

MSync协议说明

MSync是基于Protobuf的私有二进制协议。其基本结构如下：

Header + Payload

Header

Header部分内容用来标识协议版本，用户信息及payload中的内容属于哪个command。

```
optional Version version          ; //协议版本号
optional JID    guid              ; //用户id
optional string auth              ; //用户token
optional uint32 compress_algorimth ; //暂未使用
optional uint32 crypto            ; //暂未使用
optional string user_agent        ; //暂未使用
optional uint64 pov               ; //暂未使用
optional Command command          ; //用户协议命令
optional uint32 device_id         ; //设备id
```

Command是自定义的与服务器交互的主要信令，有如下几个：

```
enum Command {
    SYNC          ; //Sync命令，用来跟服务器同步数据
    UNREAD        ; //Unread命令，用来查询服务器是否有未同步数据
    NOTICE       ; //Notice命令，用来通知客户端有新数据同步
    PROVISION     ; //Provision命令，用来登录服务器做鉴权使用
}
```

Payload

根据不同的command命令，payload中会填充不同的数据，依次有如下几种：

Provision

```

message Provision
{
    optional OsType          os_type          ; //客户端类型
    optional string          version          ; //SDK版本号
    optional NetworkType     network_type     ; //SDK网络类型，暂未使用
    optional string          app_sign         ; //SDK签名，暂未使用
    repeated CompressType    compress_type    ; //压缩类型
    repeated EncryptType     encrypt_type     ; //加密类型
    optional string          encrypt_key      ; //加密key
    optional Status          status          ; //服务器返回登录状态 etc.
    optional string          device_uuid      ; //用于区分多设备同时登录时的设备
    optional bool            is_manual_login  ; //标识是否自动登录
    optional string          password         ; //用户密码，备用
    optional string          device_name      ; //设备名称
    optional string          resource         ; //用于多设备登录时服务器生成的resource id
}

```

Notice

```

message CommNotice {
    optional JID queue ; //通知某个消息队列中有新消息
}

```

Unread

```

message CommUnreadUL {
    // 客户端发送给服务器的请求，内容为空
}

message CommUnreadDL {
    optional Status      status      ; //服务器返回Unread状态
    repeated MetaQueue   unread      ; //服务器返回有新消息的队列
    optional uint64      timestamp   ; //服务器时间戳。
}

```

Sync

```

message CommSync {
    repeat Meta metas          ; //服务器与客户端直接同步的实际数据内容
    optional uint64 key        ; //标识消息内容的起点
    optional JID queue        ; //消息队列id
    optional Status status    ; //同步内容的状态
    optional uint64 meta_id    ; //同步数据id
    optional uint64 server_id  ; //服务器端生成的消息id
    optional uint64 next_key   ; //标识下一次同步数据的id
    optional bool is_last     ; //标识数据同步是否已经结束
    optional uint64 timestamp  ; //服务器时间戳。

    optional bool is_roam      ; //标识是否漫游消息
    optional uint64 last_full_roam_key ; //上次拉取的漫游消息的标识
}

message Meta
{
    enum Namespace {
        STATISTIC ; //客户端数据上报和服务器封禁等事件通知
        CHAT       ; //消息
        MUC         ; //群组事件
        ROSTER      ; //好友事件
        CONFERENCE  ; //实时音视频事件
    }

    optional uint64 id          ; // Meta id
    optional JID from          ; // Meta数据的发出方
    optional JID to            ; // Meta数据的接收方
    optional uint64 timestamp; // 服务器时间戳
    optional Namespace ns      ; // Meta数据类型
    optional bytes payload     ; // Meta中的数据
}

```

消息格式

```

message MessageBody
{
    optional Type type          ; //单聊，群聊，聊天室等消息类型
    optional JID from          ; //消息的发出方
    optional JID to            ; //消息的接收方
    repeated Content contents   ; //消息内容
    repeated KeyValue ext       ; // 消息扩展
    optional uint64 ack_message_id ; //消息回执id
}

```

```

message Content
{
    message Size
    {
        optional double width ; // 图片的宽度
        optional double height ; //图片的高度
    }

    optional Type type ; // 消息内容类型: 文本, 图片, 语音等消息类型

    //text body field
    optional string text ; // 消息文本内容

    //location body field
    optional double latitude ; // 位置消息的纬度
    optional double longitude ; // 位置消息的经度
    optional string address ; // 位置消息的地址

    //file body field
    optional string displayName ; // 附件消息的文件名
    optional string remotePath ; // 附件消息的服务器地址
    optional string secretKey ; // 附件消息的secret key
    optional int32 fileLength ; // 附件消息的文件长度

    //command body field
    optional string action ; // Cmd消息的action参数
    repeated KeyValue params ; // Cmd消息的内容

    //voice and video field
    optional int32 duration ; // 视频消息的时长

    //size field for video and image
    optional Size size ; // 图片的大小

    //thumbnail field
    optional string thumbnailRemotePath ; // 缩略图的服务器地址
    optional string thumbnailSecretKey ; // 缩略图的secret key
    optional string thumbnailDisplayName ; // 缩略图的名字
    optional int32 thumbnailFileLength ; // 缩略图的文件大小
    optional Size thumbnailSize ; // 缩略图的大小
}

```

错误码

```

message Status {
  enum ErrorCode {
    OK                ; //操作成功
    FAIL              ; //操作失败
    UNAUTHORIZED      ; //未授权
    MISSING_PARAMETER ; //缺少参数
    WRONG_PARAMETER   ; //错误的参数
    REDIRECT           ; //重定向, 暂未使用
    TOKEN_EXPIRED      ; //Token过期
    PERMISSION_DENIED ; // 用户权限不足
    NO_ROUTE           ; // 暂未使用
    UNKNOWN_COMMAND    ; // 未知的命令
    PB_PARSER_ERROR    ; // 协议解析错误
    BIND_ANOTHER_DEVICE ; // 用户已经在其他设备登录
    IM_FORBIDDEN       ; // 用户被禁止使用
    TOO_MANY_DEVICES   ; // 用户登录过多设备
    PLATFORM_LIMIT     ; // 平台限制
    USER_MUTED        ; // 用户被群组禁言
  };
};

```

实际Log例子:

Provision

```

[2017/10/26 12:29:16:302]: log: level: 1, area: 1, SEND:
{ verison : MSYNC_V1, guid : easemob-demo#chatdemoui_zl2@easemob.com/android_3c4df
774-1a75-3b61-a912-ffc4d280adae, auth : ***, compress_algorimth : 0, command : PRO
VISION, payload : { os_type : 1, version : 3.3.4, compress_type : [ COMPRESS_NONE
], device_uuid : 3c4df774-1a75-3b61-a912-ffc4d280adae, is_manual_login : true, dev
ice_name : LGENexus 5, resource : android_3c4df774-1a75-3b61-a912-ffc4d280adae, pa
ssword : *** } }

[2017/10/26 12:29:16:366]: log: level: 1, area: 1, RECV:
{ verison : MSYNC_V1, command : PROVISION, payload : { compress_type : [ COMPRESS_
NONE ], status : { error_code : 0 }, resource : android_3c4df774-1a75-3b61-a912-ff
c4d280adae } }

```

发送消息

```
[2017/10/26 12:27:38:982]: log: level: 1, area: 1, SEND:
{ verison : MSYNC_V1, compress_algorimth : 0, command : SYNC, payload : { meta : {
  id : 15089920589600003, from : z12/android_3c4df774-1a75-3b61-a912-ffc4d280adae,
to : z13, ns : CHAT, payload : { chattype : CHAT, from : z12, to : z13, contents :
  [ { contenttype : TEXT, text : 456 } ] } } } } }

[2017/10/26 12:27:39:033]: log: level: 1, area: 1, RECV:
{ verison : MSYNC_V1, command : SYNC, payload : { status : { error_code : 0 }, met
a_id : 15089920589600003, server_id : 392677097702688788, timestamp : 150899206143
6 } }
```

接收消息

```
[2017/10/26 12:27:28:743]: log: level: 1, area: 1, RECV:
{ verison : MSYNC_V1, command : NOTICE, payload : { queue : z13/webim_device_uuid
} }

[2017/10/26 12:27:28:744]: log: level: 1, area: 1, SEND:
{ verison : MSYNC_V1, compress_algorimth : 0, command : SYNC, payload : { queue :
z13/webim_device_uuid } }

[2017/10/26 12:27:28:760]: log: level: 1, area: 1, RECV:
{ verison : MSYNC_V1, command : SYNC, payload : { status : { error_code : 0 }, met
as : [ { id : 392677053607969272, from : easemob-demo#chatdemoui_z13@easemob.com/w
ebim_device_uuid, to : easemob-demo#chatdemoui_z12@easemob.com, timestamp : 150899
2051149, ns : CHAT, payload : { chattype : CHAT, from : z13, to : z12, contents :
[ { contenttype : TEXT, text : 123 } ], exts : [ { key : weichat, type : 8, value
: {"originType":"webim"} } ] } } ], next_key : 392677053607969272, queue : z13, ti
mestamp : 1508992051178 } }
```