

<http://www.cocoachina.com>

iOS开发 (<http://www.cocoachina.com/ios/>) Swift (<http://www.cocoachina.com/swift/>) App Store研究 (<http://www.cocoachina.com/appstore/>)

产品设计 (<http://www.cocoachina.com/design/>) 应用 (<http://www.cocoachina.com/review/>) VR (<http://www.cocoachina.com/vr/>)

游戏开发 (<http://www.cocoachina.com/game/>) 苹果相关 (<http://www.cocoachina.com/apple/>) 安卓相关 (<http://www.cocoachina.com/android/>)

营销推广 (<http://www.cocoachina.com/market/>) 业界动态 (<http://www.cocoachina.com/industry/>)

程序人生 (<http://www.cocoachina.com/programmer/>)

博客 (<http://blog.cocoachina.com/227971/1>) > iOS剪切板 UIPasteboard

iOS剪切板 UIPasteboard

2016-10-12 12:11

tianya2416 阅读: 624 评论: 0 赞: 2

UIPasteboard 剪切板

写博客 (/writer)

<http://blog.cocoachina.com/227971/>

tianya2416

<http://blog.cocoachina.com/227971/>

阅读: 6142

评论: 2

赞: 24

文章目录

- 1、前言
- 2、iOS自带剪切板操作的原生UI控件
- 3、系统的剪切板 UIPasteboard

版权声明: 本文为博主原创文章, 未经博主授权不得转载。

1、前言

iOS在某些特定时刻需要把内容从一个app拷贝到另一个app 这时候我们就可以使用剪切板UIPasteboard

2、iOS自带剪切板操作的原生UI控件

在iOS中下面三个控件, 自身就有复制-粘贴的功能: (1)、UITextView (2)、UITextField (3)、UIWebView

3、系统的剪切板UIPasteboard

UIPasteboard类有3个初始化方法:

```
//获取系统级别的剪切板
+ (UIPasteboard *)generalPasteboard;
//获取一个自定义的剪切板 name参数为此剪切板的名称 create参数用于设置当这个剪切板不存在时 是否进行创建
+ (nullable UIPasteboard *)pasteboardWithName:(NSString *)pasteboardName create:(BOOL)create;
//获取一个应用内可用的剪切板
+ (UIPasteboard *)pasteboardWithUniqueName;
```

上面3个初始化方法, 分别获取或创建3个级别不同的剪切板, 系统级别的剪切板在整个设备中共享, 即是应用程序被删掉, 其向系统级的剪切板中写入的数据依然在。自定义的剪切板通过一个特定的名称字符串进行创建, 它在应用程序内或者同一开发者开发的其他应用程序中可以进行数据共享。第3个方法创建的剪切板等价于使用第2个方法创建的剪切板, 只是其名称字符串为nil, 它通常用于当前应用内部。

注意: 使用第3个方法创建的剪切板默认是不进行数据持久化的, 及当应用程序退出后, 剪切板中内容将别抹去。若要实现持久化, 需要设置persistent属性为YES。

```
UIPasteboard中常用方法及属性如下:
//剪切板的名称
@property(readonly, nonatomic) NSString *name;
//根据名称删除一个剪切板
+ (void)removePasteboardWithName:(NSString *)pasteboardName;
//是否进行持久化
@property(getter=isPersistent, nonatomic) BOOL persistent;
//此剪切板的改变次数 系统级别的剪切板只有当设备重新启动时 这个值才会清零
@property(readonly, nonatomic) NSInteger changeCount;
```

下面这些方法用于设置与获取剪切板中的数据:

最新一组数据对象的存取:

```
//获取剪切板中最新数据的类型
- (NSArray<NSString> *)pasteboardTypes;
//获取剪切板中最新数据对象是否包含某一类型的数据
- (BOOL)containsPasteboardTypes:(NSArray<NSString> *)pasteboardTypes;
//将剪切板中最新数据对象某一类型的数据取出
- (nullable NSData *)dataForPasteboardType:(NSString *)pasteboardType;
//将剪切板中最新数据对象某一类型的值取出
- (nullable id)valueForPasteboardType:(NSString *)pasteboardType;
//为剪切板中最新数据对应的某一数据类型设置值
- (void)setValue:(id)value forPasteboardType:(NSString *)pasteboardType;
//为剪切板中最新数据对应的某一数据类型设置数据
- (void)setData:(NSData *)data forPasteboardType:(NSString *)pasteboardType;
```

多组数据对象的存取:

```
//数据组数
@property(readonly, nonatomic) NSInteger numberOfItems;
//获取一组数据对象包含的数据类型
- (nullable NSArray *)pasteboardTypesForItemSet:(nullable NSIndexSet *)itemSet;
//获取一组数据对象中是否包含某些数据类型
- (BOOL)containsPasteboardTypes:(NSArray<NSString> *)pasteboardTypes inItemSet:(nullable NSIndexSet *)itemSet;
//根据数据类型获取一组数据对象
- (nullable NSIndexSet *)itemSetWithPasteboardTypes:(NSArray *)pasteboardTypes;
//根据数据类型获取一组数据的值
- (nullable NSArray *)valuesForPasteboardType:(NSString *)pasteboardType inItemSet:(nullable NSIndexSet *)itemSet;
//根据数据类型获取一组数据的NSData数据
- (nullable NSArray *)dataForPasteboardType:(NSString *)pasteboardType inItemSet:(nullable NSIndexSet *)itemSet;
//所有数据对象
@property(nonaatomic, copy) NSArray *items;
//添加一组数据对象
- (void)addItem:(NSDictionary *)item;
```

上面方法中很多需要传入数据类型参数, 这些参数是系统定义好的一些字符窜, 如下:

```
//所有字符串类型数据的类型定义字符串数组
UIKIT_EXTERN NSArray<NSString> *UIPasteboardTypeListString;
//所有URL类型数据的类型定义字符串数组
UIKIT_EXTERN NSArray<NSString> *UIPasteboardTypeListURL;
//所有图片数据的类型定义字符串数组
UIKIT_EXTERN NSArray<NSString> *UIPasteboardTypeListImage;
//所有颜色数据的类型定义字符串数组
UIKIT_EXTERN NSArray<NSString> *UIPasteboardTypeListColor;
```

相比于上面两组方法, 下面这些方法更加面向对象, 在开发中使用更加方便与快捷:

```
//获取或设置剪切板中的字符串数据
@property(nullable, nonatomic, copy) NSString *string;
//获取或设置剪切板中的字符串数组
@property(nullable, nonatomic, copy) NSArray<NSString> *strings;
//获取或设置剪切板中的URL数据
@property(nullable, nonatomic, copy) NSURL *URL;
//获取或设置剪切板中的URL数组
@property(nullable, nonatomic, copy) NSArray<NSURL> *URLs;
//获取或设置剪切板中的图片数据
@property(nullable, nonatomic, copy) UIImage *image;
//获取或设置剪切板中的图片数组
@property(nullable, nonatomic, copy) NSArray<UIImage> *images;
//获取或设置剪切板中的颜色数据
@property(nullable, nonatomic, copy) UIColor *color;
//获取或设置剪切板中的颜色数组
@property(nullable, nonatomic, copy) NSArray<UIColor> *colors;
```

对剪切板的某些操作会触发如下通知:

```
//剪切板内容发生变化时发送的通知
UIKIT_EXTERN NSString *const UIPasteboardChangedNotification;
//剪切板数据类型键值增加时发送的通知
UIKIT_EXTERN NSString *const UIPasteboardChangedTypesAddedKey;
//剪切板数据类型键值移除时发送的通知
UIKIT_EXTERN NSString *const UIPasteboardChangedTypesRemovedKey;
//剪切板被删除时发送的通知
UIKIT_EXTERN NSString *const UIPasteboardRemovedNotification;
```

未完待续...

收藏

分享到:

2

相关博文

UILabel 实现长按复制功能 (<http://blog.cocoachina.com/articl...>)

我来说两句



你怎么看? 快来评论一下吧!

您还没有登录! 请 [登录 \(/signin\)](#) 或 [注册 \(/signup\)](#)

发表评论

所有评论 (0)

关于我们 (<http://www.cocoachina.com/aboutus/index.html>)

商务合作 (<http://www.cocoachina.com/contactus/>)

联系我们 (<mailto:support@cocoachina.com>)

合作伙伴

北京触控科技有限公司版权所有
(<http://www.cocoachina.com/corps/>)
©2015 Chukong Technologies, Inc.

京ICP备 11006519号 京ICP证 100954号 (<http://www.miibeian.gov.cn/>) 京公网安备11010502020289 (http://182.131.21.137/ccnt-apply/admin/business/preview/business-preview?lookUrlRFID.action?main_id=A580B24B1AE846CE9DA949026D63939F)