

安装Chrome浏览器下载资源送30个下载分！

2011移动开发者大会亮点之二：七大论坛神秘嘉宾闪亮登场！

500元移动大会门票开抢！

"IT适合你吗？"智力挑战

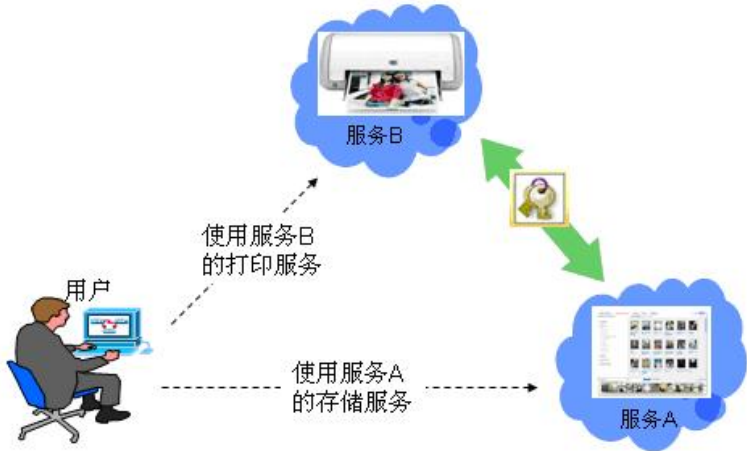
原 OAUTH协议简介

分类：Open API 2009-03-08 12:00 44934人阅读 评论(50) 收藏 举报

摘要：OAUTH协议为用户资源的授权提供了一个安全的、开放而又简易的标准。与以往的授权方式不同之处是OAUTH的授权不会使第三方触及到用户的帐号信息（如用户名与密码），即第三方无需使用用户的信息与密码就可以申请获得该用户资源的授权，因此OAUTH是安全的。同时，任何第三方都可以使用OAUTH认证服务，任何服务提供商都可以实现自身的OAUTH认证服务，因而OAUTH是开放的。业界提供了OAUTH的多种实现如PHP，JavaScript，Java，Ruby等各种语言开发包，大大节约了程序员的时间，因而OAUTH是简易的。目前互联网很多服务如Open API，很多大头公司如Google，Yahoo，Microsoft等都提供了OAUTH认证服务，这些都足以说明OAUTH标准逐渐成为开放资源授权的标准。

一、OAUTH产生的背景

典型案例：如果一个用户拥有两项服务：一项服务是图片在线存储服务A，另一个是图片在线打印服务B。如下图所示。由于服务A与服务B是由两家不同的服务提供商提供的，所以用户在这两家服务提供商的网站上各自注册了两个用户，假设这两个用户名各不相同，密码也各不相同。当用户要使用服务B打印存储在服务A上的图片时，用户该如何处理？法一：用户可能先将待打印的图片从服务A上下载下来并上传到服务B上打印，这种方式安全但处理比较繁琐，效率低下；法二：用户将在服务A上注册的用户名与密码提供给服务B，服务B使用用户的帐号再去服务A处下载待打印的图片，这种方式效率是提高了，但是安全性大大降低了，服务B可以使用用户的用户名与密码去服务A上查看甚至篡改用户的资源。



很多公司和个人都尝试解决这类问题，包括Google、Yahoo、Microsoft，这也促使OAUTH项目组的产生。OAuth是由Blaine Cook、Chris Messina、Larry Halff 及David Recordon共同发起的，目的在于为API访问授权提供一个开放的标准。OAuth规范的1.0版于2007年12月4日发布。通过官方网址：<http://oauth.net>可以阅读更多的相关信息。

二、OAUTH简介

个人资料

hereweare2009

访问：63412次
积分：424分
排名：第15561名

原创：8篇 转载：3篇
译文：0篇 评论：58条

文章搜索

文章分类

Java(2)
Open API(7)
云计算(1)
分布式存储(1)
虚拟化(1)

文章存档

2011年03月(1)
2009年08月(3)
2009年06月(1)
2009年05月(1)
2009年04月(1)

展开

阅读排行

OAUTH协议简介 (44933)
Google OAUTH + Open... (3791)
Google Open API授权认证体... (3595)

[淘宝 TOP 平台Open API入门篇 \(3300\)](#)
[淘宝Open API入门教程 \(3246\)](#)
[OpenID简介 \(1534\)](#)
[Amazon EBS弹性块存储服务初探 \(1083\)](#)
[Antrlr--看Hibernate3如何... \(646\)](#)
[REST初探 \(610\)](#)
[Google AppEngine vs.... \(342\)](#)

评论排行

[OAUTH协议简介 \(50\)](#)
[淘宝Open API入门教程 \(4\)](#)
[Google OAUTH + Openl... \(1\)](#)
[Google Open API授权认证体... \(1\)](#)
[淘宝 TOP 平台Open API入门篇 \(1\)](#)
[Google AppEngine vs.... \(1\)](#)
[OpenID简介 \(0\)](#)
[优秀程序员的十个习惯 \(0\)](#)
[Antrlr--看Hibernate3如何... \(0\)](#)
[REST初探 \(0\)](#)

最新评论

[OAUTH协议简介](#)
 gjlsx: 好文章，不错vuo
[OAUTH协议简介](#)
 xiaoyu90520: 技术贴，好文章。
[Google OAUTH + OpenID解决方案](#)
 xiha211: import os
[OAUTH协议简介](#)
 SMJ371817369: 刚才了解，谢谢！
[OAUTH协议简介](#)
 Arrui: 高质量博文，赞一个～
[OAUTH协议简介](#)
 xiang18711501944: 请问如果我想实现我的网站与腾讯互联 具体步骤是怎样啊，会不会很麻烦？楼主说的OAUTH 支持多种 ...
[OAUTH协议简介](#)
 a131988: 好文
[OAUTH协议简介](#)
 freemancqcsdn: 好文，赞。
[OAUTH协议简介](#)
 GQB_CMD: 好文章，顶起！！
[OAUTH协议简介](#)
 tecancy: 好文章

在官方网站的首页，可以看到下面这段简介：

An open protocol to allow secure API authorization in a simple and standard method from desktop and web applications.

大概意思是说OAUTH是一种开放的协议，为桌面程序或者基于BS的web应用提供了一种简单的，标准的方式去访问需要用户授权的API服务。OAUTH类似于Flickr Auth、Google's AuthSub、Yahoo's BBAuth、Facebook Auth等。OAUTH认证授权具有以下特点：

1. 简单：不管是OAUTH服务提供者还是应用开发者，都很容易于理解与使用；
2. 安全：没有涉及到用户密钥等信息，更安全更灵活；
3. 开放：任何服务提供商都可以实现OAUTH，任何软件开发商都可以使用OAUTH；

三、OAUTH相关术语

在弄清楚OAUTH流程之前，我们先了解下OAUTH的一些术语的定义：

- OAUTH相关的三个URL：
 - Request Token URL: 获取未授权的Request Token服务地址；
 - User Authorization URL: 获取用户授权的Request Token服务地址；
 - Access Token URL: 用授权的Request Token换取Access Token的服务地址；
- OAUTH相关的参数定义：
 - oauth_consumer_key: 使用者的ID，OAUTH服务的直接使用者是开发者开发出来的应用。所以该参数值的获取一般是要去OAUTH服务提供商处注册一个应用，再获取该应用的oauth_consumer_key。如Yahoo该值的注册地址为：<https://developer.yahoo.com/dashboard/>
 - oauth_consumer_secret: oauth_consumer_key对应的密钥。
 - oauth_signature_method: 请求串的签名方法，应用每次向OAUTH三个服务地址发送请求时，必须对请求进行签名。签名的方法有：HMAC-SHA1、RSA-SHA1与PLAINTEXT等三种。
 - oauth_signature: 用上面的签名方法对请求的签名。
 - oauth_timestamp: 发起请求的时间戳，其值是距1970 00:00:00 GMT的秒数，必须是大于0的整数。本次请求的时间戳必须大于或者等于上次的时间戳。
 - oauth_nonce: 随机生成的字符串，用于防止请求的重放，防止外界的非法攻击。
 - oauth_version: OAUTH的版本号，可选，其值必须为1.0。

OAUTH HTTP响应代码：

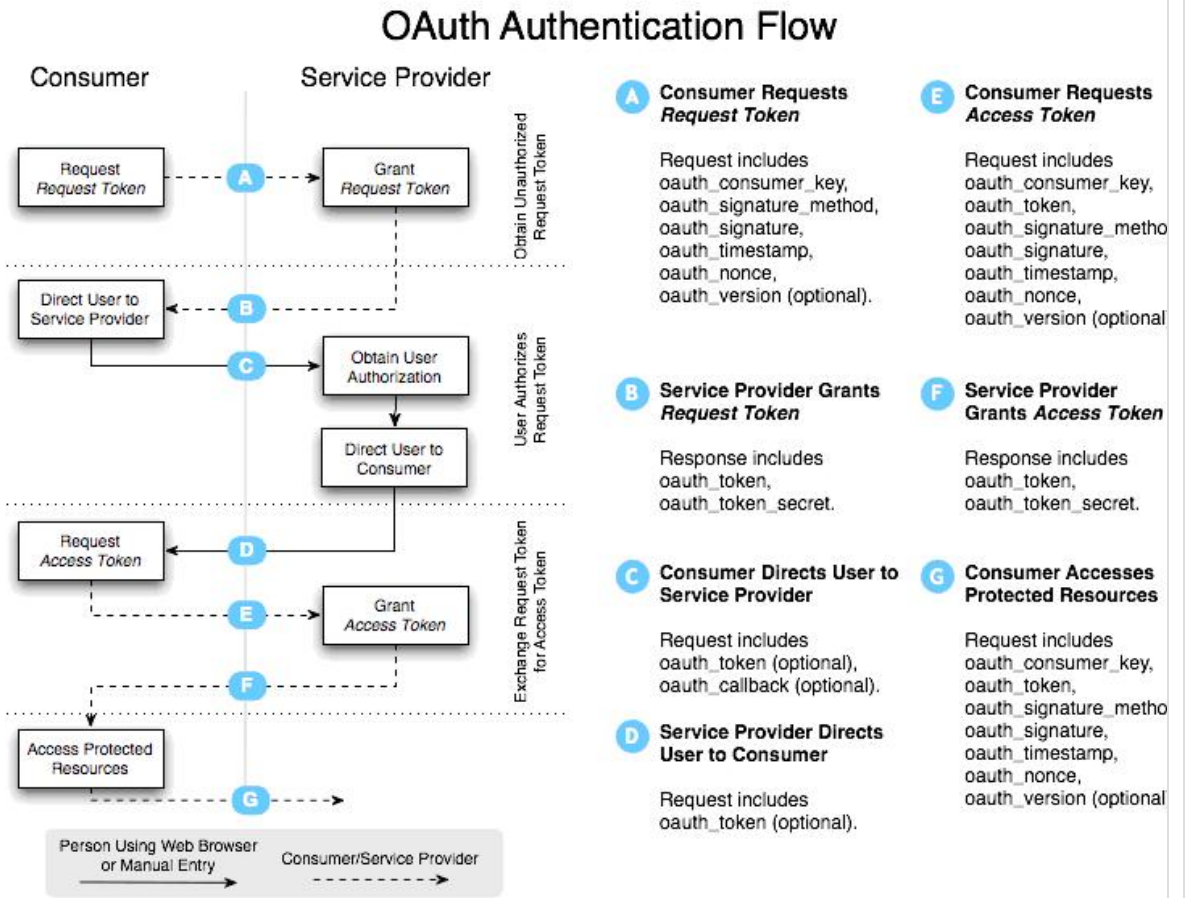
- HTTP 400 Bad Request 请求错误
 - Unsupported parameter 参数错误
 - Unsupported signature method 签名方法错误
 - Missing required parameter 参数丢失
 - Duplicated OAuth Protocol Parameter 参数重复
- HTTP 401 Unauthorized 未授权
 - Invalid Consumer Key 非法key
 - Invalid / expired Token 失效或者非法的token
 - Invalid signature 签名非法
 - Invalid / used nonce 非法的nonce

四、OAUTH认证授权流程

在弄清楚了OAUTH的术语后，我们可以对OAUTH认证授权的流程进行初步认识。其实，简单的来说，OAUTH认证授权就三个步骤，三句话可以概括：

1. 获取未授权的Request Token
2. 获取用户授权的Request Token
3. 用授权的Request Token换取Access Token

当应用拿到Access Token后，就可以有权访问用户授权的资源了。大家肯能看出来，这三个步骤不就是对应OAUTH的三个URL服务地址嘛。一点没错，上面的三个步骤中，每个步骤分别请求一个URL，并且收到相关信息，并且拿到上步的相关信息去请求接下来的URL直到拿到Access Token。具体的步骤如下图所示：



具体每步执行信息如下：

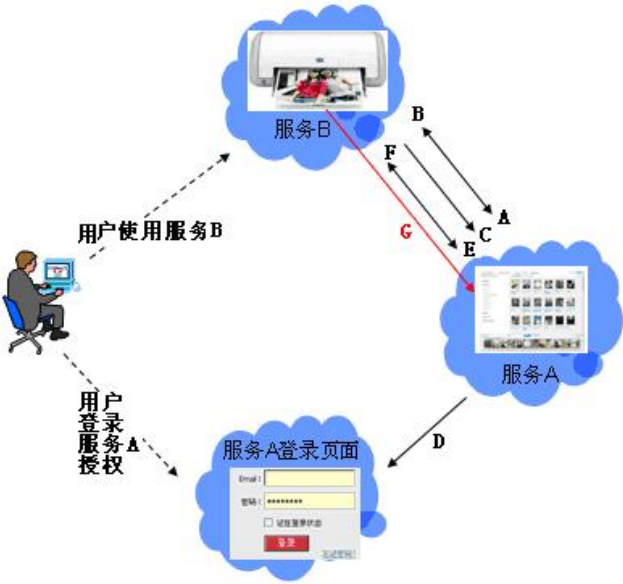
- 使用者（第三方软件）向OAUTH服务提供商请求未授权的Request Token。向Request Token URL发起请求，请求需要带上的参数见上图。
- OAUTH服务提供商同意使用者的请求，并向其颁发未经用户授权的oauth_token与对应的oauth_token_secret，并返回给使用者。
- 使用者向OAUTH服务提供商请求用户授权的Request Token。向User Authorization URL发起请求，请求带上上步拿到的未授权的token与其密钥。
- OAUTH服务提供商将引导用户授权。该过程可能会提示用户，你想将哪些受保护的资源授权给该应用。此步可能会返回授权的Request Token也可能不返回。如Yahoo OAUTH就不会返回任何信息给使用者。

E. Request Token 授权后，使用者将向Access Token URL发起请求，将上步授权的Request Token换取成Access Token。请求的参数见上图，这个比第一步A多了一个参数就是Request Token。

F. OAUTH服务提供商同意使用者的请求，并向其颁发Access Token与对应的密钥，并返回给使用者。

G. 使用者以后就可以使用上步返回的Access Token访问用户授权的资源。

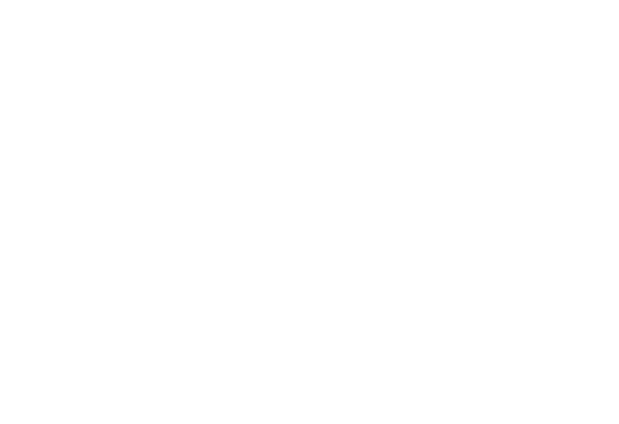
从上面的步骤可以看出，用户始终没有将其用户名与密码等信息提供给使用者（第三方软件），从而更安全。
 用OAUTH实现背景一节中的典型案例：当服务B（打印服务）要访问用户的服务A（图片服务）时，通过OAUTH机制，服务B向服务A请求未经用户授权的Request Token后，服务A将引导用户在服务A的网站上登录，并询问用户是否将图片服务授权给服务B。用户同意后，服务B就可以访问用户在服务A上的图片服务。整个过程服务B没有触及到用户在服务A的帐号信息。如下图所示，图中的字母对应OAUTH流程中的字母：

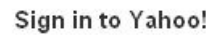


五、OAUTH服务提供商

OAUTH标准提出到现在不到两年，但取得了很大成功。不仅提供了各种语言的版本库，甚至Google，Yahoo，Microsoft等等互联网大头都实现了OAUTH协议。由于OAUTH的client包有很多，所以我们就没有必要再去自己写，避免重复造轮子，直接拿过来用就行了。我使用了这些库去访问Yahoo OAUTH服务，很不错哦！下面就贴出一些图片跟大家一起分享下！

下图是OAUTH服务提供商引导用户登录（若用户开始没有登录）





Yahoo! ID:

(e.g. free2rhyme@yahoo.com)

Password:

--

☐ Keep me signed in

for 2 weeks unless I sign out. **Info**
[Uncheck if on a shared computer]

Sign

[Forget your ID or password?](#) | [Help](#)

Don't have a Yahoo! ID?

Signing up is easy.

Sign

Welcome, [redacted]
[[Sign out](#), [My Account](#)]

[Yahoo! Home](#) - [H](#)

Allow this application to access your Yahoo! account.

You should not allow access if you do not trust this application with your data.

Developer's Description:
(Not reviewed by Yahoo!)

You are giving this developer and application access to:

[More Info](#)

 Yahoo! Profiles

- Obtain your shared and public profile data and profile data your Connections share. ?

 **Yahoo! Updates**

- Share your activities and manage the updates you receive. ?

 Yahoo! Status

- Obtain and update your status message. ?

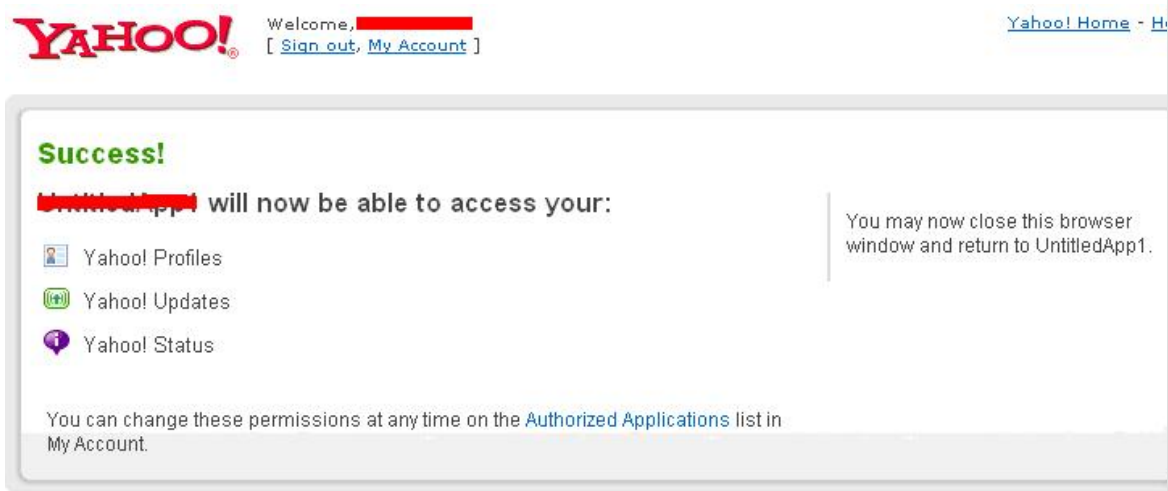
☒ I have read and agree to the [Yahoo! Additional Terms of Service](#)

☒ Allow access for 30 days only

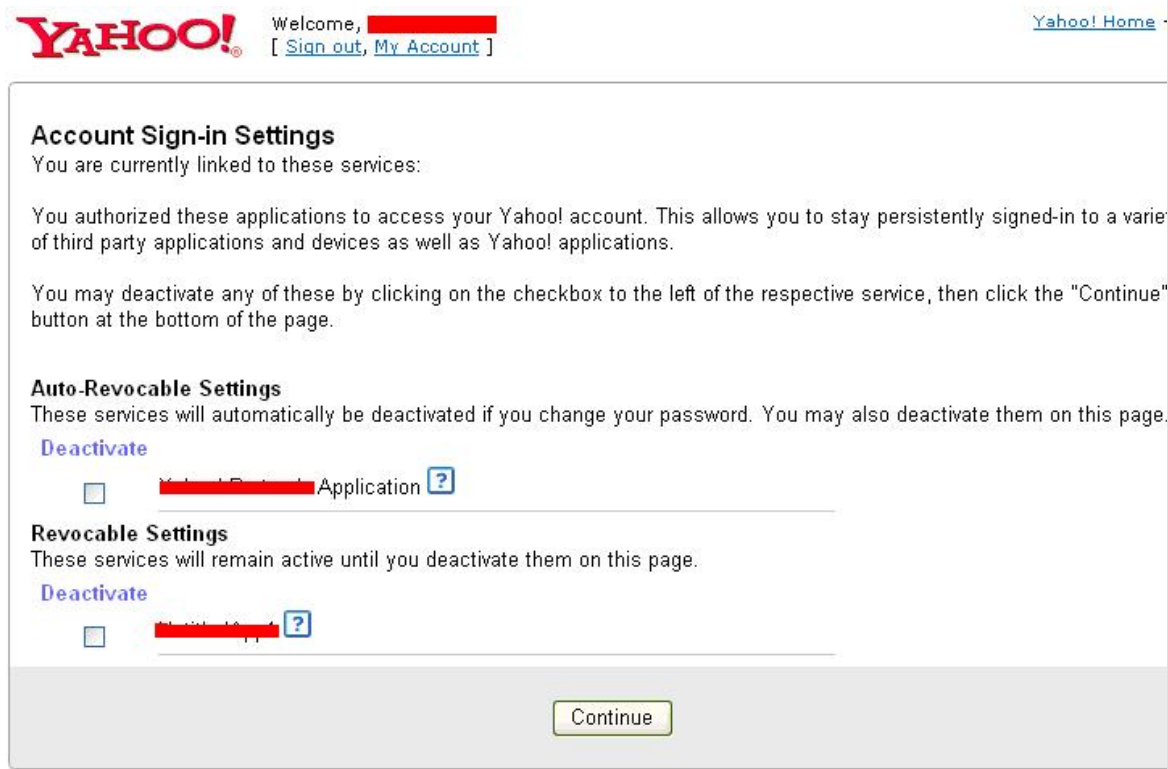
Allow Access

Cancel

下图提示用户已授权成功的信息






一些服务提供商不仅仅实现了OAUTH协议上的功能，还提供了一些更友好的服务，比如管理第三方软件的授权服务。下图就是YAHOO管理软件授权的页面，用户可以取消都某些应用的授权。




下一篇：[Google Open API授权认证体系](#)




分享到： 




查看评论





48  gjlsx 4  16:15 





 vuoto



- 




如何 使用oauth2.0 在 本地 使用http://www.f4tuan.com 本地
 
- 34 [fushituan](#) 2011-05-04 11:36
 
- 






如何 使用oauth2.0 在 本地 使用http://www.f4tuan.com 本地
 
- 33 [xiaoanian](#) 2011-04-03 00:44
 
- 








- 32 [MageShuai](#) 2011-04-01 13:17
 
- 






- 31 [Atwind](#) 2011-03-18 11:33
 
- 




APP 使用 AccessKey 使用
- 30 [chenfeng0104](#) 2011-03-10 17:45
 
- 










- 29 [think_try](#) 2011-03-06 11:13
 
- 








- 28 [huanglongmiss](#) 2011-02-21 11:49
 
- 


- 27 [pxwtf](#) 2011-02-14 12:34
 
- 
- 26 [ganziyi](#) 2011-01-10 10:12
 
- 


-
- 25 [hw0328](#) 2010-12-29 09:56
 
- 


- 24 [gauzeehom](#) 2010-12-21 17:28
 
- 



- 23 [mhfly54xfx](#) 2010-12-14 17:55
 
- 
- 22 [yliulldxz](#) 2010-12-08 15:06
 
- 
- 21 [syslogin](#) 2010-10-20 12:03
 
- 


- 20 [bellowWorld](#) 2010-09-30 20:51
 
- 




19 [fengmk2](#) 2010-09-27 09:23



oauth: http://bodianyong.com/ 😄



18 [xiaomeipingping](#) 2010-08-25 18:00



17 [tianhonghui](#) 2010-08-19 17:23



16 [hotnet522](#) 2010-08-17 16:58



Access Token



15 [gigi_1122](#) 2010-07-26 16:33



👉



14 2010-07-13 14:06



oauth

http://www.1smw.com 👨‍👩‍👧‍👦



13 [wgw335363240](#) 2010-07-11 23:38



12 [sol_dark](#) 2010-05-18 17:38



MSNQQOAUTH 🌹

10 [hereweare2009](#) 2010-04-07 19:45



hotnet522 2010-4-7 12:26:16 IP:
OAUTH

Oauth



!



8 [hotnet522](#) 2010-04-07 12:26





OAUTH










7 2010-03-31 15:20








6 2010-03-24 00:02










5    2010-03-22 16:30 








4  xumingming64398966 2010-03-21 13:05  








3    2010-03-11 23:26  



2  k463903468 2010-02-23 10:08  



1  dushimin920 2009-07-23 09:27  




您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

专区推荐内容

 MeeGo华丽转身，泰泽闪亮登场

 有一种速度叫Chrome，你体验，我送分！

 MeeGo开发常用测试工具及流程

 MeeGo开发中一些小技巧

 【示例演示】提升硬件及场景条件下的质量和性能目标


 【提升性能小工具】基于图像的后处理过滤技术

热门招聘职位

 【杭州贯通】急聘系统分析及.NET/JavaScript等各类网站开发人员

 招募“草根”手机应用提供商！——高薪网络兼职编辑

 【北塔软件】诚招JAVA软件开发/软件测试/FLEX开发/.net开发工程师

 【巨人网络】急聘Flash以及Web前台开发人才！

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告
北京创新乐知信息技术有限公司 版权所有，京 ICP 证 070598 号
世纪乐知(北京)网络技术有限公司 提供技术支持
江苏乐知网络技术有限公司 提供商务支持

 Email:webmaster@csdn.net

Copyright © 1999-2011, CSDN.NET, All Rights Reserved

