<https://github.com/JabberYQ/ELFileManager>

# 文章地址

[自造小轮子：文件管理工具及UI层文件浏览器](http://www.jianshu.com/p/3894413a5bd6)

# 正文

### 总目录界面

大致说一下思路，在Cache目录下找到ELFileCache文件夹，如果没有就创建一个，然后接下来的文件处理都会放在这个文件夹中。

### a.获取所有文件

封装方法：

/\*\*

获取到当前路径下的所有文件（包括文件夹）

@param path 文件夹路径

@return 返回 ELFileModel 对象数组

\*/

- (NSArray \*)getAllFileWithPath:(NSString \*)path;

使用方法：

- (void)getAllFile

{

self.fileManager = [ELFileManager shareManager];

self.files = [NSMutableArray arrayWithArray:[self.fileManager getAllFileWithPath:self.homePath]];

dispatch\_async(dispatch\_get\_main\_queue(), ^{

[self.tv reloadData];

});

}

### b.新建文件夹

/\*\*

新建一个文件夹

@param path 文件所在的文件夹路径

@param name 文件夹名称

@return 返回 成功与否

\*/

- (BOOL)createFolderToPath:(NSString \*)path folderName:(NSString \*)name;

int num = arc4random() % 100;

NSString \*name = [NSString stringWithFormat:@"新建的文件夹--%d", num];

if ([self.fileManager createFolderToPath:self.homePath folderName:name]) {

[self getAllFile];

} else {

NSLog(@"创建失败");

}

### c.新建文件

/\*\*

新建一个文件到指定目录

@param path 文件所在的文件夹路径

@param name 文件名称

@return 返回 成功与否

\*/

- (BOOL)createFileToPath:(NSString \*)path fileName:(NSString \*)name;

if ([self.fileManager createFileToPath:self.homePath fileName:@"hello.api"]) {

[self getAllFile];

} else {

NSLog(@"创建失败");

}

### d.下载文件（边下边写）

内部使用的是NSFileHandle类实现。 封装方法：

/\*\*

往文件里追加内容

@param contentData 追加的内容

@param filePath 文件路径

\*/

- (void)seriesWriteContent:(NSData \*)contentData intoHandleFile:(NSString \*)filePath;

使用方法：

// 下载任务开启

NSURLSession \*session = [NSURLSession sessionWithConfiguration:[NSURLSessionConfiguration defaultSessionConfiguration] delegate:self delegateQueue:[[NSOperationQueue alloc] init]];

NSURLSessionDataTask \*dataTask = [session dataTaskWithRequest:[NSMutableURLRequest requestWithURL:[NSURL URLWithString:@"http://120.25.226.186:32812/resources/videos/minion\_02.mp4"]]];

[dataTask resume];

// 在代理方法中使用

- (void)URLSession:(NSURLSession \*)session dataTask:(NSURLSessionDataTask \*)dataTask didReceiveData:(NSData \*)data

{

NSString\* filePath = [self.homePath stringByAppendingPathComponent:\_downloadFileName];

[self.fileManager seriesWriteContent:data intoHandleFile:filePath]; // 写入

\_currentLen += data.length;

double progress = (double)\_currentLen / \_totolLen;

dispatch\_async(dispatch\_get\_main\_queue(), ^{

self.progress.progress = progress;

});

}

### d.搜索

/\*\*

搜索表面的文件（不会搜索当前文件夹里的文件夹内文件。。）

@param searchText 搜索的文字

@param folderPath 搜索的文件夹路径

@return 返回 ELFileModel 对象数组

\*/

- (NSArray \*)searchSurfaceFile:(NSString \*)searchText folderPath:(NSString \*)folderPath;

/\*\*

搜索深度的文件 (会把当前文件夹里的所有文件都查一遍 包括文件夹里的文件夹)

@param searchText 搜索的文字

@param folderPath 搜索的文件夹路径

@return 返回 ELFileModel 对象数组

\*/

- (NSArray \*)searchDeepFile:(NSString \*)searchText folderPath:(NSString \*)folderPath;

NSArray \*resFiles = [self.fileManager searchDeepFile:@"的" folderPath:self.homePath];

for (ELFileModel \*file in resFiles) {

NSLog(@"%@", file.name);

}

搜索功能分了深度搜索和表面搜索。深度搜索能搜索到文件夹内部的内容。

### e.更多功能

## ELFileModel类

typedef NS\_ENUM(NSInteger, ELFileType) {

ELFileTypeUnknown = -1, //其他

ELFileTypeAll = 0, //所有

ELFileTypeImage = 1, //图片

ELFileTypeTxt = 2, //文档

ELFileTypeVoice = 3, //音乐

ELFileTypeAudioVidio = 4, //视频

ELFileTypeApplication = 5, //应用

ELFileTypeDirectory = 6, //目录

ELFileTypePDF = 7, //PDF

ELFileTypePPT = 8, //PPT

ELFileTypeWord = 9, //Word

ELFileTypeXLS = 10, //XLS

};

@interface ELFileModel : NSObject

//文件路径

@property (copy, nonatomic) NSString \*filePath; ///< 全路径

//文件URL

@property (copy, nonatomic) NSString \*fileUrl;

@property (copy, nonatomic) NSString \*name; ///< 文件名称

@property (copy, nonatomic) NSString \*fileSize; ///< 大小用字符表述

@property (nonatomic, assign) float fileSizefloat; ///< 大小用float

@property (copy, nonatomic) NSString \*modTime; ///< 修改时间

@property (copy, nonatomic) NSString \*creatTime; ///< 修改时间

@property (assign, nonatomic) ELFileType fileType;

@property (nonatomic, strong) NSDictionary \*attributes; ///<文件属性

/\*\*

初始化方法

@param filePath 全路径

@return 自身对象

\*/

- (instancetype)initWithFilePath:(NSString \*)filePath;

文件类的所有属性如上，通过- (instancetype)initWithFilePath:(NSString \*)filePath;方法初始化，就可获得完整的对象。其中文件类型的判断方法如下：（以图片类型为例子）

static const UInt8 IMAGES\_TYPES\_COUNT = 8;

static const NSString \*IMAGES\_TYPES[IMAGES\_TYPES\_COUNT] = {@"png", @"PNG", @"jpg",@",JPG", @"jpeg", @"JPEG" ,@"gif", @"GIF"};

NSArray \*imageTypesArray = [NSArray arrayWithObjects: IMAGES\_TYPES count: IMAGES\_TYPES\_COUNT];

if ([imageTypesArray containsObject:pathExtension]) {

return ELFileTypeImage;

}