iOS 归档反归档 详解


作者 唐半仙丶 (/u/a8a031000319) + 关注
2016.05.23 22:30 字数 1105 阅读 1067 评论 0 喜欢 7
(/u/a8a031000319)

> 创建一个 Person 类

```
//定义基本属性
@property(nonatomic,strong)NSString *name;
@property(nonatomic,strong)NSString *gender;
@property(nonatomic,assign)int age;
```

1,首先,复杂对象所属的类要遵循<NSCoding>协议

```
@interface Person : NSObject<NSCoding>
```

2,-(void)encodeWithCoder:(NSCoder *)aCoder;序列化/归档

```
/归档(序列化)
//对person对象进行归档时,此方法执行
//对person中想要进行归档的所有属性,进行序列化操作
-(void)encodeWithCoder:(NSCoder *)aCoder
{
    [aCoder encodeObject:self.name forKey:@"name"];
    [aCoder encodeObject:self.gender forKey:@"gender"];
    [aCoder encodeInt:self.age forKey:@"age"];
}
```

3:-(instancetype)initWithCoder:(NSCoder *)aDecoder;反序列化

```
//反归档(反序列化)
//对person对象进行反归档时,该方法执行
//创建一个新的person对象,所有属性都是通过反序列化得到
-(instancetype)initWithCoder:(NSCoder *)aDecoder
{
    if (self = [super init]) {
        self.name = [aDecoder decodeObjectForKey:@"name"];
        self.gender = [aDecoder decodeObjectForKey:@"gender"];
        self.age = [aDecoder decodeIntForKey:@"age"];
    }

    return self;
}
```

> 复杂对象写入文件

```
#pragma mark  --复杂对象写入文件
-(void)archiver
{
    Person *per = [Person new];
    per.name = @"小美眉";
    per.gender = @"女";
    per.age = 18;


    //准备路径:
    NSString *path = NSHomeDirectory();
    NSLog(@"%@",path);

    path = [path stringByAppendingPathComponent:@"singeGirl.txt"];
    //1:准备存储数据的对象
    NSMutableData *data = [NSMutableData data];
    //2:创建归档对象
    NSKeyedArchiver *archiver = [[NSKeyedArchiver alloc] initForWritingWithMutabl
eData:data];
    //3:开始归档
    [archiver encodeObject:per forKey:@"person"];
    //4:完成归档
    [archiver finishEncoding];
    //5:写入文件当中
    BOOL result = [data writeToFile:path atomically:YES];
    if (result) {
        NSLog(@"归档成功:%@",path);
    }else
    {
        NSLog(@"归档不成功!!!");
    }
```

```
#pragma mark ---  反归档/反序列化/解码/解档 ----
    //准备解档路径
    NSData *myData = [NSData dataWithContentsOfFile:path];
    //创建反归档对象
    NSKeyedUnarchiver *unarchiver = [[NSKeyedUnarchiver alloc] initForReadingWith
Data:myData];
    //反归档
    Person *aper = [Person new];
    aper = [unarchiver decodeObjectForKey:@"person"];
    //完成反归档
    [unarchiver finishDecoding];
    //测试
    NSLog(@"%@",aper.name);
    NSLog(@"%@",aper.gender);
}
```

> Foundation框架对象 归档

```
-(void)foundationClassArchiver
{
   //获取文件路径
    NSString *documentPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirec
tory, NSUserDomainMask, YES) firstObject];
    //
    NSString *filePath = [documentPath stringByAppendingPathComponent:@"archiverF
ile"];
    //归档
    NSArray *archiverArray = @[@"小萝莉",@"小正太",@"UI",@"OC"];
    BOOL result = [NSKeyedArchiver archiveRootObject:archiverArray toFile:filePat
h];

    if (result) {
        NSLog(@"归档成功:%@",filePath);
    }else
    {
        NSLog(@"归档失败");
    }

    //反归档
    NSArray *unarchiverArr = [NSKeyedUnarchiver unarchiveObjectWithFile:filePath]
;

    NSLog(@"%@",unarchiverArr);
}
```

> 对自定义的内容进行归档

```
//获取文件路径
    NSString *documentPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirec
tory, NSUserDomainMask, YES) lastObject];

    //在document文件夹下,创建新的文件
    NSString *filePath = [documentPath stringByAppendingPathComponent:@"customFil
e"];

    //1:使用Data对象进行归档
    NSMutableData *archiverData = [NSMutableData data];
    //2:创建归档对象
    NSKeyedArchiver *archiver = [[NSKeyedArchiver alloc] initForWritingWithMutabl
eData:archiverData];
    //3:自己定义数据内容(数据内容以键值对的形式存在)
    [archiver encodeObject:@"尼古拉斯.赵四" forKey:@"ZS"];
    [archiver encodeObject:@[@"爱新觉罗",@"努尔哈赤",@"叶赫那拉"] forKey:@"name"];
    [archiver encodeInt:28 forKey:@"age"];
    //4:完成归档
    [archiver finishEncoding];
    //5:写入
    BOOL result = [archiverData writeToFile:filePath atomically:YES];

    if (result) {
        NSLog(@"归档成功:%@",filePath);
    }


    //反归档
    //1:读取文件,生成NSData类型
    NSData *unarchiverData = [NSData dataWithContentsOfFile:filePath];
    //2:创建反归档对象
    NSKeyedUnarchiver *unarchiver = [[NSKeyedUnarchiver alloc] initForReadingWith
Data:unarchiverData];
    //3:反归档.根据可以访问
    NSString *ZS = [unarchiver decodeObjectForKey:@"ZS"];
    NSLog(@"%@,%@",ZS,filePath);
```

> NSUserDefaults

```
-(void)writeNSUserDf
{

   /*
    NSUserDefaults是一个单例,在整个应用程序当中只有一个实例对象,他可以用于数据的永久保存,简单实用,这是它可以让数据自由传递的一个前提.
    NSUserDefaults:它可以存储一些类型的数据:NSNumber,NSString,NSData,NSArray......
    */

//    NSUserDefaults
    NSArray *arrays = @[@"橘子",@"香蕉",@"西瓜",@"草莓",@"大苹果"];
    NSUserDefaults *user = [NSUserDefaults standardUserDefaults];
    //存
    [user setObject:arrays forKey:@"userarrays"];

    //取
    NSArray *strings = [user objectForKey:@"userarrays"];
    NSLog(@"%@",strings);



#warning 注意:对相同的key赋值约等于一次覆盖,要保证每一个key的唯一性
//    NSUserDefaults 存储的对象完全是不可变的(这一点十分关键,如果弄错会出现bug),例如,如果我想要存储一个NSMutableArray对象.我必须先创建一个不可变数组(NSArray),然后在去存入
    NSMutableArray *mutableArray = [NSMutableArray arrayWithObjects:@"123",@"456",@"789", nil];
    NSArray *array = [NSArray arrayWithArray:mutableArray];

    NSUserDefaults *user2 = [NSUserDefaults standardUserDefaults];

    [user2 setObject:array forKey:@"我们这里存放的一定是不可变的"];


    //自定义数据类型存储到NSUserDefaults
    Person *per = [Person new];
    per.name = @"小白";
    per.age = 14;
    per.gender = @"男";
    //创建存放person的数组
    NSMutableArray *dataArray = [NSMutableArray arrayWithCapacity:0];
    //将person类型转换成为NSData类型
    NSData *data = [NSKeyedArchiver archivedDataWithRootObject:per];
    //将data存入到数组当中
    [dataArray addObject:data];
    NSUserDefaults *user1 = [NSUserDefaults standardUserDefaults];
    [user1 setObject:data forKey:@"person"];
    NSLog(@"%@",user1);

    //取出数据
    NSUserDefaults *user3 = [NSUserDefaults standardUserDefaults];
    NSData *data3 = [user3 objectForKey:@"person"];
    Person *per3 = [NSKeyedUnarchiver unarchiveObjectWithData:data3];
    NSLog(@"%@",per3);



}
```


唐半仙丶 (/u/a8a031000319)
写了 11373 字，被 32 人关注，获得了 67 个喜欢
(/u/a8a031000319)

+关注

坚志而勇为，谓之刚。刚，生人之德也。

♡ 喜欢 (/sign_in) | 7

更多分享

(http://cwb.assets.jianshu.io/notes/images/4095116/

登录 (/sign_in) 后发表评论

评论

智慧如你，不想发表一点想法 (/sign_in)咩~