

iOS应用开发流程之加密工具类

浏览：160 发布日期：2016-09-09 分类：[ios](#)

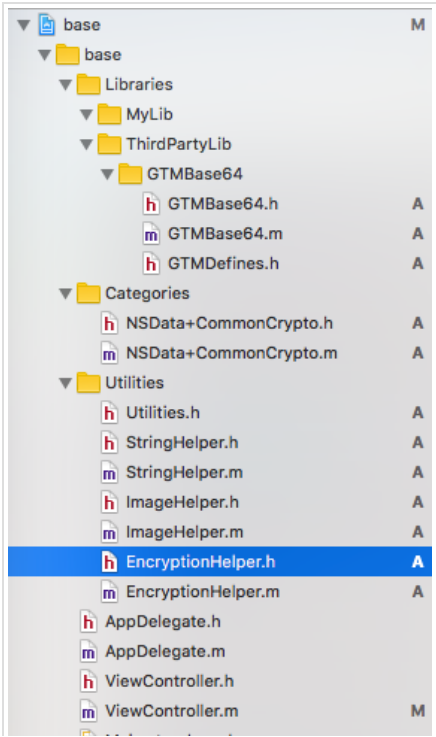
我建议先将基础的工具加入项目，后续的开发效率会呈指数增长。如果在专注功能开发过程中，才发现缺少大量常用的工具，不仅会打断思路，还会拖慢开发节奏。

当然，在每个项目开始的时候，不可能将全部工具都准备充分，只能依据个人的经验来评估需要提前准备的工具。

一个好的工匠，必须要有一个好的工具箱，并且还要不断优化它。

稍微扩展一下项目架构。

- 1.在base目录下为项目增加“Utilities”文件夹作为工具箱，并创建“Utilities.h”头文件，在PrefixHeader.pch中引用该头文件
- 2.暂时先创建三个NSObject子类“StringHelper”，“ImageHelper”，“EncryptionHelper”，作为三个工具类，并加入Utilities.h的引用列表
- 3.在base目录下再增加两个结构“Categories”和“Libraries”，分别用于存放分类和类库
- 4.我进一步将类库分为了第三方类库和自创建类库



收藏	赞	浏览
3	0	160

1

--

热门推荐

- 1 Android常用的工具类
- 2 JavaScript-数组去重由慢...
- 3 12个用得着的JQuery代码...
- 4 简单又好用的聊天室技术一...
- 5 让广大开发者相见恨晚的A...

最新更新

- 1 gulp前端构建工具白话讲...
- 2 javascript基础之String
- 3 react-router 按需加载
- 4 「daza.io」这将是独立...
- 5 立足Docker运行MySQL：...

加密工具类EncryptionHelper

暂时记录了base64编码和解码，MD2、MD4、MD5、SHA1、SHA224、SHA256、SHA384、SHA512加密，AES、DES加密和解密。

1.base64编码和解码

因为使用了Google的GTMBase64类库，所以讲此类库加入第三方库目录。

此类库代码中使用了autorelease将对象加入自动释放池，在ARC的项目中，可以将用到此类代码清除。

也可以在项目Target -> Build Phases -> Compile Sources，找到GTMBase64.m，双击后，在右边输入框内输入-fno-objc-arc，将此文件移除ARC管理。

在EncryptionHelper.m中引用GTMBase64.h后，可以实现如下方法：

```
+ (NSString *)base64EncodeWithString:(NSString *)string
{
    NSData *data = [string dataUsingEncoding:NSUTF8StringEncoding];

    NSString *result = [GTMBase64 stringByEncodingData:data];

    return result;
}

+ (NSString *)decodeBase64WithString:(NSString *)string
{
    NSData *data = [GTMBase64 decodeString:string];

    NSString *result = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];

    return result;
}
```

2.MD5加密

因为不可逆（大概可以理解为无数多解的二元一次方程），所以就只有加密算法可记录

```

+ (NSString *)MD5SumWithString:(NSString *)string
{
    NSData *data = [string dataUsingEncoding:NSUTF8StringEncoding];

    uint8_t buffer[CC_MD5_DIGEST_LENGTH];

    CC_MD5(data.bytes, (CC_LONG)data.length, buffer);

    //如下注释代码与上述加密代码等效
    // data = [data MD5Sum];
    // Byte *buffer = (Byte *)data.bytes;

    NSMutableString *result = [NSMutableString stringWithCapacity:CC_MD5_DIGEST_LENGTH];

    for(int i = 0; i < CC_MD5_DIGEST_LENGTH; i++){
        [result appendFormat:@"%02x", buffer[i]];
    }

    return result;
}

```

说明：

A.需要特别强调的一点，看到许多开发者都采用如下方法将NSString转为NSData：

```
const char *cstr = [string cStringUsingEncoding:NSUTF8StringEncoding];
```

```
NSData *data = [ NSData dataWithBytes :cstr length :string.length];
```

我不建议采用上述代码，经过测试，当字符串为纯英文字母、数字、符号时候，结果无异；但是当字符串包含中文时候，string.length取到的长度就有问题，因为一个中文字符转为NSData以后长度大于1，所以上述方法将导致中文转化后，内容丢失！

建议使用如下方法：NSData *data = [string dataUsingEncoding:NSUTF8StringEncoding];

B.MD2、MD4、SHA1、SHA224、SHA256、SHA384、SHA512加密算法，只需要将示例代码中buffer的长度 CC_MD5_DIGEST_LENGTH 和加密方法 CC_MD5 更换为对应名称即可。

C.注释代码作用与前两行一致。不过需要引用分类 NSData+CommonCrypto.h。

D.data.bytes可以强转为(Byte *)和(char *)，可以参考下述代码

E.result字符串的预留长度为CC_MD5_DIGEST_LENGTH * 2，因为buffer数组中元素类型为uint8_t（也就是unsigned char），在字符串格式化时候，以16进制形式显示两位，所以长度为2倍

F.%02x占位符表示以16进制显示两位，位数不足左边补0

3.AES、DES加密和解密

AES 和 DES 加密过程：string -> data -> AES/DES encrypt -> base64 encode -> string; 解密过程为逆向

```

+ (NSString *)AES256EncryptedString:(NSString *)string usingKey:(NSString
{
    NSData *data = [string dataUsingEncoding:NSUTF8StringEncoding];
    NSData *encryptedData = [data AES256EncryptedDataUsingKey:key error:nil];
    NSData *base64Data = [GTMBASE64 encodeData:encryptedData];

    NSString *result = [[NSString alloc] initWithData:base64Data encoding:NSUTF8StringEncoding];

    // const char *buffer = (char *)base64Data.bytes;
    // result = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    // result = [[NSString alloc] initWithBytes:base64Data.bytes length:base64Data.length encoding:NSUTF8StringEncoding];

    return result;
}

+ (NSString *)decryptedAES256String:(NSString *)string usingKey:(NSString *)key
{
    NSData *base64Data = [string dataUsingEncoding:NSUTF8StringEncoding];
    NSData *decryptedData = [GTMBASE64 decodeData:base64Data];
    NSData *data = [decryptedData decryptedAES256DataUsingKey:key error:nil];

    NSString *result = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];

    return result;
}

```

此处增加了一个分类NSData+CommonCrypto，并在EncryptionHelper.m中引用，提供了AES和DES加密解密算法，感兴趣的朋友可以细看。

Github地址：<https://github.com/ALongWay/AESCrypt-ObjC>

说明：

A.AES加密方法中，生成result字符串，罗列了3种方式

B.DES加密过程中，只需将 AES256EncryptedDataUsingKey替换为 DESEncryptedDataUsingKey方法

C.DES解密过程中，只需将 decryptedAES256DataUsingKey替换为 decryptedDESDDataUsingKey方法

罗列一下测试代码输出结果：

```

NSString *message = @"测试各种加密解密方法abc123+="/";
NSString *key = @"xyz123这是key";

NSString *base64Msg = [EncryptionHelper base64EncodeWithString:message];
NSString *decodeMsg = [EncryptionHelper decodeBase64WithString:base64Msg];


NSString *md5Msg = [EncryptionHelper MD5SumWithString:message];
NSString *sha1Msg = [EncryptionHelper SHA1HashWithString:message];
NSString *sha256Msg = [EncryptionHelper SHA256HashWithString:message];

NSString *aes256Msg = [EncryptionHelper AES256EncryptedString:message usingKey:key];
NSString *decryptedAESMsg = [EncryptionHelper decryptedAES256String:aes256Msg usingKey:key];

NSString *desMsg = [EncryptionHelper DESEncryptedString:message usingKey:key];
NSString *decryptedDESMsg = [EncryptionHelper decryptedDESString:desMsg usingKey:key];

```

```
2016-09-08 17:46:43.505 base[7128:9029057] message:测试各种加密解密方法abc123
2016-09-08 17:46:43.506 base[7128:9029057] base64Msg:5rWL6K+V5ZCE56eN5Yqg5
2016-09-08 17:46:43.506 base[7128:9029057] decodeMsg:测试各种加密解密方法abc1
2016-09-08 17:46:43.506 base[7128:9029057] md5Msg:55d86b31f8cf9b3007a30366
2016-09-08 17:46:43.506 base[7128:9029057] sha1Msg:1465c756f0eb64c7cccf175
2016-09-08 17:46:43.506 base[7128:9029057] sha256Msg:297050d60416bbf0a8485
2016-09-08 17:46:43.506 base[7128:9029057] aes256Msg:seDPZ6RLHavkLYxJOP9uM
2016-09-08 17:46:43.506 base[7128:9029057] decryptedAESMsg:测试各种加密解密方
2016-09-08 17:46:43.506 base[7128:9029057] desMsg:uXGaL4ChwSJt5HoD+gU9T/9L
2016-09-08 17:46:43.507 base[7128:9029057] decryptedDESMsg:测试各种加密解密方
```



来自：<http://www.cnblogs.com/ALongWay/p/5853970.html>

X枫林提供全面的网络编程、脚本编程、网页制作、网页特效，网站建设为站长与网络编程从业者提供学习资料。

天朝-备0101001号-01 本站由菊爆大队支持维护，站内内容全部来源网络，如果侵犯了您的权益请邮件致songshoukui@yeah.net