接口文档

目录

[接口文档 1](#_Toc520470224)

[修订记录 1](#_Toc520470225)

[1 简介 1](#_Toc520470226)

[1.1 http接口 1](#_Toc520470227)

[1.2 消息组件 1](#_Toc520470228)

[2 接口 3](#_Toc520470229)

[2.1 卡口动态 3](#_Toc520470230)

[2.1.1 船舶实时动态 3](#_Toc520470231)

[2.2 业务处理 4](#_Toc520470232)

[2.2.1 告警信息查询 4](#_Toc520470233)

[2.2.2 卡口区域配置 6](#_Toc520470234)

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 修订版本 | 修订时间 | 修订人 | 备注 |
| V1.0 | 2018-07-25 | 宋晓彤 |  |
| V1.1 | 2018-07-27 | 宋晓彤 | 增加卡口区域配置接口 |
|  |  |  |  |

不好的点：全部使用post、没有产品经理提供的需求

# 简介

## http接口

该文档描述WBS水上卡口系统后台接口规范，后台接口遵循RESTful软件架构设计风格。

## 消息组件

* + 1. **通讯方式**

通讯采用ZeroMQ；数据序列化采用Protocol Buffer

请求类通讯，端口：5858；

数据类通讯，端口：5555；

控制类通讯，端口：5556；

IP为内河VTS平台后台服务器的IP；

* + 1. **ZeroMQ简介**

ZeroMQ号称史上最快中间件。

它对socket通信进行了封装，使得我们不需要写socket函数调用就能完成复杂的网络通信。

它跟Socket的区别是：普通的socket是端到端的（1:1的关系），而ZMQ却是可以N:M的关系，人们对BSD套接字的了解较多的是点对点的连接，点对点连接需要显式地建立连接、销毁连接、选择协议（TCP/UDP）和处理错误等，而ZMQ屏蔽了这些细节，让你的网络编程更为简单。

它是一个消息处理队列库，可在多个线程、内核和主机之间弹性伸缩。和一般意义上的消息队列产品不同的是，它没有消息队列服务器，而更像是一个网络通信库。从网络通信的角度看，它处于会话层之上，应用层之下，属于传输层。

* + 1. **ZeroMQ的消息模型**

ZeroMQ将消息通信分为4种模型，分别是一对一结对模型（Exclusive-Pair）、请求回应模型（Request-Reply）、发布订阅模型（Publish-Subscribe）、推拉模型（Push-Pull）。这4种模型总结出了通用的网络通信模型，在实际中可以根据应用需要，组合其中的2种或多种模型来形成自己的解决方案。

* + 1. **Google Protocol Buffer简介**

Google Protocol Buffer( 简称 Protobuf) 是 Google 公司内部的混合语言数据标准，目前已经正在使用的有超过 48,162 种报文格式定义和超过 12,183 个 .proto 文件。他们用于 RPC 系统和持续数据存储系统。

Protocol Buffers 是一种轻便高效的结构化数据存储格式，可以用于结构化数据串行化，或者说序列化。它很适合做数据存储或 RPC 数据交换格式。可用于通讯协议、数据存储等领域的语言无关、平台无关、可扩展的序列化结构数据格式。目前提供了 C++、Java、Python 三种语言的 API。

* + 1. **Protobuf 的优点**

Protobuf 有如 XML，不过它更小、更快、也更简单。你可以定义自己的数据结构，然后使用代码生成器生成的代码来读写这个数据结构。你甚至可以在无需重新部署程序的情况下更新数据结构。只需使用 Protobuf 对数据结构进行一次描述，即可利用各种不同语言或从各种不同数据流中对你的结构化数据轻松读写。

它有一个非常棒的特性，即“向后”兼容性好，人们不必破坏已部署的、依靠“老”数据格式的程序就可以对数据结构进行升级。这样您的程序就可以不必担心因为消息结构的改变而造成的大规模的代码重构或者迁移的问题。因为添加新的消息中的 field 并不会引起已经发布的程序的任何改变。

Protobuf 语义更清晰，无需类似 XML 解析器的东西（因为 Protobuf 编译器会将 .proto 文件编译生成对应的数据访问类以对 Protobuf 数据进行序列化、反序列化操作）。

使用 Protobuf 无需学习复杂的文档对象模型，Protobuf 的编程模式比较友好，简单易学，同时它拥有良好的文档和示例，对于喜欢简单事物的人们而言，Protobuf 比其他的技术更加有吸引力。

* + 1. **请求类端口数据帧格式定义:**

请求方应该按照如下数据帧格式发送数据到请求端口:

第一帧：([service名称]/[接口的编码])

第二帧：[数据（接口的参数—json格式），如无参数，就用空字符串（””或“ ”）]

第三帧：[接口参数类型（全路径，即为请求参数中的第一个路径），如无参数类型，就用空字符串（””或“ ”）]

备注：/ 是必须的

空字符串: ””或“ ”

返回结果：json格式的字符串

* + 1. **数据类和控制类端口数据帧格式定义**

数据接收方应该按照如下数据帧格式接收数据：

第一帧：数据的特征码（Topic）

第二帧：数据的时间戳，单位ms（MSecsSinceEpoch 中国时区）

第三帧：Protobuf序列化后的数据块

# 接口

### 告警信息查询

#### 2.1.2.1分页查询事件，附带图片信息

##### 接口描述

根据检索条件分页查询指定类型的告警信息，并且附带事件的图片信息

##### 请求方式

restful http接口

URL:http://localhost:port/zchx/shipevent/page

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"page": "1",

" rows": “10”,

“t”:{

“mmsi”:”1”,

“trackNumber”:”1”,

“eventType”:”1”,

“shipType”:”1”,

“name”:”1”,

“mixTime”:”1”

“maxTime”:”1”

}

}

##### 参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| page | Integer | Y | 当前页数 |
| rows | Integer | Y | 每页的行数 |
| t | ShipEvent | Y | 查询实体（条件拼装） |

ShipEvent参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| mmsi | String | N | Ais Mmsi编号 |
| trackNumber | String | N | 雷达编号 |
| eventType | int | N | 时间类型 |
| shipType | int | N | 船舶类型 |
| name | String | N | 船舶名称 |
| mixTime | Long | N | 查询时间范围上限 |
| maxTime | Long | N | 查询时间范围下限 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | WebPage | 响应数据 | | 分页结果包装类 | |

WebPage参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 描述 | 备注 |
| total | Integer | 总条数 |  |
| Rows | List<ShipEvent> | 行list | 具体的数据 |

ShipEvent参数格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| Id | Integer | | 事件id |  |
| operTime | long | 操作时间 | | 等于创建时间 | |
| creatorId | Integer | 操作人id | |  | |
| createTime | long | 事件创建时间 | |  | |
| shipType | Integer | 船只类型 | |  | |
| name | String | 船名 | |  | |
| mmsi | String | mmsi | |  | |
| trackNumber | String | trackNumber | |  | |
| eventType | Integer | 事件类型 | |  | |
| photoInfos | photoInfo | 附带的照片信息 | |  | |
| eventId | Integer | 事件id | |  | |
| useStatus | Integer | 使用状态 | | 未使用:1,已使用:2 | |
| photoUrl | String | 照片地址路径 | |  | |
| wgs84Lat | Double | 纬度 | |  | |
| wgs84Lon | Double | 经度 | |  | |

##### 响应结果示例

{

"code": 1000,

"msg": "OK",

"data": {

"total": "1",

“rows”: [{

"id": 1,

"del": 1,

"operTime": 1533622660000,

"opertorId": 1,

"createTime": 20180807101531,

"creatorId": 1,

"shipType": "1",

"name": "ww",

"mmsi": "370569000",

"trackNumber": "4552",

"eventType": 4,

"photoInfos": [

{

"eventId": 1,

"useStatus": 1,

"photoUrl": "123",

"wgs84Lat": 1,

"wgs84Lon": 1

},

{

"eventId": 1,

"useStatus": 1,

"photoUrl": "456",

"wgs84Lat": 2,

"wgs84Lon": 2

}

],

}],

}

}

#### 2.1.2.2根据事件id查询事件信息，附带雷达，ais轨迹信息

##### 接口描述

根据事件id查询事件信息，附带雷达，ais轨迹信息

##### 请求方式

restful http接口

URL:http://localhost:port/zchx/shipevent/tracks

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"id": "1",

" maxTrackTime ": “10131231231”,

" minTrackTime ": “102312312312”,

}

##### 参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | Y | 事件id |
| minTrackTime | Long | Y | 事件轨迹范围上限(时间戳) |
| maxTrackTime | Long | Y | 事件轨迹范围下限（时间戳） |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | ShipEvent | 响应数据 | |  | |

ShipEvent参数类型

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| Id | Integer | | 事件id |  |
| operTime | long | 操作时间 | | 等于创建时间 | |
| creatorId | Integer | 操作人id | |  | |
| createTime | long | 事件创建时间 | |  | |
| shipType | Integer | 船只类型 | |  | |
| name | String | 船名 | |  | |
| mmsi | String | mmsi | |  | |
| trackNumber | String | trackNumber | |  | |
| eventType | Integer | 事件类型 | |  | |
| aisTracks | AISHistoryTracks | Ais轨迹的protbuf | |  | |
| radarTracks | RadarHistoryTracks | 雷达轨迹的protbuf | |  | |

##### 响应结果示例

{

"code": 1000,

"message": "OK",

"data": {

"id": 1,

"del": 1,

"operTime": 1533622660000,

"opertorId": 1,

"createTime": 20180807101531,

"creatorId": 1,

"shipType": "1",

"name": "ww",

"mmsi": "370569000",

"trackNumber": "4552",

"eventType": 4,

"aisTracks":"Ci8NmpmFQREAAACgVtJaQBkAAABASsMyQCWamadDLQCAqEMwAD0A",

"radarTracks":"CjkIyCMR1kGPIgYkNEAZI4paeVB3W0AlAPUDRygAOAFAAUljWYdFJLs6"

}

}

### 卡口区域配置

#### 区域新增

##### 接口描述

提供新增卡口区域功能，可以在添加卡口区域的同时添加区域拆分的网格数据

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/create

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"createAreaGrids": [{

"areaId": 1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"tilt": 0.1,

"name": "新增区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}, {

"areaId": 1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"tilt": 0.1,

"name": "新增区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}, {

"areaId": 1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"tilt": 0.1,

"name": "新增区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}, {

"areaId": 1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"tilt": 0.1,

"name": "新增区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}

],

"name": "新增测试拍照区域",

"type": 1

}

##### 参数格式

CreateArea防区对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| name | String | Y | 防区名称 |
| coordinates | List<String> | Y | 防区坐标点 |
| type | Integer | Y | 防区类型 |
| direction | Integer | N | 方向(1正，0逆向） |
| allowShipType | Integer | N | 允许船只类型 |
| allowMaxSpeed | Double | N | 允许的最大速度 |
| allowMinSpeed | Double | N | 允许的最小速度 |
| createAreaGrids | List<CreateAreaGrid> | N | 防区网格 |
|  |  |  |  |

CreateAreaGrid区域网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| name | String | Y | 网格名称 |
| areaId | Integer | Y | 所属区域编号 |
| coordinates | List<String> | Y | 网格坐标点 |
| type | Integer | N | 网格类型 |
| pan | Float | N | 相机参数-左右位置 |
| tilt | Float | N | 相机参数-上下位置 |
| zoom | Float | N | 相机参数-放大倍数 |
| runMode | String | N | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM |
| speedOption | Float | N | 相机参数-速度选项 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域编辑

##### 接口描述

区域编辑，根据传入的区域信息进行区域的数据更新，如果有传入区域网格信息则对网格信息进行更新、新增、删除等操作

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/update

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"allowMaxSpeed": 9.9,

"allowMinSpeed": 0.1,

"allowShipType": 2,

"coordinates": [

"111.111,222.222",

"333.333,444.444",

"555.555,666.666",

"777.777,888.888"

],

"createAreaGrids": [{

"areaId": 1,

"coordinates": [

"111.111,222.222",

"333.333,444.444",

"555.555,666.666",

"777.777,888.888"

],

"tilt": 0.1,

"name": "新增区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}

],

"delGridIds": [

6

],

"direction": 1,

"id": 1,

"name": "修改区域名称",

"type": 2,

"updateAreaGrids": [{

"areaId": 1,

"coordinates": [

"111.111,222.222",

"333.333,444.444",

"555.555,666.666",

"777.777,888.888"

],

"id": 4,

"tilt": 0.1,

"name": "修改区域网格名称",

"pan": 0.1,

"runMode": "AM",

"speedOption": 0.1,

"type": 2,

"zoom": 0.1

}

]

}

##### 参数格式

UpdateArea防区对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | Y | 防区编号 |
| name | String | N | 防区名称 |
| coordinates | List<String> | N | 防区坐标点 |
| type | Integer | N | 防区类型 |
| direction | Integer | N | 方向(1正，0逆向） |
| allowShipType | Integer | N | 允许船只类型 |
| allowMaxSpeed | Double | N | 允许的最大速度 |
| allowMinSpeed | Double | N | 允许的最小速度 |
| createAreaGrids | List< CreateAreaGrid> | N | 要新增的网格区域 |
| updateAreaGrids | List<UpdateAreaGrid> | N | 要更新的网格区域 |
| delGridIds | List<Integer> | N | 要删除的防区网格编号 |

CreateAreaGrid区域网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| name | String | Y | 网格名称 |
| areaId | Integer | Y | 所属区域编号 |
| coordinates | List<String> | Y | 网格坐标点 |
| type | Integer | N | 网格类型 |
| pan | Float | N | 相机参数-左右位置 |
| tilt | Float | N | 相机参数-上下位置 |
| zoom | Float | N | 相机参数-放大倍数 |
| runMode | String | N | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM |
| speedOption | Float | N | 相机参数-速度选项 |

UpdateAreaGrid区域对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | Y | 网格编号 |
| areaId | Integer | Y | 所属区域编号 |
| name | String | N | 网格名称 |
| coordinates | List<String> | N | 网格坐标点 |
| type | Integer | N | 网格类型 |
| pan | Float | N | 相机参数-左右位置 |
| tilt | Float | N | 相机参数-上下位置 |
| zoom | Float | N | 相机参数-放大倍数 |
| runMode | String | N | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM |
| speedOption | Float | N | 相机参数-速度选项 |
|  |  |  |  |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域删除

##### 接口描述

根据区域编号删除指定区域和下属的区域网格

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/delete

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"id":1

}

##### 参数格式

DeleteArea删除防区对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | N | 防区编号 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域查询

##### 接口描述

根据区域编号查询所属的区域信息合下属的区域网格信息

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/read

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"id":1

}

##### 参数格式

ReadArea查询防区对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | N | 防区编号 |
| portId | Integer | Y | 卡口编号 |
| type | Integer | N | 卡口类型 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | InspectArea | 响应数据 | | XXX | |

InspectArea防区对象

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| id | Integer | | 防区编号 |  |
| portId | String | | 卡口编号 |  |
| name | String | | 防区名称 |  |
| coordinates | List<String> | | 防区坐标点 |  |
| type | Integer | 防区类型 | |  | |
| direction | Integer | 方向(1正，0逆向） | |  | |
| allowShipType | Integer | 允许船只类型 | |  | |
| allowMaxSpeed | Double | 允许的最大速度 | |  | |
| allowMinSpeed | Double | 允许的最小速度 | |  | |
| inspectAreaGrids | List<InspectAreaGrid> | 区域网格信息 | |  | |
|  |  |  | |  | |
|  |  |  | |  | |

InspectAreaGrid防区网格对象

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| id | Integer | | 网格编号 |  |
| name | String | | 网格名称 |  |
| coordinates | List<String> | | 网格坐标点 |  |
| type | Integer | 网格类型 | |  | |
| areaId | Integer | 防区编号 | |  | |
| pan | Float | 相机参数-左右位置 | |  | |
| tilt | Float | 相机参数-上下位置 | |  | |
| zoom | Float | 相机参数-放大倍数 | |  | |
| runMode | String | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM | |  | |
| speedOption | Float | 相机参数-速度选项 | |  | |
|  |  |  | |  | |

##### 响应结果示例

{

"code": 1000,

"message": "SUCCESS",

"data": [

{

"allowMaxSpeed": null,

"allowMinSpeed": null,

"inspectAreaGrids": [

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 12,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 13,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 14,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 15,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

}

],

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增测试拍照区域",

"id": 2,

"allowShipType": null,

"portId": "1",

"type": 1,

"direction": null

}

]

}

#### 区域网格新增

##### 接口描述

根据传入的网格数据进行网格数据的新增

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/create\_grids

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"list":[

{

"areaId":1,

"coordinates":[

"123.333,444.444",

"123.333,444.444",

"123.333,444.444",

"123.333,444.444"

],

"name":"A1",

"type":1

},

{

"areaId":1,

"coordinates":[

"123.333,444.444",

"123.333,444.444",

"123.333,444.444",

"123.333,444.444"

],

"name":"A1",

"type":1

}

]

}

##### 参数格式

CreateAreaGrids区域网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| list | List<CreateAreaGrid> | Y | 新增区域网格对象结构 |

List<CreateAreaGrid>

CreateAreaGrid区域网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| name | String | Y | 网格名称 |
| areaId | Integer | Y | 所属区域编号 |
| coordinates | List<String> | Y | 网格坐标点 |
| type | Integer | N | 网格类型 |
| pan | Float | N | 相机参数-左右位置 |
| tilt | Float | N | 相机参数-上下位置 |
| zoom | Float | N | 相机参数-放大倍数 |
| runMode | String | N | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM |
| speedOption | Float | N | 相机参数-速度选项 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域网格编辑

##### 接口描述

根据传入的区域网格数据进行网格数据的更新

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/update\_grids

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"list":[

{

"areaId":1,

"coordinates":[

"123.333,444.444",

"123.333,444.444",

"123.333,444.444",

"123.333,444.444"

],

"id":5,

"name":"A1",

"pan":2.3,

"runMode":"AM",

"speedOption":9.2,

"tilt":2.3,

"type":1,

"zoom":3.4

},

{

"areaId":1,

"coordinates":[

"123.333,444.444",

"123.333,444.444",

"123.333,444.444",

"123.333,444.444"

],

"id":5,

"name":"A1",

"pan":2.3,

"runMode":"AM",

"speedOption":9.2,

"tilt":2.3,

"type":1,

"zoom":3.4

}

]

}

##### 参数格式

UpdateAreaGrids区域网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| list | List< UpdateAreaGrid> | Y | 更新区域网格对象结构 |

List<UpdateAreaGrid>

UpdateAreaGrid区域对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | Integer | Y | 网格编号 |
| areaId | Integer | Y | 所属区域编号 |
| name | String | N | 网格名称 |
| coordinates | List<String> | N | 网格坐标点 |
| type | Integer | N | 网格类型 |
| pan | Float | N | 相机参数-左右位置 |
| tilt | Float | N | 相机参数-上下位置 |
| zoom | Float | N | 相机参数-放大倍数 |
| runMode | String | N | 相机参数-运动模式 绝对移动：AM，相对移动：RM，连续移动：CM，坐标移动：GM |
| speedOption | Float | N | 相机参数-速度选项 |
|  |  |  |  |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域网格删除

##### 接口描述

根据区域网格编号进行网格数据删除

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/delete\_grids

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"ids":[

1,

2,

3,

4,

5

]

}

##### 参数格式

DeleteAreaGrid删除防区网格对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| ids | List<Integer> | Y | 区域网格编号 |
|  |  |  |  |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | XXX | 响应数据 | | XXX | |

##### 响应结果示例

{

"code": 1000,

"msg": "操作成功",

"data": null

}

#### 区域网格查询

##### 接口描述

根据输入的检索条件进行制定卡口区域的网格列表查询

##### 请求方式

restful http接口

URL:http://localhost:port/service/zchx/area/read\_grids

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"areaId":1,

"type":1

}

##### 参数格式

ReadAreaGrid区域对象结构

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| type | Integer | N | 网格类型 |
| areaId | Integer | Y | 区域编号 |

##### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| msg | String | | 提示信息 |  |
| data | List<InspectAreaGrid> | 响应数据 | | 区域网格对象 |

InspectAreaGrid结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| id | Integer | | 区域网格编号 |  |
| name | String | | 网格名称 |  |
| coordinates | String | | 网格坐标点 |  |
| type | Integer | | 网格类型 |  |
| areaId | Integer | 区域编号 | |  |

##### 响应结果示例

{

"code": 1000,

"message": "SUCCESS",

"data": [

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 12,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 13,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 14,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

},

{

"areaId": 2,

"tilt": 0.1,

"coordinates": [

"111.333,123.123",

"111.333,123.123",

"111.333,123.123",

"111.333,123.123"

],

"name": "新增区域网格名称",

"zoom": 0.1,

"id": 15,

"pan": 0.1,

"type": 2,

"runMode": "AM",

"speedOption": 0.1

}

]

}

### 自动拍照

#### 拍照请求

##### 接口描述

根据雷达目标判断船舶是否处于拍照区域内,如果在将该目标推送至相机服务端请求拍照。

##### 推送方式

Type:ZMQ.PUB

Port：5858

Topic：AutoPhoto

接收到的三帧数据结构

第一帧: Topic

第二帧: UTC时间戳

第三帧: proto数据结构的序列化字节

#### 照片回传数据接收

##### 接口描述

接收相机客户端根据上报目标拍照之后回传的照片数据做相应处理。

##### 订阅方式

Type:ZMQ.SUB

Port：5555

Topic：PhotoPush

三帧数据结构

第一帧: Topic

第二帧: UTC时间戳

第三帧: proto数据结构的序列化字节

## 船名识别

### 手动识别船名

#### 接口描述

人口识别船名之后，请求后台修改卡口事件的船名。

#### 请求方式

restful http接口

URL:http://localhost:port/zchx-wbs-server/zchx/shipevent/update

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"id": 9,

" name": “xjjj”

}

#### 参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| id | int | Y | 参数1 |
| name | String | Y | 参数2 |

#### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| message | String | | 提示信息 | OK |
| data | 1 | 响应数据 | | 影响行数 | |

#### 响应结果示例

{"code":1000,"message":"OK","data":1}

## 航行通告

### 播放VHF语音

#### 接口描述

提供给客户端调用，提供VHF通告和警告功能，基于上传的targetId事件编号进行VHF语音消息的捆绑，播报的次数可以通过参数指定

#### 请求方式

restful http接口

URL:http://localhost:port/zchx-wbs-server/zchx/vhf/play\_vhf\_msg

Method:POST

Content-type:application/json;charset=utf-8

Content:

{

"count":4,

"msg":"哈哈",

"targetId":"2",

"type":1

}

#### 参数格式

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必填(y/n) | 描述 |
| targetId | String | N | 目标编号 |
| msg | String | Y | 通告文字 |
| type | Integer | Y | 事件类型1：通告，2：警告 |
| count | Integer |  | 通告次数 |

#### 响应结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | | 描述 | 备注 |
| code | String | | 操作结果 | 1000：成功  1001：失败 |
| message | String | | 提示信息 |  |
| data | Object | 附加数据 | |  | |

#### 响应结果示例

{

"code": 1000,

"message": "SUCCESS",

"data": null

}