# 统一指令集，多种通信协议，为各种场景优化

[English](https://docs.qq.com/doc/DS3lxV2F3UWtnQmlG)

1. 元宇宙通信协议概述

元宇宙通信协议是一套简单易用且稳定可靠的通信协议。可通过元宇宙通信协议获取信息或传输数据等工作。在元宇宙通信协议的支持下，拥有更丰富的应用场景。

最新更新：

元宇宙基础API接口（可通过此服务器获取其他节点信息，遍历元宇宙）：

<http://sg.zyinfo.pro:8081/api>

API服务器开源代码：

<https://gitee.com/hayoou/metaverse_api_server_go>

[元宇宙通信协议 C# 实现](https://gitee.com/hayoou/meta_api_protocal_c_sharp)

特别接口，支持文本和基本json输出：

任意接口发送 get\_meta\_api\_info:text 或者 ｛"get\_meta\_api\_info":"json"｝或url加入后加入查询字段 &get\_meta\_api\_info=text或json，即可获取基本API文本或json格式信息，示例 2.5.2 章节 0x00指令内容：

meta\_api\_ver: 支持的元宇宙协议版本,id:meta\_example\_id, 自定义id或元宇宙统一对象idname:对象名称,meta\_api\_class\_name：元宇宙统一对象分类名称,meta\_api\_class\_id：元宇宙统一对象类型id,info\_url:介绍链接\*,api\_info: API 接口简述（web页面，人类可读）,api\_url: API 接口\*（URL形式，可选）,...以上换行符统一为\n，json返回的为上述字段的json格式。若需要屏蔽转发数据中包含的上述字符串或者只响应近期的交互，建议有时间同步的机器只响应带时间戳（string类型），或者时间戳自增的上述指令。如 get\_meta\_api\_info:text,time\_stamp:111111111,CRC\_16:AB12，同样，建议请求方携带上述信息。

通信协议默认为二进制数据流传输，如需JSON形式打包协议发送，使用下方示例：msg\_base64（为协议包二进制流的base64编码）须实现解析支持，DATA\_base64为数据段的base64编码，发送方可选择msg\_base64或下方字段发送。精简版与标准版一致。

{  
 "meta\_api\_ver": "1.0",  
 "msg\_base64": "",  
  
 "SOF": 59994,  
 "Version": 16, ...  
 "DATA\_base64": "" ...  
  
}

元宇宙通信协议的数据包格式有两种，一种是精简版的，版本号为0~3，一种是标准格式，版本号>=16。可以根据实际需要，选择相应版本。标准版适用于高质量信道环境下，大规模数据传输。协议处理建议使用状态机进行接收逻辑控制。

注：未特别指定，协议内数据低字节在前

# 2. 元宇宙通信协议描述

## 2.1 精简版协议数据格式

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOF  起始帧头 | Ver\_Length  版本/帧长度 | SEQ  序列号 | CmdSet\*  指令集 | CmdID\*  指令ID | DATA  数据 | CRC-16/32  16位校验 |
| 1-Byte | 1~2-Byte | 0~2-Byte | 0~2-Byte | 0~2-Byte | n-Byte | 2~4-Byte |

## 2.2 精简版协议字段说明

|  |  |  |
| --- | --- | --- |
| 域 | 大小 | 描述 |
| SOF  起始帧头 | 1 | 发送方帧头为 0xFA  应答帧为 0xFB |
| Ver\_Length  版本/帧长度 | 1~2 | 版本0，1：  [7:6] - 版本号  [5:0] - 数据帧的长度 ，最长64字节（包含起始帧头）  版本2：  [15:14] - 版本号，默认为 0B10  [13] - 加密标志 (默认AES256)  [12] - 保留  [11:0] - 数据帧的长度 ，最长4096字节（包含起始帧头）  高字节在前 |
| CmdSet\*  指令集 | 0~2 | 命令集\*  版本0：0字节  版本1：1字节  版本2：2字节 |
| CmdID\*  指令ID | 0~2 | 命令码\*  版本0：0字节  版本1：1字节  版本2：2字节 |
| SEQ  序列号 | 0~2 | 版本2 使用2字节：  [15] - 所有数据分包传输完成  [14:0] - 序列号，每次传输不同的指令或者回复不同的指令，序列号置0 |
| DATA | n | 数据段 见下文描述 |
| CRC-16 | 2~4 | 帧校验 （整个帧）  版本2为CRC32（4字节） |

## 2.3 标准版协议数据格式

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SOF  起始帧头 | Ver  版本号 | DataType  数据类型 | Length  数据长度 | CmdType  指令类型 | ENC  加密 | CmdSet  指令集 | CmdID  指令 | Reseved  保留 | Extend  扩展字节 | SEQ  序列号 | CRC-16  校验 | DATA\* | CRC-32\*  全帧校验 |
| 2-Byte | 1-Byte | 1-Byte | 2-Byte | 1-Byte | 1-Byte | 2-Byte | 2-Byte | 2-Byte | Ex\_len-Byte | 2-Byte | 2-Byte | n-Byte | 4-Byte |

## 2.4 标准版协议字段说明

|  |  |  |  |
| --- | --- | --- | --- |
| 域 | 偏移量 | 大小 | 描述 |
| SOF  帧头 | 0 | 2 | 帧头固定为 0xEA5A, 按数据流顺序是 0x5A ，0xEA |
| Version  版本号 | 2 | 1 | [7:6] - 保留 ，默认0  [6:0] - 版本号 16~63，默认16 |
| DataType  数据类型 | 3 | 1 | 0：文本  1：二进制（默认）  2：使用XML格式传递  3：使用JSON key-value格式传递  4：使用protobuf进行数据交互 ，schema 描述  5：使用protobuf进行数据交互 数据段 |
| Length  数据长度 | 4 | 2 | 数据帧的长度 n （包含起始帧头）  不建议一个数据包超过1024字节 |
| CmdType  指令类型 | 6 | 1 | [4:0] - 回应类型  0 - 数据发送之后不需要回应  1 - 数据发送之后需要回应，但是不应答也没关系  2-31- 数据发送之后必须要回应  [5] - 帧类型  0 - 命令帧  1 - 应答帧  [6] - 应答帧时，错误标记，数据段为错误码和错误信息，json格式，{"error\_code":1,"error\_msg":""}  [7] - 保留 默认为 0 |
| ENC  加密 | 7 | 1 | [7:5] - 加密类型  0 - 不加密  1 - AES256 加密  [4:0] - 加密时的补充字节长度（加密必须 16 字节对齐） |
| CmdSet | 8 | 2 | 命令集\* |
| CmdID | 10 | 2 | 命令码\* |
| Reseved  保留字节 | 12 | 2 | [15:4] - 保留，默认 0  [3:0] - 扩展字节长度 Ex\_len |
| Extend | 14 | Ex\_len | 扩展字节  版本16 默认为 0 |
| SEQ | 14+Ex\_len | 2 | 序列号  [15] - 所有数据分包传输完成（单独包，此处置1）  [14:0] - 序列号，每次传输不同的指令或者回复不同的指令，序列号置0 |
| CRC-16 | 16+Ex\_len | 2 | 帧头校验 |
| DATA | 18+Ex\_len | n | 数据段 见下文描述 |
| CRC-32 | 18+n+  Ex\_len | 4 | 帧校验 （整个帧） |

目前规划使用协议版本有：

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 通信接口 | 协议版本实现 | 描述 |
| 物联网 | 串行接口，CAN等 | 1 ，2（必须）  0，16（可选） | 便于精简的数据传输  建议实现 版本16的协议 |
| 互联网 | http，socket等 | 2,16（必须）  0~1（可选） | 基础数据流，优先使用 版本16协议。大规模数据只用版本16协议。 |

## 2.5 细节描述

### 2.5.1 命令集及命令 ID

相关指令设计，请注意避免设计返回值为1个字节的，容易和错误码混淆。

使用的命令集和命令码列表如下表所示：

默认指令集：所有设备实现此指令集的交互。

|  |  |  |
| --- | --- | --- |
| 项目 | 指令集实现 | 描述 |
| 物联网 | 0x00  （实现基础功能\*） | 统一通用的指令集，部分没有的功能和属性可以不实现\* |
| 互联网 | 0x00  （必须） | 统一通用的指令集 |

### 2.5.2 指令集 0x00设计内容：

|  |  |
| --- | --- |
| CmdID | 描述 |
| 0x00 | 获取对象信息\*  发送：  [0]字节，二进制：  0x00-请求人可读的数据（必须实现，UTF-8编码）  0x01-请求JSON结构数据（必须实现，UTF-8编码）  0x02-请求XML结构数据  0x03-提交JSON格式的请求，后续数据为json字符串，请求结果为JSON数据  0x04-提交JSON格式自身的信息，后续数据为json字符串  [1-n]字节 field\_name （可选）：特别选定的字段数据，字符串类型，末尾填充0x00  [1-n]字节 json字符串（可选）：json格式的查询请求（扩展）  返回：  至少提供如下字段信息：  meta\_api\_ver: 支持的元宇宙协议版本，默认1.0  id: 对象 ID ，最长512字节  name:对象名称  meta\_api\_class\_name：元宇宙统一对象分类名称(最长512字节，使用[群体协作文档注册](https://docs.qq.com/sheet/DS21mUEtNYUhEVENU)）  meta\_api\_class\_id：元宇宙统一对象类型id（最长512字节，同上，通过协作文档注册，[申请统一的对象类型id标识符](https://docs.qq.com/sheet/DS21mUEtNYUhEVENU)），同一类型id，接口和交互方式一致  info\_url:介绍链接\*  api\_info: API 接口简述（web页面，人类可读，必须）  api\_url: API 接口\*（URL形式，可选）  get\_meta\_api\_info:json 请求方若收到此字段，需通过0x04指令提交自己的信息  可加入更多[自定义信息](https://thoughts.aliyun.com/share/61954da2c1a410001add844d#title=Metaverse_API_basic_information_primitive_description(schema)_V1.0) |
| 0x01 | 请求重传指定序列号的数据帧  根据携带的数据，指出相应序列号，请求发送端重传  发送：  指令集：与接收帧相同  指令ID：与接收帧相同  序列帧：0  数据段：[0-1]字节 ： 序列号 |
| 0x02 | 请求数据全部重传\*  根据携带的数据，请求发送端重传所有该指令的数据。  发送：  指令集：与接收帧相同  指令ID：与接收帧相同  序列帧：0 |
| 0x03 | 数据接收/执行出错，用于精简版协议\*  设备端发送：  [0] 字节：错误码  [1] 字节：执行错误的指令集  字节：执行错误的指令码 |
| 0x04 | 请求接入\*  0x00：请求接入  返回：  0x01：接受请求  0x02：拒绝请求 |
| 0x05 | 加密请求  [0] 字节：  0x00：请求加密数据  [1~9] 字节：加密算法标识 ，“AES256” 等,末尾补0  [10~42] 字节：第一次握手可发送密钥key（非绝对安全）  [50~n] 字节：自己的id数据，经过加密后的  返回：  0x01：接受请求（更新密钥key）  0x02：拒绝请求 |
| 0x06 | 连接拓扑 遍历  [0] 字节：查询数量  [1-4]字节:偏移量  [5]字节：查询方式：  0x00:连接创建时间  0x01:id排序  [6]字节：是否只选中内部对象（对象内生成的对象）  0x00：否，查看所有连接的对象  0x01：是，只筛选内部对象  返回，json数据结构：  total\_connector:总的对象连接数量  列表，为每个对象输出如下信息：  id: 连接的对象id  meta\_api\_class\_id:连接的对象统一元宇宙类型id  api\_url：API接口  meta\_api\_ver: 支持的元宇宙协议版本  route\_counter:路由次数，中继跳转次数  测试：<http://42.194.159.204:8081/api?do=get_nodes&limit=10&offset=0> |
| 0x07 | 查询指定对象id信息  json格式请求数据（均为 string类型）：  data\_type:  0x00-请求人可读的数据  0x01-请求JSON结构数据  0x02-请求XML结构数据  field\_name：特别选定的字段数据，字符串类型，末尾填充0x00  object\_id：查询指定对象id ，字符串，末尾填充0x00  meta\_api\_class\_id:元宇宙唯一分类id  limit:查询数量 string  offset: 偏移量 string  返回：  查询该对象的信息，转发查询指令的执行结果  测试：<http://42.194.159.204:8081/api?do=search_nodes&object_id=meta-api-server-id-42.194.159.204&field_name=&meta_api_class_id=&limit=10&offset=0> |
| 0x08 | 断开连接\*  0x00：直接断开  0x01：删除连接信息再断开 |
| 0x09 | 请求代理中转数据  发送 json 格式数据包：含 url,timeout,post\_data,id（目标id）,data\_packet\_base64(数据帧所有内容，base64编码，包含cmd\_set,cmd\_id,data等信息）。  返回：  代理请求的数据结果 |
| 0x10 | 主机模式，请勿打扰\*  请求对象不发出任何信息，除非自己请求  发送：0x00：请勿打扰，0x01：恢复正常模式  应答：0x00：成功 ，0x01：失败 |
| 0x11 | 获取支持的[元宇宙API标准定义原型](https://thoughts.aliyun.com/share/61954da2c1a410001add844d#title=Metaverse_API_basic_information_primitive_description(schema)_V1.0)，默认 json 格式  发送：0x00：json 格式，0x01：xml格式 |
| 0x20 | 获取交互设备属性（无该属性可无需实现相关接口）：  发送：[0] 字节：0x00 获取交互属性，0x01 为json格式  [1-2] 字节：设备id，0x00 或留空 为查询任一设备  应答：  设备方发送（可主动发送）：参考 [General\_input\_device\_interface\_data\_define\_通用输入设备交互接口定义](https://thoughts.aliyun.com/share/6195b0acc1a410001add8451#title=General_input_device_interface_data_define_通用输入设备交互接口定义)  [0-1] 字节：设备id  [1-3] 字节：数据属性  1 - 键盘 10字节  2 - 鼠标 10字节  3 - 加速度传感器 12字节 （X（2字节），Y，Z，FREQ，SEQ，TIME（设备启动后的时间戳，毫秒 4字节））  4 - 陀螺仪传感器 （参考加速度传感器）  5 - 游戏手柄 （参考 USB 游戏手柄定义）  6 - 方向盘  7 - 射击游戏玩具枪模拟设备  [4-14] 字节：接口类型，USB，socket 等  [15-55] 字节： 接口参数（USB\_ID，socket ip:port 等） |
| 0x21 | 交互设备数据（设备方主动发送）  [0-1] 字节：设备id  [1-n] 字节：交互设备数据（默认为USB定义的数据格式）  [n+1 - n+5] 字节 ：设备启动后的时间戳，毫秒 4字节 |
| 0x22 | 获取当前可交互的对象id 列表，JSON格式  列表，将下方属性绑定在一组：  target\_id: 当前对象内部绑定的id，4字节，随着数字环境（场景id）变化，会有改动  id:当前对象的id （数字环境供应方和创建方需尽量避免此id重复，使其唯一） |
| 0x23 | 交互/互动接口，二进制模式：  [0-3] 字节：自己的 target\_id  [4-7] 字节：交互目标的target\_id  [8-9] 字节：interact\_type 交互类型  1 - 执行交互  2 - 尝试交互，非造成重大破坏或移除数据等方式  3 - 取消交互  4 - 强制交互  [10-11] 字节：interact\_function 交互功能  0~1000 ：默认功能，详见：[Interaction\_function\_list](https://thoughts.aliyun.com/share/6195ddf08fd3d9001a4f67b5#title=Interaction_function_list)  1000~2000：自定义功能  [12-n] 字节：interact\_data 交互数据  json格式：  携带 meta\_id 和上述数据 |
| 0x24 | 场景更新通知  指令：0x22 的 target\_id 失效，需要重新获取  发送：  [0-7] 字节：场景id （必须）  [8] 字节：失效数量，0 - 全部失效 ，1 - 失效 n个，不超过250个 ，255 - 失效id列表 JSON格式  [9 - 4\*n+9] 字节：失效的n个 target\_id ，交互目标 id  或[9 - n+9] 字节：失效id列表 JSON格式 |
| 0x25 | 场景切换，双方均可发送场景id进行切换交互  发送：  [0-7] 字节：场景id （必须）  [8] 字节：场景类型  0 - 临时场景  1 - 后续可用的场景，建议缓存对象数据  2 - 常规场景，建议存储到磁盘  3 - 永久场景，建议存储到磁盘  [9]字节：场景变化量，处理需求  0 - 几乎无变化 （较小的负荷）  1 - 手机端能支持的处理  2 - 建议使用计算机进行处理  3 - 高性能计算机，带显示加速的场景  4 - 特别高性能计算机 如需要 RTX2080显卡  5 - 高性能服务集群  [10-13] 字节：带宽需求：bps  [14-21] 字节：存储需求：Bytes  [22-29] 字节：内存需求：Bytes  [30-33] 字节：交互对象数量估计  [34-37] 字节：对象数量估计  [38-45] 字节：unix时间戳 ms |
| 0x26 | 场景信息获取:  发送：  [0] 字节：功能 0 - 遍历场景id ，1 - 获取指定场景id信息 ，2 - 获取场景信息 JSON格式，指定查询需附带场景id  [1-8] 字节：场景id （可选）  返回：  功能 0 遍历场景id列表，8字节/个  功能1 指令0x25对应的数据  功能2 JSON格式 ，信息集列表，指令0x25对应的数据(data\_base64字段) + 额外的信息 如title，meta\_scenes\_id [元宇宙统一场景id](https://docs.qq.com/sheet/DS21mUEtNYUhEVENU) 等 |
| 0x27 | 调试模式  [0-1] 字节：调试功能，0 - 退出调试模式 ，1 - 调试模式 ，2 - 调试模式 命令行  [2-n] 字节：调试数据（开发商自定义）  返回： 调试数据 |
| 0x28 | 交互数据范围校准。确保当前交互设备/工具 数据范围一致，校准开始后，用户操作输入设备，最小到最大范围，校准时间建议不超过20秒  发送：[0]字节：0 - 退出校准模式 1 - 启动校准  应答：[0]字节：0 - 校准失败 1- 校准完成 3 - 校准进度  [1]字节：校准进度 % |
| 0x29 | 适用于精简协议版本1，2 的大规模批量数据包发送，补充部分，非必须实现  注：常规使用小规模数据传输用版本1，稍多数据传输使用版本2  通知开启批量传输：  [0] 字节：批量传输控制 0 - 开启批量传输 ，1 - 结束批量传输 ，2 - 查询接收状态（返回下列数据，偏移量相同），3 - 同步序列号（接收方检查是否接收到此序列号，若前期有缺失，使用0x01指令 请求重传指定序列号的数据帧）  [1-2] 字节 CmdSet，  [3-4] 字节 CmdID，  [5-8] 字节 数据包数量（开启时可设0，结束时设置），  [9-12] 字节 本批次所有数据包 CRC-32，  [13-16] 字节 传输批次 id  [17-20] 字节 序列号 |
| 0x30 | 认证请求，希望认证该对象是否可信，是否与网络上的公钥吻合  发送：[0]字节：0 - 关闭/取消认证 ，1 - 启动认证，2 - 获取 mata\_api\_public\_key  [1-100]字节：待认证的字符串  回复：[0] 字节 应答码  [1-n]字节：加密的字符串 |
| 0xF0 | 开关机（设备/对象可不执行）\*  0x00：请求关闭  0x01：关闭完成（回复）  0x02：请求启动  0x03：启动完成（回复）  0x04：请求进入待机模式  0x05：已进入待机模式（回复） |
| 0xF1 | 重启（设备/对象可不执行）  1： 重启  2：重启完成（回复） |
| 0xF3 | 提示信息\*  连接方发送相关提示信息，如告警，关机，准备断开等  [0]字节：重要程度 0 - 不重要 ，1 - 边角提示 ，2 - 提示 ，3- 较重要，居中提示，4 - 需要确认 ，5 - 需要认真阅读  [1-n]字节：提示信息 |

以上应答/回复数据内容均需要通过通信协议打包。

指令集申请：

保留指令集 ：0x00-0x6F，0x80~0xFF ，0x8000 ~ 0xFFFF

新增指令集请通过[协作文档注册](https://docs.qq.com/sheet/DS21mUEtNYUhEVENU)。

生产商/创造者 可自定义 0x70-0x7F 指令集，无需提交申请。建议实现公开的支持广泛的指令集。

### 2.5.3 默认应答返回码

目前，支持的返回码如下表所示：

对于包含序列号的数据包，应答时需要在帧头设置该序列号

|  |  |
| --- | --- |
| 错误码值 | 含义 |
| 0x00 | 指令执行成功 |
| 0x01 | 指令解析错误 |
| 0x02 | 指令执行失败 |
| 0xFF | 未定义的错误 |

# 3.CRC校验

数据包中使用的 CRC16 和 CRC32 参数如下

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Width | Poly | Init | RefIn | RefOut | XorOut |
| CRC16 | 16 | 0x8005 | 0xc55c | True | True | 0x0000 |
| CRC32 | 32 | 0x04c11db7 | 0xc55c0000 | True | True | 0x00000000 |

## 3.1 CRC 代码示例

本协议中使用的 CRC16 可以参考：custom\_crc16.c, custom\_crc16.h

本协议中使用的 CRC32 可以参考：custom\_crc32.c, custom\_crc32.h

链接：<https://pan.baidu.com/s/15PgpC4OkKF3XO_Iv4XAKjA>

提取码：5566

# 4.开源代码

## 4.1 通信协议基础接口与代码示例

元宇宙基础API接口（可通过此服务器获取其他节点信息，遍历元宇宙）：

<http://sg.zyinfo.pro:8081/api>

示例：

获取API基本信息：

<http://sg.zyinfo.pro:8081/api?get_meta_api_info=text>

获取连接的节点信息：

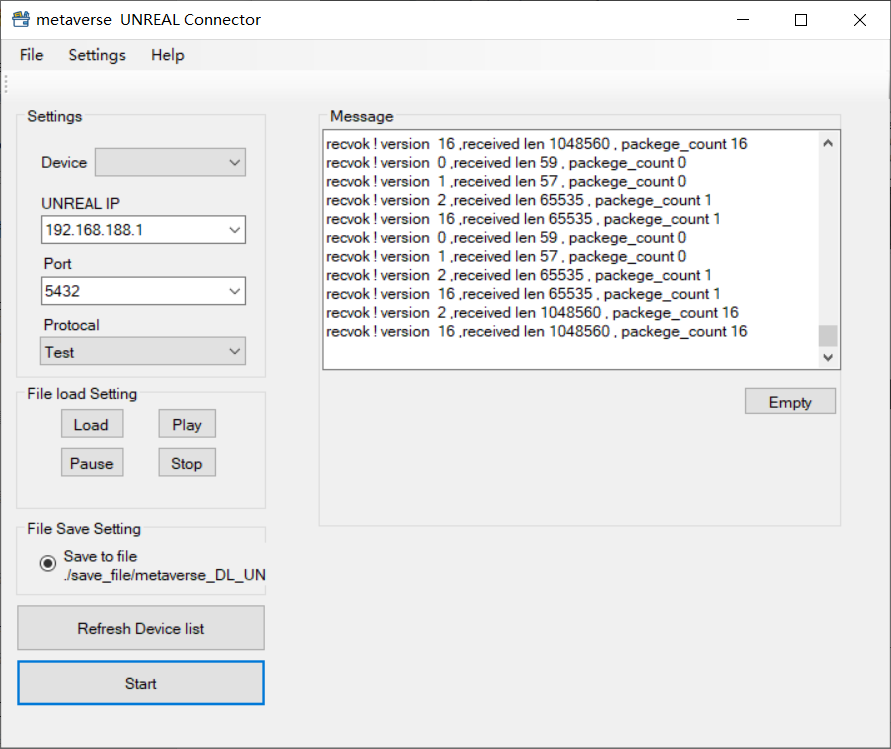
<http://sg.zyinfo.pro:8081/api?do=get_nodes&limit=10&offset=0>

搜索节点信息：[测试](http://42.194.159.204:8081/api?do=search_nodes&object_id=&field_name=id,nm,mci&meta_api_class_id=meta-api-server-class&limit=10&offset=0)

API服务器开源代码：

<https://gitee.com/hayoou/metaverse_api_server_go>

## 4.2 通信协议代码示例



[元宇宙通信协议 C# 实现](https://gitee.com/hayoou/meta_api_protocal_c_sharp)

主页：

[Metaverse standard API 元宇宙接口标准](https://thoughts.aliyun.com/share/6195068ebdc2c4001aea0058#title=Metaverse_standard_API)