

第十一讲 \$resource 数据交互插件

学习要点:

1. \$resource 简介
2. \$resource 的使用
3. 何时可以使用 Angular 资源 (选看)
4. Angularjs 自定义指令(后期讲)

主讲教师: (树根)

合作网站: www.phonegap100.com (PhoneGap 中文网)

合作网站: www.itying.com (IT 营)

1. \$resource 简介

\$resource:

\$resource 服务的核心价值在于: 提供给开发者与 **RESTful** 风格 WebServices 交互的更好的用户体验, 它封装了较为低级的\$http, 这样就不需要前端开发者写大量的异步请求代码了。

RESTful 架构:

- (1) 每一个 URI 代表一种资源;
- (2) 客户端和服务端之间, 传递这种资源的某种表现层;
- (3) 客户端通过四个 HTTP 动词, 对服务器端资源进行操作, 实现"表现层状态转化"。

RESTful 架构详细了解:

<http://www.ruanyifeng.com/blog/2011/09/restful.html>

2. \$resource 的使用

1. 引入 angular-resource.min.js
2. 定义模块时加载 ngResource

```
var app = angular.module('app',['ngResource']);
```

3. 写代码

<http://www.thinksaas.cn/group/topic/348581/>

[http://docs.angularjs.cn/api/ngResource/service/\\$resource](http://docs.angularjs.cn/api/ngResource/service/$resource)

```
$resource(url, [paramDefaults], [actions], options);
```

支持方法:

```
{
'get':    {method:'GET'},
'save':   {method:'POST'},
'query':  {method:'GET', isArray:true},
'remove': {method:'DELETE'},
'delete': {method:'DELETE'}
};
```

```
var User = $resource('/user/:userId', {userId:'@id'});
var user = User.get({userId:123}, function() {
    user.abc = true;
    user.$save();
});
```

后端支持跨域请求 **php** 为例子:

```
header('Access-Control-Allow-Origin: *');
```

3. 何时可以使用 Angular 资源（选看）

只有服务端按照 RESTful 的方式工作的时候，你才可以使用 Angular 资源。对于信用卡场景，它需要：

1. 一个到/user/123/card 的 GET 请求，它会返回用户 123 的信用卡列表。
2. 一个到/user/123/card/15 的 GET 请求，它会返回用户 123 的 ID 为 15 的信用卡。

3. 一个到/user/123/card 的 POST 请求，在 POST 的数据中带有信用卡信息，它将会为用户 123 的 ID 创建一张新的信用卡。

4. 一个到/user/123/card/15 的 POST 请求，POST 的数据中带有信用卡信息，它将会更新用户 123 的 ID 为 15 的信用卡信息。

5. 一个到/user/123/card/15 的 DELETE 请求，它将会删除用户 123 的 ID 为 15 的信用卡信息。

在学习如何使用 `ngResource` 方法创建资源之前，我们先来看看使用基本的 `$http` 服务创建类似的东西需要做些什么事情。对于我们的信用卡资源来说，除了要能够对它进行"change"（收费）操作之外，我们还要能够 `get`（获取）、`query`（查询）以及 `save`（保存）信用卡。

```
myAppModule.factory('CreditCard', ['$http', function($http) {
  var baseUrl = '/user/123/card';
  return {
    get: function(cardId) {
      return $http.get(baseUrl + '/' + cardId);
    },
    save: function(card) {
      var url = card.id ? baseUrl + '/' + card.id : baseUrl;
      return $http.post(url, card);
    },
    query: function() {
      return $http.get(baseUrl);
    },
    charge: function(card) {
      return $http.post(baseUrl + '/' + card.id, card, {params: {charge: true}});
    }
  };
}]);
```

除了这种方式之外，还可以简单地创建一个 `Angular` 服务，这个服务将会通过以下方式来描述应用所提供的资源：

```
myAppModule.factory('CreditCard', ['$resource', function($resource) {
  return $resource('/usr/:userId/card/:cardId',
    {userId: 123, cardId: '@id'},
```

```
{charge: {method: 'POST', params: {charge: true}, isArray: false}};  
});
```

感谢收看本次教程

本教程由 **phonegap 中文网(phonegap100.com)**

www.itying.com 提供

我是主讲老师： 树根

我的邮箱： **phonegap100@qq.com**