

Script

脚本之家

www.jb51.net

快云

快云服务器买多久送多久

JD购物卡 千万豪礼大放送！立即购买>>

阿里云

12核 / 16G / 2T

百兆独享 · 猛戳我

网页制作

网络编程

脚本专栏

脚本下载

数据库

CMS教程

电子书籍

平面设计

媒体动画

操作系统

网站运营

网络安全

在线手册

C#教程 vb vb.net C 语言 Java编程 Delphi java Android IOS Swift Scala 易语言 其它相关

您的位置： 首页 → 软件编程 → IOS → 正文内容 iOS的Keychain

请输入关键词 搜索

低价出售流量10000IP只需8元

装软件赚钱，每台电脑秒赚6.3元

众生网络 基于云计算的互联网基础服务运

枫信科技-江苏双线10M保证-399/元

鼎点网络百兆独享服务器仅需999元

DDoS高防IP，300G无限防

美国服务器，368元/月起，高速，稳定

服务器租用/托管-域名空间/认准腾佑科技

免备案国内虚拟主机-163cdn

[香港双高防]无视CC★DDOS/堪比广东！

16核独服350/固态硬盘独服288/创梦网络

微子网络 湛江双线高防 抗15G 700元起

BGP四线 亿恩1U服务器托管3999元/年

免备vps20/百独799/双线350/45互联

畅游网络 百独服务器 包跑满 998元

美国云20元/香港云29元/韩国云55元

脚本之家 百兆服务器租用、托管

九九数据 一 工信部认可正规资质IDC接入

韩国香港美国站群服务器 巨牛网络

服务器租用 199元起

西部数码

云主机

中国云主机领导品牌

仅需48元

上云市场

千款软件任您选

阿里云

4折

限量抢

1G仅售98元

7 阿里云 周年庆

感恩特惠

9.20-9.30

云服务器85折再送6个月，抢千元大红包

阿里云

230万用户信赖

详解iOS开发中Keychain的相关使用

作者：whj 字体：[增加 减小] 类型：转载 时间：2015-10-26 我要评论

这篇文章主要介绍了iOS开发中Keychain的相关使用,文中列举了一个使用Keychain来保存密码的例子,需要的朋友可以参考下

学习电脑编程

软件工程师薪水

app制作

app开发报价单

迷你公寓

ios编程入门

软件工程师月薪

加密码

u盘制作

ios培训班

艺术类出国留学

小字符喷码机

一、Keychain 基础

根据苹果的介绍，iOS设备中的Keychain是一个安全的存储容器，可以用来为不同应用保存敏感信息比如用户名，密码，网络密码，认证令牌。苹果自己用keychain来保存Wi-Fi网络密码，VPN凭证等等。它是一个sqlite数据库，位于/private/var/Keychains/keychain-2.db，其保存的所有数据都是加密过的。

开发者通常会希望能够利用操作系统提供的功能来保存凭证（credentials）而不是把它们（凭证）保存到NSUserDefaults,plist文件等地方。保存这些数据的原因是开发者不想用户每次都要登录，因此会把认证信息保存到设备上的某个地方并且在用户再次打开应用的时候用这些数据自动登录。Keychain的信息是存在于每个应用（app）的沙盒之外的。

通过keychain access groups可以在应用之间共享keychain中的数据。要求在保存数据到keychain的时候指定group。把数据保存到keychain的最好方法就是用苹果提供的KeychainItemWrapper。可以到这下载例子工程。第一步就是创建这个类的实例。

代码如下：

复制代码

KeychainItemWrapper \*wrapper = [[KeychainItemWrapper alloc] initWithIdentifier:@"Password" accessGroup:nil];

MacKeeper



清理您的 Mac

大家感兴趣的内容

1 iOS内存错误EXC\_BAD\_ACCESS的解决

2 iOS开发中ViewController的页面跳

3 IOS开发代码分享之设置UISearchBar

4 iOS开发中WebView的基本使用方法

5 iOS开发之路--仿网易抽屉效果

6 IOS开发代码分享之用nstimer实现

7 IOS获取各种文件目录路径的方法

8 iOS开发中实现显示gif图片的方法

9 iOS中使用schema协议调用APP和使

10 如何用IOS调用WebService ( SOAP接

标识符（Identifier）在后面我们要从keychain中取数据的时候会用到。如果你想要在应用之间共享信息，那么你需要指定访问组（access group）。有同样的访问组的应用能够访问同样的keychain信息。

代码如下：

[复制代码](#)

```
KeychainItemWrapper *wrapper = [[KeychainItemWrapper alloc] initWithIdentifier:@"Account Number" accessGroup:@"YOUR_APP_ID_HERE.com.yourcompany.GenericKeychainSuite"];
```

要把信息保存到keychain中，使用 setObject:forKey: 方法。在这里，(id)kSecAttrAccount 是一个预先定义好的键（key），我们可以用它来保存账号名称。kSecClass指定了我们要保存的某类信息，在这里是一个通用的密码。kSecValueData可以被用来保存任意的数据，在这里是一个密码。

代码如下：

[复制代码](#)

```
[wrapper setObject:kSecClassGenericPassword forKey:(id)kSecClass];  
[wrapper setObject:@"username" forKey:(id)kSecAttrAccount];  
[wrapper setObject:@"password" forKey:(id)kSecValueData];  
[wrapper setObject:(id)kSecAttrAccessibleAlwaysThisDeviceOnly forKey:(id)kSecAttrAccessible];
```

kSecAttrAccessibleIn变量用来指定这个应用合适需要访问这个数据。我们需要对这个选项特别注意，并且使用最严格的选项。这个键（key）可以设置6种值。

当然，我们应该绝对不要使用kSecAttrAccessibleAlways。一个安全点的选项是kSecAttrAccessibleWhenUnlocked。有些选项是以 ThisDeviceOnly 结尾的，如果选中了这个选项，那么数据就会被以硬件相关的密钥（key）加密，因此不能被传输到或者被其他设备看到。即使它们提供了进一步的安全性，使用它们可能不是一个好主意，除非你有一个更好的理由不允许数据在备份之间迁移。

要从keychain中获取数据，可以用 NSString \*accountName = [wrapper objectForKey:(id)kSecAttrAccount];

钥匙串中的条目称为SecItem，但它是存储在CFDictionary中的。SecItemRef类型并不存在。SecItem有五类：通用密码、互联网密码、证书、密钥和身份。在大多数情况下，我们用到的都是通用密码。许多问题都是开发人员尝试用互联网密码造成的。互联网密码要复杂得多，而且相比之下优势寥寥无几，除非开发Web浏览器，否则没必要用它。KeyChainItemWrapper只使用通用密码，这也是我喜欢它的原因之一。iOS应用很少将密钥和身份存储起来，所以我们在本书中不会讨论这方面的内容。只有公钥的证书通常应该存储在文件中，而不是钥匙串中。

最后，我们需要在钥匙串中搜索需要的内容。密钥有很多个部分可用来搜索，但最好的办法是将自己的标识符赋给它，然后搜索。通用密码条目都包含属性kSecAttrGeneric，可以用它来存储标识符。这也是KeyChainItemWrapper的处理方式。

钥匙串中的条目都有几个可搜索的\*\*属性\*\*和一个加密过的\*\*值\*\*。对于通用密码条目，比较重要的属性有账户（kSecAttrAccount）、服务（kSecAttrService）和标识符（kSecAttrGeneric）。而值通常是密码。

说明：

## MacKeeper



清理您的 Mac

阿里云

0基础1小时建站？  
上阿里云看看



### 最近更新的内容

百度地图PC端判断用户是否在配送范围内  
iOS中使用JSPatch框架使Objective-C与Java  
iOS中使用MD5加密字符串  
iOS实现裁剪框和图片剪裁功能  
汇总ios开发逆向传值的方法  
ios动态设置lbl文字标签的高度  
IOS改变UISearchBar中搜索框的高度  
iOS简单画板开发案例分享  
IOS开发中NSURL的基本操作及用法详解  
Android NavigationController 右滑手势

### 常用在线小工具

JavaScript正则在线测试工具  
在线图片格式转换(jpg/bmp/gif/png)工具  
JSON代码工具  
在线XML/JSON互相转换工具  
Unix时间戳(timestamp)转换工具  
CSS代码工具  
JavaScript代码格式化工具  
在线JSON代码检验/检验/美化/格式化  
在线JS脚本校验器错误  
JavaScript压缩/格式化/加密工具



的方法来操作KeyChain:

[复制代码](#)

```
ryRef query, CTypeRef *result);
```

```
butes, CTypeRef *result);
```

(CFDictionaryRef query, CFDictionaryRef attributesToUpdate)

达内培训  
(CFDictionaryRef query)

### 三、Keychain使用

引入Security包，引入文件 `#import <Security/Security.h>`

添加

代码如下:

[复制代码](#)

```
- (IBAction)add:()sender {
    (nameField.text.length > 0 && passwordField.text.length > 0) {
        NSMutableDictionary* dic = [NSMutableDictionary dictionary];
        [dic setObject:(())kSecClassGenericPassword forKey:(())kSecClass];
        [dic setObject:nameField.text forKey:(())kSecAttrAccount];
        [dic setObject:[passwordField.text dataUsingEncoding:NSUTF8StringEncoding] forKey:(())kSec
cValueData];
    }
}
```



```
        OSStatus s = SecItemAdd((CFDictionaryRef)dic, NULL);
        NSLog(,s);
    }
}
```

## 查找

代码如下:

[复制代码](#)

```
- (IBAction)sel:()sender {
    NSDictionary* query = [NSDictionary dictionaryWithObjectsAndKeys:kSecClassGenericPassword,kSecClass,
        kSecMatchLimitAll,kSecMatchLimit,
        kCFBooleanTrue,kSecReturnAttributes,nil];
    CTypeRef result = nil;
    OSStatus s = SecItemCopyMatching((CFDictionaryRef)query, &result);
    NSLog(,s);
    NSLog(,result);
}

- (IBAction)sname:()sender {
    (nameField.text.length >) {
        NSDictionary* query = [NSDictionary dictionaryWithObjectsAndKeys:kSecClassGenericPassword,kSecClass,
            nameField.text,kSecAttrAccount,
            kCFBooleanTrue,kSecReturnAttributes,nil];
        CTypeRef result = nil;
        OSStatus s = SecItemCopyMatching((CFDictionaryRef)query, &result);
        NSLog(,s);    NSLog(,result);
        (s == noErr) {
            NSMutableDictionary* dic = [NSMutableDictionary dictionaryWithDictionary:result];
            [dic setObject:(kCFBooleanTrue forKey:kSecReturnData];
            [dic setObject:[query objectForKey:kSecClass] forKey:kSecClass];
            NSData* data = nil;
            (SecItemCopyMatching((CFDictionaryRef)dic, (CTypeRef*)&data) == noErr) {
                (data.length)
                NSLog(,[[[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding] autorelease]);
            }
        }
    }
}
```

## 修改

代码如下:

[复制代码](#)

```
- (IBAction)update:()sender {
    (nameField.text.length > && passwordField.text.length > ) {
        NSDictionary* query = [NSDictionary dictionaryWithObjectsAndKeys:kSecClassGenericPassword,kSecClass,
            nameField.text,kSecAttrAccount,
            kCFBooleanTrue,kSecReturnAttributes,nil];
```

```

CTypeRef result = nil;
(SecItemCopyMatching((CFDictionaryRef)query, &result) == noErr)
{
    NSMutableDictionary* update = [NSMutableDictionary dictionaryWithDictionary:(NSDi
ctionary*)result];
    [update setObject:[query objectForKey:kSecClass] forKey:kSecClass];
    [update setObject:[passwordField.text dataUsingEncoding:NSUTF8StringEncoding] forKey:kSe
cValueData];
    [update removeObjectForKey:kSecClass];
    TARGET_IPHONE_SIMULATOR
    [update removeObjectForKey:()kSecAttrAccessGroup];

    NSMutableDictionary* updateItem = [NSMutableDictionary dictionaryWithDictionary:r
esult];
    [updateItem setObject:[query objectForKey:()kSecClass] forKey:()kSecClass];
    OSStatus status = SecItemUpdate((CFDictionaryRef)updateItem, (CFDictionaryRef)up
date);
    NSLog(,status);
}

```

## 删除

代码如下:

[复制代码](#)

```

- (IBAction)del:(id)sender {
    (nameField.text.length >) {
        NSDictionary* query = [NSDictionary dictionaryWithObjectsAndKeys:kSecClassGenericPass
word,kSecClass,
                                nameField.text,kSecAttrAccount,nil];
        OSStatus status = SecItemDelete((CFDictionaryRef)query);
        NSLog(,status);    }
}

```

## 四、保存密码实例

来看一下使用keychain保存密码的例子:

代码如下:

[复制代码](#)

```

@implementation WQKeyChain
+ (NSMutableDictionary *)getKeychainQuery:(NSString *)service {
return [NSMutableDictionary dictionaryWithObjectsAndKeys:
    (__bridge_transfer id)kSecClassGenericPassword,(__bridge_transfer id)kSecClass,
    service, (__bridge_transfer id)kSecAttrService,
    service, (__bridge_transfer id)kSecAttrAccount,
    (__bridge_transfer id)kSecAttrAccessibleAfterFirstUnlock,(__bridge_transfer id)kSecAttrAccessible
,
    nil];
}

+ (void)save:(NSString *)service data:(id)data {
    //Get search dictionary
    NSMutableDictionary *keychainQuery = [self getKeychainQuery:service];
    //Delete old item before add new item
    SecItemDelete((__bridge_retained CFDictionaryRef)keychainQuery);
    //Add new object to search dictionary(Attention:the data format)
    [keychainQuery setObject:[NSKeyedArchiver archivedDataWithRootObject:data] forKey:(__bridge_tr

```



```

transfer id)kSecValueData];
//Add item to keychain with the search dictionary
SecItemAdd((__bridge_retained CFDictionaryRef)keychainQuery, NULL);
}

+ (id)load:(NSString *)service {
    id ret = nil;
    NSMutableDictionary *keychainQuery = [self getKeychainQuery:service];
    //Configure the search setting
    [keychainQuery setObject:(id)kCFBooleanTrue forKey:(__bridge_transfer id)kSecReturnData];
    [keychainQuery setObject:(__bridge_transfer id)kSecMatchLimitOne forKey:(__bridge_transfer id)kSecMatchLimit];
    CFDataRef keyData = NULL;
    if (SecItemCopyMatching((__bridge_retained CFDictionaryRef)keychainQuery, (CTypeRef *)&keyData) == noErr) {
        @try {
            ret = [NSKeyedUnarchiver unarchiveObjectWithData:(__bridge_transfer NSData *)keyData];
        } @catch (NSEException *e) {
            NSLog(@"Unarchive of %@ failed: %@", service, e);
        } @finally {
        }
    }
    return ret;
}

+ (void)delete:(NSString *)service {
    NSMutableDictionary *keychainQuery = [self getKeychainQuery:service];
    SecItemDelete((__bridge_retained CFDictionaryRef)keychainQuery);
}

@end

```

代码如下:

[复制代码](#)

```

@interface WQUserDataManager : NSObject

/**
 * @brief 存储密码
 *
 * @param password 密码内容
 */
+ (void)savePassWord:(NSString *)password;

/**
 * @brief 读取密码
 *
 * @return 密码内容
 */
+ (id)readPassWord;

/**
 * @brief 删除密码数据
 */
+ (void)deletePassWord;

```

```
@end
```

代码如下:

[复制代码](#)

```
#import "WQUserDataManager.h"

@implementation WQUserDataManager

static NSString * const KEY_IN_KEYCHAIN = @"com.wuqian.app.allinfo";
static NSString * const KEY_PASSWORD = @"com.wuqian.app.password";

+(void)savePassWord:(NSString *)password
{
    NSMutableDictionary *usernamepasswordKVPairs = [NSMutableDictionary dictionary];
    [usernamepasswordKVPairs setObject:password forKey:KEY_PASSWORD];
    [WQKeyChain save:KEY_IN_KEYCHAIN data:usernamepasswordKVPairs];
}

+(id)readPassWord
{
    NSMutableDictionary *usernamepasswordKVPair = (NSMutableDictionary *)[WQKeyChain load:KEY_IN_KEYCHAIN];
    return [usernamepasswordKVPair objectForKey:KEY_PASSWORD];
}

+(void)deletePassWord
{
    [WQKeyChain delete:KEY_IN_KEYCHAIN];
}

@end
```

实现一个简单的界面，把设定的密码存起来，然后立即读取显示出来看看效果

代码如下:

[复制代码](#)

```
-(IBAction)btnAciton:(id)sender
{
    [WQUserDataManager savePassWord:self.textfield.text];
    self.label.text = [WQUserDataManager readPassWord];
}
```



X





X



迷你公寓

集成吊顶加盟

ui设计师是做什..

集成墙面哪家好

集成墙面装修

什么叫ui..

app开发报价单

集成墙面价格

嘉兴二手房

学习电脑编程

艺术类出国留学

界面设计..

ui设计教程

集成灶品牌排行榜

ui设计作品

app开发公司

erp管理系统

快速开发平台

oa系统

web前端工程师

后台模板

ui设计欣赏

p2p网贷系统

u盘定制

Tags: iOS Keychain

相关文章

<a href="#">深入解析iOS应用开发中对设计模式中的桥接模式的使用</a>	2016-03-03
<a href="#">IOS开发代码分享之用NSTimer实现倒计时功能</a>	2014-09-09
<a href="#">iOS App初次启动时的用户引导页制作实例分享</a>	2016-03-03
<a href="#">IOS代码笔记之下拉菜单效果</a>	2016-07-07
<a href="#">ios下移动文件方法汇总</a>	2015-05-05

<a href="#">IOS中实现图片点击全屏预览</a>	2015-11-11
<a href="#">理解Objective-C的变量以及面相对象的继承特性</a>	2016-01-01
<a href="#">ios开发中时间转换的方法集锦</a>	2015-05-05
<a href="#">iOS App设计模式开发中对迭代器模式的使用示例</a>	2016-03-03
<a href="#">iOS 中使用tableView实现右滑显示选择功能</a>	2016-07-07



学习电脑编程



软件工程师薪水



墙面集成



软件工程师月薪



迷你公寓



集成墙面哪家好



集成吊顶加盟



C++培训



u盘价格



集成墙面价格



集成墙面批发



集成墙面怎么样

最新评论

评论(0人参与，0条评论)



来说两句吧...

微博登录

QQ登录

手机登录

还没有评论，快来抢沙发吧！

Powered by 畅言