其他属性参考API。

注意：自定义指令的名称需要使用驼峰式命名法命名，在调用的时候，把大写字母变小写，并在前面加一个 - 。例如：

```
directive ( "myDirectiveMethod" function() {} );
```

则调用的时候：my-directive-method