Multimedia Management Tutorial (Album)

Overview

The **IAlbumManager** class is used to manage aircraft albums and includes the APIs for obtaining/downloading album files or folders and canceling downloads.

API Calling Procedure

Obtaining The Album File List

```
/**
 * Obtains the album file list.
 * @param type File type [MediaTypeEnum].
 * @param storageType Storage type [StorageTypeEnum].
 * @param albumName Folder name. If set to null, all files will be obtained.
 * @param offset Offset.
 * @param count Request count.
 * @param order Sorting by date modified (descending) [OrderTypeEnum].
 * @param callback Callback.
 */
fun getMediaFileList(
    type: MediaTypeEnum,
    storageType: StorageTypeEnum,
    albumName: String? = null,
    offset: Int = 0,
    count: Int,
    order: OrderTypeEnum = OrderTypeEnum.NORMAL,
    callback: CommonCallbacks.CompletionCallbackWithParam<AlbumResultBean>
)
```

Sample Code

```
AlbumManager.getAlbumManager().getMediaFileList(
    type,
    storageType,
    albumName,
    offset,
    count,
    order,
    object :
        CommonCallbacks.CompletionCallbackWithProgressAndParam<AlbumResultBean>
{
        override fun onProgressUpdate(progress: Double) {}
        override fun onSuccess(autelAlbumBeans: AlbumResultBean?) {

        }

        override fun onFailure(error: IAutelCode, msg: String?) {
        }
    }
)
```

Obtaining The Album Folder List

```
/**
 * Obtains the album folder list.
 * @param storageType Storage type [StorageTypeEnum].
 * @param order Sorting by date modified (descending) [OrderTypeEnum].
 * @param callback Callback.
 */
fun getMediaFolderList(
    storageType: StorageTypeEnum,
    order: OrderTypeEnum,
    callback: CommonCallbacks.CompletionCallbackWithParam<AlbumFolderResultBean>
)
```

Sample Code

```
AlbumManager.getAlbumManager().getMediaFolderList(storageType, sortType,
    object : CommonCallbacks.CompletionCallbackWithParam<AlbumFolderResultBean>
{
        override fun onSuccess(t: AlbumFolderResultBean?) {

        }

        override fun onFailure(error: IAutelCode, msg: String?) {

        }
    })
```

Deleting Album Files/Folders

```
/**
 * Deletes specified album files or folders.
 * @param indexId File/Folder IDs.
 * @param storageType Storage type.
 * @param callback Deletion result.
 */
fun deleteMediaFile(
    indexId: Int,
    callback: CommonCallbacks.CompletionCallback
)
```

Sample Code

```
AlbumManager.getAlbumManager().deleteMediaFile(
    indexId,
    object : CommonCallbacks.CompletionCallback {
        override fun onSuccess() {
            it.resume(Pair(index, true))
        }

        override fun onFailure(code: IAutelCode, msg: String?) {
            it.resume(Pair(index, false))
        }

    })
```

Downloading The Specified Album File

```
/**
 * Downloads the specified album file.
 * @param sourcePath Path of the files to download from.
 * @param destPath Path of the files to download to.
 * @param callback Download callback.
 */
fun downloadMediaFile(
    sourcePath:String,
    destPath:String,
    callback: CommonCallbacks.DownLoadCallbackWithProgress<Double>
): BaseRequest
```

Sample Code

```
AlbumManager.getAlbumManager().downloadMediaFile(
    sourcePath,
    destPath,
    object : CommonCallbacks.DownLoadCallbackWithProgress<Double> {

        override fun onSuccess(file: File?) {

        }

        override fun onProgressUpdate(progress: Double, speed: Double) {

        }

        override fun onFailure(error: IAutelCode) {

        }
    })
```

Canceling Download

```
/***
 * Cancels the download request.
 * @param request Request for file download.
 */
fun cancelDownload(request: BaseRequest)
```

Sample Code

```
AlbumManager.getAlbumManager().cancelDownload(baseRequest)
```