

代码整理标签记录

一、下载流程概述

下有下载模块结构，可以划分为三个部分：

1、添加视频下载，并启动视频下载任务

- 业务类：视频下载管理类 **VideoDownloadManager**
- 相关数据类：
 - i. VideoForDownload > 视频下载单位，与视频一一对应，并处理对应下载任务。
- 函数操控流程：
 - i. 设置支持下载同步任务数：VideoDownloadManager.maxDownloadTaskNumber = 1;
 - ii. 添加下载视频：[VideoDownloadManager.addDownloadVideo:video];
 - iii. 启动视频下载：
 - a. [VideoDownloadManager.startDownloadVideo:video
retryStatus:DownloadRetryStatusNone];

生成VideoForDownload,做相应初始化配置，并添加到管理队列。
 - b. [VideoDownloadManager.reNewDownloadTasks]

刷新下载任务队列，以及刷新本地数据。
 - c. VideoDownloadManager.downloadingVideo.startDownloadVideo

对应downloadingVideo调用开始下载视频的任务。

2、获取下载地址，并根据地址启动分段下载

- 业务类：视频下载类：**VideoForDownload**
- 相关数据和功能类：

- i. BZXParseHelper > 获取下载六地址功能类
- ii. FileDownloadManager > 下载任务管理类

- 函数操控流程

- i. 获取下载流地址: if(VideoFileTypeUnkonwn) VideoForDownload.fecthDownloadUrl {获取到.m3u8UrlArray}

BZXParseHelper工具类根据类信息获取下载流地址:
VideoForDownload.m3u8UrlArray

- ii. 流地址成功后更新视频下载管理列表数据: [VideoDownloadManager.videosMap setObject:video forKey:itemKey];
 - iii. 根据VideoForDownload的下载地址数组, 开始进行按序分段下载。

else(M3U8|MP4|~) VideoForDownload.startDownloadSegment

- iv. 同步下载事件到VideoDownloadManager:

代理协议: @protocol BZXVideoForDownloadDelegate.

3、资源下载, 并进行代理同步、数据更新

- 业务类: 文件下载类: **FileDownloadManager**

- 相关数据和功能类:

- i. VideoForDownload > 对应下载视频资源, 代理下载资源回调,
 - ii. FileDownloadManager > 下载任务管理类, 管理fileDownloader数据及对应的下载task网络任务。
 - iii. FileDownloader > 对应着一个下载单位, 作为一条下载内容业务数据的存储传递类。

- 函数操控流程:

- i. 下载管理类启动一条下载任务: FileDownloadManager.downloadFileWithURL:url delegate:VideoDownloadManager header:videoForDownload.header

参数为: 下载地址url、代理videoForDownload、请求头
videoForDownload.header

- ii. 为对应下载url创建FileDownloader实例:

a. 生成资源文件路径, [FileDownloadManager.filePathForURL:url]

- b. 创建FileDownloader: [FileDownloader alloc] initWithURL:url filePath:[self filePathForURL:url]
- c. 将下载单元存储类添加到下载管理数组序列中, _downloaders.addObject:downloader

iii. 根据下载单元数据类, 启动网络下载任务。

[self startDownloadWithObject:downloader]

- a. 查询已经存在的NSURLSessionDownloadTask
 - a. 根据下载地址url从现有任务序列(_downloadTasks)寻找,
 - b. 根据系在地址url寻找已有缓存数据resumeData, 若resumeData存在则利用缓存数据找回[self downloadTaskWithResumeData:resumeData]
- b. 没有找着NSURLSessionDownloadTask, 则新建下载任务
- c. 启动下载任务[currentDownloadTask resume], 并将之添加到任务管理队列 _downloadTasks[keyForUrl] = currentDownloadTask。
- d. 同步对应FileDownloader.tate状态。

iv. 同步网络请求数据到VideoForDownload:

下载同步代理协议: @protocol BZXFileDownloadManagerDelegate

二、对下载数据及存储路径的版本历史说明

1 下载模块代码版本

- 1. download1.0 准确替换历史版本已无法最终, 但是可以确定现在可以将相关陈旧文件及目录删整
- 2. download2.0 现在使用的下载代码版本, 后期迭代中有不断更新代码

2 文件存储方式及路径版本

版本号已经不知, 调整日期为change_20170303, 前后对比如下:

实体描述	调整 前	调整 后	补充
目录	Library/	Documents /	根据沙盒文档, 调整后为佳
缓存Dir	DQFileDown loadTemp	DQFileDow nloadTemp	实际上都无用, 断点缓存如今使用文件存储参看DQFileDownloadTempresumeDatas.plist

下载文件 存储dir	DQFileDownloadCache	DQFileDownloadCache	规则文件命名A
下载完成 视频存储 Dir	DQVideoDownloadCache	DQVideoDownloadCache	规则文件命名B

- 规则文件命名A: Document/DQFileDownloadCache/[url.absoluteString md5]
- 规则文件命名B: Document/DQFileDownloadCache/[NSString stringWithFormat:@"aid_%@_porder_%@_%@",video.aid,video.order,videoName]/[video.m3u8UrlArray indexOfObject:url]+type(M3U8-"",MP4-".mp4",Unknown-"")

根据更改日期也可以得出结论: change_20170303对应版本保留的新旧版本数据兼容代码可以删除。

3 补充下载完成后资源移动路径逻辑说明:

文件下载完成 (FileDownloadManager) :

1. 得到用户定义临时文件地址: filePath = currentDownloader.filePath = /Documents/DQFileDownloadCache/dcbe3897ae0c5b75094254832971bbde desURL = [NSURL fileURLWithPath:filePath];
2. 获取下载缓存文件地址: downloadURL = /Library/Caches/com.apple.netsessiond/Downloads/com.ysdqTV.iphone.client/CFNetworkDownload_fQM3Y.tmp
3. 将下载缓存从缓存地址移动至Documents临时文件中 [[NSFileManager defaultManager] moveItemAtURL:downloadURL toURL:desURL error:&fileMoveErr];

代理通知 (VideoForDownload)

BZXFileDownloadManager:didFinishedOfURL:downloader:

1. 通过downloader.filePath获取Documents临时文件地址 downloadFilePath = /Documents/DQFileDownloadCache/dcbe3897ae0c5b75094254832971bbde
2. 根据下载url及VideoForDownload自身信息获取最终资源存放地址 filePath = [self filePathForUrl:url.absoluteString] = Documents/DQVideoDownloadCache/aid_2_723251_porder_3_不伦食堂-第3集/0.mp4;
3. 将Documents临时文件转移到最终下载文件规则拼接的地址 [[NSFileManager defaultManager] moveItemAtPath:downldFiElPath toPath:filePath error:&error];

三、管理操作类、数据类|文件

1. DQFileDownloader

经过整理发现，一个FileDownloader对应着一个下载单位，但是FileDownloader中原先的下载行为代码已经废弃无用，所以他并不是一个行为(网络请求、数据解析、复杂逻辑处理)类，而是一个下载单位的数据存储类。

因而相关的弃用下载代码都可以删除、整理，具体可删除代码内容标注在属性方法列表中。

属性

属性名	类型	访问权限	效用	补充
state	BZXFileDownloadState	read-write	文件下载状态	
url	NSURL	readonly	下载地址	作为FileDownloadManager中寻找FileDownloader的key值
filePath	NSString	readonly	保存文件路径	
cachePath	NSString	readonly	缓存路径	暂无有效调用
userAgent	NSString	readonly		
cookie	NSString	readonly		
header	NSDictionary	readonly	请求头	
progress	double	readonly	拓展方法，获取进度	
isDownloading	BOOL	readonly	拓展方法，判断是否下载中	
hasFinished	BOOL	readonly	拓展方法，判断是否下载完成	
		read-write		

fileSize	long long	e	文件大小	
finishe dSize	long long	read-writ e	已完成的大小	
cacheC apacity	long long	read-writ e	缓存大小 以M 为单位	可删除
retryCo unt	NSInteger	read-writ e		
connec tion	NSURLCon nection	private	下载连接	相关代码现已废弃，可以删整
delegat e	id	read-writ e	回调函数	可以删除

方法

方法名	调用关系(调用者)	效用	补充
initWithURL: filePath:cac hePath: cacheCapacity:	FileDownloadMa nager	初始化，根据参 数为只读属性赋 初值	暂无调用（上级函数 暂无调用）
initWithURL: filePath:cac hePath: cacheCapacity:u serAgent:cookie:	FileDownloadMa nager	同上	暂无调用（上级函数 暂无调用）
initWithURL: filePath:cac hePath: cacheCapacity:h eader:	FileDownloadMa nager	同上	目前实际调用方法
start	无调用	- (void)start { ret urn;}	可以完全删除
cancel	FileDownloadMa nager	对state属性修改 ，并造成修改回 调，其余代码无 用	可删整
BZXFileDownloaderDidS tarted:	代理函数	修改state属性完 成后进行回调，	可删整，对应删整代 理实现FileDownload

		别无它用	Manager类里的代码
BZXFileDownloaderDidDownloadData:	代理函数	修改state后回调，同上	可删整 同上
BZXFileDownloaderDidChangeProgress:	代理函数	修改state后回调，同上	可删整 同上
BZXFileDownloaderDidFinished:	代理函数	修改state后回调，同上	可删整 同上
BZXFileDownloaderDidFailed:	代理函数	修改state后回调，同上	可删整 同上
connectionDidFinishLoading:	私有代理 NSURLConnectionDelegate	connectionq请求代理，已无用	可删除
connection: didReceiveData:	同上		可删除
connection: didReceiveResponse:	同上		可删除
creatConnection	内部调用私有方法	创建connection，实际上已经无可用时机	可删除
saveDataToFile	内部调用私有方法	文件句柄操作，实际已经无效	可删除

删整策略

1. 属性和一些直观的方法

这些内容不涉及文件之间的复杂讯息交互，可以直接连锁删除。
如connection属性，及其相关的所有函数。

2. BZXFileDownloaderDelegate

说明：1. BZXFileDownloaderDelegate 是FileDownloader声明的协议，FileDownloader所有者和协议方法代理者都是FileDownloadManager。2. 该代理的触发发生在FileDownloader的setState方法 以及NSURLConnectionDelegate回调中，如今后者已经废弃，再使用协议就再无必要，

-
1. 将FileDownloader中协议方法全部删除，包裹setState方法也没必要重写。

2. FileDownloadManager中保留BZXFileDownloaderDelegate声明的方法实现，添加一个方法传参为FileDownloader和State，其中调用逻辑与FileDownloader.setState一致。当FileDownloadManager中调用fileDownloader更改state时候改为调用该方法，调用set方法时候示例代码如下：

```
- (void)fileDownloader:(BZXFileDownloader *)downloader changeState:(BZXFileDownloadState)state {
    downloader.state = state;
    switch (state) {
        case BZXFileDownloadState_Downloading: {
            [self BZXFileDownloaderDidStarted:downloader];
            break;
        }
        // 剩余代码，此处逻辑与FileDownloader.setState中逻辑保持一致。
    }
}
```

2. FileDownloadManager,

处理VideoForDownload类发起的下载任务，并通过回调函数同步下载变化讯息；
执行下载任务并与下载单位关联管理（NSURLSessionDownloadTask <--> FileDownloader）；

结构上已经稳定，小部分版本遗留代码可以进行删除更改

属性|成员变量

属性、变量名	类型	访问权限	效用	补充
currentSession	NSURLSession	readonly	下载请求Session	可以不暴露
resumeDatas	MuDictionary	private	暂无使用到	可删除
downloadTasks	MuDictionary	private	存放DownloadTask	
notificationObservers	MuArray	private	通知容器	

_fileFolderPa th	NSSt ring	private	资源存储目录Document/DQF ileDownloadCache	
_cacheFolde rPath	NSSt ring	private	缓存目录Document/DQFileD ownloadTemp	无实际效用，可删 除，
_resumeData FilePath	NSSt ring	private	断载缓存文件plist Document /DQFileDownloadTempresu meDatas.plist	
_downloader s	MuAr ray	private	存放FileDownloader	便于索引寻去File Downloader
_downloader ForURL	MuDi ction ary	private	键值存放FileDownloader	便于键值(url)寻找 FileDownloader
_downloadD elegates	MuAr ray	private	存放代理对象	代理协议 BZXFile DownloadManage rDelegate
_downloadU RLs	MuAr ray	private	暂无使用	可删除

方法

方法名	调用关系(调用者)	效用	补充
downloadFileWithURL:	VideoForDownlo ad	判断启动 下载任务 ，并返回Fi leDownloa der	暂无调用
downloadFileWithURL: d elegat:	VideoForDownlo ad	同上	暂无调用
downloadFileWithURL: d elegat: userAgent: cooki e:	VideoForDownlo ad	同上	暂无调用
downloaderWithDelegate : forURL:	VideoForDownlo ad	同上	暂无调用

downloadFileWithURL: delegate: header	VideoForDownload	同上	现在实际使用方法
cancelForDelegate:	VideoForDownload	取消下载任务，移除相关对应代理	这个方法名字更改为cancelDownload更恰当
resumeTaskForDelegate:	VideoForDownload	恢复下载	暂无调用（上级函数暂无调用）
suspendTaskForDelegate:	VideoForDownload	暂停挂起下载任务	暂无调用（上级函数暂无调用）
deleteDownloadForURL:	VideoDownloadHelper、VideoForDownload、self	根据URL清除所有相关下载内容	
saveResumeData: forURL:	VideoDownloadHelper self	保持下载缓存	可以不必暴露（外部调用的代码其实已经无调用链延续）
deleteDownloadTempWithResumeData:	VideoDownloadHelper	移除resumeData 对应的temp文件	（外部调用的代码其实已经无调用链延续）可以考虑移除
filesMoveDir: toDir:	self	文件目录转移	1.此功能可以封装在共有工具类，譬如Path中；2.调用链版本已不存在，相关代码可以删除。
BZXFileDownloadManager: didStartedOfURL: downloader:	协议代理函数	同步下载状态变化讯息给VideoForDownload	
BZXFileDownloadManager: didDownloadDataOfURL: downloader:	同上	同上	

BZXFileDownloadManager: didChangedProgressOfURL: downloader:	同上	同上	
BZXFileDownloadManager: didStartedOfURL: didFinishedOfURL: downloader:	同上	同上	
BZXFileDownloadManager: didStartedOfURL: didDidFaileOfURL: downloader:	同上	同上	
BZXFileDownloadManager: didStartedOfURL: didStopDownloadOfURL: downloader:	同上	同上	

3. VideoForDownload

对应一个完整的视频下载对象

方法

分组列出相关方法

路径处理相关

返回值	方法名	调用关系	效用	补充
NSString	filePathForUrl:	private	根据url生成完整的下载资源文件地址， Document/DQFileDownloadCache/[video folderPath]/[video videoFileNameForUrl:]	
NSString	filePathAtIndex:	public	根据索引生成完整的下载资源文件地址， Document/DQFileDownloadCache/[video folderPath]/[video videoFileNameAtIndex:]	可以删除，用filePathForUrl:取代， 统一代码。

NSString	folderPath :	private	生成下载资源文件路径，Document/D QFileDownloadCache/[video folderP ath]	
NSString	videoFile NameFor Url:	private	根据url生成视频文件名， [video.m3u8 UrlArray indexOfObject:url]+type(M3 U8-"" ,MP4-".mp4",Unknown-"")	
NSString	videoFile NameAtIn dex:	public	根据url索引生成视频文件名 ， type(M 3U8-"" ,MP4-".mp4",Unknown-"")	
NSString	filePathFo rURLAtIn dex:	private	无调用， 废弃代码	可以删除
BOOL	isVideoOf UrlDownl oaded:	private	判断本视频的某个url是否已下载完成	
NSString	localM3U 8FilePath	public	生成本地M3U8文件地址	
void	checkExis tFilByLoc alUrl:com plete:	public	检查文件是否都存在 计算文件大小,bl ock返回检查信息	可做优化,封 装到共享工 具类中
NSArr ay	getTotalS egmentSt reamList:	public	返回多段源的多个下载地址对应的所 有本地文件路径	

4. BZXVideoDownloadHelpler

数据转换处理 | 数据读取处理

返回值	方法名	调用关系、位置	效用	补充
BZXVi deo	+createDownloadVideo:episo deIndex:	publick	生成下载中间数 据类BZXVideo	可做 优化 删整
			读取Download2.	

NSDictionary	readAllOldDownloadDataToVi doeDownloadMap	public	0之前点老版本数 据	可删除
--------------	--	--------	---------------	-----

四、业务辅助类|文件

1. FileHelper

磁盘空间检查 & 路径视频文件验证

方法

返回值	方法名	调用关系、位置	效用	补充
uint64_t	+getFreeDiskspaceMB	点击下载入口	publick	
long long int	+folderSize:	暂无调用	获取指定路径 文件大小	
BOOL	+CheckVideoFile IsAvalible	FileDownloadManager.URLSession 下载完成回调	路径视频文件 验证	

2. Reachability

网络检查.

方法

返回值	方法名	调用关系、位置	效用	补充
Reachability	reachabilityForInternetConnection	点击下载入口	网络判断	

3. BZXCachePromptView

点击完下载操作，成功添加下载任务的提示弹框。

4. BZXReminderNetState

下载网络状态弹框提示。参看VideoObjectState.reachabilityChanged

5. BZXVideoObjectState

用于处理网络监听之后的下载网络弹框状态。参看VideoDownloadManager.addNotifications

可做做删改优化，具体参考可优化点注释。

6. BZXAppSettings

全局变量存储 & 下载操作弹框方法接口

7. AppDelegate

方法

返回值	方法名	调用关系、位置	效用	补充
void	observeDownloadManager	appdelegate	函数功效是监听下载管理的可用空间	名字可以优化为observeDownloadManagerFreeSpace

属性

属性、变量名	类型	访问权限	效用	补充
b3GDownloadEnable	BOOL	public	3g下载开关	
downloadResolution	BZXVideoResolution	public	清晰度	
downloadFolderPath	NSString	public	下载路径	新版本已经废弃，可以删除

方法

返回值	方法名	调用关系、位置	效用	补充
void	tipAllowDownloadPresentViewController	点击下载入口	弹出提示供用户选择是否设置允许移动数	下载操作提示弹框建议与BZXCachePromptView合并统一整改到一处。

五、可以直接废弃的类及函数

通过整理验证，一下类及代码，以及它们逻辑相关的代码都可以进行删除：

1 BZXDownloadManager类

2 BZXJSPatchManager类

3 BZXOldDataHelp类

4 BZXM3U8Downloader类

5 BZXM3U8VideoDownloader类

6 函数readAllOldDataDownloadDataToVidoeDownloadMap，及调用它的相关逻辑代码

7 函数checkOldDataDir，及调用它的相关逻辑代码

8 函数readOldData，及调用它的相关逻辑代码

9 函数checkOldDataDir

六、 摘录废弃代码及可整改代码注释：

由于版本迭代沉积多，废弃代码很多，整改基本上是添加一点代码来实现大批量废弃代码删除的过程。一下记录为原有代码中review时候添加的整改注释：

第五中提到废弃类，在这里没有再做注释记录，如BZXDownloadManager 。

- BZXAppDelegate
 - i. // FIXME: long{ 根据历史版本情况（Created by xuewu.long on 17/3/6） ，下面的数据

新旧版本兼容操作代码可以废弃。 } [self adjustOldVersionSSARSDataStore];

- BZXAppService

- i. // FIXME: long{ 建议更换一个名字，函数功效是监听下载管理的freeSpace的操作，observeDownloadManagerFreeSpace } (void)observeDownloadManager {

- BZXPlayerViewController2

- i. // FIXME: long{ .aDownloadObj 在工程中无实际传值或存值作用，属当废弃代码} @property (nonatomic, strong) BZXDownloadObject *aDownloadObj;

- BZXFileDownloader

- i. // FIXME: long{ NSURLConnection.connection 相关代码并无使用，应当整理删除，包括NSURLConnectionDataDelegate及其相关 } @property (nonatomic,strong) NSURLConnection *connection;///
//<下载连接

- BZXFileDownloadManager

- i. // FIXME: long{ downloadFileWithURL: delegate: header: 是目前唯一使用到的外部接口，其余代码如果确认无用可以删除，如果有必要保留也不应该暴露 } BZXFileDownloadManager.h文件
- ii. // FIXME: long{ deleteDownloadForURL此处变量名不匹配，cachePath|filePath 书写颠倒 } NSString *cachePath = [self filePathForURL:url]; NSString *filePath = [self cachePathForURL:url];
- iii. // FIXME: long{ filesMoveDir:,这种通用功能代码在可以统一做共有工具代码封装，留出接口函数即可。即可封装在Path工具类中。 } (BOOL)filesMoveDir:(NSString*)dirPath toDir:(NSString*)desDir {
- iv. // FIXME: long{ checkOldDownloadDir,如今已经不存在原有版本用户，此处老版本兼容处理的代码可以废弃。相关调动代码可移除 } (void)checkOldDownloadDir {
- v. // FIXME: long{ creatFolderAtPath,这种通用功能代码在VideoDownloadManager中也存在，可以统一做共有工具代码封装，流出接口函数即可。即可封装在Path工具类中。 } (BOOL)creatFolderAtPath:(NSString*)FolderPath {

- BZXVideoDownloadHelper

- i. // FIXME: long{ createDownloadVideo:观看可知Video对象的这个方法完全可以通过在Video类中添加初始化方法实现，然后可以移除VideoDownloadHelper类，文件结构更合理。 } (BZXVideo)createDownloadVideo:(BZXDetailRequest2Item)item episodeIndex: (NSInteger)i;
- ii. // FIXME: long{ readAllOldDownloadDataToVidoeDownloadMap 上级函数无调用，可

废弃，所以这个函数也可考虑废弃不用 }

(NSDictionary*)readAllOldDataToVideoDownloadMap {

- iii. // FIXME: long{ downloadVideosMapFormDQDownloaders 上级函数无调用，可废弃，所以这个函数也可考虑废弃不用 }

(NSMutableDictionary*)downloadVideosMapFormDQDownloaders {

- iv. // FIXME: long{ downloadVideosFormDownloadObjects: 上级函数已经无调用链延续，确认无用应当抛弃 } (NSMutableDictionary*)downloadVideosFormDownloadObjects {
- v. // FIXME: long{ getVideoDownloadObject: 上级函数已经无调用链延续，确认无用应当抛弃 } (NSMutableDictionary*)getVideoDownloadObject:(NSArray)downloadObjects {
- vi. // FIXME: long{ videoObjectM3U8Info: 上级函数已经无调用链延续，确认无用应当抛弃 } (void)videoObjectM3U8Info:(BZXDownloadObject)obj videoObj:
(BZXVideoForDownload)videoObj {
- vii. // FIXME: long{ downloadObjectsAfterCheckAndMoveSegmentStream: 上级函数已经无调用链延续，确认无用应当抛弃 }
(NSArray)downloadObjectsAfterCheckAndMoveSegmentStream:
(NSArray)downloadObjS {

- BZXVideoDownloadManager

- i. // FIXME: long{ kDownloadVideoAddedNewNotification 此通知并无监听注册，应当废弃 } [[NSNotificationCenter defaultCenter]
postNotificationName:kDownloadVideoAddedNewNotification object:self];
- ii. // FIXME: long{ isVideoDownloaded 判断是否下载，不建议在缓存数组中做遍历，使用sqlite的查询优势做查询更易统一处理 }
 - (BOOL)isVideoDownloaded:(BZXVideo*)video {
- iii. // FIXME: long{ VideoObjectState的创建可以移动到VideoForDownload的初始化中，他们的逻辑关系，不必暴露在外。再看它的代理函数，可以得知目前其代理并无意义，相关代码可以废弃。 } videoForDownload = [[BZXVideoForDownload alloc]
initWithVideo:video]; BZXVideoObjectState *state = [[BZXVideoObjectState alloc] init];
state.delegate = self;
- iv. // FIXME: long{ checkOldDownloadDir, 本方法经过一年多的版本迭代，已经可以移除，原有下载路径已经毫无意义 } // 检查旧版本的文件目录，将原有文件转移至新的文件下载目录,并追加数据。 [self checkOldDownloadDir];
- v. // FIXME: long{ readOldData,这个函数在工程中没有调用处理，相关代码可以考虑删除! }
 - (void)readOldData {
- vi. // FIXME: long{ hasMove可作为单列的一个属性，简化代码，而不应该使用userdefault【占用userdefault资源，代码结构复杂化】 } NSString
*kHasMoveOldDownloadFolderAtV2 = @"kHasMoveOldDownloadFolderAtV2";

- vii. // FIXME: long{ hasReaded可作为单列的一个属性，进行一次判断，而不应该使用 userdefault 【占用userdefault资源，代码结构复杂化】 } NSString
 *kHasReadOldDataBeforeV2 = @"kHasReadOldDataBeforeV2";
- viii. // FIXME: long{ 详看readAllOldDataToVideoDownloadMap 可知，该函数返回是一个空字典，函数无效用，确认可废弃。}NSDictionary *downloadVideosMap = [BZXVideoDownloadHelper readAllOldDataToVideoDownloadMap];
 self.videosMap = [downloadVideosMap mutableCopy];
- ix. // FIXME: long{ checkOldDownloadDir 代码对应版本已经不存在，相关代码可以进行删除了。 }
 ▪ (void)checkOldDownloadDir {
- x. // FIXME: long{ addSkipBackupAttributeToItemAtPath,这种通用功能代码在 FileDownloadManager中也存在，可以统一做共有工具代码封装，流出接口函数即可。即可封装在Path工具类中。 } (BOOL)addSkipBackupAttributeToItemAtPath:(NSString*)filePathString {

- BZXVideoForDownload

- i. // FIXME: long{checkExistFilByLocalUrl 此功能当属通用功能封装与共用工具类中，譬如Path类 } (void)checkExistFilByLocalUrl:(NSArray*)urls complete:(void (^)(bool isExist,unsigned long long fileSize))acomplete {
- ii. // FIXME: long{ currentDownloadLinkShellUrl 并没有赋值操作，此处代码是有什么效用？ } //下载结束stop当前通道的加速 if (![isEmpty]([BZXDownloadSingelton shared].currentDownloadLinkShellUrl)) { [ysdq_cdeService_ stopPlay: [BZXDownloadSingelton shared].currentDownloadLinkShellUrl]; }
- iii. // FIXME: long{ resumeTaskForDelegate:suspendTaskForDelegate: 两个方法为何要如此先后匹配使用？ } [BZXDispatch async:{ [[BZXFileDownloadManager sharedManager] suspendTaskForDelegate:self]; [BZXDispatch after:allSuspendTime execute:{ self.isTaskSuspending = NO; [[BZXFileDownloadManager sharedManager] resumeTaskForDelegate:self]; }]; }];