

MQTT 消息规则

1. Topic 规范

Topic 格式：

```
snlp/v1/{device_id}/{direct}/{msg_type}
```

1.1 字段说明

字段	说明
snlp	固定前缀
v1	协议版本，当前为 v1
{device_id}	设备 ID，统一使用设备 UUID，不可修改
{direct}	消息方向，取值为 rpt 或 cmd
{msg_type}	消息类型，如 state、event、log、ack、operate、media

1.2 消息方向

direct	方向	说明
rpt	设备上报到服务端	用于状态、事件、日志、ACK 等上报
cmd	服务端下发到设备	用于操作命令、OTA 等下发

1.3 Topic 列表

方向	Topic	说明
rpt	snlp/v1/{device_id}/rpt/state	设备状态上报，周期为 1 秒
rpt	snlp/v1/{device_id}/rpt/event	事件上报，如 ASR 识别结果
rpt	snlp/v1/{device_id}/rpt/log	日志上报
rpt	snlp/v1/{device_id}/rpt/ack	命令执行结果 ACK 上报
cmd	snlp/v1/{device_id}/cmd/operate	操作类命令下发，如播放、停止、下载、音量、闹钟、重启等
cmd	snlp/v1/{device_id}/cmd/media	OTA 任务下发，仅用于 OTA_UPDATE

1.4 设备订阅要求

设备端必须分别订阅以下两个 Topic：

```
snlp/v1/{device_id}/cmd/operate
snlp/v1/{device_id}/cmd/media
```

其中 `{device_id}` 为设备 UUID。

1.5 服务端订阅建议

服务端如需接收所有设备上报，可订阅：

```
snlp/v1/+/rpt/#
```

2. Payload 基本规范

Payload 使用 JSON 格式，并包含统一 Header。

2.1 编码约定

- Payload 统一使用 JSON 格式。
- 字符串字段使用双引号。
- 时间字段统一使用 Unix 时间戳，单位为秒。
- 需要关联请求和响应时，使用 `req_id`。
- 设备 ID 统一使用设备 UUID。

3. 服务端下发命令格式

适用 Topic：

```
snlp/v1/{device_id}/cmd/operate
snlp/v1/{device_id}/cmd/media
```

消息格式：

```
{
  "header": {
    "req_id": "UUID-xxxx-xxxx",
    "device_id": "device-uuid",
    "timestamp": 1718365200,
    "ttl": 300
  },
  "action": "ACTION_TYPE",
  "payload": {}
}
```

3.1 Header 字段

字段	类型	必填	说明
req_id	string	是	请求 ID，用于关联命令与 ACK
device_id	string	是	目标设备 UUID，应与 Topic 中 {device_id} 一致
timestamp	number	是	Unix 秒级时间戳
ttl	number	是	命令有效期，单位为秒

3.2 TTL 过期规则

设备收到命令后，应按以下规则判断命令是否过期：

当前设备时间 > timestamp + ttl

如果命令已过期，设备直接丢弃。

3.3 命令字段

字段	类型	必填	说明
action	string	是	命令类型，如 PLAY_MEDIA、SYNC_CLOCK、SET_ALARM
payload	object	是	命令参数，不同 action 对应不同结构

3.4 命令 Topic 归属

Action	下发 Topic	说明
SET_VOLUME	snlp/v1/{device_id}/cmd/operate	设置音量
SET_POWER	snlp/v1/{device_id}/cmd/operate	设置电源状态
SET_CLOCK	snlp/v1/{device_id}/cmd/operate	设置设备时钟
PLAY_MEDIA	snlp/v1/{device_id}/cmd/operate	播放 SD 卡本地媒体
STOP_MEDIA	snlp/v1/{device_id}/cmd/operate	停止播放媒体
DOWNLOAD	snlp/v1/{device_id}/cmd/operate	下载文件到 SD 卡
DELETE_FILE	snlp/v1/{device_id}/cmd/operate	删除 SD 卡文件
UPDATE_MANIFEST	snlp/v1/{device_id}/cmd/operate	更新资源清单
SET_ALARM	snlp/v1/{device_id}/cmd/operate	设置闹钟
REBOOT	snlp/v1/{device_id}/cmd/operate	延迟重启
OTA_UPDATE	snlp/v1/{device_id}/cmd/media	OTA 升级

4. 设备上报消息格式

适用 Topic:

```
snlp/v1/{device_id}/rpt/state
snlp/v1/{device_id}/rpt/event
snlp/v1/{device_id}/rpt/log
snlp/v1/{device_id}/rpt/ack
```

消息格式:

```
{
  "header": {
    "device_id": "device-uuid",
    "timestamp": 1718365205
  },
  "msg_type": "state",
  "payload": {}
}
```

4.1 Header 字段

字段	类型	必填	说明
device_id	string	是	设备 UUID, 应与 Topic 中 {device_id} 一致
timestamp	number	是	Unix 秒级时间戳

4.2 上报字段

字段	类型	必填	说明
msg_type	string	是	上报类型, 如 state、event、ack、log
payload	object	是	上报内容

5. Action 列表

Action	Payload 示例	说明
SET_VOLUME	{ "level": 50 }	设置系统音量, 范围 0-100
SET_POWER	{ "state": "sleep/wake" }	设置电源状态, 如休眠或唤醒
SET_CLOCK	{ "ts": 1718365200 }	设置设备系统时间和 RTC 时间

Action	Payload 示例	说明
PLAY_MEDIA	<code>{ "path": "/data/unit1/lesson1.mp3", "volume": 60, "loop": false }</code>	播放 SD 卡本地媒体文件，支持循环播放
STOP_MEDIA	<code>{}</code>	停止播放媒体
DOWNLOAD	<code>{ "url": "...", "target_path": "/data/unit1/lesson1.mp3", "md5": "..."} </code>	下载文件到 SD 卡
DELETE_FILE	<code>{ "path": "/data/unit1/lesson1.mp3" }</code>	删除 SD 卡文件
UPDATE_MANIFEST	<code>{ "course_id": "...", "assets": [...] }</code>	更新资源清单
SET_ALARM	<code>{ "id": "alarm_01", "time": "07:00:00", "repeat": [1, 2, 3, 4, 5], "action": "add" }</code>	添加或删除闹钟
OTA_UPDATE	<code>{ "url": "...", "version": "1.0.2" }</code>	OTA 升级
REBOOT	<code>{ "delay": 5 }</code>	延迟重启，单位为秒

更新资源清单中的 `assets` 为 JSON 数组，终端收到后存入本地文件，实现资源的管理与对比。具体每个成员的格式待定。

6. 各 Action 详细说明

6.1 SET_VOLUME

Payload:

```
{  
  "level": 50  
}
```

字段	类型	必填	说明
level	number	是	系统音量，范围 0-100

6.2 SET_POWER

Payload:

```
{  
  "state": "sleep"  
}
```

字段	类型	必填	说明
state	string	是	sleep 表示进入待机；wake 表示待命状态

6.3 SET_CLOCK

Payload:

```
{
  "ts": 1718365200
}
```

字段	类型	必填	说明
ts	number	是	Unix 秒级时间戳

6.4 PLAY_MEDIA

Payload:

```
{
  "path": "/data/unit1/lesson1.mp3",
  "volume": 60,
  "loop": false
}
```

字段	类型	必填	说明
path	string	是	媒体文件业务路径，例如 /data/xx.mp3
volume	number	否	目标音频播放音量，范围 0-100。仅对本次播放生效，不覆盖系统音量设置
loop	boolean	否	是否循环播放，默认 false

6.5 STOP_MEDIA

Payload:

```
{}
```

6.6 DOWNLOAD

Payload:

```
{
  "url": "https://example.com/unit1/lesson1.mp3",
  "target_path": "/data/unit1/lesson1.mp3",
  "md5": "...
}
```

字段	类型	必填	说明
url	string	是	下载地址

字段	类型	必填	说明
target_path	string	是	目标业务路径，例如 /data/xx.mp3
md5	string	否	文件 MD5，用于下载后校验

设备应根据 target_path 自动创建必要的目录。

6.7 DELETE_FILE

Payload:

```
{
  "path": "/data/unit1/lesson1.mp3"
}
```

字段	类型	必填	说明
path	string	是	待删除文件业务路径，例如 /data/xx.mp3

6.8 UPDATE_MANIFEST

Payload:

```
{
  "course_id": "course_001",
  "assets": []
}
```

字段	类型	必填	说明
course_id	string	是	课程或资源包 ID
assets	array	是	资源列表，成员格式待定

6.9 SET_ALARM

Payload:

```
{
  "id": "alarm_01",
  "time": "07:00:00",
  "repeat": [1, 2, 3, 4, 5],
  "action": "add"
}
```

字段	类型	必填	说明
id	string	是	闹钟 ID 名称，由服务端指定，设备端需保存并用于后续删除或更新

字段	类型	必填	说明
<code>time</code>	string	<code>add</code> 时必填	闹钟时间，支持 <code>HH:MM:SS</code>
<code>repeat</code>	array	<code>add</code> 时必填	星期重复规则，取值范围 <code>1-7</code> ，对应周一到周日
<code>action</code>	string	是	<code>add</code> 表示添加或更新； <code>del</code> 表示删除

`repeat` 映射规则：`1` 表示周一，`2` 表示周二，`3` 表示周三，`4` 表示周四，`5` 表示周五，`6` 表示周六，`7` 表示周日。

6.10 OTA_UPDATE

Payload:

```
{
  "url": "https://example.com/firmware.bin",
  "version": "1.0.2"
}
```

字段	类型	必填	说明
<code>url</code>	string	是	固件下载地址
<code>version</code>	string	否	目标固件版本

OTA 升级开始前，设备通过 `rpt/ack` 返回任务已开始的 ACK。OTA 下载和写入进度接近完成、设备即将重启前，设备通过 `rpt/ack` 返回完成 ACK。

6.11 REBOOT

Payload:

```
{
  "delay": 5
}
```

字段	类型	必填	说明
<code>delay</code>	number	否	延迟重启时间，单位为秒，默认立即重启

7. 设备状态上报

适用 Topic:

```
snlp/v1/{device_id}/rpt/state
```

状态上报周期为 1 秒。

Payload 示例:

```
{
  "cpu": 15,
  "mem": 45,
  "battery": 80,
  "net": "wifi",
  "mode": "待命"
}
```

7.1 状态字段

字段	类型	必填	说明
cpu	number	是	CPU 使用率，单位为百分比
mem	number	是	内存使用率，单位为百分比
battery	number	是	电池电量，范围 0-100
net	string	是	网络类型，如 wifi
mode	string	是	设备模式，枚举见下表

7.2 mode 枚举

mode	说明
使用中	设备正在交互、播放或执行任务
待命	小智等待唤醒
待机	设备断开 Wi-Fi，进入低功耗模式

设备进入待机前，应发送一次状态上报心跳包，mode 为 待机。

8. 事件上报

适用 Topic:

```
snlp/v1/{device_id}/rpt/event
```

8.1 ASR 识别结果上报

当设备获得识别结果文本时，可通过事件上报发送给服务端。

Payload 示例:

```
{
  "event": "asr_result",
  "text": "打开音乐"
}
```

字段	类型	必填	说明
event	string	是	事件类型，ASR 识别结果固定为 asr_result
text	string	是	识别结果文本

confidence 字段舍弃，不需要上报。

9. ACK 上报

适用 Topic:

```
snlp/v1/{device_id}/rpt/ack
```

ACK 用于上报命令执行结果或任务进度。

Payload 示例:

```
{
  "req_id": "UUID-xxxx-xxxx",
  "status": "success",
  "progress": 100,
  "error_code": 0
}
```

9.1 ACK 字段

字段	类型	必填	说明
req_id	string	是	原命令的请求 ID
status	string	是	执行状态，取值为 success 或 failed
progress	number	否	进度，范围 0-100
error_code	number	否	错误码，成功时为 0

9.2 ACK 简化规则

ACK 状态只需要 success 和 failed 两种。

总体原则:

```
命令未被设备接受：不返回 ACK
命令已被设备接受并执行成功：返回 success ACK
命令已被设备接受但执行过程中失败：返回 failed ACK
```

ACK 中应包含原命令的 req_id，用于服务端关联命令和执行结果。

9.3 不返回 ACK 的场景

以下场景设备不返回 ACK：

场景	说明
JSON 格式错误	设备无法可靠解析命令内容，可能无法获取 req_id
缺少 req_id	ACK 无法关联原始请求
不支持的 action	设备忽略不支持的命令
必填参数缺失或参数类型错误	命令未被设备接受
TTL 过期	设备直接丢弃过期命令

9.4 返回 failed ACK 的场景

以下场景表示命令已被设备接受，但执行过程中失败，设备应返回 failed ACK：

Action	failed 场景示例
DOWNLOAD	URL 无法访问、HTTP 状态码异常、网络中断、下载超时、SD 卡写入失败、SD 卡空间不足、MD5 校验失败
OTA_UPDATE	固件 URL 下载失败、固件校验失败、OTA 写入失败、固件版本不匹配、固件格式错误
PLAY_MEDIA	指定文件不存在、SD 卡未挂载、文件格式不支持、解码器初始化失败、播放器任务启动失败
DELETE_FILE	文件不存在、路径非法、文件被占用、SD 卡不可写、删除操作失败
SET_ALARM	闹钟数量超过上限、NVS 保存失败、RTC 写入失败
SET_CLOCK	系统时间设置失败、DS3231 RTC 写入失败
REBOOT	延迟重启任务无法创建或当前状态禁止重启

下载类任务可通过 rpt/ack 上报任务执行进度，progress 表示进度，范围为 0-100。

10. 下载与进度上报约定

- 下载类 Payload 包含 url、md5、target_path 等字段。
- target_path 使用本地路径
- 设备通过 rpt/ack 上报任务执行进度。
- progress 表示进度，范围为 0-100。
- ACK 中应包含原命令的 req_id，用于服务端关联命令和执行结果。

示例：

```
{  
  "req_id": "UUID-xxxx-xxxx",  
  "status": "success",  
  "progress": 100,  
  "error_code": 0  
}
```