目录

1、说明

简介

这是上位机SDK，用户可通过该SDK开发用于控制播放盒的上位机软件，支持：

- Win7/10
  - 32位
  - 64位
- Linux
  - Ubuntu18.04
  - 中标麒麟V7

- Android 5.0及以上版本
    - armeabi-v7a
    - arm64-v8a
- Mac（待支持。。。）
- iOS
    - 11.0及以上版本

注意事项

在线手册

可在此处查阅最新的开发手册：ViplexCore

Linux

调用方需要链接thread库：-lpthread
静态加载时调用方需要设置为QT程序（qmake示例）：CONFIG += qt

2、全局错误码

错误码	别名	错误解释
0x00(0)	OK	成功
0x01(1)	ERR_COMMON	通用错误
0x02(2)	ERR_NULL	参数为空
0x03(3)	ERR_INVALID_PARAM	无效参数
0x04(4)	ERR_TIMEOUT	超时
0x05(5)	ERR_IO_EXCEPTION	IO异常
0x06(6)	ERR_INTERRUPTED	中断异常
0x07(7)	ERR_INCOMPLETED	未完成
0x08(8)	ERR_INCOMPLETED_SET	未完成的设置异常
0x09(9)	ERR_ALREADY_DONE	已经执行过
0x0A(10)	ERR_SECURITY	安全问题
0x0B(11)	ERR_PERMISSION_DENIED	权限不够（授权失败）
0x0C(12)	ERR_NOT_IMPLEMENTED	未实现的功能
0x0D(13)	ERR_REMOTE_EXCEPTION	远程过程调用异常， 一般用于进程间通信异常
0x0F(15)	ERR_UNSUPPORTEDENCODING	不支持的编码异常
0x10(16)	ERR_JSON_EXCEPTION	Json异常
0x11(17)	ERR_FORBIDDEN	禁止访问
0x12(18)	ERR_NO_SESSION	无会话连接异常

错误码	别名	错误解释
0x13(19)	ERR_NOT_EXISTED	不存在的异常
0x14(20)	ERR_NO_SPACE	无会话连接异常
0x15(21)	ERR_DATABASE_EXCEPTION	数据库异常
0x16(22)	ERR_TOO_FREQUENTLY	操作过于频繁
0x17(23)	ERR_ALREADY_EXISTED	已经存在
0x18(24)	ERR_VERIFY_FAILED	校验失败 （例如文件MD5不一致，系统升级包校验失败）
0x19(25)	ERR_FILE_ILLEGAL	升级包不合法或错误
0x20(32)	ERR_SIGNATURE_NO_MATCH	签名不匹配（升级包）
0x21(33)	ERR_ACCOUNT_NOT_EXIST	账号不存在
0x22(34)	ERR_SCREEN_NOT_CONFIG	未配屏
0x23(35)	ERR_NETWORK	网络异常
0x24(36)	ERR_UNSUPPORTED	终端不支持
0x25(37)	ERR_LORA_SLAVE_UNSUPPORTED	射频从设备不支持 (例如给从设备设置音量，亮度，时间，时区)
0x33(51)	ERR_NOT_ONE_FPGA	升级时，通过产品和平台验证的FPGA的个数不是一个
0x34(52)	ERR_NOT_SUPPORT_PRODUCT	升级校验时，终端产品不支持
0x35(53)	ERR_VERSION_LOW	待升级软件版本低于当前终端已安装版本
0x36(54)	ERR_NOT_SUPPORT_PLATFORM	升级校验时，终端平台不支持
0x42(66)	ERR_BEYOND_AUTHORIZED_COUNT	注册时，超出授权数量
0x43(67)	ERR_LOGIN_LOCKED	登录时，密码已经连续错误3次
0xFF06(65286)	LOG_INIT_ERR	初始化sdk--- 日志异常
0xFF07(65287)	DB_INIT_ERR	初始化sdk---数据库异常
0xFF08(65288)	VALUE_RANGE_ERROR	参数不符合规格-----值超出有效范围
0xFF14(65300)	RemoteHostClosedError	终端已断开---tcp连接被断开（服务端断开）
0xFF18(65304)	SocketTimeoutError	连接失败—socket超时
0xFF21(65313)	SN_IS_ILLEGAL	登录-----sn非法
0xFF22(65314)	ERR_PATH_NOT_EXIST	升级----- 获取文件列表时传递的路径错误
0xFF23(65315)	ERR_FILE_NUZIP_FAILED	升级----- 获取压缩包json文件内容时压缩文件解压异常
0xFF24(65316)	ERR_DESCRIPTION_JSON_NULL	升级-----获取压缩包json文件内容时Json文件内容为空
0xFF25(65317)	PARAM_IS_EXCEPTION	参数不符合规格-----修改密码----参数异常
0xFF27(65319)	NO_MATCH_FILE	在指定路径下没有找到指定后缀文件
0xFF28(65320)	UPDATE_PACKAGE_UPLOADING	升级-----升级OS或APP时，安装包开始上传

错误码	别名	错误解释
0xFF29(65321)	START_UPDATE	升级-----升级OS或APP时，安装包上传完毕，开始升级
0xFF30(65328)	DIR_NOT_EXIST	生成节目时，指定路径不存在
0xFF3a(65338)	CREATE_JSONFILE_FAILED	生成节目，写入Json文件时，文件夹打开失败导致数据写入失败
0xFF3b(65339)	UPLOAD_FILE	FTP上传文件没有传完，会实时上报这个状态码
0xFF3c(65340)	CREATEPROGRAM_ERROR	创建节目时，需要从数据库读取已有programId,读取错误，创建节目失败
0xFF3e(65342)	PROGRAMTEMPLATE_DATABASE_EMPTY	添加模板时，数据库中没有系统默认的模板
0xFF3f(65343)	ERR_CHANGEPWD_OLDPWDERR	原密码不符合规则-----修改密码-----旧密码错误
0xFF40(65344)	ERR_UPDATEPACKAGE_TRANSPORT	升级-----升级OS或APP时，安装包上传完毕
0xFF41(65345)	TEMPLATEID_IS_ERROR	编辑模板时，传入模板ID有误，数据库中没有这个ID
0xFF43(65347)	ERR_INSERTERROR	操作失败---更新数据库--------插入数据库失败
0xFF44(65348)	ERR_DELETEERROR	操作失败---删除数据库失败.
0xFF45(65349)	ERR_TCP_UNCONNECTED	终端已断开 ----TCP 未连接
0xFF46(65350)	ERR_NOT_LOGIN	终端已断开----未登录屏体
0XFF47(65351)	ERR_PWDERROR_CLEARPASSWORD	登录接口返回------ 终端返回用户名和密码错误，但是登录入参是记住密码的，因此帮助handy清除密码，使用特殊的code标识
0xFF48(65352)	OLD_DATABASE_ISEMPTY	数据迁移时，旧数据中并没有数据，传入参数为空
0xFF49(65353)	ERR_ALREADY_LOGIN	已经登录过了，请不要再次执行登录操作
0XFF51(65361)	ERR_UNGETFIRMWARE_VERSIONISNULL	请先调用获取固件版本信息接口
0XFF52(65362)	MEDIA_UPDATE_FINISH	媒体上传成功
0XFF53(65363)	ERR_FTP_SEND_ERROR	ftp发送失败，网络断开
0xFF54(65364)	ERR_VERISON_ROLLBACK	升级版本回退错误
0xFF55(65365)	ERR_NOT_SUPPORT_VERIFY	当前终端版本不支持升级可行性验证
0xFF56(65366)	ERR_FEEDBACK_FAILED	用户反馈上传失败
0xFF57(65367)	ERR_NETWORKCONNECT	网络检测异常
0xFF58(65368)	USB_NO_AVAILABLE_SPACE	U盘空间不足

错误码	别名	错误解释
0xFF59(65369)	UPDATE_PACKAGE_DOWNLOADING	升级包下载中
0xFF5A(65370)	ERR_GET_ONLINE_PACKAGE_FAILED	获取线上升级包失败
0xFF5B(65371)	ERR_STOP_DOWNLOAD_UPDATE_PACKAGE	终止升级包下载
0xFF5C(65372)	ERR_DOWNLOAD_UPDATE_PACKAGE_FAILED	升级包下载失败
0xFF5D(65373)	ERR_LOCAL_FILE	本地文件打开失败，下载文件时无法打开本地文件写入数据
0xFF5E(65374)	ERR_CURL_INIT	curl初始化失败（http访问初始化失败）
0xFF5F(65375)	ERR_FILE_VERIFY_IFAILED	文件校验失败
0xFFFF(65535)	CALLBACK_TIMEOUT	接口超时

3、修改记录

- 2020年

日期	修改人	修改内容
6月19日	张华龙	搭建手册

4、接口示例

4.1、接口示例

SDK提供的功能接口大部分形如：void nvSetColorTemperature(const char *data, ExportViplexCallback callback)

简要描述：

- 获取箱体温度的接口

请求URL： 此项为函数定义 功能接口同时支持同步和异步模式，形如表格：

类型	说明
同步	void nvGetScreenPowerModeSync(const char *data, ExportViplexCallback callback)
异步	void nvGetScreenPowerModeAsync(const char *data, ExportViplexCallback callback)

为了文档更精简并突出关键信息，**在后续的接口文档中只列出不带同异步标识的函数名，如：**

- void nvGetScreenPowerMode(const char *data, ExportViplexCallback callback)

函数参数

由于SDK提供的功能接口都是同样的样式，为了文档更精简并突出关键信息，**在后续的接口文档中不再提供此项信息**

Header名	是否必选	类型	说明
data	必选	string	请求的JSON参数，参数示例和字段描述详见请求参数示例和参数
callback	必选	ExportViplexCallback	接收返回值的回调函数

请求参数示例

```
{
  "sn":"BZSA17332J0A20002272",
  "colorTemperatureInfo":{
    "colorTemperature":6500
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
colorTemperatureInfo	必选	Object	JSON对象
colorTemperature	必选	int	色温值

回调函数返回值

接口调用都是通过ExportViplexCallback回调函数参数提供，为了文档更精简并突出关键信息，**在后续的接口文档中不再提供此项信息，并将code归入到返回参数说明里**

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
result	string	返回信息，返回值示例和字段描述详见**返回示例**和**返回参数说明**

返回示例

```
""
```

返回参数说明

参数名	类型	说明

备注

-

4.1、初始化

4.1.1、（IOS）设置移动端平台类型

简要描述：

- 如果调用方是 ios 系统，则需要调用接口，其他平台忽略此接口

请求URL：

- void nvSetPlatform(const char *platform);

请求方式：

-

参数：

参数名	是否必选	类型	说明
platform	必选	string	由于ios系统锁屏/后台的特殊机制，因此在udp搜索时需要一些特殊处理 填写ios即可

返回示例

```
""
```

返回参数说明

参数名	类型	说明

备注

-

4.1.2、SDK初始化

简要描述：

- SDK初始化接口,调用SDK接口前必须调用此接口完成SDK初始化

请求URL：

- int nvInit(const char *sdkRootDir, const int type, const int platform);

请求方式：

-

参数：

参数名	是否必选	类型	说明
sdkRootDir	必选	string	存放SDK日志、数据库的路径
type	必选	int	调用方来源是第三方还是Nova 1nova自己的平台 0第三方平台
platform	必选	int	1： 移动终端发来的（如手机） 2： 表示传统电脑 3： 表示平板 4： 表示vnnox端发来的 5： 来自iCare

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0初始化成功 65286日志初始化失败 65287数据库初始化失败

备注

-

4.2、音量调节

4.2.1、获取音量

简要描述：

- 获取音量的接口

请求URL：

- void nvGetVolume(const char * data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn": "BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "ratio":75.0
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回data详情
ratio	float	音量百分比

备注

-

4.2.2、设置音量

简要描述：

- 设置音量的接口

请求URL：

- void nvSetVolume(const char * data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn": "BZSA17332J0A20002272",
  "volumeInfo": {
    "ratio": 10
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
volumeInfo	必选	Object	请求信息对象
ratio	必选	float	音量百分比

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回data详情

备注

-

4.2.1、设置定时音量的方案

4.2.1.1、设置定时音量的方案

简要描述：

- 定时设置音量的的接口

请求URL：

- void nvSetTimingVolume(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
```

```
  "sn":"BZSA79353N1310006847",
  "data":{
    "type":"VOLUME",
    "source":{
      "type":0,
      "platform":1
    },
    "enable":true,
    "conditions":[
      {
        "cron":[
          "0 15 10 ? * *"
        ],
        "value":50.0,
        "enable":true,
        "startTime":"2020-09-01 00:00:00",
        "endTime":"4016-06-06 24:00:00"
      },
      {
        "cron":[
          "0 15 10 ? * *"
        ],
        "value":0,
        "enable":true,
        "startTime":"2020-09-01 00:00:00",
        "endTime":"4016-06-06 24:00:00"
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	固定为"VOLUME"
type	必选	int	1nova自己的平台,0第三方平台
platform	必选	int	1移动终端发来的,2表示传统电脑,3表示平板,4表示web端发来的
enable	必选	bool	按条件执行的使能开关
cron	必选	stringarray	重复次数，每条条件使用cron表达式数组表示
value	必选	float	音量值,0-100
enable	必选	bool	该条件执行的使能开关
startTime	必选	string	有效期起始时间yyyy-MM-dd

参数名	是否必选	类型	说明
sn	必选	string	有效期起始时间yyyy-MM-dd

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功

备注

-

4.2.2、获取定时音量的方案

4.2.2.1、获取定时音量的方案

简要描述：

- 定时获取音量的的接口

请求URL：

- void nvGetTimingVolume(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA79353N1310006847"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "conditions":[
```

```
    {
      "value":50.0,
      "cron":[
        "0 15 10 ? * *"
      ],
      "enable":true,
      "endTime":"4016-06-06 24:00:00",
      "startTime":"2020-09-01 00:00:00"
    },
    {
      "value":0.0,
      "cron":[
        "0 15 10 ? * *"
      ],
      "enable":true,
      "endTime":"4016-06-06 24:00:00",
      "startTime":"2020-09-01 00:00:00"
    }
  ],
  "enable":true,
  "source":{
    "platform":1,
    "type":1
  },
  "type":"VOLUME"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
type	string	固定为"VOLUME"
type	int	1nova自己的平台,0第三方平台
platform	int	1移动终端发来的,2表示传统电脑,3表示平板,4表示web端发来的
enable	bool	按条件执行的使能开关
cron	stringarray	重复次数，每条条件使用cron表达式数组表示
value	float	音量值,0-100
enable	bool	该条件执行的使能开关
startTime	string	有效期起始时间yyyy-MM-dd
sn	string	有效期起始时间yyyy-MM-dd

备注

-

4.3、App管理

4.3.1、安装APP

简要描述：

- 安装APP

请求URL：

- void startUploadApk(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{"sn":"BZSA17332J0A20002272", "apkFile":"data/data/com.example.myapp/app_flutter/NETEASE.apk","packageName":"","startAfterInstall":true}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
apkFile	必选	string	待安装app路径
packageName	非必选	string	app名称
startAfterInstall	必选	bool	安装完成是否启动app

返回示例

```
{
  "data":"success"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.3.2、获取已安装的APP信息

简要描述：

- 获取已安装的APP信息

请求URL：

- void getInstalledPackageInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{"appInfos":[{"packageName":"nova.priv.terminal.syssetting","versionName":"1.0.2.0501","versionCode":102,"label":"systemSetting","isNetWorkPermission":true},
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
packageName	string	包名
versionName	tring	版本名称
versionCode	int	版本编号
label	string	APP名称
isNetWorkPermission	boolean	是否允许访问网络

备注

-

4.3.3、获取正在运行的APP版本信息

简要描述：

- 获取正在运行的APP版本信息

请求URL：

- void getRunningPackageInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{"appInfos":[{"packageName":"nova.priv.terminal.syssetting","versionName":"1.0.2.0501","versionCode":102,"label":"systemSetting","isNetWorkPermission":true},
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
packageName	string	包名
versionName	tring	版本名称
versionCode	int	版本编号
label	string	APP名称
isNetWorkPermission	boolean	是否允许访问网络

备注

-

4.3.4、停止运行APP

简要描述：

- 停止运行APP

请求URL：

- void forceStopApp(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{"sn":"BZSA17332J0A20002272", "packageName":"nova.priv.terminal.screenService"}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
packageName	必选	string	要停止app的包名称

返回示例

```
{
  "data":"success"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.3.5、卸载安卓程序

简要描述：

- 卸载安卓程序

请求URL：

- void uninstallPackage(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{"sn":"BZSA17332J0A20002272", "packageName":"nova.priv.terminal.screenService"}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
packageName	必选	string	要卸载app的包名称

返回示例

```
{
  "data":"success"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.4、多屏拼接

4.4.1、获取拼接参数

简要描述：

- 获取多屏拼接参数的接口

请求URL：

- void nvGetSpliceInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "height":0,
  "isScale":false,
  "offsetX":0,
  "offsetY":0,
  "order":0,
  "orderNum":0,
  "videoSource":0,
  "width":0
}
```

返回参数说明

参数名	类型	说明
order	int	拼接屏的顺序
width	int	播放窗口宽度
height	int	播放窗口高度
offsetX	int	视频源x偏移量
offsetY	int	视频源y偏移量
videoSource	int	视频源:内部0,HDMI1
orderNum	int	拼接数量
isScale	bool	是否缩放,true为全屏缩放,false为不缩放
code	int	错误码：0获取成功65535请求超时

备注

-

4.4.2、设置拼接参数

简要描述：

- 设置多屏拼接参数的接口

请求URL：

- void void nvSetSpliceInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "SpliceInfo":{
```

```
    "height":400,
    "isScale":false,
    "offsetX":0,
    "offsetY":0,
    "order":0,
    "orderNum":0,
    "videoSource":0,
    "width":400
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
order	必选	int	拼接屏的顺序
width	必选	int	播放窗口宽度
height	必选	int	播放窗口高度
offsetX	必选	int	视频源x偏移量
offsetY	必选	int	视频源y偏移量
videoSource	必选	int	视频源:内部0,HDMI1
orderNum	int	必选	拼接数量
isScale	bool	必选	是否缩放,true为全屏缩放,false为不缩放

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.5、视频源控制

4.5.1、获取当前视频源

简要描述：

- 获取当前视频源

请求URL：

- void getVideoControlInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "type":"VIDEO_SOURCE_SWITCH",
  "source":{
    "type":1,
    "platform":1
  },
  "videoMode":1,
  "videoSource":0,
  "isScale":false,
  "offsetX":0,
  "offsetY":0,
  "conditions":[
    {
      "cron":"0 0 12 ? * *",
      "source":0,
      "enable":true
    },
    {
      "cron":"0 0 12 ? * *",
      "source":1,
      "enable":true
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
type	String	固定值："VIDEO_SOURCE_SWITCH"
source	object	下发命令的来源
type	int	1：nova自己的平台，
platform	int	0：未知，1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示web端发来的，5：终端本身
videoMode	int	模式，HDMI优先
videoSource	int	视频源
isScale	boolean	是否缩放，true为全屏缩放，false为不缩放
offsetX	int	偏移X
offsetY	int	偏移Y
conditions	object	任务列表
cron	String	cron表达式，用户表示开始时间和重复
source	int	视频源
enable	boolean	该条定时任务是否生效

备注

-

4.5.2、设置当前视频源

简要描述：

- 设置当前视频源

请求URL：

- void setVideoControlInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
   "videdSourceInfo":{
    "type":"VIDEO_SOURCE_SWITCH",
    "source":{
      "type":1,
      "platform":1
    },
    "videoMode":1,
```

```
    "videoSource":0,
    "isScale":false,
    "offsetX":0,
    "offsetY":0,
    "conditions":[
      {
        "cron":"0 0 12 ? * *",
        "source":0,
        "enable":true
      },
      {
        "cron":"0 0 12 ? * *",
        "source":1,
        "enable":true
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	String	固定值："VIDEO_SOURCE_SWITCH"	*
source	object	下发命令的来源	*
type	int	1：nova自己的平台,	0：第三方平台
platform	int	0：未知，1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示web端发来的，5：终端本身	*
videoMode	int	模式，HDMI优先	0，手动
videoSource	int	视频源	SOURCE_INSIDE
isScale	boolean	是否缩放，true为全屏缩放，false为不缩放	*
offsetX	int	偏移X	*
offsetY	int	偏移Y	*
conditions	object	任务列表	*
cron	String	cron表达式，用户表示开始时间和重复	*
source	int	视频源	SOURCE_INSIDE
enable	boolean	该条定时任务是否生效	*

返回示例

```
{
  "data":"success"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.5.1、EDID控制

4.5.1.1、获取EDID

简要描述：

- 获取EDID的接口

请求URL：

- void nvGetVideoEDID(const char *data, ExportViplexCallback callback)

请求方式：

- 请求参数示例

```
{
  "sn":"BZWA17422J1X20000093"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "fieldRate":60,
  "height":900,
  "width":1600
}
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
code	int	错误码：0获取成功
fieldRate	int	场频，即刷新频率
height	int	显示屏高度
height	int	显示屏宽度

备注

-

4.5.1.2、设置EDID

简要描述：

- 设置EDID的接口

请求URL：

- void nvSetVideoEDID(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "taskInfo":{
    "width":1920,
    "height":1080
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	设置信息
width	必选	int	显示屏宽度
height	必选	int	显示屏高度
fieldRate	必选	int	场频，即刷新频率

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功

备注

-

4.6、显示屏用户信息

4.6.1、修改配显示屏用户信息

简要描述：

- 修改配显示屏用户信息的接口

请求URL：

- void nvSetUserInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "userInfo":{
    "aliasName":"easy-pluto",
    "registerAddress":{
      "country":"中国",
      "province":"山西省",
      "city":"西安市",
      "county":"雁塔区",
      "address":"科技二路，西安软件园"
    }
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
userInfo	必选	Object	详情

参数名	是否必选	类型	说明
aliasName	必选	string	显示屏别名
registerAddress	必选	object	注册地址
country	必选	string	国家
province	必选	string	省份
city	必选	string	城市
county	必选	string	县，区
address	必选	string	详细地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.6.2、获取显示屏用户信息

简要描述：

- 获取显示屏用户信息的接口

请求URL：

- void nvGetUserInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "aliasName":"easy-pluto",
  "registerAddress":{
    "country":"中国",
    "province":"山西省",
    "city":"西安市",
    "county":"雁塔区",
    "address":"科技二路，西安软件园"
  }
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
aliasName	string	显示屏别名
registerAddress	object	注册地址
country	string	国家
province	string	省份
city	string	城市
county	string	县，区
address	string	详细地址

备注

-

4.7、搜索连接登录

4.7.1、搜索屏体

简要描述：

- 搜索当前wifi网络下的所有屏体

请求URL：

- void nvSerchTerminalAsync(ExportViplexCallback callback)

请求方式：

- 返回示例

```
{
  "sn":"20393844393033",
  "productName":"AX200",
  "width":400,
  "height":800,
  "rotation":0,
  "aliasName":"XianYataScreen",
  "logined":true,
  "username":[
    "admin"
  ],
  "tcpPort":16603,
  "ftpPort":16602,
  "syssetFtpPort":16604,
  "syssetTcpPort":16605,
  "key":"novaStar",
  "platform":"rk3368",
  "privacy":true,
  "terminalState":0,
  "ignoreTime":100029432434,
  "hasPassword":true,
  "password":"12345678"
}
```


返回参数说明

参数名	类型	说明
code	int	错误码： 0搜索成功，result即为搜索返回内容。 65535搜索超时，代表在4s时间内未能搜索到任何终端
sn	string	产品序列号
productName	string	产品名称
width	int	显示屏宽度,
height	int	显示屏高度,
rotation	int	旋转角度：0，90，180，270
aliasName	string	显示屏别名
logined	boolean	是否有人已经登陆
username	string_array	若已经有人登陆，代表已经登陆的用户名列表
tcpPort	int	tcp连接端口
ftpPort	int	ftp连接端口
syssetFtpPort	int	系统设置ftp端口
syssetTcpPort	int	系统设置tcp端口
key	string	终端返回的key,参与ftp的密码生产规则
platform	string	系统平台信息（暂时会出现rk3368,

参数名	类型	说明
privacy	boolean	True表示终端支持加密模式，当然如果是新的终端一定是True
password	string	密码
terminalState	int	当前屏体状态，terminalState共有6个值,分别代表： 0：未连接任何终端，当前处于未连接状态 1：连接成功但是未登录状态 2：被他人登录状态 3：登录成功状态 4：密码错误状态 5：离线状态，处于搜索不到的状态，但是曾经被搜索到过
ignoreTime	long	记录点击忽略密码的时间戳
hasPassword	boolean	是否记住密码，true表示记住，false，表示没有

备注

-

4.7.2、从数据库获取所有终端

简要描述：

- 查询所有曾经搜索到的终端信息

请求URL：

- void nvFindAllTerminalsAsync(ExportViplexCallback resultCallBack);

请求方式：

- **返回示例**


```
[
  {
    "aliasName":"Taurus-49999787",
    "ftpPort":16602,
    "hasPassWord":false,
    "height":1080,
    "ignoreTime":0,
    "ip":"172.16.9.205",
    "key":"novaStar",
    "logined":false,
    "loginedUsernames":[
      ""
    ],
    "password":"",
```

```
    "platform":"rk312x",
    "privacy":true,
    "productName":"TC300",
    "sn":"BZSA07194A0049999787",
    "syssetFtpPort":16604,
    "syssetTcpPort":16605,
    "tcpPort":16603,
    "terminalState":5,
    "username":[

    ],
    "width":1920
  }
]
```

返回参数说明

参数名	类型	说明
code	int	错误码: 0获取成功
sn	string	产品序列号
productName	string	产品名称
width	int	显示屏宽度,
height	int	显示屏高度,
rotation	int	旋转角度：0，90，180，270
aliasName	string	显示屏别名
logined	boolean	是否有人已经登陆
username	string_array	若已经有人登陆，代表已经登陆的用户名列表
tcpPort	int	tcp连接端口
ftpPort	int	ftp连接端口
syssetFtpPort	int	系统设置ftp端口
syssetTcpPort	int	系统设置tcp端口
key	string	终端返回的key,参与ftp的密码生产规则
platform	string	系统平台信息（暂时会出现rk3368,
privacy	boolean	True表示终端支持加密模式，当然如果是新的终端一定是True
password	string	密码
terminalState	int	当前屏体状态（数据库中搜索到的一直为5）
ignoreTime	long	记录点击忽略密码的时间戳
hasPassword	boolean	是否记住密码，true表示记住，false，表示没有
configInfo	object	支持模块
installedPackageVersions	object	已安装版本信息
productInfo	object	设备信息
firmwareInfo	object	固件版本信息

参数名	类型	说明
screenMosaicInfo	object	多屏拼接信息

备注

-

4.7.3、登录

简要描述：

- 登陆接口和tcp连接是合并在一起的。假如某个时间tcp突然断开了，错误信息会从这个接口返回。

请求URL：

- void nvLoginAsync(const char *data,ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "username":"admin",
  "password":"password",
  "loginType":0,
  "rememberPwd":0
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
username	必选	string	用户名
password	必选	string	密码
loginType	必选	int	0：登陆到屏体管理 1：登陆到系统设置（暗门登录） 2：登陆到诊断模块 3：LCT登录
loginType	必选	int	是否记住密码 0表示No 1表示Yes

返回示例

```
{
  "logined":true,
  "password":"123456",
  "sn":"BZSA17332J0A20002272",
  "username":"admin",
  "validation":true,
  "validition":true,
  "loginedUsernames":[
    "admin"
  ],
  "encrypt":""
}
```

返回参数说明

参数名	类型	说明
code	int	错误码: 0不代表登录成功，具体看备注 65313请求参数中SN无效 65300长连接过程中socket断开，可能是网络原因，也可能是终端断开了socket连接 **注意：当错误码code=65300时，data为sn，目的是告诉上位机哪个终端断开了** 65304执行登录动作时tcp连接失败，一般是因为网络问题（或者ip为空，端口占用等原因） 65353已经登录过了，请不要再进行二次登录 65351记住密码登录时，用户名或密码错误，有可能密码被其他人修改了，上位机可根据此错误码让用户重新手动输入密码 16密码连续错误3次，进入防暴力机制，1min内不允许登录，data返回：{"errorDiscription":"Continuein33seconds.","mRemainTime":33}，mRemainTime代表33s后才可以继续登录
sn	string	产品序列号
username	string	用户名
password	string	密码
validition	boolean	用户名和密码是否有效（旧版本遗留字段）
validation	boolean	用户名和密码是否有效（新版本增加的字段）
logined	boolean	是否成功登陆
loginedUsernames	string_array	已经登陆的账户名称,当logined为false，validation为true时，此字段有效
encrypt	string	密码的暗码，用于找回密码,当logined为false，validation为falsee时，此字段有效

备注

- 判断是否登录成功说明:
  当返回 code == 0时：
  1、当 logined 为 true,且 validition 也为 true 时，代表登录成功

2、当 logined 为 false，但 validition 为 true 时，代表终端被其他人登录
3、当 logined 为 false，且 validition 也为 false 时，代表密码错误

请重点关注错误码

4.7.4、修改密码

简要描述：

- 修改屏体的登录密码

请求URL：

- void nvChangePassWordAsync(const char *data,ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"12345787454544",
  "reSetInfo":{
    "username":"admin",
    "password":"123456",
    "newPassword":"12345678"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
username	必选	string	用户名
password	必选	string	旧密码
newPassword	必选	string	新密码

返回示例

```
changepassword success
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
code	int	错误码： 0修改成功

备注

-

4.7.5、删除屏体

简要描述：

- 删除某个屏体

请求URL：

- void nvDelTerminalInfoAsync(const char *data,ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "type":"deleteTerminal"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	"deleteTerminal"代表此次操作是"删除屏体（终端）"

返回示例

```
Success
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0删除成功 65348删除失败 12参数type错误，未实现的接口
data	string	错误码描述，不用关注

备注

-

4.7.6、屏体信息存储（数据继承）

简要描述：

- 此接口是旧版本屏体数据继承

请求URL：

- void nvSetScreenInfoAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
[
  {
    "aliasName":"Taurus-10006848",
    "screenMosaicInfo":{
      "order":456,
      "videoSource":127
    },
    "ftpPort":16602,
    "hasPassWord":false,
    "height":400,
    "ignoreTime":0,
    "ip":"192.168.1.113",
    "key":"novaStar",
    "logined":false,
    "password":"",
    "productName":"T6",
    "sn":"BZSA79353N1310006848",
    "syssetFtpPort":16604,
    "syssetTcpPort":16605,
    "tcpPort":16603,
    "terminalState":0,
    "width":400
  }
]
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
productName	可选	string	产品名称
width	可选	int	显示屏宽度,
height	可选	int	显示屏高度,
rotation	可选	int	旋转角度：0，90，180，270
aliasName	可选	string	显示屏别名
logined	可选	boolean	是否有人已经登陆
username	可选	string_array	若已经有人登陆，代表已经登陆的用户名列表
tcpPort	可选	int	tcp连接端口
ftpPort	可选	int	ftp连接端口
syssetFtpPort	可选	int	系统设置ftp端口
syssetTcpPort	可选	int	系统设置tcp端口
key	可选	string	终端返回的key,参与ftp的密码生产规则
screenMosaicInfo	可选	object	拼接屏信息
order	可选	int	拼接屏顺序
videoSource	可选	int	视频源个数

返回示例

```
Success
```

返回参数说明

参数名	类型	说明
code	int	错误码: 0存储成功 65347存储失败
data	string	错误码描述，不用关注

备注

-

4.7.7、清除密码

简要描述：

- 清除屏体密码

请求URL：

- void nvDelTerminalInfoAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "type":"clearPassWord"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	"clearPassWord"代表此次操作是"清除密码"

返回示例

```
Success
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0清除成功 65348删除失败 12参数type错误，未实现的接口
data	string	错误码描述，不用关注

备注

-

4.7.8、获取拼接屏信息

简要描述：

- 获取拼接屏信息

请求URL：

- void nvGetTerminalInfoAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
```

```
  "type":"getSystemInfo"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	"getSystemInfo"代表此次操作是"获取系统信息"

返回示例

```
{
  "mosaicWidth":400,
  "mosaicHeight":400,
  "mosaicNum":10
}
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0获取成功 12参数type错误，未实现的接口
mosaicWidth	int	拼接宽度
mosaicHeight	int	拼接高度
mosaicNum	int	拼接总数

备注

-

4.7.9、设置弱密码忽略时间

简要描述：

- 设置“点击忽略弱密码选项”的时间戳后，调用“搜索屏体”时返回设置的字段，否则调用“搜索屏体”时返回空字符串。
  设置此时间戳的目的是在未来某段时间内不再弹出“是否忽略弱密码模态框”。

请求URL：

- `void nvSetTerminalInfoAsync(const char *data, ExportViplexCallback resultCallBack);`

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "type":"setIgnoreTime",
  "value": 123456789012345
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	"getSystemInfo"代表此次操作是"设置弱密码忽略时间"
value	必选	int	要设置的时间戳（毫秒）,比如：1113259324242

返回示例

```
Success
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0设置成功 65347设置失败 12参数type错误，未实现的接口
data	string	错误码描述，不用关注

备注

-

4.7.10、设置系统信息

简要描述：

- 设置“系统信息”。设置一些无关sn的配置信息，详见下面参数

请求URL：

- void nvSetSystemInfoAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "mosaicWidth":400,
  "mosaicHeight":400,
  "lang":"cn",
  "demo":true,
  "userType":"user"
}
```

参数：

参数名	是否必选	类型	说明
mosaicWidth	必选	int	拼接宽度
mosaicHeight	必选	int	拼接高度
lang	必选	string	多语言
demo	必选	bool	是否是demo
userType	必选	string	用户模式

返回示例

```
Success
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0设置成功 65347设置失败
data	string	错误码描述，不用关注

备注

-

4.7.11、退出登录

简要描述：

- 注意：退出登录后，返回成功或失败，login接口的长连接自动失效，不会再返回数据

请求URL：

- void nvLogoutAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "loginType":0
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
loginType	必选	int	0： 断开屏体管理 1： 断开系统设置（暗门） 2： 断开诊断模块 3： 断开LCT登录

返回示例

```
disconnetSuccess
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0退出成功
data	string	错误码描述，不用关注

备注

-

4.8、屏体其他项

4.8.1、固件版本信息

简要描述：

- 获取固件版本信息接口，所有版本的终端均有此接口，返回终端的版本信息。
  1.2.2 版本碰到缺少 mac 字段

请求URL：

- `void nvGetFirmwareInfosAsync(const char *data, ExportViplexCallback resultCallBack);`

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "fpga":"1.0.0.1",
  "model":"AX100V010002CN0501",
  "mainVersion":"1.0.1",
  "productName":"T3",
  "aliasName":"easy-PH0J05447J0330015353",
  "registerAddress":"西安软件园",
  "mac":"30:5A:3A:04:62:C3"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0获取成功 65535接口超时
fpga	string	FPGA版本
model	string	系统版本
mainVersion	string	终端软件主版本
productName	string	产品名称
aliasName	string	屏体别名
registerAddress	string	屏体地址
mac	string	屏体mac地址

备注

-

4.8.2、已安装的软件版本信息

简要描述：

- 获取终端已经安装的软件版本信息

请求URL：

- void nvGetInstalledPackageVersionsAsync(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "packageInfo":{
    "packageName":[
      "nova.priv.terminal.syssetting",
      "nova.priv.terminal.easypluto"
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
packageInfo	必选	object	*
packageName	必选	string_array	软件包名数组

返回示例

```
{
  "result":[
    {
      "packageName":"nova.priv.terminal.syssetting",
      "versionName":"1.0.2.0501",
      "versionCode":102
    },
    {
      "packageName":"nova.priv.terminal.easypluto",
      "versionName":"1.0.2.0502",
      "versionCode":102
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码: 0获取成功 65535接口超时
packageName	string	包名
versionName	string	版本名称
int	string	版本号

备注

-

4.8.3、设备信息

简要描述:

- 获取终端设备信息

请求URL:

- void nvGetProductInfo Async(const char *data, ExportViplexCallback resultCallBack);

请求方式:

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数:

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "productInfo":{
    "productName":"",
    "modelId":10043
  },
  "configInfo":{
    "videoSwitch":false,
    "displayDevice":"LED",
```

```
    "portConfig":[
      {
        "portNO":1,
        "isMainPort":true,
        "belongMainPort":0
      },
      {
        "portNO":2,
        "isMainPort":true,
        "belongMainPort":0
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
code	int	错误码： 0获取成功 65535接口超时
productInfo	object	产品信息
productName	string	产品名称，例如T3,T6
modelId	int	产品编号
configInfo	object	配置信息
videoSwitch	bool	是否支持视频切换，true表示支持，false表示不支持
displayDevice	string	显示设备，有两种LED、LCD
portConfig	object_array	网口配置
portNO	int	网口编号，例如
isMainPort	bool	是否为主网口
belongMainPort	int	对应的主网口，如果该网口是主网口，该字段无效

备注

-

4.8.4、支持的模块信息

简要描述：

- **注意：获取支持的模块信息时首先要调用“获取固件版本信息”接口得到版本号,不然没有版本号无法获取支持的模块信息。**
  字段说明以showdoc为准

请求URL：

- void nvGetconfiguration Async(const char *data, ExportViplexCallback resultCallBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "config_info":{
    "REBOOT":true,
    "SCREENPOWER":true,
    "SYNC_PLAY":true,
    "SPOTS":true,
    "VIDEO_SOURCE_SWITCH":true,
    "UPDATE":true,
    "SCREENSHOT":true,
    "TIMEZONE":true,
    "POWER":true,
    "VOICE":true,
    "LIGHT":true,
    "UPDATEOS":true,
    "CARE":true,
    "VNNOX":true,
    "PLAYLOG":true,
    "COLORTEMPERATURE":true,
    "INDICATORLIGHT":true,
    "NEXUSFONT":true,
    "TERMINALINFO":true,
    "NEWPROTOCOL":true,
    "RELAYPOWER":true,
    "RELAYPOWERCONFIG":true,
    "FIXEDPOINT":true,
    "RESET":true,
    "Brightness":{
```

```
    "Validity":true,
    "CompleteCron":true,
    "FloatValue":true
  },
  "WIFI":true,
  "CustomResolution":true,
  "NetWork":{
    "IsSupport":true,
    "Wifi":true,
    "Wired":true,
    "AP":true,
    "Mobile":true,
    "IsSupportNetWorkCheck":true,
    "WiFiApSwitch":false
  },
  "VideoSource":{
    "IsSupport":true,
    "Hdmi":true,
    "Input":{
      "IsSupport":true,
      "Hdmi":true
    },
    "Output":{
      "IsSupport":true,
      "HdmiToLvds":true
    }
  },
  "RadioFrequencyManage":{
    "IsSupport":true,
    "IsSupportLoraInfo":true,
    "IsSupportProduct":true
  },
  "Upgrade":{
    "IsSupportCheckValid":true
  },
  "SourceOutMode":{
    "IsSupportChangeSourceOutMode":true
  },
  "ZipRunningLog":{
    "IsSupportZipRunningLogInfo":true
  },
  "ReceiveCard":{
    "IsSupport":true,
    "Config":true,
```

```
      "Temperature":true
    },
    "ScreenConfig":{
      "IsSupport":true,
      "Config":true,
      "ScreenJoint":true,
      "IsSupportProduct":true
    },
    "Monitoring":{
      "IsSupport":true,
      "ClearMedia":true,
      "Memory":true,
      "CPU":true,
      "AmbientBrightness":true
    },
    "TimeControl":{
      "IsSupport":true,
      "Manual":true,
      "NTP":true,
      "Lora":true,
      "DayLightOffset":false
    },
    "Resolution":{
      "IsSupport":true,
      "CustomResolution":true
    },
    "SensorBoard":{
      "IsSupport":true,
      "SensorInfo":true,
      "IsSupportProduct":true
    },
    "Rotation":{
      "IsSupport":false,
      "IsSupportProduct":false
    },
    "PlayManager":{
      "IsSupport":true,
      "ProgramListManager":true
    },
    "InfraredDetector":{
      "IsSupport":true,
      "IsSupportProduct":true
    }
  }
```

```
}
```

返回参数说明

参数名	类型	说明

备注

-

4.8.1、校时

4.8.1.1、获取时区

简要描述：

- 获取时区的接口

请求URL：

- void nvGetCurrentTimeAndZone(const char * data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "utcTimeMillis":1585190230828,
  "timeZone":"Europe/London",
  "currentTime":"2020-03-26 02:37:10",
  "isTimeOffsetEnable":true,
  "beginTime":"",
  "endTime":"",
  "timeOffsetValue":0,
```

```
  "gmt":"GMT 08:00"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
utcTimeMillis	long	utc时间的毫秒表示
timeZone	string	时区，例如"Asia/Shanghai"
gmt	string	时区gmt，例如"GMT
isTimeOffsetEnable	boolean	是否开启时间补偿，开启为true，否则false
beginTime	string	补时开始时间，传递开始的月和天，例如"2020-05-23"
endTime	string	补时结束时间，传递开始的月和天，例如"2020-12-23"
timeOffsetValue	long	时间补偿值，以秒为单位，带符号数。例如：

备注

-

4.8.1.2、设置卡上的时间

简要描述：

- 设置卡上的时间的接口

请求URL：

- void nvCalibrateTime(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "currentTime":"2020-03-26 02:37:10",
  "timeZoneInfo":{
    "utcTimeMillis":1585190230828,
    "timeZone":"Europe/London",
    "isTimeOffsetEnable":true,
    "beginTime":"",
    "endTime":"",
    "timeOffsetValue":0,
    "gmt":"GMT 08:00"
```

```
    }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
currentTime	必选	string	当前时区对应的时间时区
timeZoneInfo	必选	Object	时区详情
utcTimeMillis	必选	long	utc时间的毫秒表示
timeZone	必选	string	时区，例如"Asia/Shanghai"
gmt	必选	string	时区gmt，例如"GMT
isTimeOffsetEnable	必选	boolean	是否开启时间补偿，开启为true，否则false
beginTime	必选	string	补时开始时间，传递开始的月和天，例如"2020-05-23"
endTime	必选	string	补时结束时间，传递开始的月和天，例如"2020-05-23"
timeOffsetValue	必选	long	时间补偿值，以秒为单位，带符号数。例如：

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述
utcTimeMillis	long	utc时间的毫秒表示
timeZone	string	时区，例如"Asia/Shanghai"
gmt	string	时区gmt，例如"GMT
isTimeOffsetEnable	boolean	是否开启时间补偿，开启为true，否则false
timeOffsetValue	long	时间补偿值，以秒为单位，带符号数。例如：

备注

-

4.8.2、NTP校时

4.8.2.1、获取对时配置

简要描述：

- 获取对时配置，通过此接口可获取到ntp对时和Lora对时taskArray数组，不区分版本，低版本通过旧网络对时协议获取到信息后，转换为0x99协议的形式

请求URL：

- void nvGetNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "source":{
    "type":1,
    "platform":1
  },
  "taskArray":[
    {
      "type":"LORA_SYNC",
      "action":5,
      "status":1,
      "data":{
        "enable":true,
        "address":1,
        "channel":23,
        "mode":"MASTER",
        "groupId":"novad101",
        "regulation":{
          "timeEnable":false,
          "brightnessEnable":true,
          "volumeEnable":true,
          "environmentalMonitoring":true
        }
```

```
      }
    },
    {
      "type":"NTP_CONFIG",
      "action":5,
      "status":1,
      "data":{
        "enable":true,
        "server":"http://ntpsss.net"
      }
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
source	object	下发命令的来源
type	int	1：nova自己的平台,0：第三方平台
platform	int	1:手机、2:CS、3:平板、4:VNNOX、5:Care、6:LCT、7:Lora
taskArray	object	任务数组
type	string	表征业务类型,固定值:"NTP_CONFIG",或者固定值:"LORA_SYNC"
action	int	表征此命令的动作，固定值：ACTION_SET(5)
status	int	成功或失败,0:未知状态;1:成功;2:失败
data	object	NTP校时配置项
enable	boolean	ntp是否使能
server	string	ntp服务器地址
data	object	射频同步配置项
enable	boolean	射频同步使能
mode	string	主从模式"MASTER"or"SLAVE"
address	int	目标地址
channel	int	目标信道
groupId	string	组id，用户划分设备组,(字符串格式，最大长度为10个字节，由上位机做限制)
regulation	object	同步使能规则
timeEnable	boolean	时间同步使能
brightnessEnable	boolean	亮度同步使能
volumeEnable	boolean	音量同步使能
environmentalMonitoring	boolean	环境检测数据同步使能

备注

-

4.8.2.2、NTP校时配置

简要描述：

- 设置对时配置，包括配置ntp对时和lora对时，以数组的形式下发，也可以单独下发ntp或者lora

请求URL：

- void nvSetNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "TimingInfo":{
    "source":{
      "type":1,
      "platform":1
    },
    "taskArray":[
      {
        "type":"LORA_SYNC",
        "action":4,
        "data":{
          "enable":true,
          "address":1,
          "channel":23,
          "mode":"MASTER",
          "groupId":"novad101",
          "regulation":{
            "timeEnable":false,
            "brightnessEnable":true,
            "volumeEnable":true,
            "environmentalMonitoring":true
          }
        }
      },
      {
        "type":"NTP_CONFIG",
        "action":4,
```

```
        "data":{
          "enable":true,
          "server":"http://ntpsss.net"
        }
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
TimingInfo	必选	string	对时配置对象
source	必选	object	下发命令的来源
type	必选	int	1：nova自己的平台,0：第三方平台
platform	必选	int	1：手机、2：CS、3：平板、4：VNNOX、5：Care、6：LCT、7:Lora
taskArray	必选	object	任务数组
type	必选	string	表征业务类型,固定值："NTP_CONFIG"，或者固定值："LORA_SYNC"
action	必选	int	表征此命令的动作，固定值：ACTION_SET
data	必选	object	NTP校时配置项
enable	必选	boolean	ntp是否使能
server	必选	string	ntp服务器地址
data	必选	object	射频同步配置项
enable	必选	boolean	射频同步使能
mode	必选	string	主从模式"MASTER"or"SLAVE"
address	必选	int	目标地址
channel	必选	int	目标信道
groupId	必选	string	组id，用户划分设备组,(字符串格式，最大长度为10个字节，由上位机做限制)
regulation	必选	object	同步使能规则
timeEnable	必选	boolean	时间同步使能
brightnessEnable	必选	boolean	亮度同步使能
volumeEnable	必选	boolean	音量同步使能
environmentalMonitoring	必选	boolean	环境检测数据同步使能

返回示例

```
{
  "taskArray":[
    {
      "action":4,
      "errorCode":0,
```

```
      "status":1,
      "type":"NTP_CONFIG"
    },
    {
      "action":4,
      "errorCode":0,
      "status":1,
      "type":"LORA_SYNC"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述
type	string	表征业务类型,固定值："NTP_CONFIG"或"LORA_SYNC"
action	int	表征此命令的动作，固定值：ACTION_GET(4)
status	int	成功或失败，0:未知状态；1:成功；2:失败
errorCode	int	错误码

备注

-

4.9、显示屏亮度&环境亮度

4.9.1、模式切换

4.9.1.1、设置模式

简要描述：

- 设置模式的接口

请求URL：

- void nvSetBrightnessAdjustMode(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "brightnessAdjustModeInfo":{
    "mode":"MANUALLY"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
brightnessAdjustModeInfo	必选	Object	请求信息对象
mode	必选	string	MANUALLY表示手动设置,AUTO表示自动

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	错误码对应的具体描述

备注

-

4.9.1.2、获取模式

简要描述：

- 获取模式的接口

请求URL：

- void nvGetBrightnessAdjustMode(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "mode":"MANUALLY ",
  "source":{
    "platform":1,
    "type":1
  },
  "type":"SCREEN_BRIGHTNESS"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述
type	String	表示类型，当为开关屏时为SCREENPOWER,
source	object	表示任务的发布来源,我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等
type	number	1：nova自己的平台,0：第三方平台
platform	number	表示任务的发布来源，我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等
mode	String	MANUALLY表示手动设置,AUTO:表示自动

备注

-

4.9.2、手动控制亮度

4.9.2.1、手动设置亮度

简要描述：

- 手动设置亮度的接口

请求URL：

- void nvSetScreenBrightness(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "screenBrightnessInfo":{
    "ratio":45.0
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
screenBrightnessInfo	必选	Object	请求信息对象
ratio	必选	float	显示屏亮度的值

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	错误码对应的具体描述

备注

-

4.9.2.2、手动获取亮度

简要描述：

- 手动获取亮度的接口

请求URL：

- void nvGetScreenBrightness(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "ratio":45.0,
  "source":{
    "type":1,
    "platform":1
  }
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回data详情
ratio	float	显示屏亮度的值
source	Object	表示任务的发布来源，我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等。
type	int	1：nova自己的平台，
platform	int	1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示vnnox端发来的。5:来自iCare

备注

-

4.9.3、定时或自动亮度调节

4.9.3.1、获取亮度调节方案

简要描述：

- 获取亮度调节方案的接口

请求URL：

- void nvGetBrightnessPolicy(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "brightnessPolicyInfo":{
    "isSupportCompleteCron":false
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
brightnessPolicyInfo	必选	Object	请求信息对象
isSupportCompleteCron	非必选	bool	是否支持完整cron表达式调节及有效期，不传此参数，默认为false

返回示例

```
{
  "enable":true,
  "source":{
    "platform":1,
    "type":1
  },
  "conditions":[
    {
      "opticalFailureInfo":{
        "enable":true,
        "screenBrightness":30
      },
      "enable":true,
      "crons":[
        "0 15 10 ? ** *"
      ],
      "startTime":"2017-09-01 00:00 : 00",
      "endTime":"4016-06-06 23:59:59",
      "segments":[
        {
          "screenBrightness":80,
```

```
          "environmentBrightness":12000
        },
        {
          "screenBrightness":60,
          "environmentBrightness":10000
        }
      ],
      "type":2,
      "args":[
        12000,
        20,
        80,
        40,
        10
      ]
    },
    {
      "opticalFailureInfo":{
        "enable":true,
        "screenBrightness":30
      },
      "enable":true,
      "crons":[
        "0 15 10 ? **"
      ],
      "startTime":"2017-09-01 00:00:00",
      "endTime":"4016-06-06 23:59:59",
      "segments":[

      ],
      "type":1,
      "args":[
        30
      ]
    }
  ],
  "type":"BRIGHTNESS",
  "segmentConfig":{
    "opticalFailureInfo":{
      "enable":true,
      "screenBrightness":30
    },
    "segments":[
      {
```

```
        "screenBrightness":80,
        "environmentBrightness":12000
      },
      {
        "screenBrightness":60,
        "environmentBrightness":10000
      }
    ],
    "args":[
      12000,
      20,
      80,
      40,
      10
    ]
  },
  "timeStamp":"2018-07-19 11:27:15"
}
```


返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回参数
type	string	固定为"BRIGHTNESS"
source	object	表示任务的发布来源,我们现在终端的节目可以来自不同的来源,如PC、移动终端、云服务等
type	number	1：nova自己的平台,0：第三方平台
platform	number	1：移动终端发来的(如手机)，2：表示传统电脑，3：表示平板，4：表示web端发来的。
enable	boolean	使能开关，
conditions	object	按照条件执行的条件集合，我们支持多个条件触发。
type	number	调节类型，1：定时调节，2：自动调节
cron	array	重复次数，每条条件使用cron表达式数组表示，当为数组时，cron表达式之间使用或的关系
startTime	string	策略有效期开始时间yyyy-MM-dd
endTime	string	策略有效期结束时间yyyy-MM-dd
args	object	亮度调节参数，定时调试时args只有一个值为亮度百分比,自动调节时args含有5个值,分别为最大环境亮度,最小环境亮度,最大屏体亮度,最小屏体亮度,

参数名	类型	说明
segments	object	亮度调节参数的分段详细设置
environmentBrightness	number	环境亮度值
screenBrightness	number	对应环境亮度所要调节的屏体亮度值
opticalFailureInfo	object	自动亮度调节时，获取环境亮度失败，是否调到固定值的选项，该字段只对自动亮度调节有效。
enable	boolean	自动亮度调节时，获取环境亮度失败，是否调到固定值
screenBrightness	number	需要调节到的固定亮度值
enable	boolean	该条方案的使能开关
segmentConfig	object	根据环境亮度配置的分段表数据segmentsConfig字段，内部字段参见自动亮度调节的参数说明
timeStamp	string	当前数据的时间戳，为后期可能使用

备注

-

4.9.3.2、设置亮度调节方案

简要描述：

- 1.终端版本大于等于1.3.1
  ：支持starttime，endtime，args中最大屏体亮度，最小屏体亮度支持浮点型。screenBrightness支持浮点型。终端版本小于1.3.1不支持starttime、endtime以及屏体亮度支持浮点型。当下发定时任务时，args仅有一个参数，代表亮度 大于等于1.3.1
  时类型为浮点型，小于1.3.1时为整型
  。2.是否支持浮点型根据支持的模块判断是否是浮点型,支持：args中最大屏体亮度，最小屏体亮度支持浮点型。screenBrightness支持浮点型。不支持
  :args中最大屏体亮度，最小屏体亮度，screenBrightness只能为int类型。

请求URL：

- void nvSetBrightnessPolicy(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "taskInfo":{
    "type":"BRIGHTNESS",
    "source":{
      "type":0,
      "platform":1
    },
    "enable":true,
    "conditions":[
      {
        "type":2,
        "cron":[
          "0 15 10 ? ** *"
        ],
        "args":[
          12000,
          20,
          80,
          40,
          10
        ],
        "startTime":"2017-09-01 00:00 : 00",
        "endTime":"4016-06-06 23:59:59",
        "enable":true
      },
      {
        "type":1,
        "cron":[
          "0 15 10 ? **"
        ],
        "startTime":"2017-09-01 00:00:00",
        "endTime":"4016-06-06 23:59:59",
        "args":[
          30
        ],
        "enable":true
      }
    ],
    "segmentConfig":{
      "args":[
        12000,
        20,
        80,
```

```
        40,
        10
      ],
      "segments":[
        {
          "environmentBrightness":12000,
          "screenBrightness":80
        },
        {
          "environmentBrightness":10000,
          "screenBrightness":60
        }
      ],
      "opticalFailureInfo":{
        "enable":true,
        "screenBrightness":30
      }
    },
    "timeStamp":"2018-07-19 11:27:15"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	请求信息对象
type	必选	string	固定为"BRIGHTNESS"
source	必选	object	表示任务的发布来源，我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等
type	必选	number	1：nova自己的平台，
platform	必选	number	1：移动终端发来的(如手机)，2：表示传统电脑，3：表示平板，4：表示web端发来的
enable	必选	boolean	使能开关
conditions	必选	object	按照条件执行的条件集合，我们支持多个条件触发
type	必选	number	调节类型，1：定时调节，2：自动调节
cron	必选	array	重复次数，每条条件使用cron表达式数组表示，当为数组时，cron表达式之间使用或的关系
args	必选	array	亮度调节参数，定时调试时args只有一个值为亮度百分比，自动调

参数名	是否必选	类型	说明
			节时args含有5个值，分别为最大环境亮度，最小环境亮度，最大屏体亮度，最小屏体亮度，分段数
startTime	必选	string	策略有效期开始时间yyyy-MM-dd
endTime	必选	string	策略有效期结束时间yyyy-MM-dd
segments	非必选	object	亮度调节参数的分段详细设置,(1.3.1以下在每条定时任务里面,>=1.3.1和定时列表平级)
environmentBrightness	非必选	int	环境亮度值
screenBrightness	非必选	number	对应环境亮度所要调节的屏体亮度值
opticalFailureInfo	非必选	object	自动亮度调节时,获取环境亮度失败,是否调到固定值的选项,该字段只对自动亮度调节有效
enable	必选	boolean	自动亮度调节时,获取环境亮度失败,是否调到固定值
screenBrightness	非必选	number	需要调节到的固定亮度值
enable	必选	boolean	该条方案的使能开关
segmentConfig	非必选	object	根据环境亮度配置的分段表数据,内部字段参见自动亮度调节的参数说明
timeStamp	必选	string	当前数据的时间戳,为后期可能使用

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	错误码对应的具体描述

备注

-

4.9.4、环境亮度

4.9.4.1、获取环境亮度

简要描述：

- 获取环境亮度的接口

请求URL：

- void getEnvironmentBrightness(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":1000
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述
value	int	环境亮度的值,单位LUX

备注

-

4.10、温度&色温

4.10.1、色温

4.10.1.1、获取色温

简要描述：

- 获取色温的接口

请求URL：

- void nvGetColorTemperature(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "colorTemperature":9300
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
colorTemperature	int	色温值

备注

-

4.10.1.2、设置色温

简要描述：

- 当前终端无法做较为准确的色温设置，所以选定了3种色温，分别为9300、6500、4700，故回读亦是如此

请求URL：

- void nvSetColorTemperature(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "colorTemperatureInfo":{
    "colorTemperature":6500
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
colorTemperatureInfo	必选	Object	色温详情
colorTemperature	必选	int	色温值

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.10.2、温度

4.10.2.1、获取箱体温度

简要描述：

- 获取箱体温度的接口

请求URL：

- void nvGetScreenUnitTemp(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "temps":[
    {
      "bx":0,
      "by":0,
      "value":20
    },
    {
      "bx":128,
      "by":0,
      "value":25
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述
temps	object	数组对象
bx	int	X坐标
by	int	y坐标
value	float	温度值

备注

-

4.11、开关屏管理

4.11.1、模式切换

4.11.1.1、设置模式

简要描述：

- 设置模式的接口

请求URL：

- void nvSetScreenPowerMode(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "taskInfo":{
    "mode":"MANUALLY"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
mode	必选	string	MANUALLY

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.11.1.2、获取模式

简要描述：

- 获取模式的接口

请求URL：

- void nvGetScreenPowerMode(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "mode":"MANUALLY",
  "source":{
    "platform":1,
    "type":1
  },
  "type":"SCREENPOWER"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
type	String	表示类型，当为开关屏时为SCREENPOWER,
source	object	详情参照附录定义
type	number	1：nova自己的平台,0：第三方平台。
platform	number	1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示we

参数名	类型	说明
		b端发来的。
mode	String	MANUALLY表示手动设置,AUTO：表示自动

备注

-

4.11.2、手动开关屏

4.11.2.1、设置手动开关屏状态

简要描述：

- 设置手动开关屏状态的接口

请求URL：

- void nvSetScreenPowerState(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "taskInfo":{
    "state":"open"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
state	必选	string	显示屏开关状态,"OPEN":开,"CLOSE":关

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.11.2.2、获取手动开关屏状态

简要描述：

- 获取手动开关屏状态的接口

请求URL：

- void nvGetScreenPowerState(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "state":"UNKNOW"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
state	string	显示屏开关状态，"OPEN":开，"CLOSE":关

备注

-

4.11.3、定时开关屏

4.11.3.1、获取定时开关屏

简要描述：

- 获取定时开关屏的接口

请求URL：

- void nvGetScreenPowerPolicy(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "type":"SCREENPOWER",
  "source":{
    "type":0,
    "platform":1
  },
  "enable":true,
  "conditions":[
    {
      "cron":[
        "0 15 10 ? * *",
        "0 0 12 * * ?"
      ],
      "action":"OPEN",
      "enable":true
    },
```

```
    {
      "cron":[
        "0 15 10 ? * *",
        "0 0 12 * * ?"
      ],
      "action":"CLOSE",
      "enable":true
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
type	string	固定为"SCREENPOWER"
source	object	表示任务的发布来源，我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等。
type	number	1：nova自己的平台,0：第三方平台。
platform	number	1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示web端发来的。
enable	boolean	按条件执行的使能开关
conditions	object	按照条件执行的任务集合，我们支持多个任务。
cron	(string)array	重复次数，每条条件使用cron表达式数组表示，当为数组时，cron表达式之间使用或的关系
action	String	"OPEN":开，
enable	boolean	该条件执行的使能开关

备注

-

4.11.3.2、设置开关屏状态

简要描述：

- 设置开关屏状态的接口

请求URL：

- void nvSetScreenPowerPolicy(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "taskInfo":{
    "type":"SCREENPOWER",
    "source":{
      "type":0,
      "platform":1
    },
    "enable":true,
    "conditions":[
      {
        "cron":[
          "0 15 10 ? * *",
          "0 0 12 * * ?"
        ],
        "action":"OPEN",
        "enable":true
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
type	必选	string	固定为"SCREENPOWER"
source	非必选	object	表示任务的发布来源，我们现在终端的节目可以来自不同的来源。如PC、移动终端、云服务等。
type	非必选	number	1：nova自己的平台，
platform	非必选	number	1：移动终端发来的(如手机)，2：表示传统电脑，3：表示平板，4：表示web端发来的。
enable	必选	boolean	按条件执行的使能开关
conditions	必选	object	按照条件执行的任务集合，我们支持多个任务。
cron	必选	(string)array	重复次数，每条条件使用cron表达式数组表示，当为数组时，cron表达式之间使用或的关系
action	必选	String	"OPEN":开，
enable	必选	boolean	该条件执行的使能开关

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.12、清单管理

4.12.1、清单回读

简要描述：

- 获取节目清单的接口

请求URL：

- void nvGetProgramInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "programInfos":[
    {
      "identifier":"62932662fb727f42d695201aa91e32c8",
      "name":"PlayLists1",
```

```
"thumbnailUrl":"program_868679025220684_1/program_thumb.png",
      "statusCode":1,
      "source":0
    },
    {
      "identifier":"62932662fb727f42d695201aa91e32c8",
      "name":"PlayLists2",

"thumbnailUrl":"program_868679025220684_2/program_thumb.png",
      "statusCode":1,
      "source":1
    }
  ]
}
```


返回参数说明

参数名	类型	说明
name	string	播放清单的名称
thumbnailUrl	string	播放清单缩略图绝对路径
statusCode	int	清单播放状态(0表示停止播放;1表示正在播放;2表示暂停播放)
identifier	string	播放方案的唯一标识，这里指的是内容上的，内容相同，则播放方案标识一样，目前存放planList
source	int	清单的来源(0表示局域网发送的节目；1表示Vnnox发送的节目;)
code	int	错误码：0获取成功65535请求超时

备注

-

4.12.2、清单删除

简要描述：

- 删除节目清单的接口

请求URL：

- void nvDeletePlayList(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "taskInfo":{
    "solutions":[
      {
        "name":"program1",
        "identifier":"ccdc78e192d8d97bfd01c82881038a39"
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
identifier	必选	string	播放方案的唯一标识
name	必选	string	播放清单的名称

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.12.3、清单播放

简要描述：

- 获取节目清单的接口

请求URL：

- void nvStartPlay(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "identifier":"62932662fb727f42d695201aa91e32c8"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
identifier	必选	string	播放方案的唯一标识

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.12.4、清单暂停播放

简要描述：

- 暂停节目播放的接口

请求URL：

- void nvPausePlay(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "identifier":"62932662fb727f42d695201aa91e32c8"
}
```

参数：

参数名	是否必选	类型	说明

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
identifier	必选	string	播放方案的唯一标识

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.12.5、清单恢复播放

简要描述：

- 恢复节目播放的接口

请求URL：

- void nvResumePlay(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "identifier":"62932662fb727f42d695201aa91e32c8"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
identifier	必选	string	播放方案的唯一标识

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.13、高级特性

4.13.1、设置同步播放开关

简要描述：

- 设置同步播放开关的接口

请求URL：

- void setSyncPlay(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "enable":true
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
enable	必选	bool	开启关闭同步播放

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.13.2、获取同步播放配置

简要描述：

- 获取同步播放配置

请求URL：

- void getSyncPlay(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{"data":{{"enable":false}}}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0
enable	bool	终端是否开启同步播放

备注

-

4.13.3、恢复出厂设置

简要描述：

- 恢复出厂设置

请求URL：

- void setReBootWipeUserData(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "setInfo":{
    "reason":"terminal connect failed"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
reason	必选	string	恢复出厂设置的原因

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.13.4、清除所有媒体

简要描述：

- 清除所有媒体

请求URL：

- void clearAllMedias(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.13.5、OTG USB 状态获取

简要描述：

- OTG USB 状态获取

请求URL：

- void getOTGUSBMode(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":1.0
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
value	double	USB是否可用，1表示USB不可用，2表示USB可用，adb可调式

备注

-

4.13.6、OTG USB 状态设置

简要描述：

- OTG USB 状态设置

请求URL：

- void setOTGUSBMode(const char *data, ExportViplexCallback callBack)

请求方式：

- 请求参数示例

```
{
  "sn":"BZSA17332J0A20002272",
  "modeInfo":{
    "value":2.0
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
value	必选	double	USB是否可用，1表示USB不可用，2表示USB可用，adb可调式

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.13.7、设置当前分辨率

简要描述：

- 设置当前分辨率

请求URL：

- void setCurrentResolution(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "resolutionInfo":{
    "displayMode":1,
    "resolutionValue":"1280X720p-60"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
displayMode必选	int	显示模式，默认	DISPLAY_INTERFACE_TV(1)
resolutionValue	必选	string	分辨率的值

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.13.8、获取当前分辨率

简要描述：

- 获取当前分辨率

请求URL：

- void getCurrentResolution(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{"sn":"BZSA17332J0A20002272", "requestInfo":{"displayMode":1}}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
displayMode	必选	int	显示模式

返回示例

```
{
  "value":"960x2048p-60"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
value	string	分辨率的值

备注

-

4.13.9、获取终端所支持的分辨率

简要描述：

- 获取终端所支持的分辨率

请求URL：

- void getSupportedResolution(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{"sn":"BZSA17332J0A20002272", "requestInfo":{"displayMode":1}}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
displayMode	必选	int	显示模式

返回示例

```
{
  "result":[
    {
      "value":"2048x256p-60"
    },
    {
      "value":"4096x512p-60"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
value	string	分辨率的值

备注

-

4.13.10、获取终端的输出状态

简要描述：

- 获取终端的输出状态

请求URL：

- void getHdmiOutputStatus(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0
value	int	1表示当前是HDMI，0表示RGB

备注

-

4.13.11、设置终端的输出状态

简要描述：

- 设置终端的输出状态

请求URL：

- void setHdmiOutputStatus(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "info":{
    "value":1
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
value	必选	int	1表示当前是HDMI，0表示RGB

返回示例

```
"sccuss"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0

备注

-

4.13.12、设置自定义分辨率

简要描述：

- 设置自定义分辨率

请求URL：

- void setCustomResolution(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "info":{
    "displayMode":1,
    "width":1920,
    "height":1080
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
displayMode	必选	int	显示模式
width	必选	int	显示宽度
height	必选	int	显示高度

返回示例

```
"sccuss"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0

备注

-

4.13.1、重启

4.13.1.1、获取重启任务

简要描述：

- 获取已经设置的重启任务的接口

请求URL：

- void nvGetReBootTaskAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "conditions":[
    {
      "cron":[
        "0 15 10 ? * *",
        "0 0 12 * * ?"
      ],
      "enable":true
    }
  ],
  "executionType":"BY_CONDITIONS",
  "reason":"Just to test",
  "source":{
    "platform":1,
    "type":0
  },
  "type":"REBOOT"
}
```

返回参数说明

参数名	类型	说明
cron	stringarray	每条条件使用cron表达式数组表示，当为数组时，cron表达式之间使用或的关系
enable	bool	该条件是否启用
executionType	string	IMMEDIATELY立即执行;BY_CONDITIONS按照条件执行
reason	string	重启的原因
platform	int	1移动终端发来的(如手机),2表示传统电脑，3表示平板，4表示web端发来的
type	int	1nova自己的平台,0第三方平台
type	string	固定为"REBOOT"
code	int	错误码：0获取成功65535请求超时

备注

-

4.13.1.2、设置重启任务

简要描述：

- 设置重启的接口

请求URL：

- void void nvSetReBootTask(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "taskInfo":{
    "type":"REBOOT",
    "source":{
      "type":0,
      "platform":1
    },
    "executionType":"IMMEDIATELY",
    "reason":"Just to test",
    "conditions":[
      {
        "cron":[
          "0 15 10 ? * *",
          "0 0 12 * * ?"
        ],
        "enable":true
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
cron	必选	stringarray	每条条件使用cron表达式数组表示，当为数组时，cr

参数名	是否必选	类型	说明
			on表达式之间使用或的关系
enable	必选	该条件是否启用	*
executionType	必选	string	IMMEDIATELY立即执行;BY_CONDITIONS按照条件执行
reason	必选	string	重启的原因
platform	必选	int	1移动终端发来的(如手机),2表示传统电脑，3表示平板，4表示web端发来的
type	必选	int	1nova自己的平台,0第三方平台
type	必选	string	固定为"REBOOT"

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.14、网络配置

4.14.1、WIFI

4.14.1.1、获取WIFI列表

简要描述：

- 获取WIFI列表的接口

请求URL：

- void nvGetWifiList(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
```

```
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "result":[
    {
      "ssid":"AirStation",
      "bssid":"00:24:a5:bc:f4:56",
      "level":4,
      "security":"PSK:WPA2",
      "state":5
    },
    {
      "ssid":"ESAY_PLUTO",
      "bssid":"c8:3a:35:47:ee:08",
      "level":3,
      "security":"WPA",
      "state":0
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
result	jsonarray	wifi列表，json数组
ssid	String	wifi的ssid
bssid	String	wifi的bssid
level	int	wifi信号等级
security	String	密码加密方式，PSK:WPA，PSK:WPA2，PSK:WPA_WPA2，EAP，WEP，NONE
state	int	wifi状态

备注

-

4.14.1.2、连接wifi

简要描述：

- 连接wifi的接口

请求URL：

- void nvConnectWifiNetwork(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "taskInfo":{
    "ssid":"AP10006847",
    "password":"12345678"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
ssid	必选	String	wifi的ssid
password	必选	String	密码

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.14.1.3、断开连接

简要描述：

- 断开连接的接口

请求URL：

- void nvDisconnectWifiNetwork(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "taskInfo":{
    "ssid":"AP10006847"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
ssid	必选	String	wifi的ssid

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.14.1.4、获取wifi的开启状态

简要描述：

- 获取wifi的开启状态的接口

请求URL：

- void nvGetWifiEnabled(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "state":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
state	int	1表示开启，0表示关闭

备注

-

4.14.1.5、设置WIFI开启状态

简要描述：

- 设置WIFI开启状态的接口

请求URL：

- void nvSetWifiEnabled(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "taskInfo":{
    "state":1
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
state	必选	int	1表示开启，0表示关闭

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.14.1.6、忘记密码

简要描述：

- 忘记密码的接口

请求URL：

- void nvSendForgetWifiCommand(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
```

```
  "sn":"",
  "taskInfo":{
    "ssid":"AP10006847"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
ssid	必选	String	wifi的ssid

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时19不存在的异常
data	Object	错误码对应的具体描述

备注

- 如果忘记的wifi的密码为空时返回ErrorCode为： ERR_NOT_EXISTED

4.14.2、WIFI切换

4.14.2.1、获取当前WIFI—AP/Station的状态

简要描述：

- 获取当前WIFI—AP/Station的状态的接口

请求URL：

- void nvGetWifiCurrentStatus(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
```

```
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "taskArray":[
    {
      "type":"WIFI_AP_STATION_SWITCH",
      "action":5,
      "status":1,
      "data":{
        "state":0
      }
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
taskArray	objectArray	此条任务的类型，固定值WIFI_AP_STATION_SWITCH
action	int	表示此命令的动作,ACTION_GET
status	int	此任务执行是否成功：1-成功，2-失败，3-未知
data	Object	具体的命令协议
state	int	当前WIFI—AP/Station的状态，0表示当前为wifi-ap，1表示当前为wifi-station

备注

-

4.14.3、4G网络

4.14.3.1、获取移动网络配置信息

简要描述：

- 获取移动网络配置信息的接口

请求URL：

- void nvGetMobileNetwork(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "basicConfigs":{
    "mobileData":true,
    "dataRoaming":true,
    "enable4G":true,
    "level":1
  },
  "advanced":{
    "networkType":"AUTO",
    "APN":{
      "providerName":"中国移动"
    }
  }
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
basicConfigs	object	基础配置
mobileData	boolean	移动数据开关是否打开
dataRoaming	boolean	漫游是否打开
enable4G	boolean	是否使能4G网络
level	int	4G网络的信号强度0~5

参数名	类型	说明
advanced	object	高级特性
networkType	String	网络类型，详细见说明部分(目前只支持AUTO)
APN	object	APN设置
providerName	string	sim卡的运行商名称

备注

- AUTO:模块内部自动选择

4.14.3.2、设置移动网络配置信息

简要描述：

- 设置移动网络配置信息的接口

请求URL：

- void nvSetMobileNetwork(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "mobileData":{
    "basicConfigs":{
      "mobileData":true,
      "dataRoaming":true,
      "enable4G":true
    },
    "advanced":{
      "networkType":"AUTO",
      "APN":{
        "providerName":"中国移动"
      }
    }
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

参数名	是否必选	类型	说明
mobileData	必选	Object	请求对象信息
basicConfigs	必选	object	基础配置
mobileData	必选	boolean	移动数据开关是否打开
dataRoaming	必选	boolean	漫游是否打开
enable4G	必选	boolean	是否使能4G网络
advanced	必选	object	高级特性
networkType	必选	String	网络类型，目前只支持
APN	必选	object	APN设置
providerName	string	sim卡的运行商名称	*

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述

备注

- AUTO:模块内部自动选择

4.14.3.3、获取移动上网模块是否存在

简要描述：

- 获取移动上网模块是否存在的接口

请求URL：

- void nvIsMobileModuleExisted(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "existed":true
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
existed	boolean	true存在，false不存在

备注

-

4.14.4、有线网络

4.14.4.1、设置有线网络信息

简要描述：

- 设置仔细核对IP地址等信息，否则会导致搜到不到终端

请求URL：

- void nvSetEthernetInfo(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "taskInfo":{
    "ethernets":[
      {
        "scopeId":-1,
        "name":"eth0",
        "dhcp":true,
```

```
        "ip":"172.16.9.192",
        "mask":"255.255.255.0",
        "gateWay":"172.16.9.254",
        "dns":[
          "172.16.0.201",
          "172.16.0.202"
        ]
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	详情
ethernets	必选	Array	无
scopeId	必选	int	scopeId，如果不知道，或者使用默认的，则填写-1
name	必选	string	名称，如
dhcp	必选	boolean	true
ip	必选	string	ip地址
mask	必选	string	子网掩码
gateWay	必选	string	网关
dns	(string)object	DNS	*

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	错误码对应的具体描述

备注

-

4.14.4.2、获取有线网络信息

简要描述：

- 获取有线网络信息的接口

请求URL：

- void nvGetEthernetInfo(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "ethernets":[
    {
      "scopeId":0,
      "name":"eth0",
      "dhcp":false,
      "ip":"192.168.1.127",
      "mask":"255.255.255.0",
      "gateWay":"17.16.20.1",
      "dns":[
        "211.20.1.67"
      ]
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
ethernets	object	有线网络详情
scopeId	int	scopeId，如果
name	string	名称，如
dhcp	boolean	true

参数名	类型	说明
ip	string	ip地址
mask	string	子网掩码
gateWay	string	网关
dns	(string)object	DNS

备注

-

4.14.5、WIFI-AP

4.14.5.1、设置wifi-AP信息

简要描述：

- 设置wifi-AP信息的接口

请求URL：

- void nvSetAPNetwork(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "aliasName":"AP",
  "suffix":"0016",
  "password":"12345678",
  "channelId":10
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
aliasName	必选	String	AP别名
suffix	必选	String	AP后缀
password	必选	String	AP密码,8<=密码长度<=32,数字加大小写字母，不能有特殊字符
channelId	必选	int	信道，范围为1-13,默认为6

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述

备注

-

4.14.5.2、获取WiFi-AP信息

简要描述：

- 获取WiFi-AP信息的接口

请求URL：

- void nvgGetAPNetwork(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":""
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "aliasName":"AP",
  "password":"12345678",
  "suffix":"20002272"
}
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
aliasName	String	AP别名
suffix	String	AP后缀
password	String	AP密码

备注

-

4.14.5.3、获取APN信息

简要描述：

- 获取APN信息

请求URL：

- void nvGetAPNInfoAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "APNs":[
    {
      "carrier":"Orange FR-MMS",
      "mcc":"208",
      "mnc":"01",
      "apn":"orange.acte",
      "user":"orange",
      "password":"orange",
```

```
      "server":"*",
      "port":"80",
      "mmsc":"http://mms.orange.fr",
      "mmsproxy":"192.168.010.200",
      "mmsport":"8080",
      "type":"mms",
      "isUserDefined":false,
      "isUsed":true
    },
    {
      "carrier":"中国移动彩信 (China Mobile)",
      "mcc":"460",
      "mnc":"00",
      "apn":"cmwap",
      "server":"*",
      "proxy":"10.0.0.172",
      "mmsc":"http://mmsc.monternet.com",
      "mmsproxy":"10.0.0.172",
      "mmsport":"80",
      "type":"mms",
      "protocol":"IPV4V6",
      "isUserDefined":false,
      "isUsed":true
    }
  ]
}
```


返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
APNs	Object	APN列表
carrier	string	APN名称，可为空，只用来显示apn列表中此apn的显示名字
mcc	string	由三位数组成。
mnc	string	由两位或三位组成。
apn	string	APN网络标识（接入点名称），是APN参数中的必选组成部分。此标识由运营商分配
user	string	用户名称
password	string	密码
server	string	服务器地址
proxy	string	代理服务器的地址
port	string	代理服务器的端口
mmsc	string	MMS中继服务器/多媒体消息业务中心，是彩信的交换服务器

参数名	类型	说明
mmsproxy	string	彩信代理服务器的地址
mmsport	string	彩信代理服务器的端口号
type	string	apn的接入点类型
protocol	string	支持的协议，不配置默认为IPV4
authtype	string	apn的认证协议
roamingProtocol	string	Apn漫游协议
isUserDefined	boolean	是否用户自定义
isUsed	boolean	是否正在使用

备注

-

4.14.5.4、设置APN信息

简要描述：

- 设置APN信息

请求URL：

- void nvSetAPNInfoAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17313J0820001562",
  "APNs":[
    {
      "carrier":"Orange FR-MMS",
      "mcc":"208",
      "mnc":"01",
      "apn":"orange.acte",
      "user":"orange",
      "password":"orange",
      "server":"*",
      "port":"80",
      "mmsc":"http://mms.orange.fr",
      "mmsproxy":"192.168.010.200",
      "mmsport":"8080",
      "type":"mms",
      "isUserDefined":false,
```

```
      "isUsed":true
    },
    {
      "carrier":"中国移动彩信 (China Mobile)",
      "mcc":"460",
      "mnc":"00",
      "apn":"cmwap",
      "server":"*",
      "proxy":"10.0.0.172",
      "mmsc":"http://mmsc.monternet.com",
      "mmsproxy":"10.0.0.172",
      "mmsport":"80",
      "type":"mms",
      "protocol":"IPV4V6",
      "isUserDefined":false,
      "isUsed":true
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
sn	string	产品唯一序列号	*
APNs	object	Apn信息列表	*
carrier	tring	APN名称，可为空，只用来显示apn列表中此apn的显示名字	*
mcc	string	由三位数组成。用于识别移动用户的所在国家	*
mnc	string	由两位或三位组成。用于识别移动用户的归属PLMN	*
apn	string	APN网络标识（接入点名称），是APN参数中的必选组成部分。此标识由运营商分配	*
user	string	用户名称	*
password	string	密码	*
server	string	服务器地址	*
proxy	string	代理服务器的地址	*
port	string	代理服务器的端口	*
mmsc	string	MMS中继服务器/多媒体消息业务中心，是彩信的交换服务器	*
mmsproxy	string	彩信代理服务器的地址	*
mmsport	string	彩信代理服务器的端口号	*
type	string	apn的接入点类型	*
protocol	string	支持的协议，不配置默认为IPV4	*
authtype	string	apn的认证协议	*
roamingProtocol	string	Apn漫游协议	*
isUserDefined	boolean	是否用户自定义	*
isUsed	boolean	是否正在使用	*

返回示例

```
{
  "basicConfigs":{
    "mobileData":true,
    "dataRoaming":true,
     "enable4G":true,
    "level":1
  },
   "advanced":{
    "networkType":"AUTO",
     "APN":{
       "providerName":"中国移动"
    }
  }
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	object	返回成功参数

备注

-

4.14.5.5、设置飞行模式开关状态

简要描述：

- 设置飞行模式开关状态

请求URL：

- void nvSetFlightModeAsync(const char *data, ViplexCallback callback)

请求方式：

- 请求参数示例

```
{
  "sn":"BZSA07194A0049999716",
  "enable":true
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
enable	必选	bool	设置飞行模式开关状态

返回示例

```
{
  "errorDiscription":"the description that describe the error in detail."
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	object	返回失败的data参数

备注

-

4.14.5.6、获取飞行模式开关状态

简要描述：

- 获取飞行模式开关状态

请求URL：

- void nvGetFlightModeAsync(const char *data, ViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "enable":true
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
enable	bool	飞行模式开关状态

备注

-

4.14.5.7、获取4G网络状态

简要描述：

- 获取4G网络状态

请求URL：

- nvGet4GNetworkStatusAsync(const string &data, ViplexCallback callback)

请求方式：

- 请求参数示例

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "mobileNetState":0,
  "netType":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
mobileNetState	int	4G网络状态,1正常2异常0未知（当前连接有线网或WIFI）
netType	int	网络类型,0：无网络，1：有线网，2：wifi，3：2G，4：3G,5:4G,6:未知

备注

-

4.14.5.8、获取AP开启状态

简要描述：

- 获取AP开启状态

请求URL：

- nvGetAPNetworkOpenStatusAsync(const string &data, ViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "ebable":true
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
enable	bool	AP的开启状态

备注

-

4.14.5.9、设置AP开启状态

简要描述：

- 设置AP开启状态

请求URL：

- nvSetAPNetworkOpenStatusAsync(const string &data, ViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "enable":true
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
"sccuss"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.15、配屏

4.15.1、配屏

4.15.1.1、获取配屏信息

简要描述：

- 获取配屏信息的接口

请求URL：

- void void nvGetScreenAttribute(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "screenAttributes":[
    {
      "id":1,
      "screenSource":0,
      "xCount":1,
      "yCount":1,
      "xOffset":0,
      "yOffset":0,
      "portNumber":2,
      "orders":[
        0,
        1
      ],
      "scanInfos":[
        {
          "width":128,
          "height":128,
          "x":0,
          "y":0,
```

```
            "xInPort":0,
            "yInPort":0,
            "portIndex":0,
            "connectIndex":0
          }
        ]
      }
    ]
}
```

返回参数说明

参数名	类型	说明
id	int	显示屏id
screenSource	int	显示屏id
id	int	显示屏id
id	int	配屏参数来源（0手机配屏,1LCT配屏）
xCount	int	X方向上接收卡个数
yCount	int	y方向上接收卡个数
xOffset	int	x方向上显示位置的偏移坐标
yOffset	int	y方向上显示位置的偏移坐标
portNumber	int	所使用的网口数量
orders	intArray	总共8种走线，数组内容顺序按照网口大小依次从小到大排列
scanInfos	object	接收卡信息集合
width	int	接收卡宽度
height	int	接收卡高度
x	int	接收卡所在显卡x坐标
y	int	接收卡所在显卡y坐标
xInPort	int	接收卡所在网口带载区域x坐标
yInPort	int	接收卡所在网口带载区域y坐标
portIndex	int	接收卡所在网口下标
connectIndex	int	接收卡所在网口下的连接位置
code	int	错误码：0获取成功65535请求超时

备注

-

4.15.1.2、设置配屏信息

简要描述：

- 设置配屏信息的接口

请求URL：

- void void nvSetScreenAttribute(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "screenAttributes":[
    {
      "id":0,
      "orders":[
        0,
        1
      ],
      "portNumber":1,
      "scanInfos":[
        {
          "connectIndex":0,
          "height":256,
          "portIndex":0,
          "width":256,
          "x":0,
          "xInPort":0,
          "y":0,
          "yInPort":0
        }
      ],
      "screenSource":1,
      "xCount":1,
      "xOffset":0,
      "yCount":1,
      "yOffset":0
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
id	必选	int	显示屏id

参数名	是否必选	类型	说明
screenSource	必选	int	显示屏id
id	必选	int	显示屏id
id	必选	int	配屏参数来源（0手机配屏,1LCT配屏）
xCount	必选	int	X方向上接收卡个数
yCount	必选	int	y方向上接收卡个数
xOffset	必选	int	x方向上显示位置的偏移坐标
yOffset	必选	int	y方向上显示位置的偏移坐标
portNumber	必选	int	所使用的网口数量
orders	必选	intArray	总共8种走线，数组内容顺序按照网口大小依次从小到大排列
scanInfos	必选	object	接收卡信息集合
width	必选	int	接收卡宽度
height	必选	int	接收卡高度
x	必选	int	接收卡所在显卡x坐标
y	必选	int	接收卡所在显卡y坐标
xInPort	必选	int	接收卡所在网口带载区域x坐标
yInPort	必选	int	接收卡所在网口带载区域y坐标
portIndex	必选	int	接收卡所在网口下标
connectIndex	必选	int	接收卡所在网口下的连接位置

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.15.2、接收卡文件配屏

4.15.2.1、接收卡文件配屏

简要描述：

- 接口的功能是终端可以通过接收卡文件配屏。该接口业务逻辑分为3步：1获取文件上传路径、2将本地文件上传到指定路径、3配屏

请求URL：

- void nvSetRecvCardFile(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "filePath":"C:/test",
  "fileName":"5036.rcfgx"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
filePath	必选	string	配屏文件的地址
fileName	必选	string	配屏文件的名称

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时

备注

-

4.16、icare配置

4.16.1、获取icare配置信息

简要描述：

- 获取icare配置信息的接口

请求URL：

- void nvGetIcareConfigInfo(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "requestInfo":{
    "language":"zh-cn"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
language	必选	string	zh-cn中文,en英文

返回示例

```
{
  "state":true,
  "serverNodes":[
    {
      "label":"测试",
      "node":"t.novaicare.com"
    },
    {
      "label":"美国节点",
      "node":"care-us.novaicare.com"
    },
    {
      "label":"中国节点",
      "node":"care.novaicare.com"
    }
  ],
  "url":"t.novaicare.com",
  "username":"XXX",
  "isOnline":true
}
```

返回参数说明

参数名	类型	说明
state	bool	开启状态,true为开启,false为关闭
serverNodes	object	显示屏id

参数名	类型	说明
label	string	服务器名称
node	string	服务器地址
url	string	当前连接的服务器地址
username	string	绑定的用户名
isOnline	string	是否在线,true为在线,false为不在线
code	int	错误码：0获取成功65535请求超时

备注

-

4.16.2、设置icare配置信息

简要描述：

- 设置icare配置信息的接口

请求URL：

- void nvSetIcareConfigInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "requestInfo":{
    "state":true,
    "url":"care.novaicare.com",
    "username":"XXX"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
state	必选	bool	开启状态,true为开启,false为关闭
url	必选	string	当前连接的服务器地址
username	必选	string	绑定的用户名

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功33账号不存在34未配屏65535请求超时

备注

-

4.17、终端cloud配置

4.17.1、绑定Cloud服务器

简要描述：

- 绑定Cloud服务器的接口

请求URL：

- void nvSetBindPlayer(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "playerInfo":{
    "baseUrl":"https://api-cn.vnnox.com",
    "data":{
      "playerList":[
        {

"identifier":"84f37cb1a12783654780f47da13b55db##dangjintao",
          "isUsed":true,
          "name":"dangjintao",
          "playerIdentifier":"BZSA07194A0049999716"
        }
      ],
      "token":"f2972f5a403d5e13d1c6103f91bf78a58055bfd5"
    },
    "isNewVnnox":true,
```

```
    "password":"123456",
    "username":"gmt"
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
baseUrl	必选	string	用于更换服务器时,终端访问对应服务器主机地址
password	必选	string	播放机认证的密码
username	必选	string	播放机认证的用户名
isNewVnnox	必选	bool	绑定的是否为新的vnnox
token	必选	string	播放机交互TOKEN(播放器与服务器交互,要将此TOKEN加入http协议header中)
identifier	必选	string	服务端播放机唯一标识(从VNNOX获取的Identifer
isUsed	必选	string	服务端播放机是否已绑定
name	必选	string	服务端播放机名称
playerIdentifier	必选	string	终端播放器唯一标识

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.17.2、获取播放器列表

简要描述：

- 获取播放器列表的接口

请求URL：

- void nvGetCloudPlayerList(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "host":"http://beta-rest.vnnox.com",
  "data":{
    "username":"qht1003077897",
    "password":"qht1003077897",
    "playerType":2
  }
}
```

参数：

参数名	是否必选	类型	说明
host	必选	string	服务器地址
username	必选	string	播放机认证的用户名
password	必选	string	播放机认证的密码
playerType	必选	int	1同步2异步

返回示例

```
{
  "status":[
    10000
  ],
  "data":{
    "token":"654ad5s64f65w46f5e456wa4f",
    "playerList":[
      {
        "name":"player_1",
        "identifier":"654ad-5s64f6-5w46-f5e45-6wa4f",
        "isUsed":true,
        "playerIdentifier":"33-22-44-22-44"
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
status	int	返回结果状态
token	string	播放机交互TOKEN(播放器与服务器交互)
name	string	服务端播放机名称
identifier	string	服务端播放机唯一标识

参数名	类型	说明
isUsed	bool	服务端播放机是否已绑定
token	string	终端播放器唯一标识
code	int	错误码:0获取成功65535请求超时

备注

-

4.17.3、获取播放器绑定信息

简要描述：

- 获取播放器绑定信息的接口

请求URL：

- void nvGetBindPlayer(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "isBind":true,
  "errorDescription":" ",
  "baseUrl":"http://test.vnnox.com",
  "password":"wxgaly199425",
  "username":"wxgaly",
  "status":[
    10000
  ],
  "data":{
    "token":"9e49437a8bc44ba790389524b74c76acdc7909e6",
```

```
    "playerList":[
      {
        "identifier":"103190fdfa52c37516aae26a9d00ed7c",
        "isUsed":true,
        "name":"异步播放机1",
        "playerIdentifier":"30:34:00:00:00:17"
      }
    ]
  }
}
```

返回参数说明

参数名	类型	说明
isBind	bool	终端播放器是否被绑定
errorDescription	string	播放器未被绑定的错误信息
baseUrl	string	用于更换服务器时，终端访问对应服务器主机地址
password	string	播放机认证的密码
username	string	播放机认证的用户名
status	intarray	播放机当前心跳状态(10000表示心跳正常)
token	string	播放机交互TOKEN(播放器与服务器交互)
name	string	服务端播放机名称
identifier	string	服务端播放机唯一标识
isUsed	bool	服务端播放机是否已绑定
token	string	终端播放器唯一标识
code	int	错误码:0获取成功65535请求超时

备注

-

4.18、监控

4.18.1、获取发送卡的监控信息

简要描述：

- 获取发送卡的监控信息

请求URL：

- void nvGetSendCardMonitorInfoAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "sendCardMonitorInfo":[
    {
      "isDVIConnected": true,
      "DVIRate": 60,
      "redundancePortInfo": [
        {
          "isRedundant": false,
          "devMappingList": [
            {
      "devIndex": 0,
              "devType": 1
            },
            {
      "devIndex": 0,
      "devType": 2
            },
            {
      "devIndex": 0,
      "devType":3
            }
          ],
          "deviceWorkState": 1
        },
            {
          "isRedundant": true,
          "devMappingList": [
            {
      "devIndex": 0,
      "devType": 1
            },
            {
```

```
      "devIndex": 0,
      "devType": 2
            },
            {
      "devIndex": 1,
      "devType": 3
            }
          ],
          "deviceWorkState": 0
        }
      ],
      "deviceMapList": [
        {
          "deviceIndex": 0,
          "deviceType": 1
        },
         {
          "deviceIndex": 0,
          "deviceType": 2
        }
      ],
      "deviceWorkState": 0
    }
  ]
}
```


返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时
sendCardMonitorInfo	object	发送卡的监控数据
deviceWorkState	int	工作状态，
deviceMapList	object	设备的位置索引信息
deviceIndex	int	设备所在位置编号，设备串联上的第几个
deviceType	int	设备类型
isDVIConnected	boolean	DVI源是否连接
DVIRate	int	DVI刷新率，仅在DVI连接时才有效
redundancePortInfo	object	网口信息的列表
deviceWorkState	int	工作状态，
deviceMapList	object	设备的位置索引信息
deviceIndex	int	设备所在位置编号，设备串联上的第几个
deviceType	int	设备类型
isRedundant	boolean	是否是冗余网口

备注

-

4.18.2、获取接收卡的数量及信息

简要描述：

- 获取接收卡的数量及信息

请求URL：

- void nvGetReceiverCountAndInfoAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "receiveCardRegionInfo":[
    {
      "senderIndex": 0,
      "portIndex": 0,
      "connectIndex": 0,
      "X": 0,
      "Y": 0,
      "XInPort": 0,
      "YInPort": 0,
      "width": 400,
      "height": 400,
      "rowIndexInScreen": 0,
      "colIndexInScreen": 0
    }
  ]
```

```
}
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时
receiveCardRegionInfo	object	所有接收卡的位置和大小信息
senderIndex	byte	所属发送卡序号,
portIndex	byte	所属输出口序号
connectIndex	int	位于所在输出口的第几个
X	int	X位置(在其所在显卡的X坐标)
Y	int	Y位置(在其所在显卡的Y坐标)
XInPort	int	X位置(在其所在Port带载区域的X坐标)
YInPort	int	Y位置(在其所在Port带载区域的Y坐标)
width	int	带载像素宽度
height	int	带载像素高度
rowIndexInScreen	int	位于屏体第几行
colIndexInScreen	int	位于屏体第几列

备注

-

4.18.3、根据接收卡索引获取监控信息

简要描述:

- 根据接收卡索引获取监控信息

请求URL:

- void nvGetMonitorInfoByReceiverIndexAsync(const char *data, ExportViplexCallback callBack);

请求方式:

- **请求参数示例**

```
{
  "info":{
    "receiveCardRegionInfo": [
      {
        "portIndex":0,
        "connectedIndex":0
      },
```

```
        {
          "portIndex":0,
          "connectedIndex":1
        }
      ]
    }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
receiveCardRegionInfo	必选	object	所要获取的接收卡的位置信息
portIndex	必选	byte	所属输出口序号，网口位置
connectedIndex	必选	int	位于所在输出口的第几个

返回示例

```
{
  "receiveCardRegionInfo":[
    {
      "portIndex":0,
      "connectedIndex":0
    },
    {
      "portIndex":0,
      "connectedIndex":1
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时
receiveCardRegionInfo	object	所要获取的接收卡的位置信息
portIndex	byte	所属输出口序号，网口位置
connectedIndex	int	位于所在输出口的第几个

备注

-

4.18.1、获取系统参数

4.18.1.1、获取硬盘存储信息

简要描述：

- 获取硬盘存储信息

请求URL：

- void void nvGetAvailableStorageDataAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "diskAvailableSize":3.67,
  "diskTotalSize":3.96,
  "storageInfos":[
    {
      "diskAvailableSize":3938967552,
      "diskCriticalSize":524288000,
      "diskReserveSize":0,
      "diskTotalSize":4248846336,
      "type":"LOCAL"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
diskTotalSize	float	外部存储空间总大小,单位G
diskAvailableSize	float	可用空间大小,单位G
storageInfos	object	存储空间信息

参数名	类型	说明
diskAvailableSize	float	磁盘可用空间大小,单位Byte
diskCriticalSize	float	磁盘空间不足预警阀值,当剩余存储空间小于该值,部分功能可能无法正常使用,单位Byte
diskReserveSize	float	磁盘内部预留空间,单位Byte
diskTotalSize	float	磁盘存储空间总大小,单位Byte
type	string	磁盘类型,LOCAL内部存储（SD卡）
code	int	错误码:0获取成功65535请求超时

备注

-

4.18.1.2、获取CPU使用率

简要描述：

- 获取CPU使用率的接口

请求URL：

- void nvGetCPUUsage(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":18
}
```

返回参数说明

参数名	类型	说明

参数名	类型	说明
value	float	CPU使用率
code	int	错误码:0获取成功65535请求超时

备注

-

4.18.1.3、获取CPU温度

简要描述：

- 获取CPU温度的接口

请求URL：

- void nvGetCPUTempAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":38.8
}
```

返回参数说明

参数名	类型	说明
value	float	CPU使用率
code	int	错误码:0获取成功65535请求超时

备注

-

4.18.1.4、获取可用内存

简要描述：

- 获取可用内存的接口

请求URL：

- void nvGetAvailableMemoryAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "value":18
}
```

返回参数说明

参数名	类型	说明
value	float	CPU使用率
code	int	错误码:0获取成功65535请求超时

备注

-

4.19、其他

4.19.1、获取是否夏令时

简要描述：

- 获取是否夏令时的接口

请求URL：

- void nvGetIsUseDayLightTimeAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{"isUseDayLightTimeData":{"timeZone":"Asia/Shanghai","gmt":"GMT 08:00"}
```

参数：

参数名	是否必选	类型	说明
isUseDayLightTimeData	必选	Object	获取是否夏令时信息
gmt	必选	string	时区gmt

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.1、对时服务器列表

4.19.1.1、获取时间服务器列表

简要描述：

- 1.获取对时服务器列表，由两部分组成，用户自定义及服务器获取到的列表，服务器获取到的列表会根据语言返回对应的地址，服务器列表地址如：http://download.vnnox.com/vnnox_api_host/timehost.json2.服务器获取到的列表不可修改

请求URL：

- void nvGetNetTimingListInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "lang":"en"
}
```

参数：

参数名	是否必选	类型	说明
lang	必选	string	语言，英文：en，中文：zh-cn，日文：jp，韩文：kr，西班牙语：es，法语：fr

返回示例

```
{
  "serverInfo":[
    {
      "islocal":true,
      "label":"aa",
      "node":"www.baidu.com"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明
serverInfo	array	返回服务器列表数组
islocal	bool	是否是用户自定义添加的数据，true：用户自定义，false：服务器获取
label	string	节点名称
node	string	节点地址

备注

-

4.19.1.2、添加对时服务器

简要描述：

- 添加对时服务器的接口

请求URL：

- void nvAddNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "label":"aa",
  "node":"www.baidu.com"
}
```

参数：

参数名	是否必选	类型	说明
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.1.3、删除对时服务器

简要描述：

- 删除对时服务器列表，如果存在两条node和label都完全一致的数据，则删除第一条

请求URL：

- void nvDeleteNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "label":"aa",
  "node":"www.baidu.com"
}
```

参数：

参数名	是否必选	类型	说明
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.1.4、修改对时服务器

简要描述：

- 修改对时服务器列表,修改服务器列表,如果存在两条node和label都完全一致的数据,则修改第一条

请求URL：

- void nvUpdateNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "oldServer":{
    "label":"zhangsan",
    "node":"123.123.123"
  },
  "newServer":{
    "label":"lisi",
    "node":"111.111.111"
  }
}
```

参数：

参数名	是否必选	类型	说明
oldServer	必选	Object	被修改的服务器信息
label	必选	string	节点名称
node	必选	string	节点地址
newServer	必选	Object	修改后的服务器信息
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.1.5、添加对时服务器列表

简要描述：

- 添加对时服务器列表的接口

请求URL：

- void nvAddNetTimingInfoList(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "serverInfo":[
    {
      "label":"aa",
      "node":"www.baidu.com"
    },
    {
      "label":"aa",
      "node":"www.baidu.com"
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
serverInfo	必选	array	服务器列表数组
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.2、数据迁移

4.19.2.1、数据迁移

简要描述：

- 数据迁移的接口

请求URL：

- void nvDataBaseMigration(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "templates":[
    {
      "data":[
        {
          "width":0.4,
          "height":0.4,
          "left":0.4,
          "top":0.4,
          "zindex":1,
          "index":0
        }
      ]
    }
  ],
  "size":[
    {
      "height":300,
      "width":300
    }
  ],
  "program":[
    {
      "ID":"15",
      "programName":"name",
      "PLAYLIST":"",
      "Width":"400",
      "Height":"400"
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
ID	必选	string	节目ID

参数名	是否必选	类型	说明
programName	必选	string	节目名称
PLAYLIST	必选	string	playlist.json文件中的内容
Width	必选	string	节目分辨率的宽
Height	必选	string	节目分辨率的高
templates	必选	object	自定义模版信息
width	必选	int	模板中窗口宽度
height	必选	int	模板中窗口高度
left	必选	float	模板中窗口左边占比
top	必选	float	模板中窗口上边占比
zindex	必选	int	Z序
index	必选	int	模板中窗口窗口的ID
width	必选	int	模板的宽度
templates	必选	int	自定义模版信息

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.3、下载缩略图

4.19.3.1、下载缩略图

简要描述：

- 下载缩略图的接口

请求URL：

- void nvDownLoadFiles(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "remotePathsAndFileNames":{
    "sdcard/nova/viplex_terminal/program/program_DANGJT-
P1/tu1/35384db4-d10a-4db8-a1fe-54479706376f.png":"1.png",
    "sdcard/nova/viplex_terminal/program/program_DANGJT-
P1/tu2/296653e7-814c-4fad-9198-c731c0770d8c.png":"2.png"
  },

"downLoadDirectoryPath":"/data/data/com.example.myhandy_android/app_f
lutter/log",
  "fileType":".png"
}
```


参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
remotePathsAndFileNames	必选	map	{"需要下载缩略图文件路径":"自定义文件名称"}
downLoadDirectoryPath	必选	string	指定路径，缩略图下载到这个路径下
fileType	必选	string	文件类型

返回示例


```
""
```


返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.4、获取指定路径下指定类型文件

4.19.4.1、获取指定路径下指定类型文件

简要描述：

- 获取指定路径下指定类型文件的接口

请求URL：

- void nvQueryFileByType(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "paths":[
    "C:/"
  ],
  "types":[
    ".txt"
  ],
  "pageIndex":0,
  "pageSize":100
}
```

参数：

参数名	是否必选	类型	说明
paths	必选	string	需要查询的路径
types	必选	string	需要查询的文件类型
pageIndex	必选	int	每页显示的个数(暂未用到,可设置任意值)
pageSize	必选	int	查询的最大值(暂未用到,可设置任意值)

返回示例

```
{
  "files":[
    "C:/1.txt"
  ]
}
```

返回参数说明

参数名	类型	说明
files	stringarray	查询到的指定后缀类型的文件集合
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.5、节目编辑

4.19.5.1、发布节目

简要描述：

- 发布节目的接口

请求URL：

- void nvStartTransferProgram(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
    "sn": "",
    "iconPath": "",
    "iconName": "",
    "sendProgramFilePaths": {
        "programPath": "",
        "mediasPath": {
            "C:/test/test.mp4": "test.mp4",
            "": ""
        }
    },
    "programName": "",
    "deviceIdentifier": "",
    "startPlayAfterTransferred": true,
    "insertPlay": true
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	唯一标识符
iconPath	必选	string	缩略图路径
iconName	必选	string	缩略图名称
programPath	必选	string	节目路径
mediasPath	必选	map	{媒体路径 ： 媒体名称}
programName	必选	string	节目名称
deviceIdentifier	必选	string	节目的唯一识别号
startPlayAfterTransferred	必选	bool	是否立即播放节目
insertPlay	是	bool	是否插播

备注
发布节目接口，会有很多回调。文件的实时进度对调和节目是否下发成功回调。

返回示例 文件实时进度返回示例

```
{
    "m_curBytes": 0,
    "m_totalBytes":10000
}
```

返回参数说明

参数名	类型	说明
m_curBytes	long long	当前媒体上传实时进度
m_totalBytes	long long	当前节目的所有媒体的总大小

文件发送成功回调示例

```
{"code":65362,"data":"all media update finish"}
```

返回参数说明

参数名	类型	说明
code	int	0发送成功2062文件传输终止2027本地网络连接断开2048 FTP超时
data	string	返回值错误码对应的说明

备注

-

4.19.5.2、生成节目

简要描述：

- 生成节目的接口

请求URL：

- void nvMakeProgram(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
```

```
  "programID":1,
  "outPutPath":"test",
  "mediasPath":[
    {
      "oldPath":"test",
      "newPath":"test"
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
programID	必选	int	节目id
outPutPath	必选	string	生成节目相关协议的路径
oldPath	必选	string	原文件路径(针对IOS获取不到原文件时使用)
newPath	必选	string	文件现路径(针对IOS获取不到原文件时，将源文件复制到某路径下)

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65338创建Json文件时失败
data	string	返回值错误码对应的说明

备注

-

4.19.5.3、配置默认系统模板

简要描述：

- 配置默认系统模板的接口

请求URL：

- void nvSetSystemTplInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**


```
{
  "data":[
    {
      "data":[
        {
          "width":1,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        }
      ]
    },
    {
      "data":[
        {
          "width":0.5,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        },
        {
          "width":0.5,
          "height":1,
          "left":0.5,
          "top":0,
          "zindex":1,
          "index":1
        }
      ]
    },
    {
      "data":[
        {
          "width":1,
          "height":0.5,
          "left":0,
          "top":0,
          "zindex":0,
```

```
          "index":0
        },
        {
          "width":1,
          "height":0.5,
          "left":0,
          "top":0.5,
          "zindex":1,
          "index":1
        }
      ]
    },
    {
      "data":[
        {
          "width":0.3333,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        },
        {
          "width":0.3333,
          "height":1,
          "left":0.3333,
          "top":0,
          "zindex":1,
          "index":1
        },
        {
          "width":0.3333,
          "height":1,
          "left":0.6666,
          "top":0,
          "zindex":2,
          "index":2
        }
      ]
    },
    {
      "data":[
        {
          "width":1,
```

```
          "height":0.3333,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        },
        {
          "width":1,
          "height":0.3333,
          "left":0,
          "top":0.3333,
          "zindex":1,
          "index":1
        },
        {
          "width":1,
          "height":0.3333,
          "left":0,
          "top":0.6666,
          "zindex":2,
          "index":2
        }
      ]
    },
    {
      "data":[
        {
          "width":0.5,
          "height":0.5,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        },
        {
          "width":0.5,
          "height":0.5,
          "left":0.5,
          "top":0,
          "zindex":1,
          "index":1
        },
        {
          "width":0.5,
```

```
            "height":0.5,
            "left":0,
            "top":0.5,
            "zindex":2,
            "index":2
          },
          {
            "width":0.5,
            "height":0.5,
            "left":0.5,
            "top":0.5,
            "zindex":3,
            "index":3
          }
        ]
      }
    ]
}
```

参数：

参数名	是否必选	类型	说明
width	必选	int	高度
height	必选	int	高度
left	必选	float	左边占比
top	必选	float	上边占比
zindex	必选	int	Z序
index	必选	int	小窗口的ID

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明

备注

-

4.19.5.4、添加用户自定义模板

简要描述：

- 添加用户自定义模板的接口

请求URL：

- void nvAddTpl(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "data":[
    {
      "data":[
        {
          "width":1,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        }
      ]
    }
  ],
  "size":{
    "width":400,
    "height":400
  }
}
```

参数：

参数名	是否必选	类型	说明
width	必选	int	宽度
height	必选	int	高度
left	必选	float	左边占比
top	必选	float	上边占比
zindex	必选	int	Z序
index	必选	int	小窗口的ID
width	必选	int	模板的宽度
height	必选	int	模板的高度

返回示例

```
{
  "tplID":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明
tplID	int	增家的模板ID

备注

-

4.19.5.5、编辑用户自定义模板

简要描述：

- 编辑用户自定义模板的接口

请求URL：

- void nvEditTpl(const char *data, ExportViplexCallback callBack)

请求方式：

- 请求参数示例

```
{
  "tplID":1,
  "data":[
    {
      "data":[
        {
          "width":1,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        }
      ]
```

```
    }
  ],
  "size":{
    "width":400,
    "height":400
  },
  "isSystemTpl":true
}
```

参数：

参数名	是否必选	类型	说明
tplID	必选	int	模板id
width	必选	int	宽度
height	必选	int	高度
left	必选	float	左边占比
top	必选	float	上边占比
zindex	必选	int	Z序
index	必选	int	小窗口的ID
width	必选	int	模板的宽度
height	必选	int	模板的高度

返回示例

```
{
  "tplID":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明

备注

-

4.19.5.6、删除用户自定义模板

简要描述：

- 删除用户自定义模板的接口

请求URL：

- void nvDeleteTpl(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "tplID":[
    1,
    2,
    3
  ]
}
```

参数：

参数名	是否必选	类型	说明
tplID	必选	intArray	模板id

返回示例

```
{
  "tplID":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明

备注

-

4.19.5.7、获取本地节目

简要描述：

- 获取本地节目的接口

请求URL：

- void nvGetProgram(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
""
```

参数：

参数名	是否必选	类型	说明
data	必选	string	data为任意字符串，“”或“任意string”

返回示例


```
{"name":"jiemu1","programID":1,"height":400,"width":400,"page":{"id":1,"name":"page1","widgetContainers":[{"audioGroup":"","backgroundColor":"#00000000","backgroundDrawable":"","contents":{"widgetGroups":[],"widgets":[]},"enable":true,"id":1,"itemsSource":"","layout":{"height":"100%","width":"100%","x":"0%","y":"0%"},"name":"widgetContainers1","pickCount":0,"pickPolicy":"ORDER","zOrder":0}],"widgetGroups":[],"widgets":[]}]}}
```


返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明
name	string	节目名称
programID	int	节目的ID
height	int	节目的宽
page	int	节目page信息

备注

-

4.19.5.8、删除本地节目

简要描述：

- 删除本地节目的接口

请求URL：

- void nvDeleteProgram(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "programID":[
    1,
    2,
    3
  ]
}
```

参数：

参数名	是否必选	类型	说明
programID	必选	intArray	节目的ID数组

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明

备注

-

4.19.5.9、取消节目发布

简要描述：

- 取消节目发布的接口

请求URL：

- void nvStopProgramTransfer(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
""
```

参数：

参数名	是否必选	类型	说明
data	必选	string	data为任意字符串，""或任意string

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	返回值错误码对应的说明

备注

-

4.19.5.10、获取模板列表

简要描述：

- 获取模板列表的接口

请求URL：

- void nvGetCustomerTplAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
"string"
```

参数：

参数名	是否必选	类型	说明
data	必选	string	参数data可以任意string,但是必须有该参数

返回示例

```
{
  "templates":[
    {
      "data":[
```

```
        {
          "width":1,
          "height":1,
          "left":0,
          "top":0,
          "zindex":0,
          "index":0
        }
      ]
    }
  ],
  "size":{
    "width":400,
    "height":400
  },
  "isSystemTpl":true
}
```

返回参数说明

参数名	类型	说明
width	int	宽度
height	int	高度
left	float	左边占比
top	float	上边占比
index	int	小窗口的ID
width	int	模板宽度
height	int	模板宽度
isSystemTpl	bool	是否用户自定义模板
code	int	错误码：0获取成功65535请求超时

备注

-

4.19.5.11、获取媒体文件MD5码

简要描述：

- 获取媒体文件MD5码的接口

请求URL：

- void nvGetFileMD5(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "filePath":"C:/test/media"
}
```

参数：

参数名	是否必选	类型	说明
filePath	必选	string	媒体所在路径

返回示例

```
"12d4sa654d564ddauioaj4163"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明

备注

-

4.19.5.12、创建节目

简要描述：

- 创建节目的接口

请求URL：

- void nvCreateProgram(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "name":"jiemu1",
  "width":400,
  "height":400,
  "tplID":1
```

```
}
```

参数：

参数名	是否必选	类型	说明
name	必选	String	节目名称
width	必选	int	宽度
height	必选	int	高度
tplID	必选	int	模板ID

返回示例

```
{
  "programID":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65340创建失败
data	string	返回值错误码对应的详细信息
programID	int	创建成功的节目id

备注

-

4.19.5.13、编辑节目

简要描述：

- 修改page的接口

请求URL：

- void nvSetPageParam(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "programID":1,
  "pageID":1,
  "pageInfo":{
```

```
    "name":"pageNew",
    "widgetContainers":[
      {
        "audioGroup":"",
        "backgroundColor":"#00000000",
        "backgroundDrawable":"",
        "contents":{
          "widgetGroups":[

          ],
          "widgets":[
            {
      "id":1,
      "mid":2,
      "name":"",
      "type":"PICTURE",
      "duration":10000,
      "repeatCount":1,
      "layout":{
                "x":"34.6%",
                "y":"24.0%",
                "width":"100%",
                "height":"100%"
      },
      "displayRatio":"ORIGINAL",
      "inAnimation":{
                "type":1,
                "duration":1000
      },
      "outAnimation":{
                "type":0,
                "duration":0
      },
      "dataSource":"26a0debe893d5c837270c60dde463913.png",
      "originalDataSource":"D:/nova/dog.png",
      "backgroundMusic":"/local/media/1.mp3",
      "backgroundColor":"#00ff0000",
      "backgroundDrawable":"/local/media/dog.png",
      "zOrder":100,
      "constraints":[
                {
                  "startTime":"2016-11-14T12:15:15Z 8:00",
                  "cron":[
```

```
                    ],
                    "endTime":"2017-02-10T12:15:15Z 8:00"
                }
        ],
        "fold":{
                "enable":true,
                "count":2,
                "orientation":"HORIZONTAL",
                "layoutItems":[
                  {
                    "y":0,
                    "height":200,
                    "x":0,
                    "width":200
                  },
                  {
                    "y":200,
                    "height":200,
                    "x":0,
                    "width":200
                  }
                ]
        },
        "metadata":""
              }
            ]
          },
          "enable":true,
          "id":1,
          "itemsSource":"",
          "layout":{
            "height":"100%",
            "width":"100%",
            "x":"0%",
            "y":"0%"
          },
          "name":"widgetContainers1",
          "pickCount":0,
          "pickPolicy":"ORDER",
          "zOrder":0
        }
      ]
    }
}
```

参数：

参数名	是否必选	类型	说明
programID	必选	int	节目id
pageID	必选	int	pageId(当前版本pageId为1)
pageInfo	必选	Object	page的详细信息

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时65285修改page失败
data	string	返回值错误码对应的详细信息

备注

-

4.19.6、节点服务器列表

4.19.6.1、获取节点服务器列表

简要描述：

- 获取硬盘存储信息

请求URL：

- void nvGetNodeServerList(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "lang":"en"
}
```

参数：

参数名	是否必选	类型	说明
lang	必选	string	语言,英文:en,中文:zh-cn,日文:jp,韩文:kr,西班牙语:es,法语:fr

返回示例

```
{
  "serverInfo":[
    {
      "islocal":true,
      "label":"aa",
      "node":"www.baidu.com"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
islocal	bool	外部存储空间总大小,单位G
diskAvailableSize	float	是否是用户自定义添加的数据,true:用户自定义,false:服务器获取
label	string	节点名称
node	string	节点地址
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.6.2、添加节点服务器

简要描述：

- 添加对时服务器的接口

请求URL：

- void nvAddNodeServerList(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "label":"aa",
  "node":"www.baidu.com"
}
```

参数：

参数名	是否必选	类型	说明
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.6.3、删除节点服务器

简要描述：

- 删除对时服务器列表，如果存在两条node和label都完全一致的数据，则删除第一条

请求URL：

- void nvDeleteNodeServerList(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "label":"aa",
  "node":"www.baidu.com"
}
```

参数：

参数名	是否必选	类型	说明
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.6.4、修改节点服务器

简要描述：

- 修改对时服务器列表,修改服务器列表,如果存在两条node和label都完全一致的数据,则修改第一条

请求URL：

- void nvUpdateNetTimingInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "oldServer":{
    "label":"zhangsan",
    "node":"123.123.123"
  },
  "newServer":{
    "label":"lisi",
    "node":"111.111.111"
  }
}
```

参数：

参数名	是否必选	类型	说明
oldServer	必选	Object	被修改的服务器信息
label	必选	string	节点名称
node	必选	string	节点地址
newServer	必选	Object	修改后的服务器信息

参数名	是否必选	类型	说明
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.6.5、添加服务器列表

简要描述：

- 添加服务器列表的接口

请求URL：

- void nvAddNetTimingInfoList(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "serverInfo":[
    {
      "label":"aa",
      "node":"www.baidu.com"
    },
    {
      "label":"aa",
      "node":"www.baidu.com"
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
serverInfo	必选	array	服务器列表数组
label	必选	string	节点名称
node	必选	string	节点地址

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.19.7、意见反馈

4.19.7.1、上传用户反馈信息

简要描述：

- 上传用户反馈信息的接口

请求URL：

- void nvUploadFeedbackInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"",
  "contact":"minyf@novastar.tech",
  "description":"我的意见反馈描述双卡双待的",
  "attacheds":[
    {
      "dir":"feedback/handy/20200301/abc.jpg"
    },
```

```
    {
      "dir":"feedback/handy/20200301/abc.zip"
    }
  ]
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一标识号
contact	非必选	string	联系方式
description	必选	string	意见反馈描述
attacheds	非必选	Object	附件
dir	非必选	string	上传到阿里云的文件dir目录

返回示例

```
{
  "logid":"225be01205",
  "status":0,
  "data":"ok"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	Object	返回值具体描述
logid	string	请求日志id
status	int	结果返回状态，0-成功，非0代表失败，具体参考code码含义
errmsg	string	错误时会返回错误描述

备注

-

4.19.7.2、取消上传意见反馈

简要描述：

- 取消上传意见反馈的接口

请求URL：

- void nvUploadFeedBackFileState(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "state":true
}
```

参数：

参数名	是否必选	类型	说明
state	非必选	bool	true表示继续下载，false表示停止下载，默认为false

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值具体描述

备注

-

4.19.8、绑定优化

4.19.8.1、是否公有云

简要描述：

- 是否公有云的接口

请求URL：

- void nvIsCommonCloud(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "baseUrl":"https://api-cn.vnnox.com"
```

```
}
```

参数：

参数名	是否必选	类型	说明
baseurl	必选	string	播放器绑定地址

返回示例

```
{
  "status":[
    10000
  ],
  "data":{
    "isCloud":1
  }
}
```

返回参数说明

参数名	类型	说明
status	intarray	返回服务端状态码:10000成功,10306url格式错误
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.8.2、获取播放器唯一识别号

简要描述：

- 获取播放器唯一识别号的接口

请求URL：

- void nvGetPlayerIdentifer(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "baseUrl":"https://test-rest-k8s.vnnox.com",
  "data":{
```

```
    "token":"f2972f5a403d5e13d1c6103f91bf78a58055bfd5",
    "number":1
  }
}
```

参数：

参数名	是否必选	类型	说明
baseurl	必选	string	播放器绑定地址
token	必选	string	播放机交互TOKEN(播放器与服务器交互)
number	必选	int	申请的唯一标识的数量（int最大值为1000，超过1000，按照1000进行计算

返回示例

```
{
  "status":[
    10000
  ],
  "data":{
    "identifierList":[
      "b518fd5c1ac9e6c68d20ed97c749b490"
    ]
  }
}
```

返回参数说明

参数名	类型	说明
status	intarray	返回服务端状态码:10000成功,10201http[header]中token为空10202http[header]中token错误
identifierList	stringarray	申请的唯一标识号
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.8.3、播放器名称校验

简要描述：

- 播放器名称校验的接口

请求URL：

- void nvIsExistPlayerName(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "baseUrl":"https://test-rest-k8s.vnnox.com",
  "data":{
    "token":"f2972f5a403d5e13d1c6103f91bf78a58055bfd5",
    "playerName":"dangjintao"
  }
}
```

参数：

参数名	是否必选	类型	说明
baseurl	必选	string	播放器绑定地址
token	必选	string	播放机交互TOKEN(播放器与服务器交互)
playerName	必选	string	播放器名称

返回示例

```
{
  "status":[
    10000
  ],
  "data":{
    "playerName":"dangjintao"
  }
}
```

返回参数说明

参数名	类型	说明
status	intarray	返回服务端状态码:10000成功,10201http[header]中token为空10202http[header]中token错误10209播放器名称格式错误10210播放器名称已存在
playerName	string	播放器名称
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.8.4、获取token值

简要描述：

- 播放器名称校验的接口

请求URL：

- void nvGetToken(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "baseUrl":"https://test-rest-k8s.vnnox.com",
  "data":{
    "username":"gmt",
    "password":"123456"
  }
}
```

参数：

参数名	是否必选	类型	说明
baseurl	必选	string	播放器绑定地址
username	必选	string	播放器用户名
password	必选	string	播放器密码

返回示例

```
{
  "status":[
    10000
  ],
  "data":{
    "token":"654ad5s64f65w46f5e456wa4f",
    "timeout":3600
  }
}
```

返回参数说明

参数名	类型	说明
status	intarray	返回服务端状态码:10000成功,10001用户名或密码错误

参数名	类型	说明
token	string	播放机交互TOKEN（播放器与服务器交互，要将此TOKEN加入http协议header中）
timeout	int	token的超时时间(单位s)
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.9、获取时区列表

4.19.9.1、获取时区列表

简要描述：

- 获取时区列表的接口

请求URL：

- void nvGetTimeZone(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
"anystring"
```

参数：

参数名	是否必选	类型	说明
data	必选	string	参数data必须有,但是可以为任意string

返回示例

```
{
  "Pacific/Midway":"中途岛 (UTC-11:00)",
  "Pacific/Honolulu":"檀香山 (UTC-10:00)",
  "America/Anchorage":"安克雷奇 (UTC-08:00)",
  "America/Los_Angeles":"洛杉矶/美国太平洋 (UTC-07:00)",
  "America/Tijuana":"提华纳/美国太平洋 (UTC-07:00)",
  "America/Phoenix":"凤凰城美国山区 (UTC-07:00)",
  "America/Chihuahua":"奇瓦瓦 (UTC-06:00)",
  "America/Denver":"丹佛/美国山区 (UTC-06:00)",
  "America/Costa_Rica":"哥斯达黎加/美国中部 (UTC-06:00)",
```

```
  "America/Regina":"里贾纳/美国中部 (UTC-06:00)",
  "America/Mexico_City":"墨西哥城/美国中部 (UTC-05:00)",
  "America/Chicago":"芝加哥/美国中部 (UTC-05:00)",
  "America/Bogota":"波哥大/哥伦比亚 (UTC-05:00)",
  "America/New_York":"纽约/美国东部 (UTC-04:00)",
  "America/Caracas":"加拉加斯/委内瑞拉 (UTC-04:30)",
  "America/Barbados":"巴巴多斯/大西洋 (UTC-04:00)",
  "America/Manaus":"马瑙斯/亚马逊 (UTC-04:00)",
  "America/Santiago":"圣地亚哥 (UTC-03:00)",
  "America/Sao_Paulo":"圣保罗 (UTC-03:00)",
  "America/St_Johns":"圣约翰/纽芬兰 (UTC-04:00)",
  "America/Argentina/Buenos_Aires":"布宜诺斯艾利斯 (UTC-
03:00)",
  "America/Montevideo":"蒙得维的亚/乌拉圭 (UTC-03:00)",
  "America/Godthab":"戈特霍布 (UTC-02:00)",
  "Atlantic/South_Georgia":"南乔治亚 (UTC-02:00)",
  "Atlantic/Cape_Verde":"佛得角 (UTC-01:00)",
  "Atlantic/Azores":"亚述尔群岛 (UTC 00:00)",
  "Africa/Casablanca":"卡萨布兰卡 (UTC 00:00)",
  "Europe/London":"伦敦/格林尼治 (UTC 00:00)",
  "Africa/Brazzaville":"布拉扎维/西部非洲 (UTC 01:00)",
  "Africa/Windhoek":"温得和克 (UTC 01:00)",
  "Europe/Amsterdam":"阿姆斯特丹/中欧 (UTC 02:00)",
  "Europe/Belgrade":"贝尔格莱德/中欧 (UTC 02:00)",
  "Europe/Brussels":"布鲁塞尔/中欧 (UTC 02:00)",
  "Europe/Sarajevo":"萨拉热窝/中欧 (UTC 02:00)",
  "Africa/Harare":"哈拉雷/中部非洲 (UTC 02:00)",
  "Africa/Cairo":"开罗/东欧 (UTC 02:00)",
  "Asia/Beirut":"贝鲁特/东欧 (UTC 03:00)",
  "Europe/Athens":"雅典/东欧 (UTC 03:00)",
  "Europe/Helsinki":"赫尔辛基/东欧 (UTC 03:00)",
  "Asia/Jerusalem":"耶路撒冷/以色列 (UTC 03:00)",
  "Asia/Amman":"安曼/东欧 (UTC 03:00)",
  "Europe/Minsk":"明斯克 (UTC 03:00)",
  "Asia/Baghdad":"巴格达 (UTC 03:00)",
  "Europe/Moscow":"莫斯科 (UTC 03:00)",
  "Asia/Kuwait":"科威特 (UTC 03:00)",
  "Africa/Nairobi":"内罗毕/东部非洲 (UTC 03:00)",
  "Asia/Baku":"巴库 (UTC 05:00)",
  "Asia/Tbilisi":"第比利斯 (UTC 04:00)",
  "Asia/Yerevan":"埃里温 (UTC 04:00)",
  "Asia/Dubai":"迪拜 (UTC 04:00)",
  "Asia/Tehran":"德黑兰/伊朗 (UTC 04:30)",
  "Asia/Kabul":"喀布尔/阿富汗 (UTC 04:30)",
```

```
  "Asia/Karachi":"卡拉奇 (UTC 05:00)",
  "Asia/Oral":"乌拉尔 (UTC 05:00)",
  "Asia/Yekaterinburg":"叶卡捷林堡 (UTC 05:00)",
  "Asia/Calcutta":"加尔各答 (UTC 05:30)",
  "Asia/Colombo":"科伦坡 (UTC 05:30)",
  "Asia/Katmandu":"加德满都/尼泊尔 (UTC 05:45)",
  "Asia/Almaty":"阿拉木图 (UTC 06:00)",
  "Asia/Rangoon":"仰光/缅甸 (UTC 06:30)",
  "Asia/Krasnoyarsk":"克拉斯诺亚尔斯克 (UTC 07:00)",
  "Asia/Bangkok":"曼谷 (UTC 07:00)",
  "Asia/Shanghai":"北京/中国 (UTC 08:00)",
  "Asia/Hong_Kong":"香港/中国 (UTC 08:00)",
  "Asia/Irkutsk":"伊尔库茨克 (UTC 08:00)",
  "Asia/Kuala_Lumpur":"吉隆坡 (UTC 08:00)",
  "Australia/Perth":"佩思 (UTC 08:00)",
  "Asia/Taipei":"台北时间 (UTC 08:00)",
  "Asia/Seoul":"首尔 (UTC 09:00)",
  "Asia/Tokyo":"东京/日本 (UTC 09:00)",
  "Asia/Yakutsk":"雅库茨克 (UTC 09:00)",
  "Australia/Adelaide":"阿德莱德 (UTC 09:30)",
  "Australia/Darwin":"达尔文 (UTC 09:30)",
  "Australia/Brisbane":"布里斯班 (UTC 10:00)",
  "Australia/Hobart":"霍巴特 (UTC 10:00)",
  "Australia/Sydney":"悉尼 (UTC 10:00)",
  "Asia/Vladivostok":"符拉迪沃斯托克/海参崴 (UTC 10:00)",
  "Pacific/Guam":"关岛 (UTC 10:00)",
  "Asia/Magadan":"马加丹 (UTC 10:00)",
  "Pacific/Majuro":"马朱罗 (UTC 12:00)",
  "Pacific/Auckland":"奥克兰 (UTC 12:00)",
  "Pacific/Fiji":"斐济 (UTC 12:00)",
  "Pacific/Tongatapu":"东加塔布 (UTC 13:00)"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.19.10、校时

4.19.10.1、获取是否夏令时

简要描述：

- 获取是否夏令时的接口

请求URL：

- void nvGetIsUseDayLightTimeAsync(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{"isUseDayLightTimeData":{"timeZone":"Asia/Shanghai","gmt":"GMT 08:00"}
```

参数：

参数名	是否必选	类型	说明
isUseDayLightTimeData	必选	Object	获取是否夏令时信息
gmt	必选	string	时区gmt

返回示例

```
"success"
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
data	string	返回值错误码对应的说明

备注

-

4.20、屏体管理

4.20.1、获取终端截图

简要描述：

- 获取终端截图

请求URL：

- void downLoadScreenshot(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272",
  "width":256,
  "height":256,
  "type":"PNG"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
width	必选	int	图片宽度
height	必选	int	图片高度
type	必选	enum	图片格式，PNG

返回示例

```
{
  "code":0,
  "path":"/cache/screenShots/screenshot.png"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
path	string	截图路径

备注

-

4.21、多功能卡电源管理

4.21.1、定时开关电源

4.21.1.1、获取定时开关电源状态

简要描述：

- 获取定时开关电源状态的接口

请求URL：

- `void nvGetTimingPowerSwitchStatus(const char *data, ExportViplexCallback callBack);`

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "data":[
    {
      "commands":[
        {
          "conditions":[
            {
      "action":0,
      "cron":[
                "0 15 10 ? * *",
                "0 0 12 * * ?"
      ],
      "enable":false,
      "endTime":"2018-06-07",
              "flag":"abc",
      "powerIndex":0,
      "startTime":"2018-06-06",
      "type":"power"
            },
            {
```

```
        "action":1,
        "cron":[
                  "0 15 10 ? * *",
                  "0 0 12 * * ?"
        ],
        "enable":false,
        "endTime":"2018-06-07",
        "flag":"abc",
        "powerIndex":1,
        "startTime":"2018-06-06",
        "type":"power"
              }
           ]
         }
        ],
        "connectIndex":1,
        "enable":false,
        "portIndex":0
     }
  ],
  "source":{
    "platform":1,
    "type":1
  },
  "type":"FUNCTIONPOWER"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明
type	string	FUNCTIONPOWER(固定值)
source	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
source	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
platform	int	表示任务发布来源,1:手机,2;CS,3:平板,4:VNNOX,5;Care,6:LCT
data	ObjectArray	按照条件执行的多功能卡任务集合
enable	bool	按条件执行的使能开关.如果为true,则执行conditions中的定时任务,如果为false,则不执行
portIndex	int	多功能卡的网口地址
powerIndex	int	多功能卡网口连接的设备编号
commands	ObjectArray	按照条件执行的任务集合,支持多任务
conditions	ObjectArray	按照条件执行的任务集合,支持多任务
cron	StringArray	每个控制方案使用cron表达式数组表示,当有多个cron表达式时,cron

参数名	类型	说明
		表达式之间使用或的关系
action	int	具体电源控制:0开,1关
type	string	开关的类型,以字符串的形式表征
startTime	string	有效时间的开始日期
endTime	string	有效时间的结束日期
enable	boolean	按条件执行的使能开关.如果为true,那么此条cron表达式生效,如果为false,则不生效
powerIndex	int	多功能卡电源的下标(目前支持0-7)
flag	string	此字段属于VNNOX,手机,CS专用字段

备注

-

4.21.1.2、设置定时电源开关状态

简要描述：

- 设置定时电源开关状态的接口

请求URL：

- void nvSetTimingPowerSwitchStatus(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"123456",
  "taskInfo":{
    "type":"FUNCTIONPOWER",
    "source":{
      "type":1,
      "platform":1
    },
    "data":[
      {
        "commands":[
          {
            "conditions":[
      {
                "cron":[
                  "0 15 10 ? * *",
```

```
                    "0 0 12 * * ?"
                ],
                "action":0,
                "type":"power",
                "startTime":"2018-06-06",
                "endTime":"2018-06-07",
                "enable":false,
                "powerIndex":0,
                "flag":"abc"
        },
        {
                "cron":[
                    "0 15 10 ? * *",
                    "0 0 12 * * ?"
                ],
                "action":1,
                "type":"power",
                "startTime":"2018-06-06",
                "endTime":"2018-06-07",
                "enable":false,
                "powerIndex":1,
                "flag":"abc"
        }
            ]
          }
        ],
        "enable":false,
        "portIndex":0,
        "connectIndex":1
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
type	必选	string	FUNCTIONPOWER(固定值)
source	必选	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
source	必选	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
platform	必选	int	表示任务发布来源,1:手机,2;CS,3:平板,4:VNNOX,5;Care,6:LCT
data	必选	ObjectArray	按照条件执行的多功能卡任务集合

参数名	是否必选	类型	说明
enable	必选	bool	按条件执行的使能开关.如果为true,则执行conditions中的定时任务,如果为false,则不执行
portIndex	必选	int	多功能卡的网口地址
powerIndex	必选	int	多功能卡网口连接的设备编号
commands	必选	ObjectArray	按照条件执行的任务集合,支持多任务
conditions	必选	ObjectArray	按照条件执行的任务集合,支持多任务
cron	必选	StringArray	每个控制方案使用cron表达式数组表示,当有多个cron表达式时,cron表达式之间使用或的关系
action	必选	int	具体电源控制:0开,1关
type	必选	string	开关的类型,以字符串的形式表征
startTime	必选	string	有效时间的开始日期
endTime	必选	string	有效时间的结束日期
enable	必选	boolean	按条件执行的使能开关.如果为true,那么此条cron表达式生效,如果为false,则不生效
powerIndex	必选	int	多功能卡电源的下标(目前支持0-7)
flag	必选	string	此字段属于VNNOX,手机,CS专用字段

返回示例

```
{
  "data":"12d4sa654d564ddauioaj4163"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明

备注

-

4.21.2、电源实时状态获取

4.21.2.1、获取电源实时状态获取

简要描述：

- 获取电源实时状态的接口

请求URL：

- void nvGetRealtimePowerSwitchStatus(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "source":{
    "type":1,
    "platform":1
  },
  "current_status_info":[
    {
      "portIndex":0,
      "connectIndex":1,
      "updatePowerIndexStates":[
        {
          "type":"屏体电源",
          "powerIndex:":0,
          "status":0
        },
        {
          "type":"屏体电源",
          "powerIndex:":1,
          "status":0
        },
        {
          "type":"屏体电源",
          "powerIndex:":2,
          "status":0
        },
        {
          "type":"屏体电源",
```

```
            "powerIndex:":3,
            "status":0
          },
          {
            "type":"屏体电源",
            "powerIndex:":4,
            "status":0
          },
          {
            "type":"屏体电源",
            "powerIndex:":5,
            "status":0
          },
          {
            "type":"屏体电源",
            "powerIndex:":6,
            "status":0
          },
          {
            "type":"屏体电源",
            "powerIndex:":7,
            "status":0
          }
        ]
      }
    ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明
source	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
type	Object	表示任务的发布来源.如:VNNOX,CS,LCT等
platform	int	表示任务发布来源,1:手机,2;CS,3:平板,4:VNNOX,5;Care,6:LCT
current_status_info	ObjectArray	多功能卡的各路电源状态
portIndex	int	多功能卡的网口地址
connectIndex	int	多多功能卡网口连接的设备编号
updatePowerIndexStates	ObjectArray	每张多功能卡中各路电源的状态
type	string	开关的类型,以字符串的形式表征
powerIndex	int	多功能卡电源的下标(目前支持0-7)
status	int	0:开,1:关

备注

-

4.21.3、手动开关电源

4.21.3.1、获取手动电源开关状态

简要描述：

- 获取手动电源开关状态的接口

请求URL：

- void nvGetManualPowerSwitchStatus(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "data":[
    {
      "conditions":[
        {
          "action":1,
          "powerIndex":0,
          "type":"屏体电源"
        },
        {
          "action":1,
          "powerIndex":1,
          "type":"风扇电源"
        }
```

```
      ],
      "connectIndex":1,
      "portIndex":0
    }
  ],
  "source":{
    "platform":1,
    "type":1
  },
  "type":"FUNCTIONPOWER"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明
data	ObjectArray	按照条件执行的多功能卡任务集合
conditions	ObjectArray	按照条件执行的任务集合,支持多任务
action	int	具体电源控制。0：开、1：关
powerIndex	int	开关的类型，以字符串的形式表征
type	string	多功能卡电源的下标,(目前支持0-7)
connectIndex	int	多功能卡网口连接的设备编号
portIndex	int	多功能卡的网口地址
source	Object	表示任务的发布来源。如：VNNOX、CS、LCT等
platform	int	表示任务发布来源,1：手机,2：CS,3：平板,4：VNNOX,5：Care、6：LCT
type	int	表示发布任务的平台,1：nova自己的平台0：第三方平台
type	string	FUNCTIONPOWER(固定值)

备注

-

4.21.3.2、设置手动电源开关状态

简要描述：

- 设置手动电源开关状态的接口

请求URL：

- void nvSetManualPowerSwitchStatus(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"123456",
  "data":{
    "type":"FUNCTIONPOWER",
    "source":{
      "type":1,
      "platform":1
    },
    "data":[
      {
        "conditions":[
          {
            "action":1,
            "type":"屏体电源",
            "powerIndex":0
          },
          {
            "action":1,
            "type":"风扇电源",
            "powerIndex":1
          }
        ],
        "portIndex":0,
        "connectIndex":1
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
data	必选	ObjectArray	按照条件执行的多功能卡任务集合
conditions	必选	ObjectArray	按照条件执行的任务集合,支持多任务
action	必选	int	具体电源控制。0：开、1：关
powerIndex	必选	int	开关的类型，以字符串的形式表征
type	必选	string	多功能卡电源的下标,(目前支持0-7)
connectIndex	必选	int	多功能卡网口连接的设备编号
portIndex	必选	int	多功能卡的网口地址
source	必选	Object	表示任务的发布来源。如：VNNOX、CS、LCT等

参数名	是否必选	类型	说明
platform	必选	int	表示任务发布来源,1：手机,2：CS,3：平板,4：VNNOX,5：Care、6：LCT
type	必选	int	表示发布任务的平台,1：nova自己的平台0：第三方平台
type	必选	string	FUNCTIONPOWER(固定值)

返回示例

```
{
  "data":"12d4sa654d564ddauioaj4163"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明

备注

-

4.22、VPN连接管理

4.22.1、VPN连接管理

4.22.1.1、获取VPN连接信息

简要描述：

- 获取VPN连接信息的接口

请求URL：

- void nvGetVPNConnectInfo(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例


```
{
  "connectInfo":{
    "connectStatus":2,
    "failReasonCode":0
  },
  "guid":"decf313f-1014-44ad-a059-28ed58fce170",
  "networkInfo":{
    "ipAddress":"10.0.0.2"
  },
  "source":{
    "platform":2,
    "type":1
  },
  "taskAction":"VPN_CONNECT",
  "timestamp":"2020-07-13 T02:27:47 Z 00:00",
  "vpnInfo":{
    "address":"vpn.vnnox.com",
    "isReconnect":false,
    "name":"456",
    "password":"bOwVRSe814WnE6hCkItCEgLHFZpNndYx",
    "protocolType":1,
    "redialInterval":180,
    "redialNumber":3,
    "sessionTimeout":7200,
    "sharedSecretKey":"novastar.tech",
    "username":"vnnox"
  },
  "orderId":-1
}
```


返回参数说明

参数名	类型	说明
connectInfo	object	连接上报信息
connectStatus	int	连接接状态：1、链接中2、链接成功3、断开
failReasonCode	int	失败原因code，0：正常，ERR_INTERRUPTED：超时后主动中断，ERR_ALREADY_EXISTED：命令重复，其他错误码见错误码说明
guid	string	最后一条命令的唯一标识
timestamp	string	时间戳
taskAction	string	操作：VPN_CLOSE，VPN_CONNECT

参数名	类型	说明
networkInfo	object	网络信息
ipAddress	string	Ip地址
source	object	上报资源信息
type	int	类型信息：1：nova，2：第三方平台
platform	int	1：vnnox，2：cs软件
vpnInfo	object	vpn信息
address	string	vpn地址
isReconnect	boolean	是否断线重连
name	string	vpn名称
password	string	密码
protocolType	string	vpn协议类型
redialInterval	int	重拨间隔
redialNumber	int	重播次数
sessionTimeout	int	会话时长
sharedSecretKey	string	共享秘钥
username	string	用户名

备注

-

4.22.1.2、设置VPN连接信息

简要描述：

- 设置VPN连接信息的接口

请求URL：

- void nvSetVPNConnectInfo(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "taskInfo":{
    "guid":"123456",
    "taskAction":"VPN_CLOSE",
    "source":{
      "type":1,
      "platform":2
```

```
    },
    "vpnInfo":{
      "name":"123456",
      "address":"vpn.vnnox.com",
      "protocolType":1,
      "username":"vnnox",
      "password":"123456",
      "sharedSecretKey":"novastar.tech",
      "redialNumber":3,
      "redialInterval":180,
      "sessionTimeout":7200,
      "isReconnect":false
    }
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	object	设置VPN的详细信息
guid	必选	string	命令的唯一标识
taskAction	必选	string	操作：VPN_CLOSE，VPN_CONNECT
source	非必选	object	上报资源信息
type	非必选	int	类型信息：1：nova，0：第三方平台
platform	非必选	int	平台类型：1：移动终端发来的（如手机），2：表示传统电脑，3：表示平板，4：表示web端发来的
vpnInfo	必选	object	vpn信息
name	必选	string	vpn名称
address	必选	string	vpn地址
protocolType	必选	string	vpn协议类型
username	必选	string	用户名
password	必选	string	密码
sharedSecretKey	必选	string	共享秘钥
redialNumber	必选	int	重拨次数
redialInterval	必选	int	重播间隔
sessionTimeout	必选	int	会话时长
isReconnect	必选	boolean	是否断线重连

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
data	string	成功返回md5码,失败返回错误码对应的说明

备注

-

4.23、播放日志

4.23.1、播放日志路径获取

4.23.1.1、播放日志路径获取

简要描述：

- 播放日志路径获取的接口

请求URL：

- void nvGetPlaylogPath(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "logList":[
    "2020-07-13.json"
  ],
  "url":"/sdcard/nova/viplex_terminal/play_log"
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功65535请求超时
logList	stringArray	播放日志文件名集合
url	string	播放日志目录路径

备注

-

4.24、字体

4.24.1、获取终端支持的字体

简要描述：

- 获取终端支持的字体的接口

请求URL：

- void nvGetTerminalFont(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"123456"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "supportFonts":[
    {
      "name":"Arial",
      "src":"system"
    },
    {
```

```
      "name":"Calibri",
      "src":"system"
    },
    {
      "name":"Wingdings2",
      "src":"system"
    },
    {
      "name":"SimSun",
      "src":"system"
    },
    {
      "name":"KaiTi",
      "src":"system"
    },
    {
      "name":"Microsoft YaHei",
      "src":"system"
    },
    {
      "name":"Agency FB",
      "src":"custom"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
supportFonts	ArrayObject	终端支持的字体
name	String	字体名称
src	String	字体来源，目前包括system、custom，以后也可支持来自某用户

备注

-

4.24.2、删除字体

简要描述：

- 删除字体的接口

请求URL：

- void nvDeleteFont(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "taskInfo":{
    "fonts":[
      {
        "name":"Arail"
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
taskInfo	必选	Object	字体信息
fonts	必选	object	要删除的字体列表
name	必选	string	字体名称

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功

备注

-

4.24.3、更新字体

简要描述：

- 更新字体的接口

请求URL：

- void nvUpdateFont(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZWA17422J1X20000093",
  "localFontPath":"C:\\Windows\\Fonts",
  "taskInfo":{
    "fonts":[
      {
        "name":"Agency FB",
        "style":[
          "Bold",
          "Italic",
          "Normal",
          "Underline",
          "Strikeout"
        ],
        "file":[
          "AGENCYB.TTF",
          "AGENCYB.TTF",
          "AGENCYB.TTF",
          "AGENCYB.TTF",
          "AGENCYB.TTF"
        ]
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
localFontPath	必选	string	字体存放目录
taskInfo	必选	Object	字体信息
fonts	必选	objectArray	要添加的字体列表
name	必选	string	字体名称
style	必选	stringArray	字体属性
file	必选	stringArray	字体文件名

返回示例

```
""
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功

备注

-

4.25、射频管理

4.25.1、获取Lora信息

简要描述：

- 获取Lora信息

请求URL：

- void getLoraInfo(const char *data, ExportViplexCallback callBack)

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA17332J0A20002272"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "exist":true,
  "frequency":433,
  "power":1,
  "version":1,
  "channel":23,
  "baudRate":9600,
```

```
  "transSpeed":2.4,
  "workState":0,
  "modelNum":1
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0
exist	boolean	模块是否存在
frequency	float	频率(单位：mhz)
power	float	功率(单位：w)
version	int	版本
channel	float	信道
baudRate	int	波特率
transSpeed	float	空中速率(k/bps)
workState	String	工作模式
modelNum	int	模块型号

备注

-

4.26、采集器接收器

4.26.1、添加采集接收器

简要描述：

- 添加采集接收器

请求URL：

- void nvAddCollectorAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{
  "sn":"BZSA07194A0049999716",
  "info":{
    "pickers":[
      {
```

```
        "type":"CPU_USAGE",
         "pickType":"PERIOD",
        "period":1000,
        "receivers":[
          {
            "outputDestination":"TCP_NET_WORK",
            "persistent":false
          }
        ]
      }
    ]
  }
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
info	必选	object	采集器数据实体
pickers	必选	object	采集器数据列表
type	必选	enum	采集器类型
pickType	必选	enum	PERIOD
period	必选	long	采集周期
receivers	必选	object	关联的接收器
outputDestination	必选	enum	采集输出路径包括文件、网络和控制台
persistent	必选	bool	表示是否持久

返回示例

```
"sccuss"
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.26.2、删除采集接收器

简要描述：

- 删除采集接收器

请求URL：

- void nvDeleteCollectorAsync(const char *data, ExportViplexCallback callBack);

请求方式：

- **请求参数示例**

```
{"sn":"BZSA07194A0049999716""info":{"type":["CPU_USAGE"]}}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号
info	必选	object	删除的采集器类型
type	必选	list	删除的采集器类型列表

返回示例

```
"sccuss"
```

返回参数说明

参数名	类型	说明
code	int	错误码:0获取成功65535请求超时

备注

-

4.26.3、获取采集接收器配置

简要描述：

- 获取采集接收器配置的接口

请求URL：

- void nvGetPickerReceiver(const char *data, ExportViplexCallback callback)

请求方式：

- **请求参数示例**

```
{
```

```
  "sn":"BZWA17422J1X20000093"
}
```

参数：

参数名	是否必选	类型	说明
sn	必选	string	产品唯一序列号

返回示例

```
{
  "pickers":[
    {
      "period":1000,
      "pickType":"PERIOD",
      "receivers":[
        {
          "outputDestination":"TCP_NET_WORK",
          "persistent":false
        }
      ],
      "type":"CPU_USAGE"
    }
  ]
}
```

返回参数说明

参数名	类型	说明
code	int	错误码：0获取成功
pickers	objectArray	采集器列表
period	long	采集周期，单位为毫秒，当为周期性采集时才有效
pickType	string	PERIOD：表示的是周期性采集，AUTO：表示的是自动的，根据这种类型的具体实现。
receivers	object	关联的接收器
outputDestination	string	FILE：表示采集到文件，TCP_NET_WORK：表示的是自动的，根据这种类型的具体实现，CONSOLE：控制台，如终端的控制台，一般用户调试
persistent	bool	true：表示持久的，即当断电，上电依然有效，false：表示只添加到内存的
type	string	CPU_USAGE：cpu使用率，CPU_TERMPATURE：cpu温度，MEMORY_USAGE：内存使用率，WIFI_STATE_CHANGED：wifi状态变化

备注

-

5、二次开发范例

5.1、范例运行结果

运行结果：

```
ViplexCore Demo nvSerchTerminalAsync begin...
ViplexCore Demo code:0
ViplexCore Demo
data:{"aliasName":"VPlayer_W9A7KY8J","configInfo":{"config_info":{"Br
ightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},
"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXED
POINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":f
alse,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBr
ightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Mem
ory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":fal
se,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModule
Info":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":
false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"L
og4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":fa
lse,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":
false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false
,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAY
POWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":
false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCar
d":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution
":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport
":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":fa
lse,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"I
sSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"Sensor
Board":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false
},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINF
O":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":fal
se,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgra
de":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX"
:false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":fals
e,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"
IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunning
LogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":
"","mac":"","mainVersion":"","model":"","productName":"","registerAdd
ress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTim
e":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.129","ke
y":"novaStar","logined":true,"loginedUsernames":[],"password":"","pla
```

tform":"","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"W9A7KY8J","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"width":400}
ViplexCore Demo code:0
ViplexCore Demo
data:{"aliasName":"VPlayer_W9ABGQ3H","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.139","key":"novaStar","logined":false,"loginedUsernames":[],"password":"","platform":"","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0

,"productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"W9ABGQ3H","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-10007615","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.177","key":"novaStar","logined":false,"loginedUsernames":[""],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA79353N1510007615","syssetFtpPort":16604,"syssetTcpPort":16605,

"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo
data:{"aliasName":"VPlayer_W9AB9VRN","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.73","key":"novaStar","logined":true,"loginedUsernames":[],"password":"","platform":"","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"W9AB9VRN","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"width":400}
ViplexCore Demo code:0

ViplexCore Demo data:{"aliasName":"Taurus-20004774","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.172","key":"novaStar","logined":true,"loginedUsernames":["admin"],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA17466W0X20004774","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"VPlayer_W9A61VK7","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},

"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXED
POINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":f
alse,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBr
ightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Mem
ory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":fal
se,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModule
Info":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":
false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"L
og4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":fa
lse,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":
false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false
,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAY
POWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":
false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCar
d":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution
":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport
":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":fa
lse,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"I
sSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"Sensor
Board":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false
},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINF
O":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":fal
se,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgra
de":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX"
:false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":fals
e,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"
IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunning
LogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":
"","mac":"","mainVersion":"","model":"","productName":"","registerAdd
ress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTim
e":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.98","key
":"novaStar","logined":true,"loginedUsernames":[],"password":"","plat
form":"","privacy":false,"productInfo":{"configInfo":{"displayDevice"
:"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"
productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order"
:0,"videoSource":0},"sn":"W9A61VK7","syssetFtpPort":16604,"syssetTcpP
ort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"wi
dth":775}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-
40000105","configInfo":{"config_info":{"Brightness":{"CompleteCron":f
alse,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERAT
URE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGH
T":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":fal

se},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.3","key":"novaStar","logined":false,"loginedUsernames":[""],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA07194A0040000105","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo
data:{"aliasName":"VPlayer_806D6287","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":fal

se,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.77","key":"novaStar","logined":true,"loginedUsernames":[],"password":"","platform":"","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"806D6287","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-10000177","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkChec

k":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.92","key":"novaStar","logined":true,"loginedUsernames":["admin"],"password":"","platform":"rk3368","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA07201A0010000177","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo
data:{"aliasName":"VPlayer_806C7F8B","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":

false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16612,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.69","key":"novaStar","logined":false,"loginedUsernames":[],"password":"","platform":"","privacy":false,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"VPlayer","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"806C7F8B","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16613,"terminalState":0,"username":["admin"],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-20004764","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":

false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.34","key":"novaStar","logined":true,"loginedUsernames":["admin"],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA17466W0X20004764","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-50000373","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false

,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":600,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.88","key":"novaStar","logined":true,"loginedUsernames":["admin"],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA07216J0550000373","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":800}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-50000355","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportP

roduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.28","key":"novaStar","logined":true,"loginedUsernames":["admin"],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA07216J0550000355","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-20004871","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"

TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":false,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.121","key":"novaStar","logined":false,"loginedUsernames":[""],"password":"","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA17466W0X20004871","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-49999716","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"Video

Source":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"","model":"","productName":"","registerAddress":""},"ftpPort":16602,"hasPassWord":true,"height":400,"ignoreTime":0,"installedPackageVersions":{"result":[]},"ip":"172.16.9.71","key":"novaStar","logined":false,"loginedUsernames":[""],"password":"123456","platform":"rk3368","privacy":true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[],"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"productName":"T8","screenMosaicInfo":{"order":0,"videoSource":0},"sn":"BZSA07194A0049999716","syssetFtpPort":16604,"syssetTcpPort":16605,"tcpPort":16603,"terminalState":0,"username":[],"width":400}
ViplexCore Demo code:0
ViplexCore Demo data:{"aliasName":"Taurus-20001502","configInfo":{"config_info":{"Brightness":{"CompleteCron":false,"FloatValue":false,"Validity":false},"CARE":false,"COLORTEMPERATURE":false,"CustomResolution":false,"FIXEDPOINT":false,"INDICATORLIGHT":false,"InfraredDetector":{"IsSupport":false,"IsSupportProduct":false},"LIGHT":false,"Monitoring":{"AmbientBrightness":false,"CPU":false,"ClearMedia":false,"IsSupport":false,"Memory":false},"NEWPROTOCOL":false,"NEXUSFONT":false,"Network":{"AP":false,"AirPlane":false,"Apn":false,"IsSupport":false,"IsSupportGetModuleInfo":false,"IsSupportModuleVersionUpdate":false,"IsSupportMultiDns":false,"IsSupportNetWorkCheck":false,"IsSupportSimModeSwitch":false,"Log4G":false,"Mobile":false,"NetStateFor4G":false,"Restart4GModule":false,"WiFiApSwitch":false,"Wifi":false,"Wired":false,"WorkStateFor4G":false},"PLAYLOG":false,"POWER":false,"PlayManager":{"IsSupport":false,"ProgramListManager":false},"REBOOT":false,"RELAYPOWER":false,"RELAYPOWERCONFIG":false,"RESET":false,"RadioFrequencyManage":{"IsSupport":false,"IsSupportLoraInfo":false,"IsSupportProduct":false},"ReceiveCard":{"Config":false,"IsSupport":false,"Temperature":false},"Resolution":{"CustomResolution":false,"IsSupport":false},"Rotation":{"IsSupport":false,"IsSupportProduct":false},"SCREENPOWER":false,"SCREENSHOT":false,"SPOTS":false,"SYNC_PLAY":false,"ScreenConfig":{"Config":false,"IsSupport":false,"IsSupportProduct":false,"ScreenJoint":false},"SensorBoard":{"IsSupport":false,"IsSupportProduct":false,"SensorInfo":false},"SourceOutMode":{"IsSupportChangeSourceOutMode":false},"TERMINALINFO":false,"TIMEZONE":false,"TimeControl":{"IsSupport":false,"Lora":false,"Manual":false,"NTP":false},"UPDATE":false,"UPDATEOS":false,"Upgrade":{"IsSupportCheckValid":false},"VIDEO_SOURCE_SWITCH":false,"VNNOX":false,"VOICE":false,"VideoSource":{"Hdmi":false,"Input":{"Hdmi":false,"IsSupport":false},"IsSupport":false,"Output":{"HdmiToLvds":false,"IsSupport":false}},"WIFI":false,"ZipRunningLog":{"IsSupportZipRunningLogInfo":false}},"version":

```
""},"firmwareInfo":{"aliasName":"","fpga":"","mac":"","mainVersion":"
","model":"","productName":"","registerAddress":""},"ftpPort":16602,"
hasPassWord":false,"height":256,"ignoreTime":0,"installedPackageVersi
ons":{"result":[]},"ip":"172.16.9.48","key":"novaStar","logined":fals
e,"loginedUsernames":[""],"password":"","platform":"rk3368","privacy"
:true,"productInfo":{"configInfo":{"displayDevice":"","portConfig":[]
,"videoSwitch":false},"productInfo":{"modelId":0,"productName":""}},"
productName":"T6","screenMosaicInfo":{"order":0,"videoSource":0},"sn"
:"BZSA17313J0820001502","syssetFtpPort":16604,"syssetTcpPort":16605,"
tcpPort":16603,"terminalState":0,"username":[],"width":128}

ViplexCore Demo nvLoginAsync begin...
ViplexCore Demo code:0
ViplexCore Demo
data:{"logined":true,"password":"123456","sn":"BZSA07194A0049999716",
"username":"admin","validation":true,"validition":true}

ViplexCore Demo nvSetVolumeAsync begin...
ViplexCore Demo code:0
ViplexCore Demo data:

ViplexCore Demo nvGetVolumeAsync begin...
ViplexCore Demo code:0
ViplexCore Demo data:{"ratio":60.0}
```


5.1、C/C++

5.1.1、C/C++_windows

以下代码示范了初始化，搜索设备，登录设备，设置音量和获取音量API的使用，设备返回信息见：运行结果 注意：
如果JSON参数中涉及到中文，请确保调用处是*UTF-8*编码，如： #pragma execution_character_set("utf-8")

```
#include
#include
#include
#include
#include
#include
#include
#pragma execution_character_set("utf-8")
using namespace std;
typedef void(*ExportViplexCallback)(const uint16_t, const char *);
```

```
//#define LOADING_FUNCTION_WITH_DYNAMIC
#ifndef LOADING_FUNCTION_WITH_DYNAMIC
#include "exportviplexcoreasync.h"
#pragma comment(lib, "viplexcore.lib")
#else
typedef int(*nvInitFunc)(const char*, const int, const int);
typedef void(*nvSerchTerminalAsyncFunc)(ExportViplexCallback);
typedef void(*nvLoginAsyncFunc)(const char*, ExportViplexCallback);
typedef void(*nvSetVolumeAsyncFunc)(const char*,
ExportViplexCallback);
typedef void(*nvGetVolumeAsyncFunc)(const char*,
ExportViplexCallback);
nvInitFunc nvInit;
nvSerchTerminalAsyncFunc nvSerchTerminalAsync;
nvLoginAsyncFunc nvLoginAsync;
nvSetVolumeAsyncFunc nvSetVolumeAsync;
nvGetVolumeAsyncFunc nvGetVolumeAsync;
void loadingFunctionWithDynamic()
{
    DWORD dwError = 0;
    HMODULE hDll = LoadLibraryExA("viplexcore.dll", NULL,
LOAD_WITH_ALTERED_SEARCH_PATH);//加载dll
    if (NULL == hDll)
    {
        dwError = GetLastError();
        std::string msg = std::string("ViplexCore Demo
加载dll文件失败:") + to_string(dwError);
        OutputDebugStringA(msg.c_str());
    }
    nvInit = (nvInitFunc)(GetProcAddress(hDll, "nvInit"));//加载函数
    if (NULL == nvInit)
    {
        std::string msg = std::string("ViplexCore Demo
函数nvInit加载失败");
        OutputDebugStringA(msg.c_str());
    }
    nvSerchTerminalAsync =
(nvSerchTerminalAsyncFunc)(GetProcAddress(hDll,
"nvSerchTerminalAsync"));
    if (NULL == nvSerchTerminalAsync)
    {
        std::string msg = std::string("ViplexCore Demo
函数nvInit加载失败");
        OutputDebugStringA(msg.c_str());
```

```
    }
    nvLoginAsync = (nvLoginAsyncFunc)(GetProcAddress(hDll,
"nvLoginAsync"));
    if (NULL == nvLoginAsync)
    {
        std::string msg = std::string("ViplexCore Demo
函数nvInit加载失败");
        OutputDebugStringA(msg.c_str());
    }
    nvSetVolumeAsync = (nvSetVolumeAsyncFunc)(GetProcAddress(hDll,
"nvSetVolumeAsync"));
    if (NULL == nvSetVolumeAsync)
    {
        std::string msg = std::string("ViplexCore Demo
函数nvInit加载失败");
        OutputDebugStringA(msg.c_str());
    }
    nvGetVolumeAsync = (nvGetVolumeAsyncFunc)(GetProcAddress(hDll,
"nvGetVolumeAsync"));
    if (NULL == nvSetVolumeAsync)
    {
        cout << "ViplexCore Demo 函数nvGetVolumeAsync加载失败" <<
endl;
    }
}
#endif
atomic_bool g_bAPIReturn = false;
ExportViplexCallback callback = [](const uint16_t code, const char
*data)
{
    std::string msgCode = std::string("ViplexCore Demo code:") +
to_string(code);
    std::string msgData = std::string("ViplexCore Demo data:") +
data;
    OutputDebugStringA(msgCode.c_str());
    OutputDebugStringA(msgData.c_str());
    g_bAPIReturn = true;
};
void waitAPIReturn()
{
    while (!g_bAPIReturn)
    {
        this_thread::sleep_for(chrono::seconds(1));
    }
```

```
    g_bAPIReturn = false;
}
void APITester()
{
    cout << "nvInit:" << nvInit("D:\\test", 1, 1) << endl;
    OutputDebugStringA("ViplexCore Demo nvSerchTerminalAsync begin...
");
    nvSerchTerminalAsync(callback);
    this_thread::sleep_for(chrono::seconds(5));
    g_bAPIReturn = false;
    OutputDebugStringA("ViplexCore Demo nvLoginAsync begin... ");

nvLoginAsync("{\"sn\":\"BZSA07194A0049999716\",\"username\":\"admin\"
,\"password\":\"123456\",\"loginType\":0,\"rememberPwd\":0}",
callback);
    waitAPIReturn();
    OutputDebugStringA("ViplexCore Demo nvSetVolumeAsync begin... ");

nvSetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\",\"volumeInfo\":{\"
ratio\":60.0}}", callback);
    waitAPIReturn();
    OutputDebugStringA("ViplexCore Demo nvGetVolumeAsync begin... ");
    nvGetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\"}", callback);
    waitAPIReturn();
}
int main()
{
#ifdef LOADING_FUNCTION_WITH_DYNAMIC
    loadingFunctionWithDynamic();
#endif
    APITester();
    return 0;
}
```

5.1.2、C/C++_Ubuntu

以下代码示范了初始化，搜索设备，登录设备，设置音量和获取音量API的使用，设备返回信息见：运行结果 注意：

1、如果JSON参数中涉及到中文，请确保调用处是*UTF-8*编码，如：

```
#pragma execution_character_set("utf-8")
```

2、LOADING_FUNCTION_WITH_DYNAMIC开启此宏代表动态加载（建议使用静态加载）

```
#示例 qmake:

TEMPLATE = app
CONFIG += console c++11
CONFIG -= app_bundle
CONFIG += qt

SOURCES += \
        main.cpp
unix:{
    QMAKE_LFLAGS += "-Wl,-rpath,\'\$$ORIGIN\'"
}
DEFINES -= LOADING_FUNCTION_WITH_DYNAMIC #是否动态加载

if(contains(DEFINES,LOADING_FUNCTION_WITH_DYNAMIC))
{
    LIBS +=-ldl -lpthread
}
else()
{
LIBS +=-ldl -lpthread -L/lviplexcore库路径/ -lviplexcore
INCLUDEPATH+=/头文件路径/include
}
```

```
#include
#include
#include
#include
#include
#include
#include
#include
//#include

#pragma execution_character_set("utf-8")
using namespace std;
typedef void(*ExportViplexCallback)(const uint16_t, const char *);
#ifndef LOADING_FUNCTION_WITH_DYNAMIC
#include "exportviplexcoreasync.h"
#else
typedef int(*nvInitFunc)(const char*, const int, const int);
typedef void(*nvSerchTerminalAsyncFunc)(ExportViplexCallback);
typedef void(*nvLoginAsyncFunc)(const char*, ExportViplexCallback);
typedef void(*nvSetVolumeAsyncFunc)(const char*,
ExportViplexCallback);
```

```
typedef void(*nvGetVolumeAsyncFunc)(const char*,
ExportViplexCallback);
nvInitFunc nvInit;
nvSerchTerminalAsyncFunc nvSerchTerminalAsync;
nvLoginAsyncFunc nvLoginAsync;
nvSetVolumeAsyncFunc nvSetVolumeAsync;
nvGetVolumeAsyncFunc nvGetVolumeAsync;
void loadingFunctionWithDynamic()
{
    void* handle =
dlopen("/home/qht/viplexcore/RelWithDebInfo/bin/libviplexcore.so",
RTLD_LAZY);
        if(!handle)
        {
        printf("ERROR, Message(%s).\n", dlerror());
                return ;
        }

        nvInit = (nvInitFunc)dlsym(handle, "nvInit");
        char* szError = dlerror();
        if(szError != NULL)
        {
            printf("ERROR, Message(%s).\n", szError);
            dlclose(handle);
            return ;
        }

        nvSerchTerminalAsync =
(nvSerchTerminalAsyncFunc)dlsym(handle, "nvSerchTerminalAsync");
        char* szError2 = dlerror();
        if(szError2 != NULL)
        {
            printf("ERROR, Message(%s).\n", szError2);
            dlclose(handle);
            return ;
        }

        nvLoginAsync = (nvLoginAsyncFunc)dlsym(handle,
"nvLoginAsync");
        char* szError3 = dlerror();
        if(szError3 != NULL)
        {
            printf("ERROR, Message(%s).\n", szError3);
            dlclose(handle);
```

```
            return ;
        }

        nvSetVolumeAsync = (nvSetVolumeAsyncFunc)dlsym(handle,
"nvSetVolumeAsync");
        char* szError4 = dlerror();
        if(szError4 != NULL)
        {
            printf("ERROR, Message(%s).\n", szError4);
            dlclose(handle);
            return ;
        }

        nvGetVolumeAsync = (nvGetVolumeAsyncFunc)dlsym(handle,
"nvGetVolumeAsync");
        char* szError5 = dlerror();
        if(szError5 != NULL)
        {
            printf("ERROR, Message(%s).\n", szError5);
            dlclose(handle);
            return ;
        }
}
#endif
atomic_bool g_bAPIReturn(false);
ExportViplexCallback callback = [](const uint16_t code, const char
*data)
{
    std::string msgCode = std::string("ViplexCore Demo code:") +
to_string(code);
    std::string msgData = std::string("ViplexCore Demo data:") +
data;
    std::cout << msgCode.c_str();
    std::cout << msgData.c_str();
    g_bAPIReturn = true;
};
void waitAPIReturn()
{
    while (!g_bAPIReturn)
    {
        this_thread::sleep_for(chrono::seconds(1));
    }
    g_bAPIReturn = false;
}
```

```
void APITester()
{
    cout << "nvInit:" << nvInit("D:\\test", 1, 1) << endl;
    std::cout << ("ViplexCore Demo nvSerchTerminalAsync begin... ");
    nvSerchTerminalAsync(callback);
    this_thread::sleep_for(chrono::seconds(5));
    g_bAPIReturn = false;
    std::cout << ("ViplexCore Demo nvLoginAsync begin... ");

nvLoginAsync("{\"sn\":\"BZSA07194A0049999716\",\"username\":\"admin\"
,\"password\":\"123456\",\"loginType\":0,\"rememberPwd\":0}",
callback);
    waitAPIReturn();
    std::cout << ("ViplexCore Demo nvSetVolumeAsync begin... ");

nvSetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\",\"volumeInfo\":{\"
ratio\":60.0}}", callback);
    waitAPIReturn();
    std::cout << ("ViplexCore Demo nvGetVolumeAsync begin... ");
    nvGetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\"}", callback);
    waitAPIReturn();
}
int main()
{
#ifdef LOADING_FUNCTION_WITH_DYNAMIC
    loadingFunctionWithDynamic();
#endif
    APITester();
    return 0;
}
```

5.2、C#

5.2.1、C#

以下代码示范了初始化，搜索设备，登录设备，设置音量和获取音量API的使用，设备返回信息见：运行结果 注意：
如果JSON参数中涉及到中文，请确保调用处是*UTF-8*编码。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
```

```
using System.Threading;
using System.Threading.Tasks;

namespace Api
{
    [UnmanagedFunctionPointerAttribute(CallingConvention.Cdecl,
CharSet = CharSet.Unicode)]
    public delegate void exportViplexCallback(Int16 code, string
data);

    class ApiTester
    {
        [DllImport("viplexcore.dll", EntryPoint = "nvInit", CharSet =
CharSet.Unicode, CallingConvention = CallingConvention.Cdecl)]
        public static extern int NvInit(string sdkRootDir, int type,
int platform);

        [DllImport("viplexcore.dll", EntryPoint =
"nvSerchTerminalAsync", CharSet = CharSet.Unicode, CallingConvention
= CallingConvention.Cdecl)]
        public static extern void
NvSerchTerminalAsync(exportViplexCallback resultCallback);

        [DllImport("viplexcore.dll", EntryPoint = "nvLoginAsync",
CharSet = CharSet.Unicode, CallingConvention =
CallingConvention.Cdecl)]
        public static extern void NvLoginAsync(string data,
exportViplexCallback resultCallback);

        [DllImport("viplexcore.dll", EntryPoint = "nvSetVolumeAsync",
CharSet = CharSet.Unicode, CallingConvention =
CallingConvention.Cdecl)]
        public static extern void NvSetVolumeAsync(string data,
exportViplexCallback resultCallback);

        [DllImport("viplexcore.dll", EntryPoint = "nvGetVolumeAsync",
CharSet = CharSet.Unicode, CallingConvention =
CallingConvention.Cdecl)]
        public static extern void NvGetVolumeAsync(string data,
exportViplexCallback resultCallback);

        static bool g_bAPIReturn = false;

        static void testDemoDelegate(Int16 code, string data)
```

```
        {
            string strCode = "ViplexCore Demo code:" + code;
            string strData = "ViplexCore Demo data:" + data;
            Console.WriteLine(strCode);
            Console.WriteLine(strData);
            g_bAPIReturn = true;
        }

        static void waitAPIReturn()
        {
            while (!g_bAPIReturn)
            {
                Thread.Sleep(1000);
            }
            g_bAPIReturn = false;
        }

        static void testApi()
        {
            Console.Write("nvInit:");
            Console.WriteLine(NvInit("D:/ApiTester/bin", 1, 1));

            Console.WriteLine("ViplexCore Demo nvSerchTerminalAsync
begin... ");
            NvSerchTerminalAsync(testDemoDelegate);
            Thread.Sleep(5000);
            g_bAPIReturn = false;

            Console.WriteLine("ViplexCore Demo nvLoginAsync begin...
");

NvLoginAsync("{\"sn\":\"BZSA07194A0049999716\",\"username\":\"admin\"
,\"password\":\"123456\",\"loginType\":0,\"rememberPwd\":0}",
testDemoDelegate);
            waitAPIReturn();
            Console.WriteLine("ViplexCore Demo nvSetVolumeAsync
begin... ");

NvSetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\",\"volumeInfo\":{\"
ratio\":60.0}}", testDemoDelegate);
            waitAPIReturn();
            Console.WriteLine("ViplexCore Demo nvGetVolumeAsync
begin... ");
            NvGetVolumeAsync("{\"sn\":\"BZSA07194A0049999716\"}",
```

```
testDemoDelegate);
            waitAPIReturn();

            Console.ReadKey();
        }

        static void Main(string[] args)
        {
            testApi();
        }
    }
}
```

5.3、Java（待完善）

5.4、Flutter（待完善）

5.5、JS（待完善）