
龙船数据 API 开发接口

loongship

目 录

龙船数据 API 开发接口.....	I
说明.....	II
1 基础接口.....	1
1.1 获取加密种子.....	1
1.1.1 接口功能介绍.....	1
1.1.2 接口参数说明.....	1
1.1.3 接口 URL.....	1
1.1.4 返回结果说明.....	1
2 船舶基础信息 API.....	2
2.1 船舶查询服务.....	2
2.1.1 接口功能介绍.....	2
2.1.2 接口参数说明.....	2
2.1.3 接口 URL.....	2
2.1.4 返回结果说明.....	2
2.2 船舶基本信息查询服务.....	3
2.2.1 接口功能介绍.....	3
2.2.2 接口参数说明.....	3
2.2.3 接口 URL.....	4
2.2.4 返回结果说明.....	4
2.3 船舶最新船位查询服务.....	5
2.3.1 接口功能介绍.....	5
2.3.2 接口参数说明.....	5
2.3.3 接口 URL.....	5
2.3.4 返回结果说明.....	5
2.4 船舶基本信息及最新船位查询服务.....	7
2.4.1 接口功能介绍.....	7
2.4.2 接口参数说明.....	7
2.4.3 接口 URL.....	7
2.4.4 返回结果说明.....	7
2.5 船舶历史轨迹查询服务.....	9
2.5.1 接口功能介绍.....	9
2.5.2 接口参数说明.....	9
2.5.3 接口 URL.....	9
2.5.4 返回结果说明.....	9
2.6 船舶历史航次查询服务.....	11
2.6.1 接口功能介绍.....	11
2.6.2 接口参数说明.....	11
2.6.3 接口 URL.....	11
2.6.4 返回结果说明.....	11
2.7 船舶当前航次查询服务.....	13
2.7.1 接口功能介绍.....	13
2.7.2 接口参数说明.....	13
2.7.3 接口 URL.....	13
2.7.4 返回结果说明.....	13

2.8	按区域查询船舶列表服务.....	14
2.8.1	接口功能介绍.....	14
2.8.2	接口参数说明.....	14
2.8.3	接口 URL.....	15
2.8.4	返回结果说明.....	15
2.9	根据经纬度位置查找附近船舶查询服务.....	17
2.9.1	接口功能介绍.....	17
2.9.2	接口参数说明.....	17
2.9.3	接口 URL.....	17
2.9.4	返回结果说明.....	17
3	调用示例.....	18
3.1	java 版.....	18
4	数据字典.....	21
4.1	Ais 船舶类型（2.2、2.4 接口）	21
4.2	船舶状态（2.3、2.4 接口）	21

修订记录

修订日期	版本号	修订描述	修订人	批准人
2016-12-01	V1.0	数据 API 接口定义	唐洁	
2017-07-24	V1.1	增加了 1.5 接口定义	刘江波	
2017-07-26	V1.1.1	1.3、1.5 接口增加了 shiptypename 字段	刘江波	
2017-10-18	V1.1.2	1.3、1.5 接口增加了 nationality 字段	路普	
2017-12-11	V1.2	增加了 1.6、1.7、1.8、1.9 四个接口	刘江波	
2017-12-12	V1.2.1	1.7 接口增加 startUtcTime、endUtcTime 参数	刘江波	
2017-12-21	V1.2.2	1.4、1.5 接口增加了 navStatusName 字段	刘江波	
2018-07-03	V1.2.3	1.9 接口添加返回状态码 19、20	刘江波	
2018-08-08	V1.2.4	1.7、1.8 添加 language 参数	刘江波	
2018-09-16	V1.2.5	新增 1.10, 1.11, 1.12 三个接口	唐洁	
2018-09-28	V1.2.6	新增潮汐接口,修改 1.12 接口参数	唐洁	
2018-10-09	V1.2.7	新增 1.14, 1.15 两个台风接口	唐洁	
2018-10-23	V1.2.8	新增 1.16 船舶详细档案接口	唐洁	

说明

API 是 Application Program Interface 的缩写，即应用程序接口。龙船数据 API 由 Http 接口组成的。

使用这套数据 API 接口，具体步骤如下：

第一步：船达通会给每个 API 客户分配一个唯一 Key

第二步：调用 API 时先获取加密种子，然后使用 md5 算法加密第一步分配的 Key，得到一个

加密后的报文字符串

第三步：调用每个 API 接口时，需要将第二步得到的加密后的报文字符串当作请求参数传递至此用户可以快捷的获取到船舶数据。

该文档主要面向项目经理、开发技术人员，便于了解 API 接口规范和调用。

1 基础接口

编码方式	UTF-8
返回值说明	返回值使用 JSON 方式
调用方式	建议使用 HTTP POST 方式

1.1 获取加密种子

1.1.1 接口功能介绍

此方法返回年月日字符串，如 20150324。客户在调用 api 接口时，用此字符串和分发的 key 组合并进行 md5 加密生成新字符串来传递给接口，如分发的 key 为：e10adc3949ba59abbe56e057f20f883e，则用户在使用接口时，需要先调用该接口得到 20150324，然后 md5（20150324+e10adc3949ba59abbe56e057f20f883e）生成的 key，调用数据 API 每个接口都传入该 key。

1.1.2 接口参数说明

参数名称	参数格式	参数含义	备注

1.1.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getComKey>

1.1.4 返回结果说明

```
{
  "seedStr": "20150324",
  "status": 0
}
```

返回参数	参数说明
status	返回状态
seedStr	加密种子字符串

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员

0	成功

2 船舶基础信息 API

2.1 船舶查询服务

2.1.1 接口功能介绍

根据 MMSI、船名等模糊查船，目的是获取船舶唯一 ID，对于返回值超过 100 条的，按排序优先级返回前 100 条数据。

2.1.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
term	term=福远渔	船名或 MMSI 或 IMO 或呼号	必填参数

2.1.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipId?term=qingdao&key=5cdd7336161c4ce2a27a21d770b8aa53>

2.1.4 返回结果说明

```
{
  "result": [
    {
      "callsign": "413321",
      "shipid": "412704230",
      "shipname": "FU YUAN YU 38",
      "imo": "9132961"
    }
  ],
  "status": 0
}
```

返回参数	参数说明
status	返回状态

result	返回信息(json 字符串数组)
shipid	船舶 mmsi
shipname	船舶名称
callsign	船舶呼号
imo	Imo 号

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有权限,请与管理员联系
5	请求参数 term 为空
6	请求参数 key 为空
7	没有查询到任何数据
8	查询结果集数据集合大于 20 条,按排序优先级返回前 20 条数据。

2.2 船舶基本信息查询服务

2.2.1 接口功能介绍

根据船舶 ID, 获取最新的船舶基本静态信息; (单一/批量, 批量最多不超过 20 条)

2.2.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=420368290	船舶 mmsi (批量用逗号分隔, 表示多条船舶 mmsi)	必填参数

2.2.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipBasicInformation?shipid=235057836&key=5cdd7336161c4ce2a27a21d770b8aa53>

2.2.4 返回结果说明

```
{
  "result": [
    {
      "callsign": "413321",
      "mmsi": "412704230",
      "shipname": "FU YUAN YU 38",
      "imo": "9132961",
      "shiptype": "46",
      "shiptypename": "货船",
      "length": "52",
      "breadth": "9",
      "eta": "2015-03-26 15:00",
      "dest_port": "QING DAO",
      "nationality": "中国",
      "draught": "3.8"
    }
  ],
  "status": 0
}
```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
mmsi	mmsi
shipname	船舶名称
callsign	船舶呼号
imo	Imo 号
shiptype	船舶类型
shiptypename	船舶类型名称
length	船长
breadth	船宽
eta	预到时间(mm-dd hh:MM)
dest_port	目的地
draught	吃水
nationality	船籍

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 Shipid 为空
6	请求参数 key 为空
7	没有查询到任何数据
8	批量查询结果不能超过 20 条

2.3 船舶最新船位查询服务

2.3.1 接口功能介绍

根据船舶 ID, 获取最新船位; (单一/批量, 批量最多不超过 20 条)

2.3.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=420368290	船舶 mmsi (批量用逗号分隔, 表示多条船舶 mmsi)	必填参数

2.3.3 接口 URL

URL 请求地址:

[http://api.shipdt.com/DataApiServer/
getShipLatest?shipid=235057836&key=5cdd7336161c4ce2a27a21d770b8aa53](http://api.shipdt.com/DataApiServer/getShipLatest?shipid=235057836&key=5cdd7336161c4ce2a27a21d770b8aa53)

2.3.4 返回结果说明

```
{
  "result": [
    {
      "postime": "1426572584",
      "longitude": "2944204",
      "latitude": "31424344",
```

```

        "course": "71.5",
        "heading": "71",
        "speed": "8.8",
        "navStatusName": "在航(主机推动)",
        "mmsi": "413378420",
        "navStatus": "0"
    }
],
"status": 0
}

```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
postime	位置时间 (UTC 时间)
mmsi	mmsi
longitude	经度(经度: 1/10000 分)
latitude	纬度(纬度: 1/10000 分)
course	航向(单位度)
heading	航首向(单位度)
speed	船速(单位节)
navStatus	航行状态
navStatusName	航行状态名称

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 Shipid 为空
6	请求参数 key 为空
7	没有查询到任何数据
8	批量查询结果不能超过 20 条

2.4 船舶基本信息及最新船位查询服务

2.4.1 接口功能介绍

根据船舶 ID，获取船舶基本信息及最新船位；（单一/批量，批量最多不超过 20 条），该接口返回的数据可以认为是 1.3、1.4 两个接口的合集。

2.4.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=420368290	船舶 mmsi（批量用逗号分隔，表示多条船舶 mmsi）	必填参数

2.4.3 接口 URL

URL 请求地址:

[http://api.shipdt.com/DataApiServer/
getShipLatestInformation?shipid=235057836&key=5cdd7336161c4ce2a27a21d770b8aa53](http://api.shipdt.com/DataApiServer/getShipLatestInformation?shipid=235057836&key=5cdd7336161c4ce2a27a21d770b8aa53)

2.4.4 返回结果说明

```
{
  "result": [
    {
      "imo": "",
      "eta": "",
      "breadth": "",
      "speed": "0.0",
      "draught": "",
      "postime": "1497618507",
      "mmsi": "413989528",
      "shipname": "",
      "callsign": "",
      "shiptype": "",
      "dest_port": "",
      "dest_std": "YANGPU,CN",
      "destcode": "CNYPG",
      "navStatusName": "在航(主机推动)",
    }
  ]
}
```

```

        "course": "295.3",
        "shiptypename": "货船",
        "length": "",
        "longitude": 67873850,
        "latitude": 13787304,
        "heading": "",
        "navStatus": "0"
    }
],
"status": 0
}

```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
postime	位置时间 (UTC 时间)
mmsi	mmsi
shipname	船舶名称
callsign	船舶呼号
imo	Imo 号
shiptype	船舶类型
shiptypename	船舶类型名称
length	船长
breadth	船宽
eta	预到时间(mm-dd hh:MM)
dest_port	目的地
dest_std	港口名称, 国家两字代码
destcode	港口五字代码
draught	吃水
nationality	船籍
longitude	经度(经度: 1/10000 分)
latitude	纬度(纬度: 1/10000 分)
course	航向(单位度)
heading	航首向(单位度)
speed	船速(单位节)
navStatus	航行状态
navStatusName	航行状态名称

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系

3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 Shipid 为空
6	请求参数 key 为空
7	没有查询到任何数据
8	批量查询结果不能超过 20 条

2.5 船舶历史轨迹查询服务

2.5.1 接口功能介绍

根据船舶 ID，时间范围获取船舶历史轨迹；（最多查询过去 4 个月的历史轨迹）

2.5.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=420368290	船舶 mmsi	必填参数
startUtcTime	startUtcTime=1512108577	开始时间(utc 时间戳)	必填参数
endUtcTime	endUtcTime=1512972577	结束时间(utc 时间戳)	必填参数

2.5.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipHistoryTrack?shipid=413472680&key=5cdd7336161c4ce2a27a21d770b8aa53&startUtcTime=1512108577&endUtcTime=1512972577>

2.5.4 返回结果说明

```
{
  "result": [
    {
      "course": 323.3,
      "longitude": 66258251,
      "speed": 9.3,
      "postime": "1512108963",
      "latitude": 12667825,
      "heading": 326
    },
    Object{...},
  ]
}
```

```

    Object{...},
    Object{...},
    Object{...}
  ],
  "status": "0",
  "specialResult": [
    {
      "course": 288.5,
      "longitude": 66325730,
      "speed": 0,
      "postime": "1512730392",
      "navStatus": "3",
      "latitude": 12651217,
      "heading": 279
    },
    Object{...},
    Object{...},
    Object{...},
    Object{...}
  ]
}

```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
specialResult	返回特殊点信息(在航、靠泊、停泊、锚泊)
postime	位置时间 (UTC 时间)
longitude	经度(经度: 1/10000 分)
latitude	纬度(纬度: 1/10000 分)
course	航向(单位度)
heading	航首向(单位度)
speed	船速(单位节)
navStatus	航行状态(1:在航;2:靠泊;3:锚泊;4:停泊)

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 shipid 为空
6	请求参数 key 为空

7	请求参数 startUtcTime 为空或格式不对
8	请求参数 endUtcTime 为空或格式不对
9	查询时间超过范围
10	结束时间应大于开始时间
11	没有查询到任何数据

2.6 船舶历史航次查询服务

2.6.1 接口功能介绍

根据船舶 ID，获取船舶历史航次；（最多查询过去 4 个月的历史航次）

2.6.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=422814000	船舶 mmsi	必填参数
startUtcTime	startUtcTime=1511008577	开始时间(utc 时间戳)	必填参数
endUtcTime	endUtcTime=1512972577	结束时间(utc 时间戳)	必填参数
language	language=en	港口名称语言	选填参数， 默认 cn，cn 表示返回的港口名称优先是中文， en 表示返回的港口名称是英文

2.6.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipHistoryVoyage?shipid=413472680&key=5cdd7336161c4ce2a27a21d770b8aa53&startUtcTime=1511008577&endUtcTime=1512972577&language=cn>

2.6.4 返回结果说明

```
{
  "result": [
```



```

{
  "departportname": "BANDAR-E SHAHID REAJIE",
  "ata": "1512023577",
  "departtime": "1511036961",
  "totalvoyage": 1048.17,
  "arrivedportname": "BANDAR ABBAS"
},
{
  "departportname": "BUSHEHR",
  "ata": "1511025676",
  "departtime": "1510158558",
  "totalvoyage": 669.25,
  "arrivedportname": "BANDAR-E SHAHID REAJIE"
}
],
"status": "0"
}

```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
departportname	出发港名称
arrivedportname	目的港名称
ata	实际到港时间(utc 时间)
departtime	离港时间(utc 时间)
totalvoyage	航程(单位:海里)

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 Shipid 为空
6	请求参数 key 为空
7	请求参数 startUtcTime 为空或格式不对
8	请求参数 endUtcTime 为空或格式不对
9	查询时间超过范围
10	结束时间应大于开始时间
11	没有查询到任何数据
12	请求参数 language 错误

2.7 船舶当前航次查询服务

2.7.1 接口功能介绍

根据船舶 ID，获取船舶当前航次

2.7.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
shipid	shipid=420368290	船舶 mmsi	必填参数
language	language=en	港口名称语言	选填参数， 默认 cn，cn 表示返回的 港口名称优 先是中文， en 表示返回 的港口名称 是英文

2.7.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipCurrentVoyage?shipid=413472680&key=5cdd7336161c4ce2a27a21d770b8aa53&language=cn>

2.7.4 返回结果说明

```
{
  "result": {
    "departportname": "阳江港",
    "eta": "07-29 22:30",
    "anchorportname": "湛江港",
    "anchortime": "1512730392",
```

```

    "departtime": "1511752014",
    "arrivingportname": "ZHANJIANG"
  },
  "status": "0"
}

```

返回参数	参数说明
status	返回状态
result	返回信息(json 字符串数组)
departportname	出发港名称
departtime	离港时间(utc 时间)
anchorportname	停靠泊港名称
anchortime	停靠泊时间(utc 时间)
arrivingportname	预到港名称
eta	预到时间

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 Shipid 为空
6	请求参数 key 为空
7	没有查询到任何数据
8	请求参数 language 错误

2.8 按区域查询船舶列表服务

2.8.1 接口功能介绍

根据经纬度区域, 获取船舶列表

2.8.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数

xmin	xmin=72754470	最小经度(1/10000 分)	必填参数
ymin	ymin=18905471	最小纬度(1/10000 分)	必填参数
xmax	xmax=72869930	最大经度(1/10000 分)	必填参数
ymax	ymax=18951866	最大纬度(1/10000 分)	必填参数
hour	hour=1	最近几小时船位(范围:1-24)	必填参数

说明：经度之差、纬度之差最大为 0.5 度。

2.8.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipListByRect?xmin=72754470&ymin=18905471&xmax=72869930&ymax=18951866&hour=1&key=5cdd7336161c4ce2a27a21d770b8aa53>

2.8.4 返回结果说明

```
{
  "total":129,
  "result": [
    {
      "course": "170.7",
      "breadth": 10,
      "speed": "0.1",
      "length": 49,
      "postime": "1512972823",
      "longitude": "72767680",
      "shipname": "SUYANCHENGHUO215099",
      "latitude": "18905480",
      "mmsi": "413850305",
      "shiptype": "高速船",
      "heading": "0"
    },
    Object{...},
    Object{...},
    Object{...},
    Object{...},
    Object{...},
    Object{...}
  ],
  "status": "0"
}
```

返回参数	参数说明
status	返回状态
total	返回总船舶数
result	返回信息(json 字符串数组)

mmsi	mmsi
shipname	船舶名称
shiptype	船舶类型名称(对应字典 shiptypename)
length	船长
breadth	船宽
longitude	经度(经度: 1/10000 分)
latitude	纬度(纬度: 1/10000 分)
course	航向(单位度)
heading	航首向(单位度)
speed	船速(单位节)
postime	位置时间 (UTC 时间)

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 key 为空
6	请求参数 xmin 为空
7	请求参数 ymin 为空
8	请求参数 xmax 为空
9	请求参数 ymax 为空
10	请求参数 xmin 数据类型错误
11	请求参数 ymin 数据类型错误
12	请求参数 xmax 数据类型错误
13	请求参数 ymax 数据类型错误
14	请求参数 hour 范围必须在 1-24 之间
15	请求参数 hour 数据类型错误
16	请求参数 hour 为空
17	没有查询到任何数据
18	只能获取区域内 2000 条船舶列表
19	经度之差不能超过 0.5 度
20	纬度之差不能超过 0.5 度

2.9 根据经纬度位置查找附近船舶查询服务

2.9.1 接口功能介绍

根据位置信息列出 xx 米（可变数据默认为 400 米）范围内实时的船只，返回 xx 米（可变数据默认为 400,最大不超过 10000 米）范围内的所有船只信息列表。

2.9.2 接口参数说明

参数名称	参数格式	参数含义	备注
key	key=5cdd7336161c4ce2a27a21d770b8aa53	Api 用户 32 位字符串 key(使用 md5 加密后的字符)	必填参数
lon	lon = 120.212	经度(以度为单位,正负 180 之内)	必填参数
lat	lat=40.20	纬度(以度为单位,正负 90 之内)	必填参数
distance	distance =400	距离(米)	选填参数

2.9.3 接口 URL

URL 请求地址:

<http://api.shipdt.com/DataApiServer/getShipByNear?lon=120.212&lat=40.20&distance=4000&key=5cd7336161c4ce2a27a21d770b8aa53>

2.9.4 返回结果说明

```
{
  "total":129,
  "result": [
    {
      "shipname":"BO HAI JING ZHU",
      "mmsi":"414095000",
      "imo":"9584815"
    },
    Object {...},
    Object {...},
    Object {...},
    Object {...},
    Object {...},
    Object {...}
  ],
}
```

```
"status": "0"
}
```

返回参数	参数说明
status	返回状态
total	返回总船舶数
result	返回信息(json 字符串数组)
mmsi	mmsi
shipname	船舶名称
lmo	船舶 mmsi 号

返回状态码说明:

返回参数	参数说明
-1	后台查询数据失败,请联系管理员
0	成功
1	Key 不存在,请重新获取 key 再调用,请与管理员联系
2	API 接口权限过期,请与管理员联系
3	用户 key 已被禁用,请与客户管理员联系
4	当前 api 用户已经没有任何权限,请与管理员联系
5	请求参数 key 为空
6	请求参数 lon 为空
7	请求参数 lon 数据类型错误
8	请求参数 lat 为空
9	请求参数 lat 数据类型错误
10	请求参数 distance 数据类型错误
11	请求参数 distance 范围必须在 1-10000 之间
12	没有查询到任何数据

3 调用示例

3.1 java 版

```
public final static String MD5(String s) {
    char
    hexDigits[]={ '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c',
    'd', 'e', 'f' };
    try {
        byte[] btInput = s.getBytes();
        // 获得MD5摘要算法的 MessageDigest 对象
        MessageDigest mdInst = MessageDigest.getInstance("MD5");
        // 使用指定的字节更新摘要
```

```

mdInst.update(btInput);

// 获得密文
byte[] md = mdInst.digest();

// 把密文转换成十六进制的字符串形式

int j = md.length;
char str[] = new char[j * 2];
int k = 0;
for (int i = 0; i < j; i++) {
    byte byte0 = md[i];
    str[k++] = hexDigits[byte0 >>> 4 & 0xf];
    str[k++] = hexDigits[byte0 & 0xf];
}
return new String(str);
} catch (Exception e) {
    e.printStackTrace();
    return null;
}
}

```

```

public static String getComKey() throws Exception {
    String url = "http://api.shipdt.com/DataApiServer/getComKey";
    HttpClient httpClient = new HttpClient();
    HttpMethod method = postMethod(url, null);
    httpClient.executeMethod(method);
    String response = method.getResponseBodyAsString();
    return response;
}

```

/**

* 1.1 船舶查询服务

*

* @throws Exception

*/

```

public static void getShipId() throws Exception {
    String url="http://api.shipdt.com/DataApiServer/getShipId";

```

//请求地址

String comKey=getComKey(); //获取加密种子

JSONObject json=JSONObject.fromObject(comKey);

String key="4297f44b13955235245b2497399d7a93";//船达通分配给用

户的key


```

HttpClient httpClient = new HttpClient();
Map<String, Object> map = new HashMap<String, Object>();
map.put("key", MD5(json.getString("seedStr")+key));
map.put("term", "qingdao");
HttpMethod method = postMethod(url, map);
httpClient.executeMethod(method);
String response = method.getResponseBodyAsString();
System.out.println(response);
}

```

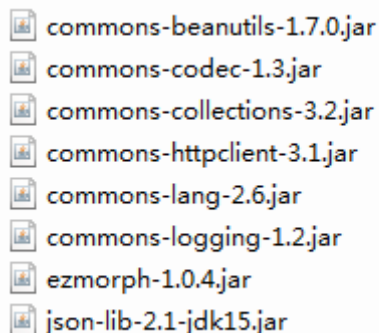
private static HttpMethod **postMethod**(String url, Map<String, Object> params)









```

    throws IOException {
        PostMethod post = new PostMethod(url);
        post.setRequestHeader("Content-Type",
            "application/x-www-form-urlencoded;charset=utf-8");
        List<NameValuePair> list = new ArrayList<NameValuePair>();
        if (params!=null&&!params.isEmpty()) {
            Set<String> keySet = params.keySet();
            for (String key : keySet) {
                NameValuePair param = new NameValuePair(key,
                    (String) params.get(key));
                list.add(param);
            }
        }
        NameValuePair[] array = new NameValuePair[list.size()];
        list.toArray(array);
        post.setRequestBody(array);
        post.releaseConnection();
        return post;
    }

```

注: 本示例依赖



-  commons-beanutils-1.7.0.jar
-  commons-codec-1.3.jar
-  commons-collections-3.2.jar
-  commons-httpclient-3.1.jar
-  commons-lang-2.6.jar
-  commons-logging-1.2.jar
-  ezmorph-1.0.4.jar
-  json-lib-2.1-jdk15.jar

依赖Jar包下载地址:

<https://share.weiyun.com/23eeff69fe61f5f40136085f2a7c7334>

4 数据字典

4.1 Ais 船舶类型（2.2、2.4 接口）

shiptype	shiptypename
46	货船
47	集装箱船
48	油船
49	渔船
50	客船
51	拖轮/引航船
52	工作船
53	其它
54	其它
56	拖引船
57	拖轮
58	引航船
59	潜水作业
60	挖泥/水下作业
61	港口供应船
62	执法船
63	医务运输船
64	军事作业船
65	防污染设备船
66	搜救船
67	高速船
68	特种船
69	交管设备
70	地效翼船
71	其它
72	18 号决议规定的船舶
73	保留
74	助航设施
75	备用
76	休闲游艇
77	帆船

4.2 船舶状态（2.3、2.4 接口）

navStatus	navStatusName
0	在航（主机推动）
1	锚泊

2	失控
3	操纵受限
4	吃水受限
5	靠泊
6	搁浅
7	从事捕鱼
8	靠船帆提供动力
9	保留将来修订之用
10	保留将来修订之用
11	预留供将来使用
12	预留供将来使用
13	预留供将来使用
14	AIS-SART
15	未定义