

交通标准协议

V1.1.0



JAVA SDK 说明

目录

目录.....	i
1 使用说明.....	1
1.1 SDK 文件	1
1.2 接口使用方法	1
1.3 使用前说明	1
2 功能接口.....	2
2.1 初始化客户端模式	2
2.2 初始化服务端模式	2
2.3 获取设备名称	3
2.4 设置设备名称	4
2.5 身份验证和安全通讯约定	4
2.6 设置连接中使用 DES 加密密码	5
2.7 发送播放列表	6
2.8 发送局部更新	6
2.9 移除局部更新	7
2.10 发送文件.....	8
2.11 手动亮度调节.....	8
2.12 手动开关屏	9
2.13 清理文件.....	9
2.14 设置自动亮度控制参数.....	10
2.15 获取自动亮度控制参数.....	11
2.16 控制屏体电源开关	11
2.17 控制本板电源开关	12
2.18 控制多功能卡电源开关.....	12
2.19 查询设备状态	13
2.20 设置日期时间	14
2.21 设备重启.....	14
2.22 恢复 IP 出厂设置	15
2.23 查询版本信息	15
2.24 设置屏体基本参数	15
2.25 获取屏幕点阵数据	16

2.26 设置音量大小	17
2.27 获取内部存储空间大小	18
2.28 设置时间时区	18
2.29 获取时间时区	19
2.30 设置 NTP 参数.....	19
2.31 获取 NTP 参数.....	20
2.32 设置高温关屏温度	21
2.33 获取高温关屏温度	21
2.34 设置网络中断后屏体播放策（虚连接）	22
2.35 FTP 发送文件.....	22
2.36 升级文件.....	23
2.37 查询当前播放内容	23
2.38 查询当前播放列表全部内容.....	24
2.39 获取当前播放截图	25
2.40 设置定时开关屏.....	25
2.41 获取定时开关屏.....	26
2.42 设置定时播放列表	27
2.43 获取定时播放列表	28
2.44 设置定时亮度	29
2.45 获取定时亮度	31
3 附录.....	31
3.1 错误码	31
3.2 时区信息.....	32

1 使用说明

1.1 SDK 文件

SDK 编译环境	JDK 1.8 x86	
SDK 版本	SDK 文件	备注
V1.1 版本	nova.traffic.jar	无

1.2 接口使用方法

import nova.traffic.NovaTraffic 后初始化 NovaTraffic 类。初始化后使用对应的功能接口。

1.3 使用前说明

目前此协议支持客户端和服务端两种方式连接。可创建线程多卡同时操作。

卡做客户端时，服务端可在公网搭建。

使用前请先确定 IP、端口、协议类型是否正确。

2 功能接口

2.1 初始化客户端模式

【功能描述】

初始化 NovaTraffic 类。

【接口定义】

public NovaTraffic(String ip, int port)

【输入参数】

参数	类型	说明	备注
ip	String	例如，192.168.0.220	控制卡 IP
port	int	例如，5000	控制卡端口

【返回参数】

【使用范例】

NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);

2.2 初始化服务端模式

【功能描述】

初始化 NovaTraffic 类。

【接口定义】

public NovaTraffic(Socket socket)

【输入参数】

参数	类型	说明	备注
----	----	----	----

socket	Socket	socket	由用户自行维护 sokcet 打开关闭
--------	--------	--------	---------------------

【返回参数】

【使用范例】

```
ServerSocket serverSocket = new ServerSocket(5000);
while (true) {
    Socket socket = serverSocket.accept();
    Thread t1 = new Thread(new Runnable() {
        public void run() {
            NovaTraffic novaTraffic = new NovaTraffic(socket);
            try {
                while (true) {
                    //业务
                }
            } catch (Exception e) {
                e.printStackTrace();
            } finally {
                try {
                    socket.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    });
    t1.start();
}
```

2.3 获取设备名称

【功能描述】

获取设备的名称。

【接口定义】

```
public String getDeviceName()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
空	String	设备名称	UTF-8 编码

- null 失败

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
String name = novaTraffic.getDeviceName();
```

2.4 设置设备名称

【功能描述】

设置设备的名称。

【接口定义】

```
public int setDeviceName(String name)
```

【输入参数】

参数	类型	说明	备注
name	String	设备名称	UTF-8 编码

【返回参数】

- 1: 成功
小于 0: 错误参考附录 1

【使用范例】

```
String name = novaTraffic.setDeviceName("Taurus50000346");
```

2.5 身份验证和安全通讯约定

【功能描述】

身份验证和安全通讯约定。JT100 1.7.0 以后版本需填写 ViPlex Express 设备密码。

【接口定义】

```
public class DeviceSecretParam {
```

```
private String password;  
private boolean isDesOpen;  
private boolean isMD5Open;  
private String desPassword;  
}  
public int setDeviceSecretParam(DeviceSecretParam mDeviceSecretParam)
```

【输入参数】

参数	类型	说明	备注
password	String	设备密码	初始密码 123456
isDesOpen	boolean	网络数据传输时是否开启 DES 加密	
isMD5Open	boolean	发送文件接口是否采用 MD5 加密	
desPassword	String	DES 密码	固定 8 位

【返回参数】

- 1: 成功
小于 0: 错误参考附录 1

【使用范例】

```
DeviceSecretParam mDeviceSecretParam = new DeviceSecretParam(  
    "123456",true, false , "12345678");  
int ret = novaTraffic.setDeviceSecretParam(mDeviceSecretParam );
```

2.6 设置连接中使用 DES 加密密码

【功能描述】

NovaTraffic 中所有的消息采用 DES 加密的密码。

【接口定义】

```
public boolean NovaTrafficSetPassword(String password)
```

【输入参数】

参数	类型	说明	备注
password	String	DES 密码	固定 8 位

【返回参数】

无

【使用范例】

```
DeviceSecretParam mDeviceSecretParam = new DeviceSecretParam(  
    "123456",true, false , "12345678");  
  
int ret = novaTraffic.setDeviceSecretParam(mDeviceSecretParam );  
  
System.out.println( "设置 des 加密结果: " +ret);  
  
novaTraffic.NovaTrafficSetPassword("12345678");
```

2.7 发送播放列表

【功能描述】

发送播放文件。如播放需要附带图片、视频等资源文件。需使用下发文件接口提前发送。

【接口定义】

```
public int sendPlayList(int id, String content)
```

【输入参数】

参数	类型	说明	备注
id	int	下发播放列表的 ID（范围 1~100）	1 代表 play001.lst。相同 ID 会覆盖之前节目
content	String	播放文件的内容,具体参考标准协议的播放格式说明	UTF-8 编码

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
String content =  
"[all]\r\nitems=1\r\n[item1]\r\nparam=100,1,1,1,0,5,1,0,1\r\nimg1=0,0,001.jpg,0,0,0\r\nimgparam1=100";  
  
int ret = novaTraffic.sendPlayList(1, content, "192.168.0.220", 5000);
```

2.8 发送局部更新

【功能描述】

发送局部更新。如播放需要附带图片、视频等资源文件。需使用下发文件接口提前发送。

【接口定义】

```
public int sendLocalUpdate(int id, String content)
```

【输入参数】

参数	类型	说明	备注
id	int	下发局部更新的区域 ID（范围 1~10）	相同 ID 会覆盖之前的播放内容
content	String	播放文件的内容，具体参考标准协议的播放格式说明	UTF-8 编码

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String content =  
"[all]\r\nitems=1\r\n[item1]\r\nparam=100,1,1,1,0,5,1,0,1\r\nimg1=0,0,001.jpg,0,0,0\r\nimgparam1=100";  
  
int ret = novaTraffic.sendLocalUpdate (1, content);
```

2.9 移除局部更新

【功能描述】

移除某个区域的局部更新。

【接口定义】

```
public int removeLocalUpdate (int id)
```

【输入参数】

参数	类型	说明	备注
id	int	移除局部更新的区域 ID（范围 1~10）	代表局部更新

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic. removeLocalUpdate (1);
```

2.10 发送文件

【功能描述】

发送本地某个资源文件。

【接口定义】

```
public int sendFile(File file, String saveFileName)
```

【输入参数】

参数	类型	说明	备注
file	File	本地文件资源	
saveFileName	String	保存到控制卡上的文件名	UTF-8

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.sendFile(new File("d:\\2.jpg"), "1.jpg");
```

2.11 手动亮度调节

【功能描述】

手动亮度调节。

【接口定义】

```
public int setBrightness(int brightness)
```

【输入参数】

参数	类型	说明	备注
brightness	int	亮度等级	范围 0~255

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBrightness(130, "192.168.0.220", 5000);
```

2.12 手动开关屏

【功能描述】

手动亮度调节。

【接口定义】

```
public int setPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	<ul style="list-style-type: none">• True: 开屏• False: 关屏	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
int ret = novaTraffic. setPower (true);
```

2.13 清理文件

【功能描述】

清理文件。

【接口定义】

```
public int cleanUpFiles(int type)
```

【输入参数】

参数	类型	说明	备注
type	int	<ul style="list-style-type: none">• 0: 清理所有媒体文件• 1: 清理无效媒体文件• 2: 清理所有文件• 3: 清理升级包	参数只有 0~3 为有效值，如传入其他 int 型数值，统一按参数 1 场景处理

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
NovaTraffic novaTraffic = new NovaTraffic("192.168.0.220", 5000);  
int ret = novaTraffic. cleanUpFiles (1);
```

2.14 设置自动亮度控制参数

【功能描述】

设置终端自动亮度控制参数。

【接口定义】

```
public class BrightnessItem{  
    private int environment;//环境亮度  
    private int screen;//屏体亮度  
}  
public int setBrightnessAuto(List<BrightnessItem> brightnessItem)
```

【输入参数】

List 必须为 8 组 或者传 null 清除。

参数	类型	说明	备注
environment	int	• 光探头采集的亮度值	0-255
screen	int	• 屏体的亮度值	0-100

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
List<BrightnessItem> brightnessItem = new ArrayList<BrightnessItem>();  
for(int i = 0 ; i < 8 ; i++) {  
    BrightnessItem item = new BrightnessItem(10 * i,10 * i);  
    brightnessItem.add(item);  
}  
Int ret = novaTraffic.setBrightnessAuto(brightnessItem);
```

2.15 获取自动亮度控制参数

【功能描述】

查询终端当前自动亮度控制参数。

【接口定义】

```
List<BrightnessItem> BrightnessItemList = novaTraffic.getBrightnessAuto()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
environment	int	• 光探头采集的亮度值	0-255
screen	int	• 屏体的亮度值	0-100

- null: 错误

【使用范例】

```
List<BrightnessItem> BrightnessItemList = novaTraffic.getBrightnessAuto();
```

2.16 控制屏体电源开关

【功能描述】

设置为关闭时，屏体先变黑，然后关闭本板电源。开屏为开启本板电源，屏体再变亮。

【接口定义】

```
public int setBoardPowerAndScreenPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	• true 开 false 关	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBoardPowerAndScreenPower(true);
```

2.17 控制本板电源开关

【功能描述】

控制卡本板继电器开关。

【接口定义】

```
public int setBoardPower(boolean power)
```

【输入参数】

参数	类型	说明	备注
power	boolean	• true 开 false 关	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setBoardPower(true);
```

2.18 控制多功能卡电源开关

【功能描述】

控制多功能卡电源的开关。

【接口定义】

```
public int setMultiFunctionCardPower(int cardIndex, int roadIndex, boolean power)
```

【输入参数】

参数	类型	说明	备注
cardIndex	int	• 多功能卡位置索引	0-255
roadIndex	int	• 电源路数索引	0-7
power	boolean	• true 开 false 关	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setMultiFunctionCardPower(0,0,true);
```

2.19 查询设备状态

【功能描述】

查询设备状态。

【接口定义】

```
public class DeviceType {  
    private Date date;  
    private boolean isBoxDoorOpen;  
    private boolean isScreenPowerOpen;  
    private int temperature;  
    private boolean isInputSourceInside;  
    private int environmentBrightness;  
    private int brightnessType;  
    private int screenBrightness;  
}  
public DeviceType getDeviceType()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
date	Date	• 时间	
isBoxDoorOpen	boolean	• 箱门是否打开	
isScreenPowerOpen	boolean	• 屏体电源状态	
temperature	int	• 温度	摄氏度
isInputSourceInside	boolean	输入源	
environmentBrightness	int	• 当前环境亮度值	0-255
brightnessType	int	• 亮度控制方式	1-自动 2-手动 3-定时
screenBrightness	int	• 亮度	0-255

- null: 错误

【使用范例】

```
DeviceType mDeviceType = novaTraffic.getDeviceType();
```

2.20 设置日期时间

【功能描述】

设置设备的日期时间。

【接口定义】

```
public int setTime(Date date)
```

【输入参数】

参数	类型	说明	备注
date	Date	• 日期（年月日时分秒）	此功能不能更改时区

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setTime(new Date());
```

2.21 设备重启

【功能描述】

重启设备。

【接口定义】

```
public int restartDevice()
```

【输入参数】

无

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.restartDevice();
```

2.22 恢复 IP 出厂设置

【功能描述】

恢复设备 IP 到出厂 IP: 192.168.0.220。恢复 IP 出厂设置无回复, 设置后连接中断。

【接口定义】

```
public void resetDeviceIpToDefault()
```

【输入参数】

无

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.resetDeviceIpToDefault ();
```

2.23 查询版本信息

【功能描述】

查询设备的软件版本信息。

【接口定义】

```
public String getDeviceVersion()
```

【输入参数】

无

【返回参数】

- null: 失败

【使用范例】

```
String verison = novaTraffic.getDeviceVersion();
```

2.24 设置屏体基本参数

【功能描述】

设置设备的屏体网络连接参数, 包括控制卡 IP 参数及其连接的上位机参数。当 ip 发生变化时会断开连接。

【接口定义】

```
class DeviceNetBasicParam {  
    private String deviceIp;  
    private int port;  
    private String mark;  
    private String gate;  
    private String serviceIp;  
}  
  
public void setDeviceNetBasicParam(DeviceNetBasicParam mDeviceBasicParam)
```

【输入参数】

参数	类型	说明	备注
deviceIp	String	• 设备 IP	例"192.168.0.220", 当只需改 port 和 serviceIp 时传 deviceIp, mark、gate 可都为 null。
port	int	• 端口	控制卡做服务端时, 为上位机连接控制卡所需端口号; 控制卡做客户端时, 为控制卡连接上位机所需端口号。无需改变可单独为 null
mark	String	• 设备的子网掩码	
gate	String	• 设备的网关	
serviceIp	String	• 控制卡做客户端时, 对应的服务器 IP	当控制卡服务端时无效, 无需改变可单独为 null

【返回参数】

- 无

【使用范例】

```
DeviceNetBasicParam mDeviceNetBasicParam = new  
DeviceNetBasicParam("192.168.0.220",5000,  
                    "255.255.255.0","192.168.0.1","192.168.0.103");  
novaTraffic.setDeviceNetBasicParam(mDeviceNetBasicParam);
```

2.25 获取屏幕点阵数据

【功能描述】

获取显示屏像素宽度和高度。

【接口定义】

```
public class DeviceSize {  
    private int width;  
    private int height;  
}  
  
public DeviceSize getDeviceSize()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
width	int	• 显示屏像素宽度	
height	int	• 显示屏像素高度	

- null: 失败

【使用范例】

```
DeviceSize mDeviceSize = novaTraffic.getDeviceSize();
```

2.26 设置音量大小

【功能描述】

设置设备音量大小。

【接口定义】

```
public int setDeviceVolume(int volume)
```

【输入参数】

参数	类型	说明	备注
volume	int	• 音量大小	取值范围: 0 - 100

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setDeviceVolume(50);
```

2.27 获取内部存储空间大小

【功能描述】

获取内部存储空间的总大小和剩余大小。

【接口定义】

```
public class DeviceMemorySize {  
    private int totolSize;  
    private int remainSize;  
}  
  
public DeviceMemorySize getDeviceMemorySize()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
totolSize	int	• 总空间大小	单位 MB
remainSize	int	• 剩余空间大小	单位 MB

- null: 失败

【使用范例】

```
DeviceMemorySize mDeviceMemorySize = novaTraffic.getDeviceMemorySize();
```

2.28 设置时间时区

【功能描述】

设置设备时间时区。

【接口定义】

```
public class TimeZoneParam {  
    private long utcTime;  
    private String timeZoneId;  
    private String gmt;  
}  
  
public int setTimeAndTimeZone(TimeZoneParam timeZoneParam)
```

【输入参数】

参数	类型	说明	备注
utcTime	long	• UTC 时间	
timeZoneId	String	• 参考附录 2	
gmt	String	• 参考附录 2	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
TimeZoneParam mTimeZoneParam = new TimeZoneParam((long)(new Date().getTime()), "Asia/Shanghai", "GMT+08:00");  
int ret = novaTraffic.setTimeAndTimeZone(mTimeZoneParam);
```

2.29 获取时间时区

【功能描述】

获取内部存储空间的总大小和剩余大小。

【接口定义】

```
public TimeZoneParam getTimeAndTimeZone()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
utcTime	long	• UTC 时间	
timeZoneId	String	• 参考附录 2	
gmt	String	• 参考附录 2	

- null: 失败

【使用范例】

```
TimeZoneParam mTimeZoneParam = novaTraffic.getTimeAndTimeZone();
```

2.30 设置 NTP 参数

【功能描述】

设置控制卡 NTP 校时服务器参数。

【接口定义】

```
public class DeviceNTPParam {  
    private boolean isOpen;  
    private String addr;  
}  
  
public int setDeviceNTPParam(DeviceNTPParam deviceNTPParam)
```

【输入参数】

参数	类型	说明	备注
isOpen	long	• 是否开启 ntp 对时	
addr	String	• Ntp 对时 url	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
DeviceNTPParam mDeviceNTPParam = new DeviceNTPParam(false,null);  
int ret = novaTraffic.setDeviceNTPParam(mDeviceNTPParam);
```

2.31 获取 NTP 参数

【功能描述】

获取控制卡当前使用的 NTP 校时服务器参数。

【接口定义】

```
public DeviceNTPParam getDeviceNTPParam()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
isOpen	long	• 是否开启 ntp 对时	
addr	String	• Ntp 对时 url	

- null: 失败

【使用范例】

```
DeviceNTPPParam mDeviceNTPPParam = novaTraffic.getDeviceNTPPParam();
```

2.32 设置高温关屏温度

【功能描述】

温度到达多少度后关屏。

【接口定义】

```
public int setCloseScreenTemperature(int value)
```

【输入参数】

参数	类型	说明	备注
value	int	• 关屏温度	系统最小值为 70，0 表示不进行关屏处理

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setCloseScreenTemperature(80);
```

2.33 获取高温关屏温度

【功能描述】

获取高温关屏温度。

【接口定义】

```
public int getCloseScreenTemperature()
```

【输入参数】

无

【返回参数】

- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.getCloseScreenTemperature();
```


2.34 设置网络中断后屏体播放策（虚连接）

【功能描述】

设置网络在虚连接检测时间阈值内未收到任何数据时（包括心跳），播放指定节目的参数。

【接口定义】

```
public int setVirtualConnection(boolean isOpen,int secondTime, int id)
```

【输入参数】

参数	类型	说明	备注
isOpen	boolean	• 是否开启虚连接	
secondTime	int	• 秒	最低 60 秒, 小于 60 秒按 60 秒计算
id	int	• 播放列表 id	

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret = novaTraffic.setVirtualConnection(false, 0,1);
```

2.35 FTP 发送文件

【功能描述】

设备下载指定 URL FTP 地址的内容。不支持多文件发送。下载完毕回传后在发送另一个下载指令。

【接口定义】

```
public int sendFtpDownloadaddress(String url)
```

【输入参数】

参数	类型	说明	备注
url	String	• FTP 文件存储路径	Utf-8 格式, 举例: ftp://user:password@192.168.0.108:21/play001.lst

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret =  
novaTraffic.sendFtpDownloadaddress("ftp://user:password@192.168.0.103:21/UPDA  
TE_OS_JT-TAURUSV010803CN0601.nuzip");
```

2.36 升级文件

【功能描述】

通过发送文件或 FTP 发送文件传输升级 NUZIP 文件后，使用此接口传入文件名，升级 APP 或系统。

【接口定义】

```
public int updateFileName(String path)
```

【输入参数】

参数	类型	说明	备注
name	String	• 升级文件名 称	Utf-8 格式， UPDATE_OS_JT-TAURUSV010803CN0601.nuzip

【返回参数】

- 1: 成功
- 小于 0: 错误参考附录 1

【使用范例】

```
int ret =  
novaTraffic.updateFileName("UPDATE_OS_JT-TAURUSV010803CN0601.nuzip");
```

2.37 查询当前播放内容

【功能描述】

查询设备当前播放的内容。切换过程中可能会回复空内容。

【接口定义】

```
public class DeviceNowPlayItem {  
    private boolean isOpen;  
    private int playId;  
    private String content;  
}
```

```
public DeviceNowPlayItem getNowPlayContent()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
isOpen	boolean	• 是否开屏	
playId	int	• 当前播放列表 id	id > 0
content	String	• 当前播放内容	

【使用范例】

```
DeviceNowPlayItem mDeviceNowPlayItem = novaTraffic.getNowPlayContent();
```

2.38 查询当前播放列表全部内容

【功能描述】

查询当前播放列表的全部内容。

【接口定义】

```
public class DeviceNowPlayList {  
    private int playId;  
    private String content;  
}  
  
public DeviceNowPlayList getNowPlayAllContent()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
playId	int	• 当前播放列表 id	id > 0
content	String	• 当前播放内容	

【使用范例】

```
DeviceNowPlayList mDeviceNowPlayList = novaTraffic.getNowPlayAllContent();
```

2.39 获取当前播放截图

【功能描述】

获取当前播放截图。

【接口定义】

```
public int getDeviceScreenshot(String filePath)
```

【输入参数】

参数	类型	说明	备注
filePath	String	• 截图文件保存路径	UTF-8 格式，举例： "d:\\2.jpg"

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
int ret = novaTraffic.getDeviceScreenshot(filePath);
```

2.40 设置定时开关屏

【功能描述】

设置终端定时开关屏的控制参数。时间段内开，时间段外关。规则覆盖操作。传 null 清除。

【规则】

比如定时开关屏策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒~K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class DevicePowerByTimeParam {  
    private Date start;  
    private Date end;  
}  
  
public int setDevicePowerByTimeList(List<DevicePowerByTimeParam> list)
```

【输入参数】

参数	类型	说明	备注
start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
String endStr = "2020.12.31 23:59:59";
SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");
sdf.setTimeZone(TimeZone.getDefault());
Date start = null;
Date end = null;
try {
    List<DevicePowerByTimeParam> mDevicePowerByTimeParamList = new
    ArrayList<DevicePowerByTimeParam>();
    start = sdf.parse(startStr);
    end = sdf.parse(endStr);
    DevicePowerByTimeParam mDevicePowerByTimeParam = new
    DevicePowerByTimeParam(start, end);
    mDevicePowerByTimeParamList.add(mDevicePowerByTimeParam);
    int ret =
    novaTraffic.setDevicePowerByTimeList(mDevicePowerByTimeParamList);
} catch (ParseException e) {
    e.printStackTrace();
}
```

2.41 获取定时开关屏

【功能描述】

获取定时开关屏参数。

【接口定义】

```
public List<DevicePowerByTimeParam> getDevicePowerByTimeList()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）

- null：失败

【使用范例】

```
List<DevicePowerByTimeParam> mDevicePowerByTimeParamList =  
novaTraffic.getDevicePowerByTimeList();
```

2.42 设置定时播放列表

【功能描述】

设置终端定时播放的控制参数。规则覆盖操作。传 null 清除。

【规则】

比如定时播放策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒~K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class PlayByTimeParam {  
    private Date start;  
    private Date end;  
    private int id;  
}  
  
public int setPlayByTimeList(List<PlayByTimeParam> list)
```

【输入参数】

参数	类型	说明	备注
----	----	----	----

start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）
Id	int	• 播放列表 id	1-100

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
String endStr = "2020.12.31 23:59:59";
SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");
sdf.setTimeZone(TimeZone.getDefault());
Date start = null;
Date end = null;
try {
    List<PlayByTimeParam> mPlayByTimeParamList = new
    ArrayList<PlayByTimeParam>();
    start = sdf.parse(startStr);
    end = sdf.parse(endStr);
    PlayByTimeParam mPlayByTimeParam = new PlayByTimeParam(start, end,1);
    mPlayByTimeParamList.add(mPlayByTimeParam);
    int ret = novaTraffic.setPlayByTimeList(mPlayByTimeParamList);
} catch (ParseException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

2.43 获取定时播放列表

【功能描述】

获取定时播放列表。

【接口定义】

```
public List<PlayByTimeParam> getPlayByTimeList()
```

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）
Id	int	• 播放列表 id	1-100

- null: 失败

【使用范例】

```
List<PlayByTimeParam> mPlayByTimeParamList = novaTraffic.getPlayByTimeList();
```

2.44 设置定时亮度

【功能描述】

设置终端定时亮度的控制参数。规则覆盖操作。传 null 清除。

【规则】

比如定时亮度调节策略为：

A 年 B 月 C 日 D 时 E 分 F 秒 ~H 年 I 月 J 日 K 时 L 分 M 秒；

那么该条定时策略可解读为：

A 年 B 月 C 日到 H 年 I 月 J 日这个时间段内，每天 D 时 E 分 F 秒~K 时 L 分 M 秒，定时策略生效。

【接口定义】

```
public class BrightnessByTimeParam {  
    Date start;  
    Date end;  
    int brightness;  
}  
  
public int setBrightnessByTimeList(List<BrightnessByTimeParam> list);
```

【输入参数】

参数	类型	说明	备注
----	----	----	----

start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）
brightness	int	• 亮度值	0-255

【返回参数】

- 1：成功
- 小于 0：错误参考附录 1

【使用范例】

```
String startStr = "2020.1.1 00:00:00";
String endStr = "2020.12.31 23:59:59";
SimpleDateFormat sdf = new SimpleDateFormat("yyyy.MM.dd HH:mm:ss");
sdf.setTimeZone(TimeZone.getDefault());
Date start = null;
Date end = null;
try {
    List<BrightnessByTimeParam> mBrightnessByTimeParamList = new
    ArrayList<BrightnessByTimeParam>();
    start = sdf.parse(startStr);
    end = sdf.parse(endStr);
    BrightnessByTimeParam mBrightnessByTimeParam = new
    BrightnessByTimeParam(start, end,50);
    mBrightnessByTimeParamList.add(mBrightnessByTimeParam);
    ret = novaTraffic.setBrightnessByTimeList(mBrightnessByTimeParamList);
    tempStr = "设置定时亮度列表结果:";
    outTest += tempStr + ret + "\n";
    System.out.println(name + tempStr +ret);
} catch (ParseException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

2.45 获取定时亮度

【功能描述】

获取定时亮度。

【接口定义】

public List<BrightnessByTimeParam> getBrightnessByTimeList()

【输入参数】

无

【返回参数】

参数	类型	说明	备注
start	Date	• 开始时间	定时功能的起始时间点（年月日时分秒）
end	Date	• 结束时间	定时功能的结束时间点（年月日时分秒）
brightness	int	• 亮度值	0-255

- null：失败

【使用范例】

```
List<BrightnessByTimeParam> mBrightnessByTimeParamList =  
novaTraffic.getBrightnessByTimeList();
```

3 附录

3.1 错误码

参数名	说明
-1	找不到 IP 对应主机

参数名	说明
-2	IO 错误
-3	文件名发送未收到回复
-4	文件名接收到错误
-5	文件块发送未收到回复
-6	文件块接收到错误
-7	文件块未收到结束标识
-8	文件块结束标识接受错误
-9	指定播放列表未收到回复
-10	指定播放列表接收错误
-11	发送未收到回复
-12	发送接收错误
-13	文件不存在
-14	SOCKET 错误
-15	传入参数错误

3.2 时区信息

时区 ID	时区 GMT 偏移	时区中文备注
Pacific/Majuro	GMT+12:00	马朱罗
Pacific/Midway	GMT-11:00	中途岛
Pacific/Honolulu	GMT-10:00	檀香山
America/Anchorage	GMT-09:00	安克雷奇
America/Los_Angeles	GMT-08:00	洛杉矶/美国太平洋
America/Tijuana	GMT-08:00	提华纳/美国太平洋
America/Phoenix	GMT-07:00	凤凰城美国山区
America/Chihuahua	GMT-07:00	奇瓦瓦

时区 ID	时区 GMT 偏移	时区中文备注
America/Denver	GMT-07:00	丹佛/美国山区
America/Costa_Rica	GMT-06:00	哥斯达黎加/美国中部
America/Chicago	GMT-06:00	芝加哥/美国中部
America/Mexico_City	GMT-06:00	墨西哥城/美国中部
America/Regina	GMT-06:00	里贾纳/美国中部
America/Bogota	GMT-05:00	波哥大/哥伦比亚
America/New_York	GMT-05:00	纽约/美国东部
America/Caracas	GMT-04:30	加拉加斯/委内瑞拉
America/Barbados	GMT-04:00	巴巴多斯/大西洋
America/Manaus	GMT-04:00	玛瑙斯/亚马逊
America/Santiago	GMT-03:00	圣地亚哥
America/St_Johns	GMT-03:30	圣约翰/纽芬兰
America/Sao_Paulo	GMT-03:00	圣保罗
America/Argentina/Buenos_Aires	GMT-03:00	布宜诺斯艾利斯
America/Godthab	GMT-03:00	戈特霍布
America/Montevideo	GMT-03:00	蒙得维的亚/乌拉圭
Atlantic/South_Georgia	GMT-02:00	南乔治亚
Atlantic/Azores	GMT-01:00	亚述尔群岛
Atlantic/Cape_Verde	GMT-01:00	佛得角
Africa/Casablanca	GMT+00:00	卡萨布兰卡
Europe/London	GMT+00:00	伦敦/格林尼治

时区 ID	时区 GMT 偏移	时区中文备注
Europe/Amsterdam	GMT+01:00	阿姆斯特丹/中欧
Europe/Belgrade	GMT+01:00	贝尔格莱德/中欧
Europe/Brussels	GMT+01:00	布鲁塞尔/中欧
Europe/Sarajevo	GMT+01:00	萨拉热窝/中欧
Africa/Windhoek	GMT+01:00	温得和克
Africa/Brazzaville	GMT+01:00	布拉扎维/西部非洲
Asia/Amman	GMT+02:00	安曼/东欧
Europe/Athens	GMT+02:00	雅典/东欧
Asia/Beirut	GMT+02:00	贝鲁特/东欧
Africa/Cairo	GMT+02:00	开罗/东欧
Europe/Helsinki	GMT+02:00	赫尔辛基/东欧
Asia/Jerusalem	GMT+02:00	耶路撒冷/以色列
Europe/Minsk	GMT+03:00	明斯克
Africa/Harare	GMT+02:00	哈拉雷/中部非洲
Asia/Baghdad	GMT+03:00	巴格达
Europe/Moscow	GMT+03:00	莫斯科
Asia/Kuwait	GMT+03:00	科威特
Africa/Nairobi	GMT+03:00	内罗毕/东部非洲
Asia/Tehran	GMT+03:30	德黑兰/伊朗
Asia/Baku	GMT+04:00	巴库
Asia/Tbilisi	GMT+04:00	第比利斯

时区 ID	时区 GMT 偏移	时区中文备注
Asia/Yerevan	GMT+04:00	埃里温
Asia/Dubai	GMT+04:00	迪拜
Asia/Kabul	GMT+04:30	喀布尔/阿富汗
Asia/Karachi	GMT+05:00	卡拉奇
Asia/Oral	GMT+05:00	乌拉尔
Asia/Yekaterinburg	GMT+05:00	叶卡捷林堡
Asia/Calcutta	GMT+05:30	加尔各答
Asia/Colombo	GMT+05:30	科伦坡
Asia/Katmandu	GMT+05:45	加德满都/尼泊尔
Asia/Almaty	GMT+06:00	阿拉木图
Asia/Rangoon	GMT+06:30	仰光/缅甸
Asia/Krasnoyarsk	GMT+07:00	克拉斯诺亚尔斯克
Asia/Bangkok	GMT+07:00	曼谷
Asia/Shanghai	GMT+08:00	北京/中国
Asia/Hong_Kong	GMT+08:00	香港/中国
Asia/Irkutsk	GMT+08:00	伊尔库茨克
Asia/Kuala_Lumpur	GMT+08:00	吉隆坡
Australia/Perth	GMT+08:00	佩思
Asia/Taipei	GMT+08:00	台北时间
Asia/Seoul	GMT+09:00	首尔
Asia/Tokyo	GMT+09:00	东京/日本

时区 ID	时区 GMT 偏移	时区中文备注
Asia/Yakutsk	GMT+09:00	雅库茨克
Australia/Adelaide	GMT+09:30	阿德莱德
Australia/Darwin	GMT+09:30	达尔文
Australia/Brisbane	GMT+10:00	布里斯班
Australia/Hobart	GMT+10:00	霍巴特
Australia/Sydney	GMT+10:00	悉尼
Asia/Vladivostok	GMT+10:00	符拉迪沃斯托克/海参崴
Pacific/Guam	GMT+10:00	关岛
Asia/Magadan	GMT+10:00	马加丹
Pacific/Auckland	GMT+12:00	奥克兰
Pacific/Fiji	GMT+12:00	斐济
Pacific/Tongatapu	GMT+13:00	东加塔布

版权所有 ©2020 西安诺瓦星云科技股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

NOVA STAR 是诺瓦科技的注册商标。

声明

欢迎您选用西安诺瓦星云科技股份有限公司（以下简称诺瓦科技）的产品，如果本文档为您了解和使用产品带来帮助和便利，我们深感欣慰。我们在编写文档时力求精确可靠，随时可能对内容进行修改或变更，恕不另行通知。如果您在使用中遇到任何问题，或者有好的建议，请按照文档提供的联系方式联系我们。对您在使用中遇到的问题，我们会尽力给予支持，对您提出的建议，我们衷心感谢并会尽快评估采纳。

24小时免费服务热线

400-696-0755

<http://www.novastar-led.cn>

西安总部

地址：西安市高新区科技二路72号西安软件园零壹广场DEF101

电话：029-68216000

官方技术交流群：104159979 / 202735063



诺瓦科技官方微信号