

• [首页](#) • [开源项目](#) • [问答](#) • [代码](#) • [博客](#) • [翻译](#) • [资讯](#) • [移动开发](#) • [招聘](#) • [城市圈](#)

当前访客身份：游客 [[登录](#) | [加入开源中国](#)]

在 37676 款开源软件中搜

软件 ▼

软件

搜索



小檀 [关注此人](#)

[关注\(124\)](#) [粉丝\(95\)](#) [积分\(96\)](#)



I have a dream.

[发送留言](#) [请教问题](#)

博客分类

- [java](#)(12)
- [Angular](#)(6)
- [javascript](#)(7)
- [MYSQL](#)(1)
- [工作日志](#)(1)
- [OC](#)(2)
- [日常记录](#)(1)

阅读排行

1. [1. Quartz2.2.1简单使用](#)
2. [2. AngularJS中service, factory, provider的区别](#)
3. [3. AngularJS 防止页面闪烁的方法](#)
4. [4. Quartz2.2.1配置文件](#)
5. [5. Angular路由的定义和使用](#)
6. [6. Spring注解方式实现任务调度](#)
7. [7. Lucene4.3.1 拼写检查SpellChecker](#)
8. [8. Lucene4.3.1 使用GroupingSearch对搜索结果进行分组](#)

最新评论

- [@娃娃脾气](#)：好！ [查看»](#)
- [@英强](#)：jQuery太强大了，解决了我们大部分的js需求。... [查看»](#)
- [@小檀](#)：引用来自“英强”的评论赞一个，支持第四种写法~... [查看»](#)
- [@英强](#)：赞一个，支持第四种写法~ [查看»](#)
- [@小檀](#)：引用来自“北落”的评论resolve的解决办法还不错... [查看»](#)
- [@北落](#)：resolve的解决办法还不错，angularjs还是提倡用p... [查看»](#)
- [@小檀](#)：引用来自“成熟的毛毛虫”的评论赞，下周要看到结... [查看»](#)
- [@成熟的毛毛虫](#)：赞，下周要看到结果喔 [查看»](#)
- [@小檀](#)：引用来自“小鱼吃虾米”的评论这个有问题吧？后... [查看»](#)
- [@小鱼吃虾米](#)：这个有问题吧？后来我又看了，教程里有说过是利... [查看»](#)

访客统计

- 今日访问：62
- 昨日访问：165
- 本周访问：227
- 本月访问：2380
- 所有访问：30920

[空间](#) » [博客](#) » [Angular](#)

原 AngularJS中service , factory , provider的区别

发表于12个月前(2014-07-26 11:35) 阅读 (5547) | 评论 (1) 9人收藏此文章, [我要收藏](#)

赞1

摘要 翻译, 原文链接: <http://www.ng-newsletter.com/advent2013/#!/day/1>

[AngularJS](#)

目录[-]

- [一、service引导](#)
- [二、service](#)
- [1、factory\(\)](#)
- [2、service\(\)](#)
- [3、provider\(\)](#)

一、service引导

刚开始学习Angular的时候, 经常被误解和被初学者问到的组件是 service(), factory(), 和 provide()这几个方法之间的差别。This is where we'll start the twenty-five days of Angular calendar.

二、service

在Angular里面, services作为单例对象在需要到的时候被创建, 只有在应用生命周期结束的时候(关闭浏览器)才会被清除。而controllers在不需要的时候就会被销毁了。

这就是为什么使用controllers在应用里面传递数据不可靠的原因, 特别是使用routing的时候。Services are designed to be the glue between controllers, the minions of data, the slaves of functionality, the worker-bees of our application (就是说services在应用的controllers、方法、数据之前起到了很关键的作用)

现在我们开始看怎么**创建service**。每个方法我们都会看到下面两个一样的参数:

- name-我们要定义的service的名字
- function-service方法

他们都创建了相同的底层对象类型。实例化后, 他们都创建了一个service, 这些对象没有什么功能上的差别。

1、factory()

Angular里面创建service最简单的方式是使用**factory()**方法。

factory()让我们通过返回一个包含service方法和数据的对象来定义一个service。在service方法里面我们可以注入services，比如 \$http 和 \$q等。

```
1 angular.module('myApp.services')
2 .factory('User', function($http) { // injectables go here
3   var backendUrl = "http://localhost:3000"; var service = { // our fa
4     user: {},
5     setName: function(newName) {
6       service.user['name'] = newName;
7     },
8     setEmail: function(newEmail) {
9       service.user['email'] = newEmail;
10    },
11    save: function() {
12      return $http.post(backendUrl + '/users', {
13        user: service.user
14      });
15    }
16  }; return service;
17 });
```

- 在应用里面使用factory()方法

在应用里面可以很容易地使用factory，需要到的时候简单地注入就可以了

```
1 angular.module('myApp')
2 .controller('MainCtrl', function($scope, User) {
3   $scope.saveUser = User.save;
4 });
```

- 什么时候使用factory()方法

在service里面当我们仅仅需要的是一个方法和数据的集合且不需要处理复杂的逻辑的时候，factory()是一个非常不错的选择。

注意：需要使用.config()来配置service的时候不能使用factory()方法

2、service()

service()通过构造函数的方式让我们创建service，我们可以使用原型模式替代JavaScript原始的对象来定义service。

和factory()方法一样我们也可以在函数的定义里面看到服务的注入

```
1 angular.module('myApp.services')
2 .service('User', function($http) { // injectables go here
3   var self = this; // Save reference
4   this.user = {};
5   this.backendUrl = "http://localhost:3000";
6   this.setName = function(newName) {
7     self.user['name'] = newName;
```

```
8     }
9     this.setEmail = function(newEmail) {
10         self.user['email'] = newEmail;
11     }
12     this.save = function() {
13         return $http.post(self.backendUrl + '/users', {
14             user: self.user
15         })
16     }
17 });
```

这里的功能和使用factory()方法的方式一样，service()方法会持有构造函数创建的对象。

- 在应用里面使用service()方法

```
1 angular.module('myApp')
2   .controller('MainCtrl', function($scope, User) {
3     $scope.saveUser = User.save;
4   });
```

- 什么时候适合使用service()方法

service()方法很适合使用在功能控制比较多的service里面

注意：需要使用.config()来配置service的时候不能使用service()方法

3、provider()

provider()是创建service最底层的方式，这也是唯一一个可以使用.config()方法配置创建service的方法

不像上面提到的方法那样，我们在定义的this.\$get()方法里面进行依赖注入

```
1 angular.module('myApp.services')
2   .provider('User', function() {
3     this.backendUrl = "http://localhost:3000";
4     this.setBackendUrl = function(newUrl) {
5       if (url) this.backendUrl = newUrl;
6     }
7     this.$get = function($http) { // injectables go here
8       var self = this;
9       var service = {
10         user: {},
11         setName: function(newName) {
12           service.user['name'] = newName;
13         },
14         setEmail: function(newEmail) {
15           service.user['email'] = newEmail;
16         },
17         save: function() {
18           return $http.post(self.backendUrl + '/users', {
19             user: service.user
20           })
21         }
22       };
23       return service;
24     }
25   });
```

- 在应用里面使用provider()方法

为了给service进行配置，我们可以将provider注入到.config()方法里面

```
1 angular.module('myApp')
2   .config(function(UserProvider) {
3     UserProvider.setBackendUrl("http://myApiBackend.com/api");
4   })
```

这样我们就可以和其他方式一样在应用里面使用这个service了

```
1 angular.module('myApp')
2   .controller('MainCtrl', function($scope, User) {
3     $scope.saveUser = User.save;
4   });
```

- 什么时候使用provider()方法

1. 当我们希望在应用开始前对service进行配置的时候就需要使用到provider()。比如，我们需要配置services在不同的部署环境里面（开发，演示，生产）使用不同的后端处理的时候就可以使用到了
2. 当我们打算发布开源provider()也是首选创建service的方法，这样就可以使用配置的方式来配置services而不是将配置数据硬编码写到代码里面。

点击查看完整的代码：<https://gist.github.com/auser/7743235>


分享到： 新浪微博  腾讯微博  1赞

声明：OSCHINA 博客文章版权属于作者，受法律保护。未经作者同意不得转载。

- [« 上一篇](#)
- [下一篇 »](#)

极光IM,
让你的App“聊”起来

JPush 极光推送



免费接入
www.jpusth.cn

评论1

-



1楼：[娃娃脾气](#) 发表于 2015-07-10 09:22 [回复此评论](#)
好！



插入： [表情](#) [开源软件](#)

发表评论

[关闭](#)插入表情

关闭相关文章阅读

- 2014/08/26 [AngularJs学习日记\[1\]：angularJs基...](#)
- 2014/04/06 [angularjs \\$http 提交post问题...](#)
- 2014/03/17 [AngularJS资料汇总](#)
- 2015/05/20 [AngularJS 的安全Apply](#)
- 2015/04/30 [angularjs directive 的几点使用技巧...](#)

© 开源中国(OSChina.NET) | [关于我们](#) | [广告联系](#) | [@新浪微博](#) | [开](#) 开源中国手机客户端：
[源中国手机版](#) | 粤ICP备12009483号-3

开源中国社区(OSChina.net)是工信部 [开源软件推进联盟](#) 指定的官方社区