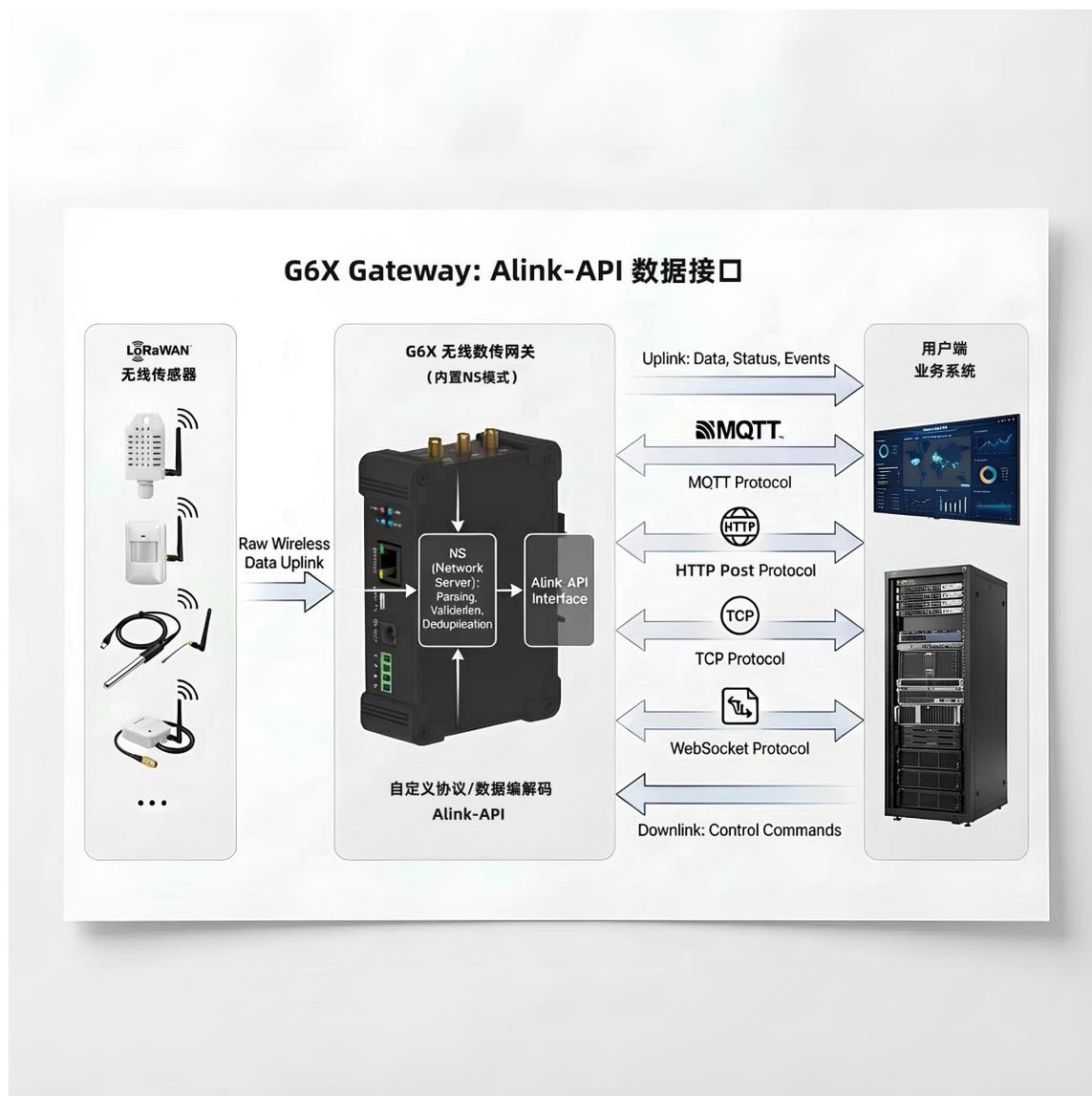


G6x 数传网关 API 接口协议及报文定义

内置 NS 模式 | 数据接口 (Alink-API)

G6X 网关内置 **NS (Network Server, 网络服务器)** 功能，可对接入的 LoRaWAN 无线节点进行统一管理与数据处理。节点的数据在进入网关后，会先由网关内部的 NS 进行解析、校验和编码，再通过网关自带的 Alink-API 接口与用户系统进行数据交互。

Alink-API 支持使用 **MQTT / HTTP / TCP / WebSocket** 协议，用户可以方便地获取节点的**上行数据、设备状态、事件信息**，同时也可以向节点发送**下行指令**，实现业务系统与 G6X 网关设备之间的双向通信。



Alink-API 协议类型配置

在网关配置页面最左侧的主导航栏中，点击 设置>数据传输>网关数据模式(内置 NS)

在同一页面向下滑动，即可找到 “Alink API 数据接口” 标题。在此区域，您可以开启 Alink API 并配置协议 (MQTT/HTTP/TCP/WS) 类型及服务器地址，数据主题等参数。

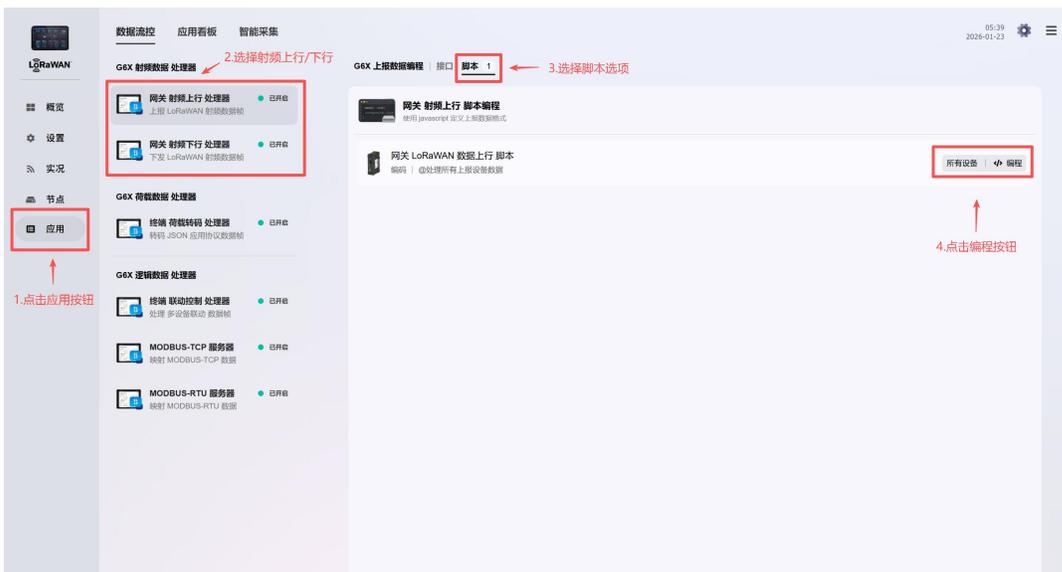


Alink-API 报文内容自定义

可编程

在网关配置页面最左侧的主导航栏中，点击 应用>网关 射频上行/下行处理器>脚本

在同一页面右侧，即可找到 “编程” 按钮。点击按钮后，您可以开启 Alink API 代码编辑器，通过 Javascript 脚本编码，使网关产生和发送 自定义 上行/下行 数据格式报文。



终端 - 射频上行报文 / Uplink

MQTT

HTTP

TCP

WS

可编程

当 LoRaWAN 终端设备完成数据采集并向网关上报时，网关在成功接收到该数据后，会生成一条 LoRaWAN 射频上行报文 (Uplink)。

该数据报文由网关内置 NS 统一处理后，通过 API 接口**上报**给用户系统。

通过该报文，用户系统可获取设备上报的数据内容及通信状态，用于业务处理、数据展示和设备监控。

上行报文 MQTT 默认主题 (可自定义)：/gateway/{网关 EUI}/ns/up1

上行 JSON 报文 默认格式 (可自定义)：

```
{
  "applicationId": "0", // 应用 ID
  "applicationName": "G6x_LoRaWAN_Uplink", // 应用名称
  "deviceName": "测试终端 OTAA", // 设备名称
  "devEui": "9cc466c19fe2da25", // 设备唯一标识 EUI
  "fCnt": 2, // 上行帧计数
  "fPort": 3, // 数据端口号
  "payload": "EjRWeJCrzQ4=", // 原始上行数据 (Base64 编码)
  "data": "1234567890abcd0e", // 原始上行数据 (字符编码)
  "rxInfo": { // 接收信息
    "gatewayId": "639c6847407c7c4c", // 接收网关 EUI
    "rssi": -39, // 信号强度
    "snr": 13.8 // 信噪比
  },
  "txInfo": { // 无线参数
    "frequency": 471.7, // 工作频点 (MHz)
    "modulation": {
      "lora": {
        "spreadingFactor": "SF7BW125", // 扩频因子/带宽
        "codeRate": "CR_4_5" // 编码率
      }
    }
  }
}
```

终端 - 射频下行报文 / Downlink

MQTT

TCP

WS

可编程

当用户系统需要向 LoRaWAN 终端设备发送控制指令或配置数据时，需要向网关推送一条 LoRaWAN 射频下行报文 (Downlink)。

该数据帧通过用户下发，由 API 接口**下发**给网关内置 NS 进行处理，并发送至指定的终端设备。通过该报文，用户系统可以实现对终端设备的远程控制和参数配置。

下发报文 MQTT 默认主题 (可自定义) : /gateway/{网关 EUI}/ns/down1

下发报文 JSON 默认格式 (可自定义) :

```
{
  "deveui": "9fe2da259fe2da25",           // 设备唯一标识 EUI
  "data": "1234567890abcdef",           // 自定义下发数据内容 (字符编码)
  "fPort": 2,                           // 数据端口号, 默认 2
  "retry": 1,                            // 数据下发次数, 默认 1
  "confirmed": false                     // 终端是否需要在下一次上报中确认
}
```

终端 - OTAA 入网报文 / Join

MQTT

HTTP

TCP

WS

当 LoRaWAN 终端设备采用 OTAA 方式完成入网流程，并成功加入网络后，网关会生成一条 OTAA 入网事件报文 (Join Event)。

该报文用于通知用户系统设备已成功入网，通过 API 接口**上报**给用户系统。

通过该事件报文，用户系统可以感知设备入网状态，用于设备初始化、状态更新及运维管理。

上报报文 MQTT 默认主题 (可自定义) : /gateway/{网关 EUI}/ns/join

上报报文 JSON 数据格式 (不可修改) :

```
{
  "ApplicationID": "0",                  // 应用 ID
  "ApplicationName": "JoinEvent",        // 应用名称
  "DeviceName": "测试终端 OTAA",        // 设备名称
  "DevEUI": "9cc466c19fe2da25",         // 设备唯一标识 EUI
  "DevAddr": "c0152925",                 // OTAA 入网为设备分配的短地址
  "Time": "2026-01-20T23:53:25.475Z"    // OTAA 入网时间
}
```

终端 - 确认回复报文 / Ack

MQTT

HTTP

TCP

WS

当 LoRaWAN 终端设备接收到一条 需要确认的下发数据帧 (Confirmed Downlink) 后, 终端会在下一次上行通信时携带确认标志进行回复。

网关在接收到该上行数据后, 将整理为一条确认信息 (Ack Event) 并通过 API 上报给用户系统。

通过该事件数据帧, 用户系统可判断下发指令是否被终端成功接收, 用于下发结果确认和通信状态分析。

上发报文 MQTT 默认主题 (可自定义) : /gateway/{网关 EUI}/ns/ack

上发报文 JSON 数据格式 (不可修改) :

```
{
  "applicationID": "0", // 应用 ID
  "applicationName": "AckEvent", // 应用名称
  "deviceName": "测试终端 OTAA", // 设备名称
  "devEUI": "9cc466c19fe2da25", // 设备唯一标识 EUI
  "fCnt": 10, // 确认的下行计数
  "acknowledged": true, // 确认标志, 固定为 true
  "time": "2026-01-21T02:31:07.656Z" // 确认时间
}
```

终端 - 状态查询 / Request+Response

MQTT

当用户系统需要查询当前网关下已接入设备的状态和基础信息时, 仅可通过 MQTT 请求方式向网关下发设备信息查询请求 (Request) 。

网关在接收到请求后, 会对请求进行处理, 并通过 MQTT 返回对应的 设备信息响应数据 (Response) 。

通过该请求与响应机制, 用户系统可以实时获取设备运行状态, 用于设备管理、状态监测和运维分析。

第一步: 用户通过 MQTT 主题下发 Request 请求

下发请求 MQTT 数据主题 (不可修改) : gateway/{网关 EUI}/api/request

下发请求 JSON 数据格式 (不可修改) :

```
{
  "id": "req-1", // 请求 ID, 网关将回复相同 ID
  "method": "GET", // 请求方法
  "path": "/api/cot/Dev/DeviceList", // 请求的接口路径
}
```

```
"headers": {}, // 头信息, 默认为空
"body": "", // 请求体, 默认为空
"query": {} // 查询参数, 默认为空
}
```

第二步：网关 MQTT 主题上报 Response 回复用户

上报回复 MQTT 数据主题（不可修改）：`gateway/{网关 EUI}/api/response`

上报回复 JSON 数据格式（不可修改）：

```
{
  "id": "req-1", // 响应的请求 ID, 与 Request 中的 id 一致
  "status_code": 200, // 请求处理结果状态码
  "headers": {
    "Access-Control-Allow-Origin": "*", // 响应头信息
    "Content-Type": "application/json" // 数据格式
  },
  "body": {
    "code": 0, // 业务处理结果 (0 表示成功)
    "data": [
      {
        "name": "J32_环境监测", // 设备名称
        "devEUI": "9135bfd79dfdafb3", // 设备唯一标识 EUI
        "lastSeen": "N/A", // 最近一次上行时间
        "lastSeenDown": "", // 最近一次下行时间
        "isOtaa": true, // 是否为 OTAA 入网方式
        "fCntUp": 0, // 上行数据帧计数
        "class": "C", // 设备类型 (Class A/B/C)
        "alive": false, // 当前是否在线
        "id": 0, // 预留字段
        "javaName": "" // 预留字段
      },
      {
        "name": "测试终端 OTAA",
        "devEUI": "9cc466c19fe2da25",
        "lastSeen": "2026/01/21 10:31:07",
        "lastSeenDown": "",
        "isOtaa": true,
      }
    ]
  }
}
```

```
"fCntUp": 13,
"class": "A",
"alive": true,
"id": 1768962667,
"javaName": ""
}
],
"msg": "ok"
},
"error": ""
}
```

网关 - 状态报文 / Status

MQTT

HTTP

TCP

WS

网关在正常运行过程中，会按照设定的周期主动上报心跳状态报文，用于告知用户系统当前网关仍处于在线运行状态。

该报文用于网关运行状态监测，不涉及具体设备或业务数据。

通过该心跳状态报文，用户系统可以实时掌握网关的在线情况，用于运行监控和运维管理。

上报报文 MQTT 默认主题（可自定义）：gateway/event/stats

上报报文 JSON 数据格式（不可修改）：

```
{
  "gatewayID": "639c6847407c7c4c", // 网关唯一标识 (Gateway EUI)
  "time": "2025-05-18T10:28:25Z", // 心跳上报时间 (UTC, ISO 8601 格式)
  "status": "00 天 16 时 09 分 13 秒" // 网关运行时长
}
```