

日志下载管理类API

文档变更记录

版本	变更说明	日期	作者	审核
1.0	创建文档	2023年5月23日	唐威	

class DeviceLogManager

method queryLogInfoList

```
fun queryLogInfoList(sn: String, deviceType: DeviceLogTypeEnum, callback: CommonCallbacks.CompletionCallbackWithParam<DeviceSysLog?>)
```

描述：查询设备日志。

输入参数：

- sn：设备SN
- deviceType: 设备类型，参考：[DeviceLogTypeEnum](#)
- callback：查询回调 参考 [CommonCallbacks](#) 回调日志对象参考：[DeviceSysLog](#)。

输出参数：无。

method downloadPowerLog

```
fun downloadPowerLog(powerLogs: List<DeviceSysOncePowerLog>, callback: CommonCallbacks.DownloadCallbackCompletionWithProgressAndFile<Double, DeviceSysLogModule>)
```

描述：按上电记录下载日志。

输入参数：

- powerLogs：上电记录列表
- callback：查询回调 参考 [CommonCallbacks](#) 回调日志对象参考：[DeviceSysLogModule](#)。

注：每完成一个DeviceSysLogModule，就会通过 onProgressFileUpdate 通知

输出参数：无。

method downloadModuleLog

```
fun downloadModuleLog(moduleLogs: List<DeviceSysLogModule>, callback:
CommonCallbacks.DownloadCallbackCompletionWithProgressAndFile<Double,
DeviceSysLogModule>)
```

描述：按模块下载日志。

输入参数：

moduleLogs: 设备模块列表（支持不同上电记录的模块放在一起下载）

callback: 查询回调 参考 [CommonCallbacks](#) 回调日志对象参考: [DeviceSysLogModule](#)。

注：每完成一个DeviceSysLogModule，就会通过 onProgressFileUpdate 通知

输出参数：无。

data class DeviceSysLog

```
public data class DeviceSysLog(var type: DeviceLogTypeEnum,
                                var sn: String,
                                var powerLogList: List<DeviceSysOncePowerLog>) {

}
```

描述：设备日志对象，里面包含所有的日志信息。

属性：

type: 设备类型，参考 [DeviceLogTypeEnum](#)

sn: 设备SN

powerLogList: 上电记录列表，参考 [DeviceSysOncePowerLog](#)

data class DeviceSysOncePowerLog

```
public data class DeviceSysOncePowerLog(var type: DeviceLogTypeEnum,
                                          var sn: String,
                                          var index: Int,
                                          var time: Long,
                                          var moduleList: List<DeviceSysLogModule>) {

}
```

描述：上电记录对象，里面包含这次上电记录全部模块的日志，设备最多存储16次上电记录，达到16次后，会依次删除离当前时间的最远的记录

属性：

type: 设备类型，参考 [DeviceLogTypeEnum](#)

sn: 设备SN

index: 上电记录次序

time: 上电时间戳

moduleList: 设备模块日志列表, 参考 [DeviceSysLogModule](#)

data class DeviceSysLogModule

```
public data class DeviceSysLogModule(  
    var mode: DeviceLogTypeEnum,  
    var sn: String,  
    var index: Int,  
    var type: DeviceLogEnum,  
    var time: Long,  
    var size: Int,  
    var downPath: String = "",  
    var name: String = "",  
    var localPath: String = "") {  
}
```

描述: 设备具体模块的日志信息

属性:

mode: 设备类型, 参考 [DeviceLogTypeEnum](#)

sn: 设备SN

index: 上电记录次序

type: 日志类型, 参考 [DeviceLogEnum](#)

time: 上电时间戳

size: 模块日志大小, 注: 执行下载操作后才有值

downPath: 日志文件下载地址, 用户不用关心

name: 日志名

localPath: 日志下载后的地址, 注: 执行下载操作后才有值

enum class DeviceLogEnum

```
enum class DeviceLogEnum (var value: String, val type: Int) {  
  
    /**  
     * unkown  
     */  
    UNKOWN("unkown", 0),  
}
```

```

/**
 * Rc Transmission
 */
RCTRANSMISSION("RcTransmission",1),

/**
 * Air Transmission
 */
AIRTRANSMISSION("AirTransmission",2),

/**
 * Air Skylink
 */
AIRSKYLINK("AirSkylink",3),

/**
 * Air Vision
 */
AIRVISION("AirVision",4),

/**
 * Rc Android
 */
RCANDROID("RcAndroid",5),

/**
 * Air Camera
 */
AIRCAMERA("AirCamera",6);
}

```

描述：设备模块枚举

enum class DeviceLogTypeEnum

```

enum class DeviceLogTypeEnum (var value: String) {
    /**
     * unkown
     */
    UNKOWN("Unkown"),

    /**
     * RC
     */
    RC("RC"),

    /**
     * Drone
     */
    DRONE("UAV");
}

```

描述：设备类型枚举

CommonCallbacks

```
class CommonCallbacks {  
    /**  
     * 带参数的完成回调  
     *  
     * the completion callback with param  
     */  
    interface CompletionCallbackWithParam<T> {  
        /**  
         * 成功回调 - callback when success  
         * @param t result return when call successfully  
         */  
        fun onSuccess(t: T?)  
        /**  
         * 失败时回调 - callback when failed  
         * @param error code when call failed  
         * @param msg msg when call failed  
         */  
        fun onFailure(error: IAuthCode, msg: String? = null)  
    }  
  
    /**  
     * 有进度的完成回调  
     *  
     * the completion callback with progress  
     */  
    interface CompletionCallbackWithProgress<T> {  
        /**  
         * 调用进度改变回调 - callback when call progress changed  
         * @param progress the progress for the call  
         */  
        fun onProgressUpdate(progress: T)  
        /**  
         * 调用成功回调 - callback when call successfully  
         */  
        fun onSuccess()  
        /**  
         * 失败时回调 - callback when failed  
         * @param error code when call failed  
         */  
        fun onFailure(error: IAuthCode)  
    }  
  
    /**  
     * the completion callback with progress  
     */  
  
    interface DownloadCallbackCompletionWithProgressAndFile<T,D> {  
        /**  
         * callback when call progress changed  
         * @param progress the progress for the call
```

```

    */
    fun onProgressUpdate(progress: T, speed: Double)

    /**
     * callback when one file has been success downed
     * @param file file path for success downloaded
     */
    fun onProgressFileUpdate(file : File?, mode: D?)

    /**
     * callback when all files have been downed
     */
    fun onSuccess()
    /**
     * callback when failed
     * @param error code when call failed
     */
    fun onFailure(error: IAutelCode)
}

/**
 * the completion callback with no param
 */
interface CompletionCallback {
    /**
     * callback when call successfully
     */
    fun onSuccess()

    /**
     * callback when failed
     * @param error code when call failed
     */
    fun onFailure(code: IAutelCode, msg: String? = null)
}
}

```

描述：回调定义