

StreamGate Public API 接入文档

面向第三方业务系统的 REST API 接入说明，包含授权、鉴权、接口字段、调用示例、错误处理和版本兼容信息。

适用版本: StreamGate 1.8.2 / 1.9.1 / 1.9.2 / 1.9.3 / 1.9.4 / 1.9.6 / 1.9.7 API 版本: v1 文档版本: 1.2

更新日期: 2026-10-13

1. 文档范围

Public API 用于第三方系统读取 StreamGate 的设备、通道、运行状态、播放地址、录像索引和告警状态。StreamGate 1.9.1 Professional / Enterprise 进一步提供设备、通道、录像、GB28181、WebHook 与 MQTT 的管理接口。

接入方式	方向与用途	验证方式
REST API	客户系统主动请求网关，读取或修改资源	Bearer Token 与所需 scope
WebHook	网关主动向客户 HTTP 接收端 POST 事件	接收端校验 HMAC 签名
MQTT	网关向 broker/topic 发布事件，客户订阅	broker 凭据、主题权限与配置的 QoS

三者不是同一接口。API 中的 `/integrations/webhooks`、`/integrations/mqtt` 用于管理订阅与投递配置，不是事件接收地址；API Token 也不替代 WebHook 签名密钥或 MQTT 凭据。

基础地址	<code>http://streamgate.example:18080</code>
接口前缀	<code>/api/public/v1</code>
数据格式	<code>application/json; charset=utf-8</code>
鉴权方式	<code>Authorization: Bearer <API_TOKEN></code>
时间格式	ISO 8601，服务端时间字段使用 UTC，例如 <code>2026-09-18T08:30:00Z</code>

示例说明： 域名、标识、Token 和业务数据均为示例。请将 `streamgate.example` 替换为实际服务器域名或 IP，并保留配置的 Web 端口。

2. 版本兼容

能力	1.8.2	1.9.1	所需 scope
查询通道	支持	支持	channels:read
查询录像索引	支持	支持	recording:read
查询告警	支持	支持	alarms:read
确认、清除告警	支持	支持	alarms:write
查询设备身份	-	1.9.1 新增	devices:read
查询通道运行状态	-	1.9.1 新增	status:read
获取播放地址	-	1.9.1 新增	playback:read
设备与 ONVIF 管理	-	1.9.1 专业版	devices:write
通道配置与云台控制	-	1.9.1 专业版	channels:write / ptz:write
录像任务与策略	-	1.9.1 专业版	recording:read / recording:write
GB28181 管理	-	1.9.1 专业版	gb28181:read / gb28181:write
WebHook、MQTT 与投递日志	-	1.9.1 专业版	integrations:read / integrations:write

第三方程序应忽略响应中未识别的字段，以便在后续兼容版本增加可选信息时保持正常运行。对 1.8.2 与 1.9.1 混合部署，建议先探测所需端点；收到 404 时按版本能力降级。

3. 授权与 Token

3.1 使用条件

- 当前 1.9.7：Public REST API、WebHook、MQTT 均不向 Free / Standard 开放；有效 Professional 试用、Professional 或 Enterprise 软件授权可使用，仍须满足 openapi 权益及运行包条件。评估视频不加水印；试用仅用于授权允许的非生产评估，期限、容量及功能范围以签发授权为准。
- REST 调用及通过 REST 管理集成还须有效 API Token 和相应 scopes；WebHook 接收端签名校验、MQTT 凭据与主题权限另行满足。控制台内部接口不属于 Public API。

- Analysis 授权不替代上述软件权益。历史 1.8.2 接口仍要求已启用 OpenAPI 的 Professional / Enterprise 授权；版本支持不代表自动授予接口权限。

3.2 创建 API Token

- 使用管理员账号登录 StreamGate Web Console。
- 进入“系统设置 > 开放接口 > API 令牌”。
- 填写第三方系统名称，并按最小权限原则选择 scope。
- 点击“创建 Token”，立即保存页面中仅显示一次的完整 Token。

重要：完整 Token 只在创建后显示一次。StreamGate 后续只显示 Token 前缀，无法恢复原文；遗失后应撤销旧 Token 并重新创建。

3.3 权限范围

Scope	用途	建议
channels:read	读取通道清单	通道同步程序
recording:read	读取录像索引	录像检索程序
alarms:read	读取告警状态	告警看板、事件同步
alarms:write	确认或清除告警	仅授予需要处置告警的系统
devices:read	读取设备身份	1.9.1 及以上
status:read	读取通道运行状态	1.9.1 及以上
playback:read	读取通道播放地址	1.9.1 及以上
playback:warm	短时预热通道来源，与地址读取分开授权	以运行时 OpenAPI 为准
devices:write	设备增删改、批量启停、ONVIF 发现与码流解析	专业版；设备密码仅写入，不在响应中返回
channels:write	通道增删改、批量启停及按需/常驻切换	专业版
ptz:write	云台控制	专业版；与通道配置写权限分离
recording:write	录像任务、录像策略及策略应用	专业版
gb28181:read	读取国标平台、目录发布和上联配置	专业版

Scope	用途	建议
<code>gb28181:write</code>	管理国标平台、目录发布和上联配置	专业版
<code>integrations:read</code>	读取 WebHook、MQTT、事件类型和投递日志	专业版
<code>integrations:write</code>	管理和测试 WebHook / MQTT、清理投递日志	专业版
<code>exports:read</code>	读取 StreamGate 运行事件	专业版；响应执行凭据脱敏
<code>*</code>	完整 Public API 权限	仅用于受控的可信集成，不建议日常使用

3.4 请求头

HTTP

```
Authorization: Bearer <API_TOKEN>
Accept: application/json
Content-Type: application/json
```

GET 请求可省略 `Content-Type`。Token 不应放入 URL、查询参数、日志或前端页面代码。

4. 快速验证

以下请求用于验证网络、授权和 Token scope。请先将地址和 Token 替换为实际值。

curl

```
curl --request GET \
  "http://streamgate.example:18080/api/public/v1/channels" \
  --header "Authorization: Bearer <API_TOKEN>" \
  --header "Accept: application/json"
```

PowerShell

```
$baseUrl = "http://streamgate.example:18080"
$headers = @{ Authorization = "Bearer <API_TOKEN>" }

Invoke-RestMethod `
  -Uri "$baseUrl/api/public/v1/channels" `
  -Headers $headers `
  -Method Get
```

预期结果：HTTP 状态码为 `200`，响应体包含 `channels` 数组。空数组表示当前没有已配置通道，不代表接口异常。

5. 查询通道

GET `/api/public/v1/channels`

1.8.2+ 通道清单

返回已配置的视频通道，不包含设备账号、密码或接入 URL。

权限

`channels:read`

响应字段

字段	类型	说明
<code>channels</code>	array	通道数组
<code>channels[].id</code>	string	通道唯一标识
<code>channels[].endpointId</code>	string	关联接入端点标识，可选
<code>channels[].deviceId</code>	string	关联设备标识，可选
<code>channels[].name</code>	string	通道名称
<code>channels[].sourceType</code>	string	通道来源类型
<code>channels[].enabled</code>	boolean	是否启用
<code>channels[].onDemand</code>	boolean	是否按需拉流

响应示例

200 OK

```
{
  "channels": [
    {
      "id": "ch-001",
      "deviceId": "device-001",
      "name": "园区东门",
      "sourceType": "rtsp",
      "enabled": true,
      "onDemand": true
    }
  ]
}
```

6. 查询录像索引

GET /api/public/v1/recordings

1.8.2+ 录像索引

按条件查询本地录像索引。该接口返回索引信息，不等同于录像文件下载授权。

权限

recording:read

查询参数

参数	类型	必填	说明
mediaResourceId	string	否	按通道 ID 筛选
status	string	否	ready、recording 或 empty
limit	integer	否	分页数量，默认 100；1.8.2 建议不超过 500，1.9.1 有效范围为 1-1000
offset	integer	否	分页偏移，默认 0

请求示例

HTTP

```
GET /api/public/v1/recordings?mediaResourceId=ch-001&status=ready&limit=20&offset=0
```

响应字段

字段	类型	说明
<code>files</code>	array	录像索引数组
<code>files[].id</code>	string	录像文件标识
<code>files[].taskId</code>	string	录像任务标识
<code>files[].mediaResourceId</code>	string	通道标识
<code>files[].path</code>	string	服务器本地路径，仅在授权范围允许时返回
<code>files[].sizeBytes</code>	integer	文件大小，单位字节
<code>files[].status</code>	string	录像状态
<code>files[].error</code>	string	异常信息，可选
<code>files[].modifiedAt</code>	string	文件修改时间
<code>files[].indexedAt</code>	string	索引时间
<code>total</code>	integer	匹配记录总数
<code>limit</code>	integer	本次分页数量
<code>offset</code>	integer	本次分页偏移

响应示例

200 OK

```
{
  "files": [
    {
      "id": "rec-20260918-0001",
      "taskId": "task-001",
      "mediaResourceId": "ch-001",
      "path": "D:\\StreamGateData\\recordings\\ch-001\\20260918-083000.mp4",
      "sizeBytes": 24873984,
      "status": "ready",
      "modifiedAt": "2026-09-18T08:35:00Z",
      "indexedAt": "2026-09-18T08:35:02Z"
    }
  ],
  "total": 1,
  "limit": 20,
  "offset": 0
}
```

7. 查询告警

GET /api/public/v1/alarms

1.8.2+ 当前告警状态

查询当前告警状态。默认只返回处于活动状态的告警。

权限

alarms:read

查询参数

参数	类型	默认值	说明
limit	integer	100	分页数量，有效范围为 1-500
offset	integer	0	分页偏移
active	boolean	true	是否只查询活动告警

参数	类型	默认值	说明
<code>includeCleared</code>	boolean	false	是否包含已清除告警
<code>severity</code>	string	-	按严重级别筛选，例如 <code>warning</code> 、 <code>error</code>
<code>mediaResourceId</code>	string	-	按通道 ID 筛选

查询已清除告警：同时设置 `active=false` 和 `includeCleared=true`。仅设置其中一个参数不会返回已清除告警。

响应字段

字段	类型	说明
<code>dedupeKey</code>	string	告警去重标识，也是确认/清除接口的路径参数
<code>firstSeenAt</code>	string	首次出现时间
<code>lastSeenAt</code>	string	最近出现时间
<code>severity</code>	string	严重级别
<code>source</code>	string	告警来源
<code>type</code>	string	告警类型；客户端应允许新增类型
<code>mediaResourceId</code>	string	关联通道 ID，可选
<code>correlationKey</code>	string	关联标识，可选
<code>message</code>	string	告警摘要
<code>fields</code>	object	告警扩展字段，键值均为字符串
<code>active</code>	boolean	是否处于活动状态
<code>acknowledged</code>	boolean	是否已确认
<code>acknowledgedAt</code>	string	确认时间，可选
<code>acknowledgedBy</code>	string	确认主体，可选
<code>ackNote</code>	string	确认备注，可选

字段	类型	说明
<code>clearedAt</code>	string	清除时间，可选
<code>clearReason</code>	string	清除原因，可选
<code>eventCount</code>	integer	同一去重键累计事件数
<code>lastEventId</code>	integer	最近事件标识

响应示例

200 OK

```
{
  "alarms": [
    {
      "dedupeKey": "channel:stream_error:ch-001",
      "firstSeenAt": "2026-09-18T08:20:11Z",
      "lastSeenAt": "2026-09-18T08:21:03Z",
      "severity": "error",
      "source": "channel",
      "type": "channel.stream_error",
      "mediaResourceId": "ch-001",
      "message": "通道接流异常",
      "active": true,
      "acknowledged": false,
      "eventCount": 3,
      "lastEventId": 128
    }
  ]
}
```

8. 确认告警

POST `/api/public/v1/alarms/{dedupeKey}/ack`

1.8.2+ 标记已确认

将指定告警标记为已确认。 `dedupeKey` 必须进行 URL 编码。

权限

alarms:write

请求体

字段	类型	必填	说明
note	string	否	确认备注

```
curl

curl --request POST \
  "http://streamgate.example:18080/api/public/v1/alarms/channel%3Astream_error%3Ach-001/ack" \
  --header "Authorization: Bearer <API_TOKEN>" \
  --header "Content-Type: application/json" \
  --data "{\"note\":\"已通知现场检查摄像机网络\"}"
```

成功时返回更新后的完整告警状态。告警不存在时返回 404。

9. 清除告警

POST /api/public/v1/alarms/{dedupeKey}/clear

1.8.2+ 结束告警状态

清除指定告警。清除属于状态变更操作，调用前应由业务系统确认处置流程已经完成。

权限

alarms:write

请求体

字段	类型	必填	说明
note	string	否	清除原因

HTTP

```
POST /api/public/v1/alarms/channel%3Astream_error%3Ach-001/clear
Authorization: Bearer <API_TOKEN>
Content-Type: application/json

{
  "note": "网络恢复，画面已确认正常"
}
```

成功时返回更新后的完整告警状态，其中 `active` 为 `false`，并包含清除时间和原因。

10. 查询设备

GET `/api/public/v1/devices`

1.9.1+ 设备身份清单

返回设备身份信息，不返回设备地址、账号、密码或接入 URL。

权限

`devices:read`

响应字段

字段	类型	说明
<code>devices[].id</code>	string	设备唯一标识
<code>devices[].name</code>	string	设备名称
<code>devices[].vendor</code>	string	设备厂商，可选
<code>devices[].protocol</code>	string	接入协议
<code>devices[].enabled</code>	boolean	是否启用

200 OK

```
{
  "devices": [
    {
      "id": "device-001",
      "name": "园区东门摄像机",
      "vendor": "generic",
      "protocol": "onvif",
      "enabled": true
    }
  ]
}
```

11. 查询运行状态

GET /api/public/v1/status

1.9.1+ 通道分发状态

返回 StreamGate 当前观测到的实时分发会话状态。该状态不主动探测设备，也不等同于设备在线状态。

权限

status:read

状态值

unknown、starting、active、draining、idle、error。客户端应允许后续增加状态值。

200 OK

```
{
  "version": "1.9.1",
  "observedAt": "2026-09-18T08:30:00Z",
  "channels": [
    {
      "id": "ch-001",
      "enabled": true,
      "state": "active"
    }
  ]
}
```

12. 获取播放地址

GET /api/public/v1/channels/{id}/playback-urls

1.9.1+ 播放地址

返回通道可用的播放地址。调用本接口不会启动码流，也不会签发播放凭据。

权限

playback:read

查询参数

参数	类型	必填	说明
quality	string	否	auto、high 或 fluent；未提供时采用已保存的 viewing.defaultQuality，显式值覆盖该默认值；无观看配置的旧通道保留原单路行为
viewportWidth	integer	否	1 至 8192；请求或已保存的默认播放码流为 auto 时参与选择
viewportHeight	integer	否	1 至 8192；请求或已保存的默认播放码流为 auto 时参与选择

在“系统设置 > 媒体与播放”的“控制台播放码流”设置 `media.defaultViewingQuality`：
auto/high/fluent 分别为自动、主码流、子码流，统一用于控制台预览、详情及标准/专业大屏。自动按窗口和展示方式选流，没有可用子码流时回退主码流。通道界面已移除默认播放选择；旧

`viewing.defaultQuality` 仅兼容未传 `quality` 的 Public API 播放地址请求，显式 `quality=high/fluent/auto` 仍覆盖它。系统控制台选择不改变 Public API、录像选源或 GB28181 供流。

响应字段

字段	类型	说明
<code>id</code>	string	逻辑通道 ID
<code>enabled</code>	boolean	通道是否启用
<code>outputStreamId</code>	string	实际输出流 ID
<code>httpFlv</code>	string	HTTP-FLV 地址；协议未启用时为空
<code>websocketFlv</code>	string	WebSocket-FLV 地址；协议未启用时为空
<code>hls</code>	string	HLS 地址；协议未启用时为空
<code>webRtc</code>	string	WebRTC 播放页面地址；协议未启用时为空
<code>whep</code>	string	WHEP 接口地址；协议未启用时为空
<code>requestedQuality</code>	string	请求的画质档位
<code>actualQuality</code>	string	实际选择的画质档位
<code>qualitySource</code>	string	画质来源
<code>qualityReason</code>	string	画质选择原因
<code>qualityFallback</code>	string	发生回退时的原因，可选
<code>fluentAvailable</code>	boolean	是否具备子码流输出
<code>playbackTokenRequired</code>	boolean	播放地址是否还需要播放凭据

鉴权边界：API Token 只用于调用 Public API，不能替代播放凭据或控制台会话。当 `playbackTokenRequired` 为 `true` 时，播放器仍需按 StreamGate 的播放鉴权配置取得有效凭据。

多窗口界面的画质切换

- 1. 宫格窗口请求 `?quality=fluent`，使用响应中的子码流地址。
- 2. 窗口放大时停止原播放器，再请求 `?quality=high` 并建立主码流播放。

- 窗口缩回宫格时按相同步骤切回 `?quality=fluent`。
- 每次读取 `actualQuality`；未配置子码流时，服务端会回退主码流并在 `qualityFallback` 中说明原因。

HTTP

```
GET /api/public/v1/channels/ch-001/playback-urls?quality=fluent
Authorization: Bearer <API_TOKEN>

GET /api/public/v1/channels/ch-001/playback-urls?quality=high
Authorization: Bearer <API_TOKEN>
```

编码边界： `high` 对应通道主码流，`fluent` 对应子码流；`quality`、原有 `high/fluent` 值、响应字段和播放 URL key 均保留兼容。选择码流不会临时创建转码策略。通道主码流使用所选输入，也可来自设备子码流，不等于设备物理主码流；子码流可来自设备或由同一原始输入独立编码，不是再次转码主码流输出。两路策略与旧配置规则见下方“通道双路输出配置”。H.265 要供不兼容浏览器播放，须为对应输出启用兼容转码，不能只切换 `quality`。

可选原生 HLS 兼容： StreamGate 1.9.6 起，仅在播放端不能使用 MSE、需要原生 HLS 时，为返回的 Gateway `hls` 播放 URL 追加 `hls_compat=fmp4`；无查询参数时使用 `?`，已有参数时使用 `&`，保留全部原始 query，尤其是播放凭据。MSE / hls.js 播放器不追加此参数。它只选择完整 fMP4 分片兼容模式，不承诺 LL-HLS 低延迟，也不替代编码兼容转码；不新增 REST 资源，不改变播放凭据或验签合同。不要把它加到 `playback-urls` API 请求上。

200 OK

```
{
  "id": "ch-001",
  "enabled": true,
  "outputStreamId": "ch-001",
  "httpFlv": "http://streamgate.example:18080/live/ch-001.flv",
  "websocketFlv": "ws://streamgate.example:18080/ws/live/ch-001.flv",
  "hls": "http://streamgate.example:18080/hls/ch-001/index.m3u8",
  "webRtc": "http://streamgate.example:18080/whip/live/ch-001",
  "whip": "http://streamgate.example:18080/whip/live/ch-001/whip",
  "requestedQuality": "high",
  "actualQuality": "high",
  "fluentAvailable": true,
  "playbackTokenRequired": false
}
```

POST /api/public/v1/channels/{id}/warm

可选：短时预热来源

StreamGate 1.9.6 起提供，属于可选接口。只有页面能在用户打开播放器前准备按需通道时，才需要调用；正常播放不需要预热，常驻通道已保持来源连接。Token 需具备独立的 `playback:warm` 权限，读取播放地址本身仍不会拉流。

`quality` 查询参数应与随后请求的播放画质一致。请求体可传 `{"ttlMillis":15000}`，默认 15 秒、最长 30 秒，响应包含 `expiresAt`。只建议对用户即将打开的画面调用。

HTTP

```
POST /api/public/v1/channels/ch-001/warm?quality=high
Authorization: Bearer <API_TOKEN>
Content-Type: application/json

{"ttlMillis":15000}
```

收到成功响应后打开播放器；预热失败时仍可按正常流程播放。不要预热全部通道或在后台持续续期。预热只保持来源流，不提前生成 LL-HLS 切片，不签发播放凭据。1.9.6 之前的版本返回 404。

播放器恢复与来源参数变化

StreamGate 1.9.6 新增既有 WebHook / MQTT 路径上的可选 `channel.media_changed` 通知，以及 `GET /api/public/v1/status` 的可选 `channels[].media` 字段。旧版本可能省略 `media`，客户端仍须接受原有 `id`、`enabled`、`state` 合同；没有新增 SSE 服务器或浏览器订阅端点。通知只作补充，自建播放器仍须使用所选播放库的事件和画面推进检测恢复连接。

播放方式	恢复依据
WebRTC	<code>connectionstatechange</code> 进入 <code>failed</code> ，或媒体轨道 <code>ended</code> 。 <code>disconnected</code> 可能是短时网络波动，应先留出恢复时间。
LL-HLS / hls.js	<code>Hls.Events.ERROR</code> 的 <code>fatal=true</code> ，按错误类型恢复或重建播放器；非致命错误由播放库自动恢复，不应每次都重新连接。
HTTP-FLV / WS-FLV	监听播放库的错误及直播连接结束事件，释放旧实例后重新连接。

连接正常但播放时间持续不推进时，也应启动有界重连；暂停或页面隐藏时不要误判。采用单一重连任务、逐步增加重试间隔，并在用户关闭画面后停止重试。分辨率变化触发的 `video.resize` 只需调整布局，画面继续推进时无需重连。

设备或通道 ID 未变时可先重用原播放地址；更换画质、地址失效或播放凭据到期时，重新获取地址或凭据。H.265 原码流仍要求查看端支持该编码，重连不能替代兼容转码。可选 `warm` 接口不是重连通知或凭据刷新接口。

可选媒体变化通知 (1.9.6)

既有 WebHook 或 MQTT 端点使用 `eventTypes:["channel.media_changed"]`，即可只订阅媒体变化通知；StreamGate 1.9.6 的 `GET /api/public/v1/integrations/event-types` 会列出此精确类型。其它旧事件类型仍可使用并与它合并订阅；`eventTypes:["*"]` 仍投递全部事件，但不支持 `channel.*`。省略/留空 `eventTypes` 沿用既有默认列表，不自动加入媒体变化通知。Public 集成配置沿用 `integrations:read / integrations:write`，状态沿用 `status:read`；仍要求有效 Professional / Enterprise 软件授权及 `openapi` 权益，媒体投递前再次检查。不新增 license feature 或播放权限。

状态字段 (channels[].media[])	含义
<code>outputStreamId</code> 、 <code>quality</code>	输出身份与 high/fluent 档位；只比较相同通道/输出。
<code>current</code> 、 <code>observedAt</code>	观测有效性与 UTC 采集时间。Core 读取失败或过期时保留旧值并标记 <code>current=false</code> ，不能据此认定停流。
<code>available</code>	已观测到活动视频描述，不代表设备健康、已解码画面或全部协议就绪。
<code>mediaGeneration</code>	Gateway 不透明比较标记，不解析或排序；重启后重新建立基线。
<code>format</code>	videoCodec、width、height 与 origin (source 或 applied)；不可用时省略。不是逐协议线路实测，不包含音频适配。
<code>effectiveVideo</code>	可选编码器创建采纳值：codec、width、height、fps、bitrateKbps、keyframeIntervalMs。不是瞬时流量或画质承诺。
<code>publisherGenerations</code> 、 <code>outputGeneration</code> 、 <code>selection</code>	可选 Core 局部发布/输出代次（十进制字符串）与已选策略；局部计数可归零，比较应使用 <code>mediaGeneration</code> 。

观察者在既有串行媒体循环读取一次 Core 聚合状态，名义间隔两秒而非通知延迟承诺；不启动来源，也不逐通道探测。Public 状态只读缓存。未观测或旧 Core 可省略媒体字段；超过十秒的观测不是 `current`。每个已配置通道最多两档缓存；删除通道时移除缓存，不保证最后通知。

去重忽略瞬时流量、包/帧/队列计数及尚未采纳的请求配置。编码/尺寸、有效编码配置/实例、来源重启身份及发布/输出代次改变可产生新标记。主码流和子码流分别比较，只改子码流输出不改主码流基线；可观测的停流/恢复及同格式重启会通知。首次为 `reason=observed`，后续为 `changed`、`stopped`、`resumed`，客户端应允许新增值；初始化中或陈旧值不虚构新格式。

Core 当前有效的 `videoEncoding.instanceId` 与 `encoderGeneration` 参与内部比较；因此来源估计导致新码率被采纳时，即使编码/尺寸不变也可通知。同一聚合快照的变化每档输出合并为一个事件；分轮观测到的变化可能产生不同事件。

沿用既有 envelope 的 `eventId`、`type`、`occurredAt`、`deploymentId`、`nodeId`、`mediaResourceId`、`source`、`action`、`severity`、`message` 与字符串值 `payload`。示例：

```
JSON

{"eventId":"evt-example","type":"channel.media_changed","occurredAt":"2026-10-01T08:00:00Z","mediaResourceId":"ch-001","source":"channel","action":"media_changed","payload":{"outputStreamId":"ch-001__fluent","quality":"fluent","mediaGeneration":"media-example-2","previousMediaGeneration":"media-example-1","available":"true","reason":"changed","observedAt":"2026-10-01T08:00:00Z","videoCodec":"h264","width":"640","height":"360","formatOrigin":"applied","bitrateKbps":"400","publisherGenerations":{"hls-rtsp":{"2\"}}
```

WebHook 沿用 HMAC 头、重试与投递记录；MQTT 沿用主题与 QoS。这是尽力投递的采样通知，不是持久变更日志：两次观测间的短暂变化、队列满、接收失败和进程重启可能漏发，重试可能重复。以 `eventId` 去重并比较通道/输出标记；可选低频状态查询可对齐最新 `current` 状态，不能回补中间变化。停流通知不是调用 `warm` 或重启来源的命令；变化/恢复时更新播放器信息，仅在播放需要时启动单一有界恢复任务，不因码率配置或尺寸改变就重连仍在推进的播放器。

新发布代次可能结束已有连接（例如 RTSP EOF），需要重连；通知不证明解码无损，不能把非 EOF 解码损坏当作预期的代次切换而忽略。

13. 设备与通道管理

以下接口适用于 StreamGate 1.9.1 及以上 Professional / Enterprise。写入请求可以提交设备凭据，响应会移除密码、密码引用、密钥、URL 用户信息和敏感查询参数。

方法与路径	Scope	用途
POST /devices	devices:write	创建设备
GET /devices/{id}	devices:read	查询单个设备配置
PUT /devices/{id}	devices:write	更新设备
DELETE /devices/{id}	devices:write	删除设备
POST /devices/batch	devices:write	批量 enable、disable 或 delete
POST /devices/onvif/discover	devices:write	发现 ONVIF 设备
POST /devices/onvif/resolve-stream	devices:write	解析 ONVIF 码流
POST /channels	channels:write	创建通道

方法与路径	Scope	用途
GET /channels/{id}	channels:read	查询通道配置
PUT /channels/{id}	channels:write	更新通道
DELETE /channels/{id}	channels:write	删除通道
POST /channels/batch	channels:write	批量 enable、disable、ondemand、always_on 或 delete
POST /channels/{id}/ptz	ptz:write	执行云台控制

创建设备示例

```
HTTP

POST /api/public/v1/devices
Authorization: Bearer <API_TOKEN>
Content-Type: application/json

{
  "name": "东门摄像机",
  "vendor": "generic",
  "protocol": "onvif",
  "address": "192.0.2.20",
  "username": "operator",
  "passwordRef": "设备密码",
  "enabled": true
}
```

通道双路输出配置

通道创建与更新使用 `channels:write`。根级 `output` 是主码流策略；可选 `viewing.fluent.output` 是子码流独立策略。两路均不修改设备编码，旧配置缺失字段时按下表保留原语义。

字段	含义
<code>viewing.defaultQuality</code>	仅兼容未传 <code>quality</code> 的 Public API 播放地址请求，不用于控制台预览/详情或大屏。沿用 <code>auto/high/fluent</code> ；high 为主码流，fluent 为子码流；保存 auto/fluent 仍要求已配置子来源，显式 query quality 覆盖它。通道 UI 已移除此选择。
<code>viewing.fluent.source</code>	<code>profile</code> ：设备子码流 Profile，另传 <code>profileToken</code> ； <code>url</code> ：设备子码流地址，另传 <code>sourceUrl</code> ； <code>output</code> ：从主码流地址的原始输入转码，不传上述来源字段。

字段	含义
<code>output.mode</code> <code>viewing.fluent.output.mode</code>	<code>""</code> / 省略: 跟随系统; <code>source</code> : 原码流; <code>compatible</code> : 兼容转码; <code>custom</code> : 始终转码。主码流与设备子码流均可选四种; 生成子码流只允许 <code>compatible/custom</code> 。界面名称变化不改 <code>custom</code> 值。
各路 <code>video</code>	<code>compatible/custom</code> 可设 <code>bitrateKbps</code> 、 <code>qualityPreset</code> 、 <code>automaticPolicy</code> 、 <code>automaticBitrate</code> 、 <code>hevcBitrateReservePercent</code> 、 <code>maxWidth/maxHeight</code> 、 <code>maxFps</code> 、 <code>keyframeIntervalMs</code> (毫秒) 和 <code>encoderPreference</code> (旧字段保留, 实际使用全局后端)。 <code>compatible</code> 仅将 H.265 转 H.264, H.264 直出时不使用这些参数; <code>source</code> /跟随系统不传视频覆盖参数。
<code>video.automaticPolicy</code>	可选的 <code>economy/balanced/quality</code> 对象, 各含 <code>sourcePercent</code> 、 <code>referenceBitrateKbps</code> 、 <code>referenceCeilingKbps</code> ; 有值的新策略优先旧强度。默认值、校验及整对象继承见下文。
<code>video.hevcBitrateReservePercent</code>	可选整数 <code>0..200</code> , 仅用于 H.265 转 H.264 的自动码率。省略继承系统, 显式 <code>0</code> 覆盖为不预留; 独立于 <code>automaticPolicy</code> , 系统缺省为 <code>35</code> 。
<code>video.automaticBitrate</code>	旧兼容对象, 含 <code>economyPercent</code> 、 <code>balancedPercent</code> 、 <code>qualityPercent</code> , 各为整数 <code>0</code> 或 <code>50..150</code> 。百分比调整内置自动目标, 不是固定 kbps 或画质分数; 整对象继承规则见下文。

“从主码流地址转码”与主码流共用原始输入并独立编码, 不串级再次编码。编辑页新建该子码流时默认 `custom`, 保持来源尺寸和帧率、档位跟随系统; 下面示例明确设置为 640 x 360、15 fps、`economy`, 作为通道请求体片段, 更新时应保留其余字段。尺寸是等比缩小上限, 帧率是上限, 不放大或补帧; 主码流策略不因子策略更改而被覆盖。

JSON

```
{
  "output": {"mode": "compatible", "video": {"qualityPreset": "balanced", "hevcBitrateReservePercent": 0}},
  "viewing": {
    "defaultQuality": "auto",
    "fluent": {
      "source": "output",
      "output": {
        "mode": "custom",
        "video": {"qualityPreset": "economy", "maxWidth": 640, "maxHeight": 360, "maxFps": 15}
      }
    }
  }
}
```

此例主码流显式关闭 H.265 码率预留; 子码流省略该字段而继承系统。不改变两路输出模式或播放选择。

配置	主码流策略	子码流策略
profile/url, 缺失 fluent.output	根 output	沿用根 output
profile/url, 显式 fluent.output={}	根 output	独立跟随系统
output 来源, 缺失 fluent.output	跟随系统	根 output (旧特殊规则)
output 来源, 显式 fluent.output	根 output	fluent.output, 须为 compatible/custom

缺失字段不等于空对象。读取不会迁移旧配置；前端首次编辑保存才显式转换，并保留此前两路语义：设备子流把原根策略保存为子策略；旧生子流把原根策略移到子策略，主码流策略保存为跟随系统。旧客户端字段与播放地址仍可使用。

通道自动码率与按需模式

两路转码的 `video.qualityPreset` 可为 `""`、`"economy"`（节省带宽）、`"balanced"`（均衡）、`"quality"`（优先画质）。省略或空值跟随系统 `media.videoQualityPreset`（默认均衡），各路显式档位覆盖系统；正数手动 `video.bitrateKbps` 优先并忽略自动策略与 H.265 预留。

在“系统设置 > 媒体与播放”的“自动转码”选择节省带宽、均衡或优先画质。旁边齿轮打开“转码设置”，分别调整三档画质强度、可选码率上限和共用关键帧间隔；全局编码后端在下方“播放与资源高级设置”选择。

自动档位不改变来源分辨率或帧率。需要缩小画面或降低帧率时，单独设置通道主/子输出的宽度、高度和帧率。主/子选择“跟随系统”时分别继承全局转码模式和档位，不会自动创建未配置的子码流。

档位	默认画质强度	x264 CRF	Intel QSV QVBR 质量值
节省带宽 <code>economy</code>	4	28	28
均衡 <code>balanced</code>	6	25	25
优先画质 <code>quality</code>	8	22	22

画质强度越高保留的细节越多，通常需要更多码率。强度不是编码器原生值：x264 使用 CRF+VBV；QSV 在支持时使用 QVBR；NVENC/AMF 使用其支持的质量优先 VBR。OpenH264 等缺少相应质量模式的后端采用码率预算降级，不能视为与 CRF 等价。不同后端的数值不代表相同画质，也不保证恢复来源已损失的细节。

三档保存在 `media.videoTranscodeParameters.automaticPolicy` 的 `economy/balanced/quality` 中。每档 `qualityLevel` 为整数 `1..10`，`0`/省略使用默认值；三档须依次不递减。可选 `maxBitrateKbps` 为 `0` 或整数 `256..64000` kbps，`0`/留空由系统估算预算；设置上限不是固定码率，也不是含音频、封装和重传的线路硬上限。后端不支持质量模式时，实际输出显示 `bitrate-fallback`。

通道各路 `video.automaticPolicy` 省略、为空或全零时继承整组系统策略，有值时整组覆盖，缺项使用内置值。通道正数 `bitrateKbps` 使用手动码率并忽略自动档位。全局编码后端对所有转码输出生效，旧通道 `encoderPreference` 仍可读取但不覆盖系统选择；通道关键帧间隔为 `0` 时使用全局值，全局默认 1000 ms。

设备 H.264 码流在“兼容转码”下保持直出，不使用编码参数；H.265 才转为 H.264。“从主码流地址转码”本身就是生成独立编码输出，即使主码流为 H.264 也会编码。两种来源的处理规则不同，不会把主流编码结果再次转码。

兼容保留旧 `sourcePercent/referenceBitrateKbps/referenceCeilingKbps`、`automaticBitrate` 和 `hevcBitrateReservePercent`，它们只参与内部预算/降级，不是画质强度。预算结合来源实测码率、实际输出尺寸和帧率计算；未测得时使用规格估计。自动策略内的旧 `maxWidth/maxHeight/maxFps` 不再改变画面，只有通道 `video` 的明确尺寸和帧率生效。

取消不改页面，应用只改草稿，保存系统设置后生效。无修改应用不制造未保存状态。只调整共用编码参数保留旧固定码率，编辑自动策略或选择“改用自动码率”才切回自动。实际采用后端、率控模式和质量值以运行状态为准，不新增 Public API 系统设置接口。

旧 `automaticBitrate` 保留兼容，仅在继承解析后没有有值的新策略时使用。各分量 `0` 或省略为内置 `100%`，显式 `100` 保留为覆盖。通道整个对象未提供、为空或全零时继承整个系统强度对象；非零对象整覆盖，缺省/零分量取内置 `100%`，不逐字段继承。例如 `{"balancedPercent":100}` 可覆盖旧系统的 90 或 120，其余档位为内置 `100%`。调整后的目标须保持节省带宽、均衡、优先画质顺序，反转组合会被拒绝；该百分比不是固定 kbps 或画质分数，不再作为当前系统齿轮 UI 的参数。

兼容模式仅在 H.265 转码时使用参数；H.264 直出和原码流不使用自动策略，也不因档位而转码。保存全局档位或策略可能短时重连实际采用变化值的活动输出，不覆盖通道显式策略或手动码率；未使用档位的变化不重建当前输出。参数不保证线路带宽或原画质，不新增 `scope` 或授权权益；保存成功不代表编码器已采用，可用时应读取 `current` 媒体状态。

`onDemand=true` 或批量 `{"action":"ondemand","ids":["ch-001"]}` 对应“按需拉流已开启”：存在预览、分发、录像、分析或 GB28181 消费者时拉流，空闲后可停流；`onDemand=false` / `always_on` 对应“按需拉流已关闭”：通道启用后保持输入连接。API 字段和值不变；正在运行的任务仍可能保持来源连接。

14. 录像管理

方法与路径	Scope	用途
<code>GET /recording/tasks</code>	<code>recording:read</code>	查询录像任务
<code>POST /recording/tasks</code>	<code>recording:write</code>	创建录像任务
<code>GET /recording/tasks/{id}</code>	<code>recording:read</code>	查询单个录像任务
<code>PUT /recording/tasks/{id}</code>	<code>recording:write</code>	更新录像任务
<code>DELETE /recording/tasks/{id}</code>	<code>recording:write</code>	删除录像任务
<code>GET /recording/policy</code>	<code>recording:read</code>	查询默认录像策略
<code>PUT /recording/policy</code>	<code>recording:write</code>	更新默认录像策略
<code>POST /recording/policy/apply</code>	<code>recording:write</code>	将策略应用到指定通道或全部通道

1.9.6 可按两级选择录像目标：先选主路或子码流，再选该路原始数据或处理后数据。

字段	值	含义
streamRole	default / secondary	主路 / 子码流
mediaSource	source / output	所选路的原始数据 / 处理后数据

设备独立子码流支持两种数据来源；从主码流地址转码生成的子码流只支持 secondary/output 。未配置子码流时不可选择 secondary ，无效组合返回 400 ，不会自动改选主路。切换目标不改变录像开关、计划或保留设置，已生成片段的来源不随当前配置改写。

旧任务省略 streamRole 时保留主路选择； mediaSource 省略或为空时保留原有处理策略。修改旧任务时应保留查询返回的未修改字段。

存储目录：录像目录由 StreamGate 所在服务器解析并校验，不能填写第三方调用方计算机的本地路径。配置了录像根目录时，任务目录必须位于该根目录内。

15. GB28181 管理

三个资源组均提供集合 GET/POST 和单项 GET/PUT/DELETE 。读操作使用 gb28181:read ，写操作使用 gb28181:write 。

资源路径	用途
/gb28181/platforms	本机 GB28181 服务平台配置
/gb28181/exposure-bindings	将 StreamGate 通道发布到国标目录
/gb28181/upstreams	配置向上级平台注册与级联

完整请求字段以随文档交付的 OpenAPI YAML 为准。平台或上联密码可在写入时提交，但响应中不会返回 passwordRef 。

1.9.6 可在向上级联配置中统一选择供流方式，所有共享通道使用同一策略，不需要逐通道设置。

供流方式	字段	上级点播
单码流·主路	mediaStreamMode=single 、 singleStreamRole=default	提供通道主路输出
单码流·子码流	mediaStreamMode=single 、 singleStreamRole=secondary	提供已配置的子码流输出

供流方式	字段	上级点播
主/子双码流	<code>mediaStreamMode=dual</code>	未带码流编号或编号 0 取主路，编号 1 取子码流

单码流模式下，合法的主/子请求均提供所选一路。未配置子码流的通道在上述策略下提供主路，不临时生成子码流；其他编号会被拒绝。旧配置未设置这两个字段时保留原有处理。此设置只控制实时级联，不改变录像选源或转码参数。

16. WebHook、MQTT 与投递日志

以下是 REST API 管理接口；事件按已配置的 WebHook 接收地址或 MQTT broker/topic 投递，方向与鉴权见文档开头。

方法与路径	Scope	用途
<code>GET /integrations/event-types</code>	<code>integrations:read</code>	查询可订阅事件类型
<code>GET /integrations/webhooks</code>	<code>integrations:read</code>	查询 WebHook
<code>POST /integrations/webhooks</code>	<code>integrations:write</code>	创建或更新 WebHook
<code>DELETE /integrations/webhooks/{id}</code>	<code>integrations:write</code>	删除 WebHook
<code>POST /integrations/webhooks/{id}/test</code>	<code>integrations:write</code>	发送测试事件
<code>GET /integrations/mqtt</code>	<code>integrations:read</code>	查询 MQTT 配置
<code>PUT /integrations/mqtt</code>	<code>integrations:write</code>	更新 MQTT 配置
<code>POST /integrations/mqtt/test</code>	<code>integrations:write</code>	发送测试消息
<code>GET /integrations/deliveries</code>	<code>integrations:read</code>	按端点、类型和状态查询投递日志
<code>DELETE /integrations/deliveries</code>	<code>integrations:write</code>	清空投递日志
<code>GET /events</code>	<code>exports:read</code>	按级别、来源、动作或通道查询运行事件

WebHook 请求体

```
{
  "name": "业务系统告警通知",
  "enabled": true,
  "url": "https://ops.example.com/streamgate/events",
  "eventTypes": ["alarm.*", "recording.error", "channel.stream_error"],
  "hmacSecret": "由双方安全保存的签名密钥",
  "timeoutSeconds": 5,
  "maxAttempts": 3
}
```

`hmacSecret` 和 MQTT `password` 只用于写入，查询和保存响应不会返回明文。测试接口会执行真实网络投递，联调时应使用专用接收端。

17. 错误处理

错误响应使用 JSON。调用方应同时判断 HTTP 状态码和 `error` 字段，不要依赖完整错误文本进行业务分支判断。

错误响应

```
{
  "error": "api token scope denied",
  "scope": "alarms:write"
}
```

状态码	含义	处理建议
400	参数或 JSON 请求体无效	检查参数类型、分页范围和 JSON 格式，不应原样重试
401	缺少 Token，或 Token 无效、停用、已撤销	检查 Authorization 请求头；必要时重新创建 Token
403	授权权益或 Token scope 不足	检查 StreamGate 授权状态和 Token 权限
404	接口、通道或告警不存在	核对版本、资源 ID 和经过 URL 编码的路径参数
405	HTTP 方法不允许	核对 GET / POST 方法
500	服务端处理失败	记录请求时间与响应摘要，查看 StreamGate 运行日志

重试建议

- GET 请求遇到网络错误或 `5xx` 时，可采用指数退避重试，例如 1 秒、2 秒、4 秒，设置最大次数。
- 不要对 `400`、`401`、`403`、`404` 进行无条件重试。
- 告警确认和清除属于状态变更操作。调用超时后，应先重新查询告警状态，再决定是否重试。

18. 安全建议

- 生产环境通过受信网络访问 StreamGate；跨网络访问时使用 HTTPS 反向代理或安全隧道。
- 为不同第三方系统分别创建 Token，避免共用。
- 只授予业务实际需要的 scope，定期撤销不再使用的 Token。
- Token 仅保存在服务端密钥配置中，不提交到代码仓库，不写入浏览器脚本。
- 调用日志中记录 Token 前缀即可，不记录完整 Authorization 请求头。
- 录像路径、告警扩展字段和播放地址可能包含部署信息，应按业务敏感数据管理。

19. 联调检查单

1. 网络连通

第三方服务器可以访问 StreamGate 的 Web 地址和端口。

2. 授权有效

授权页面显示 Professional / Enterprise 或相应有效 API 权益。

3. Token 已保存

完整 Token 已写入第三方系统的安全配置。

4. Scope 正确

Token 具备目标接口要求的最小权限。

5. 基础请求通过

`GET /channels` 返回 `200`。

6. 版本能力匹配

1.9.1 新增接口未被用于 1.8.2 实例。

7. 异常处理完成

客户端正确处理超时、401、403、404 和 5xx。

8. 凭据未泄露

请求日志、截图和配置导出中没有完整 Token。

20. 配套文件

运行中的 StreamGate 可通过 `GET /api/public/v1/openapi.json` 返回实例自带的机器可读接口描述。该描述适合工具导入和版本探测，中文接入说明以本文档为准。

Professional / Enterprise 发布包固定包含 OpenAPI YAML。Standard 包不包含 API 文档。Postman 集合仅在客户明确需要联调时单独提供；HTML 是文档维护和在线阅读版本，不进入运行包。

文件	交付方式	用途
<code>streamgate-public-api-v1.openapi.yaml</code>	Professional / Enterprise 发布包	OpenAPI 3.0 接口定义，可导入 Swagger Editor、Postman 或代码生成工具
<code>streamgate-public-api.postman_collection.json</code>	按需联调附件	Postman 请求集合，使用 <code>baseUrl</code> 和 <code>apiToken</code> 变量