

RTK 接口 API 说明:

1、初始化网络 RTK 类型, 不调用为默认 Ntrip 类型网络 RTK, 本工程同时集成了机巢 RTK, 网络 RTK, 但网络 RTK 同时集成了 Ntrip 及千寻 SDK, 用户根据自身账号类型及项目需要配置网络 RTK 类型(isQianxun 为 true 代表千寻原生 SDK, false 为 Ntrip), 用户根据需要设置网络 RTK 类型, 否则网络 RTK 默认采用 Ntrip 方式实现

```
updateNetRtkType(isQianxun:Boolean, callback: ChangeRTKConfigCallback)
```

2、该方法实现 RTK 的启用与关闭

```
fun enableRTKLocation( isEnabled: Boolean,callback: ChangeRTKConfigCallback)
```

3、通过该方法返回当前 RTK 的启用状态

```
fun isenableRTKLocation():Boolean
```

4、切换 RTK 类型, 目前只支持网络 RTK 及机巢 RTK

```
fun switchRTKSignalEnum( rtkSignalEnum: RTKSignalEnum,callback: ChangeRTKConfigCallback )
```

5、NTrip 网络 RTK 授权

```
fun autherNetRtk(userName:String, password :String, mountPoint :String?,callback: CommonCallbacks.CompletionCallback)
```

6、千寻原生 SDK 网络 RTK 授权

```
fun autherQianxunRtk(key:String, secret :String, deviceType :String,deviceId :String,callback: CommonCallbacks.CompletionCallback)
```

7、注册 RTK 信息上报监听

```
fun registerRtkInfoCallBack(mRtkCallBack: RTKReportInfoCallback)
```

8、取消注册 RTK 信息上报监听

```
fun unRegisterRtkInfoCallBack(mRtkCallBack: RTKReportInfoCallback)
```

9、配置机巢 RTK 基站工作模式

...

```
fun setNestSystemRtkMode(nestRtkModeCommand: NestRtkModeCommand, callback: CommonCallbacks.CompletionCallbackWithParam<Void>)
```

...

示例:

...

//workMode:工作模式 0.自定位模式, 1.设定坐标模式,(参数经度、纬度、高度), 2.移动模式, 3.流动站模式

```
fun setNestRtk(rtkLat: Double?, rtkLng: Double?, altitude: Double?) {  
    val nestMode = NestRtkModeCommand()  
    nestMode.workMode = 1  
    nestMode.latitude = rtkLat  
    nestMode.longitude = rtkLng  
    nestMode.altitude = altitude ?: 0.00  
}
```

```

        NestModelManager.getInstance().setNestSystemRtkMode(nestMode,
            object : CommonCallbacks.CompletionCallbackWithParam<Void> {
                override fun onSuccess(t: Void?) {

                }

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

                }

            })
    }
}
...

```

备注：

// 1、配置更新回调，开启、授权及切换 RTK 类型等配置回调

```

interface ChangeRTKConfigCallback {
    /**
     *
     * 网络 RTK 在更新配置后需进行授权验证，用户根据 isQianxun 执行网络 RTK 授权
     * isQianxun 为 true 调用 autherQianxunRtk 授权，false 调用 autherNetRtk 授权，账号
    信息由用户自己缓存，SDK 不做存储操作
     */
    fun onNeedAuterInfo(singnalEnum: RTKSignalEnum,isQianxun: Boolean)

    /**
     * 更新配置成功
     */
    fun onUpdateConfigSuccess()

    /**
     * 更新配置失败
     */
    fun onUpdateConfigFailure(error: IAutelCode, msg: String?)
}

```

// 2、RTK 数据上报监听，用户可以在需要展示差分数据及 RTK 位置信息处监听

```

interface RTKReportInfoCallback {
    /**
     * RTK 断开链接
     *
     */
    fun onRtkUnConnected()

    /**
     * Ntrip 网络挂载点列表，用户可以缓存列表用于网络 RTK 授权
     *
     * @param list 挂载点名称列表
    */
}

```

```

        */
fun onRtkMountPointList(list: ArrayList<String>?)
/**
 *
 * RTK 上报信息，详细参考 RtkReportBean 类
 * @param reportInfo
 *
 */
fun onRtkReportInfo(reportInfo: RtkReportBean)
/**
 *
 * RTK 机巢上报信息，详细参考 NestRtkStatusNotifyBean 类
 * @param reportInfo
 *
 */
fun onNestRtkReportInfo(nestRtkReportInfo: NestRtkStatusNotifyBean)
}

```

项目实例如：com.autel.sdk.debugtools.fragment.RtkFragment