船舶人工智能系统

网络通信协议

编制： 王 科 威

审核：

版本： V1.0.0

2020年5月18日

目录

[修改历史 I](#_Toc41290392)

[一 引言 1](#_Toc41290393)

[1.1 编写目的 1](#_Toc41290394)

[1.2 规范说明 1](#_Toc41290395)

[二 定义 1](#_Toc41290396)

[2.1 协议格式 1](#_Toc41290397)

[2.2 协议定义 1](#_Toc41290398)

# 修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **说明** | **作者** | **审核** | **日期** |
| V1.0.0.0 | 初稿 | 王科威 |  | 2020-05-25 |

# 引言

## 编写目的

在代码编写阶段按照该文档定义的数据内容进行数据封装和解析。

## 规范说明

该文档数据格式按照Google protocol buffers开源项目的sytex2格式要求编写。

# 定义

## 协议格式

各类消息按照统一的协议格式进行封装和解析，协议格式定义如下：

syntax = "proto2";

import "Alarm.proto";

import "Algorithm.proto";

import "Component.proto";

import "Crew.proto";

import "Device.proto";

import "Status.proto";

import "User.proto";

package msg;

message MSG

{

enum Type

{

MSG\_NONE = 0;

ALARM = 1;

ALGORITHM = 2;

COMPONENT = 3;

CREW = 4;

DEVICE = 5;

STATUS = 6;

User = 7;

}

required Type type = 1 [default = MSG\_NONE];

required int32 sequence = 2;

required int32 timestamp = 3;

optional AlarmMsg alarmmsg = 4;

optional AlgorithmRequest algorithmrequest = 5;

optional AlgorithmResponse algorithmresponse = 6;

optional ComponentRequest componentrequest = 7;

optional ComponentResponse componentresponse = 8;

optional CrewRequest crewrequest = 9;

optional CrewResponse crewresponse = 10;

optional DeviceRequest devicerequest = 11;

optional DeviceResponse deviceresponse = 12;

optional StatusRequest statusrequest = 13;

optional StatusResponse statusresponse = 14;

optional UserRequest userrequest = 15;

optional UserResponse userresponse = 16;

}

## 协议定义

### 组件类

#### 协议定义

syntax = "proto2";

package msg;

message ComponentRequest

{

enum Type

{

XMQ = 1;

WEB = 2;

HKD = 3;

DHD = 4;

ALM = 5;

}

required Type type = 1;

enum Operate

{

SIGNIN = 1;

SIGNOUT = 2;

}

required Operate operate = 2;

required string cid = 3;

optional string cname = 4;

}

message ComponentResponse

{

required int32 result = 1;

}

#### 注册请求

* type：按照实际运行的组件实例类型设置；
* operate：SIGNIN；
* cid：组件ID标识，用CPU@进程ID标识；
* cname：组件名称。

#### 注册应答

* result：错误码

#### 注销请求

* type：按照实际运行的组件实例类型设置；
* operate：SIGNOUT；
* cid：组件ID标识，用CPU@进程ID标识；
* cname：组件名称。

#### 注销应答

无

#### 心跳请求

* type：按照实际运行的组件实例类型设置；
* operate：SIGNIIN；
* cid：组件ID标识，用CPU@进程ID标识；
* cname：组件名称。

#### 心跳应答

无

### 用户类

#### 协议定义

syntax = "proto2";

package msg;

message Author

{

required bool enableconfigure = 1;

required bool enablequery = 2;

}

message User

{

required string uname = 1;

required string password = 2;

required Author author = 3;

}

message UserRequest

{

enum Operate

{

NEW = 0;

DELETE = 1;

MODIFY = 2;

QUERY = 3;

LOGIN = 4；

}

required Operate operate = 1;

optional User user = 2;

optional string uid = 3;

}

message UserResponse

{

required int32 result = 1;

optional string uid = 2;

repeated User users = 3;

}

#### 新增请求

* operate：NEW；
* user.uname：用户名；
* user.password：用户密码；
* user.author.enableconfigure：配置权限许可标识；
* user.author.enablequery：查询权限许可标识。

#### 新增应答

* result：错误码；
* uid：用户ID标识。

#### 删除请求

* operate：DELETE；
* uid：用户ID标识。

#### 删除应答

* result：错误码。

#### 修改请求

* operate：MODIFY；
* uid：用户ID标识；
* user.uname：用户名；
* user.password：用户密码；
* user.author.enableconfigure：配置权限许可标识；
* user.author.enablequery：查询权限许可标识。

#### 修改应答

* result：错误码。

#### 查询请求

* operate：MODIFY；
* uid：用户ID标识。

#### 查询应答

* result：错误码；
* users：用户信息集合。

#### 登录请求

* operate：LOGIN；
* uid：用户ID标识；
* user.uname：用户名；
* user.password：用户密码。

#### 登录应答

* result：错误码；

### 设备类

#### 协议定义

syntax = "proto2";

package msg;

message Camera

{

required string cid = 1;

required int32 index = 2;

required bool enable = 3;

optional string nickname = 4;

optional string ip = 5;

}

message Device

{

required bool enable = 1 [default = false];

enum Factory

{

HIKVISION = 1;

DAHUA = 2;

EATON = 3;

}

optional Factory factory = 2 [default = NONE];

enum Type

{

DVR = 1;

NVR = 2;

IPC = 3;

}

optional Type type = 3;

optional string name = 4;

optional string password = 5;

optional string ip = 6;

optional int32 port = 7;

optional string did = 8;

optional string nickname = 9;

repeated Camera cameras = 10;

}

message DeviceRequest

{

enum Operate

{

NEW = 0;

DELETE = 1;

MODIFY = 2;

QUERY = 3;

}

required Operate operate = 1;

optional Device device = 2;

}

message DeviceResponse

{

required int32 result = 1;

repeated Device devices = 2;

}

#### 新增请求

* operate：NEW;
* device.enable：true表示启动设备，false表示停用设备；
* device.factory：设备厂商类型；
* device.type：设备类型；
* name：设备登录名；
* password：设备登录密码；
* ip：设备IP地址；
* port：设备端口号；
* nickname：设备别名。

#### 新增应答

* result：错误码；
* device.did：设备ID标识；
* device.cameras：设备可用摄像机集合。

#### 删除请求

* operate：DELETE;
* did：设备ID标识。

#### 删除应答

* result：错误码。

#### 修改请求

* operate：MODIFY;
* device.enable：true表示启动设备，false表示停用设备；
* device.factory：设备厂商类型；
* device.type：设备类型；
* name：设备登录名；
* password：设备登录密码；
* ip：设备IP地址；
* port：设备端口号；
* nickname：设备别名；
* device.cameras：设备可用摄像机集合。

#### 修改应答

* result：错误码。

#### 查询请求

* operate：QUERY;

#### 查询应答

* result：错误码；
* device：设备集合。

### 算法类

#### 协议定义

syntax = "proto2";

package msg;

message Algorithm

{

required string cid = 1;

optional int32 gpu = 2;

optional bool enablesleep = 3;

optional bool enablefight = 4;

optional bool enablehelmet = 5;

optional bool enablephone = 6;

optional bool enableattendancein = 7;

optional bool enableattendanceout = 8;

optional float similar = 9;

}

message AlgorithmRequest

{

enum Operate

{

CONFIGURE = 1;

QUERY = 2;

}

required Operate operate = 1;

repeated Algorithm algorithms = 2;

}

#### 配置请求

* operate：CONFIGURE；
* algorithm：算法参数集合。

#### 配置应答

无

#### 查询请求

* operate：QUERY。

#### 查询应答

* result：错误码；
* algorithms：算法配置集合。

### 员工类

#### 协议定义

syntax = "proto2";

package msg;

message Crew

{

optional string uid = 1;

optional string name = 2;

optional string job = 3;

repeated string pictures = 4;

}

message CrewRequest

{

enum Operate

{

NEW = 0;

DELETE = 1;

MODIFY = 2;

QUERY = 3;

}

required Operate operate = 1;

optional Crew crew = 2;

optional string uid = 3;

}

message CrewResponse

{

required int32 result = 1;

optional string uid = 2;

repeated Crew crews = 3;

}

#### 新增请求

* operate：NEW；
* crew：员工信息。

#### 新增应答

* result：错误码；
* uid：用户ID标识。

#### 删除请求

* operate：DELETE；
* uid：用户ID标识。

#### 删除应答

* result：错误码。

#### 修改请求

* operate：MODIFY；
* crew：员工信息。

#### 修改应答

* result：错误码。

#### 查询请求

* operate：QUERY。

#### 查询应答

* result：错误码；
* crews：员工信息集合。

### 报警类

#### 协议定义

syntax = "proto2";

package msg;

message Position

{

required int32 x = 1;

required int32 y = 2;

required int32 w = 3;

required int32 h = 4;

}

message Alarm

{

enum Type

{

NONE = 0;

SLEEP = 1;

FIGHT = 2;

HELMET = 3;

PHONE = 4;

FACE = 5;

}

required Type type = 1 [default = NONE];

repeated Position posistion = 2;

optional int32 uid = 3;

}

message AlarmMsg

{

required string cid = 1;

required string time = 2;

required string picture = 3;

required Alarm alarm = 4;

}

#### 报警上传

* cid：摄像机ID标识；
* time：报警时间；
* picture：报警实时图片，以JEPG格式封装；
* alarm：报警信息。

### 状态类

#### 协议定义

syntax = "proto2";

package msg;

message StatusRequest

{

enum Operate

{

SET = 1;

QUERY = 2;

}

required Operate operate = 1;

enum Type

{

SAIL = 1;

}

required Type type = 2;

optional bool flag = 3;

}

message StatusResponse

{

required int32 result = 1;

optional bool flag = 2;

}

#### 航停请求

* operate：SET；
* type：SAIL；
* flag：true表示航行，false表示停港。

#### 航停应答

* result：错误码。

#### 查询请求

* operate：QUERY。

#### 查询应答

* result：错误码。
* flag：true表示航行，false表示停港。