

传输层协议

1. 概要

小爱 AIVS 协议的终端设备通过 **websocket** 与服务器进行通信，设备与服务端始终保持连接状态，服务端通过此连接可主动向设备端下发指令。

采用 **websocket** 通讯协议，客户端和服务端只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输，能更好的节省服务器资源和带宽。

本页面将介绍如何建立、保持连接。

2. 服务连接

在客户端，**websocket** 实例化一个新的 **websocket** 客户端对象，请求类似服务端的 **websocket** URL，客户端 **websocket** 对象会自动解析并识别为 **websocket** 请求，并连接服务端端口，完成服务端和客户端的通讯连接。

使用标准的 **websocket**，客户端和服务端在首次建立连接时，需要交换 **Header** 的数据。在通过第一个 **request** 建立 **TCP** 连接之后，之后交换的数据都不需要发送 **Header** 就能交换数据。

客户端发送数据格式如下：

```
GET /speech/v1.0/access HTTP/1.1

Upgrade: websocket

Connection: Upgrade

Host: speech.ai.xiaomi.com

Authorzation: MIOT-TOKEN-V1 app_id:xxx,access_token:yyy

Sec-WebSocket-Key: hj0eNqbhE/A0GkBXDRrYYw==

Sec-WebSocket-Version: 13
```

说明：

Upgrade:websocket 参数值表明这是 **websocket** 类型请求。

Sec-WebSocket-Key 是 **websocket** 客户端发送的一个 **base64** 编码的密文，要求服务端必须返回一个对应加密的 **Sec-WebSocket-Accept** 应答，否则客户端会抛出 **Error during WebSocket handshake** 错误，并关闭连接。

Authorization 是必须字段，用于鉴权使用，服务器通过鉴权才能完成连接，进行后续通信（开发者集成 SDK 时需配置鉴权信息，SDK 内部会自动生成 **Authorization** 并与服务器进行交互）。
服务端收到报文后返回的数据格式类似：

```
HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: K7DJLdLooIwIG/M0pvWFB3y3FE8=
```

说明：

Sec-WebSocket-Accept 的值是服务端采用与客户端一致的密钥计算出来后返回客户端的。

HTTP/1.1 101 Switching Protocols 表示服务端接受 **websocket** 协议的客户端连接，经过这样的请求响应处理后，两端的 **websocket** 连接握手成功，后续就可以进行 TCP 通讯了。

3. 服务保持

为了保持物理连接不断，设备端会每隔一段时间便向服务器发送一次 **Ping**，若服务器正常返回 **Pong** 消息，则物理连接保持；否则设备端应该立即关闭当前的物理连接，建立新的连接。