



第十讲 REST服务

教学目标

1. REST服务
2. 使用\$http服务
3. 使用\$resource服务

1

REST服务

①

什么是REST

REST ---Resource **Re**presentational **S**tate **T**ransfer

资源在网络中以某种表现形式进行状态转移。

Resource : 资源，即数据。

Representational : 某种表现形式，比如用JSON，XML等；

State Transfer : 状态变化。

1

什么是REST

REST 指的是一组架构约束条件和原则。

REST描述的是在网络中客户端和服务器的交互形式；



- **使用客户端/服务器模型。**

- **层次化的系统。**

客户端通常不会将其直接连接到终端服务器，可以直接连接到中介服务器。

- **无状态。**

在一个REST系统中，服务端并不会保存有关客户的任何状态。

- **可缓存。**

REST系统需要能够恰当地缓存请求，以提高性能。

- **统一的接口。**

一个REST系统需要使用一个统一的接口来完成子系统之间以及服务与用户之间的交互。

满足这些约束条件和原则的应用程序或设计就是 RESTful。

AngularJS实现REST服务：

- 1、**\$http服务**同步服务器并操作视图数据
- 2、**\$resource服务**更新服务器数据

2

使用\$http服务

1

列出产品数据

Demo:products.html

调用文件：

```
<script src="angular.js"></script>  
<script src="products1.js"></script>
```

视图：

```
<ng-include src="'tableView.html'"  
            ng-show="displayMode == 'list'">
```

```
</ng-include>
```

条件成立时显示外部文件
tableView.html的内容

```
<ng-include src="'editorView.html'"  
            ng-show="displayMode == 'edit'">
```

```
</ng-include>
```

条件成立时显示外部文件
editorView.html的内容

1

列出产品数据

tableView.html

使用表格输出控制器作用域的数据

```
<tr ng-repeat="item in products">
  <td>{{item.name}}</td>
  <td>{{item.category}}</td>
  <td class="text-right">{{item.price | currency}}</td>
  <td class="text-center">
    <button ng-click="deleteProduct(item)">
      删除
    </button>
    <button ng-click="editOrCreateProduct(item)">
      编辑
    </button>
  </td>
</tr>
```

1

列出产品数据

editorView.html

编辑视图

<label>名称:</label>

<input ng-model="currentProduct.name" />

双向数据绑定currentProduct

<label>类型:</label>

<input ng-model="currentProduct.category" />

双向数据绑定currentProduct

<label>价格:</label>

<input ng-model="currentProduct.price" />

双向数据绑定currentProduct

< button ng-click="saveEdit(currentProduct)">
保存</button>

<button ng-click="cancelEdit()">取消</button>

②

\$http与服务器通信

```
angular.module("exampleApp", [])  
  .constant("baseUrl",  
    "http://localhost:2403/products/")  
  // .constant (变量, 值) 定义全局变量  
  .controller("defaultCtrl", function ($scope,  
    $http, baseUrl) {})
```

服务器数据

Demo: products1.js

3

显示产品

```
$scope.displayMode = "list";  
//初始化：显示产品列表  
$scope.currentProduct = null;  
//编辑视图中的双向数据绑定的input初始化为空  
$scope.listProducts = function () {  
    $http.get(baseUrl).success(function (data)  
    {  
        $scope.products = data;  
    });  
};
```

Demo: products1.js

④

删除产品

Demo: products1.js

```
$scope.deleteProduct = function (product) {  
  $scope.products.splice($scope.products.indexOf(p  
    roduct), 1);
```

//splice(index,howmany)方法可删除从index处开
始的howmany个元素。

```
};
```

```
<button ng-click="deleteProduct(item)">  
  删除  
</button>
```

5

创建产品

Demo: products1.js

```
<button ng-click="editOrCreateProduct()">  
添加</button>
```

```
$scope.editOrCreateProduct = function (product) {  
    $scope.currentProduct = product ? product : {};  
    $scope.displayMode = "edit";  
};  
  
$scope.createProduct = function (product) {  
    $scope.products.push(product);  
    $scope.displayMode = "list";  
};
```

6

更新产品

```
$scope.updateProduct = function (product) {  
    for (var i = 0; i < $scope.products.length; i++) {  
        if ($scope.products[i].id == product.id) {  
            $scope.products[i] = product;  
            break;  
        }  
    }  
    $scope.displayMode = "list";  
};
```

Demo: products1.js

7

保存操作

```
$scope.saveEdit = function (product) {  
    if (angular.isDefined(product.id)) {  
        //若产品存在则调用更新函数  
        $scope.updateProduct(product);  
    } else {  
        //若产品不存在则调用添加函数  
        $scope.createProduct(product);  
    }  
};
```

Demo: products1.js

8

取消操作

```
$scope.cancelEdit = function () {  
    $scope.currentProduct = {};  
    $scope.displayMode = "list";  
};
```

Demo: products1.js

3

使用\$resource服务

①

安装ngResource模块

ngResource模块是一个可选的angularjs模块，

如果需要使用，我们要单独引用js:

```
<script src="angular.js"></script>
```

```
<script src="angular-resource.js"><script>
```

依赖\$http服务

②

使用\$resource服务

\$resource服务是一个创建资源对象的工厂，底层封装了**\$http 服务**,用来创建同服务器交互的对象。返回的对象包含了同后端服务进行交互的方法，可以把User对象理解成同RESTFul的后端服务进行交互的接口。

\$resource (**url**, [**paramDefaults**], [**actions**]);

url参数

- 参数需加前缀冒号 : , 例 : /user/ : username
- 可以加端口号 , 例 : http://localhost:8080/products
- 可以使用后缀

②

使用\$resource服务

paramDefaults(可选参数,object类型)

- url参数中的默认值。
- 若任一参数为函数，则每当一次请求需要获取参数值时，都将被执行（除非参数被覆盖）。
- 参数对象中的每个键值对，若在url模板中存在则会被首先绑定，多余的键将被添加到url搜索查询（?之后）。
- 若参数值以【@】作为前缀，则该参数的值将被从data对象中取出（用于非GET操作）。

例：`$resource (url+ 'uid' ,{uid:' @id' });`

data中的id将被取出放到Url中

②

使用\$resource服务

action(可选参数,object类型)——自定义方法的集合。

声明应以\$http.config的格式创建：

```
{action1: {method:?, params:?, isArray:?, headers:?, ...},  
  action2: {method:?, params:?, isArray:?, headers:?, ...},  
  ...}
```

action：动作名。该名称成为你的资源对象的方法名。

method：HTTP请求方法。

合法的方法包括GET, POST, PUT, DELETE和JSONP。

isArray：本动作的返回对象是不是一个数组，true返回数组。

3

配置\$resource服务

存取对象定义的默认动作

名称	HTTP	URL	描述
query()	GET	/products/	获取所有对象的数组
get(id)	GET	/products/<id>	获取（单个）指定ID的对象
save(product)	POST	/products/<id>	保存指定ID的对象的修改
delete(params,product)	DELETE	/products/<id>	移除指定id对象
remove(params,product)	DELETE	/products/<id>	移除指定ID的对象

③

配置\$resource服务

resource对象支持的方法

名称	描述
\$get()	从服务器刷新对象，清理任何未提交的本地改变
\$save()	保存对象到服务器
\$delete()	从服务删除对象，等效于调用\$remove()
\$remove()	服务删除对象，等效于调用\$delete()

3

配置\$resource服务

```
var user = $resource('/products/:id', {id:'@id'});
```

1、user.query()

query方法用来请求多条数据。

2、user.\$get();

get方法向url发送一个get请求，并期望一个json类型的响应。

Demo: ex.html

Demo: ex.js

3

配置\$resource服务

3、 user. \$save()

save方法会发起一个post请求

4、 user. \$delete()

delete方法一个DELETE请求

Demo: ex.html

Demo: ex.js

4

应用\$resource服务

产品

名称	类型	价格	操作
香蕉	水果	\$2.50	<button>删除</button> <button>编辑</button> <button>+</button>
苹果	水果	\$1.20	<button>删除</button> <button>编辑</button> <button>+</button>
梨	水果	\$2.02	<button>删除</button> <button>编辑</button> <button>+</button>
金枪鱼	鱼	\$20.45	<button>删除</button> <button>编辑</button> <button>+</button>
鲑鱼	鱼	\$17.93	<button>删除</button> <button>编辑</button> <button>+</button>
鳕鱼	鱼	\$12.93	<button>删除</button> <button>编辑</button> <button>+</button>

刷新添加

Demo: products2.js

4

应用\$resource服务

```
angular.module("increment", [])  
  .directive("increment", function () {  
    return {  
      restrict: "E", //指令作为元素使用  
      scope: {  
        item: "=item",  
        property: "@propertyName",  
        restful: "@restful",  
        method: "@methodName"  
      },  
      link: function (scope, element, attrs) {  
        .....  
      }  
    }  
  })
```

Demo: increment.js

本课小结

1. 使用REST服务
2. 使用\$http服务
3. 隐藏Ajax请求

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net