自动配载算法接口调用说明

|  |  |
| --- | --- |
| 项目代号 |  |
| 项目名称 | algo-transfer |
| 文档编号 |  |
| 密级 |  |
| 版本 | 4.0 |
| 日期 | 2020-03-31 |

更改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 更改内容 | 更改人/日期 | 审核人/日期 | 批准人/日期 |
|  | V1.0 | 初稿 | 刘阳  2020.03.31 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目 录**

[1 前言 4](#_Toc36539807)

[2 接口协议 4](#_Toc36539808)

[3 通用返回格式 4](#_Toc36539809)

[4 算法接口 5](#_Toc36539810)

[5 测试服务器 7](#_Toc36539811)

# 前言

本文档为不同版本调用自动配载算法接口说明文档，通过http请求的方式传递指定参数并返回配载结果

# 接口协议

协议：支持http 1.0/1.1 协议

请求方式：方法采用POST请求

数据格式：JSON

报文头：Content-Type=application/json，Content-Encoding=UTF-8

# 通用返回格式

通用返回数据结构（JSON对象）如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **必须** | **默认值** | **备注** |
| **code** | string | 是 | "0" | 成功返回0，失败返回-1或其他错误码。详见code码 |
| **message** | string | 是 | "操作成功！" | 失败返回错误信息 |
| **data** | jsonObject/array/string |  | null | 返回所请求的数据 |

code码：

|  |  |
| --- | --- |
| **code值** | **含义** |
| **0** | 成功 |
| **-1** | 失败 |

示例：

|  |
| --- |
| {  "code": "0",  "message": "操作成功！",  "data": null  } |

# 算法接口

地址：

|  |  |  |
| --- | --- | --- |
| **名称** | **值** | **备注** |
| **接口地址** | http://ip:port/api/v4/transfer/getAlgoResult | 算法接口请求 |

参数：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数（URL）** | 说明 | 必须 | 备注 |
| **shipVisitId** | 船舶艘次号Id | 是 |  |
| **facilityId** | 码头代码 | 是 |  |
| **version** | 对应的数据库版本,4.0、5.0必须传 |  | 默认7.0 |

算法配置参数：放到body中，不传执行默认参数。

示例：

{

"algoRun": {

"bayOrderType": "1",

"language": "EN"

}

}

注意：不传配置参数，body要传{"algoRun":{}} 不可以不传

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **默认值** | **含义** | **备注** |
| **bayOrderType** | string |  | 1 | 贝排序 | 1=按行左右  2=按行中间  3=按列逐层  4=按层交替  5=单侧阶梯  6=对称阶梯 |
| **language** | string |  | 中文 | 语言 | EN: 英语 |
| **bayDirectionType** | string |  | 1 | 贝填充方向 | 1=右舷优先  2=左舷优先 |
| **bayOrderTypeCustom** | string |  |  | 定义单独的排序 | 奇数贝号:方向，多个逗号分隔。如 001:1,003:2 |
| **bayDirectionTypeCustom** | string |  |  | 定义单独的方向 | 奇数贝号:方向，多个逗号分隔。如 001:1,003:2 |
| **yardOrderType** | stirng |  | 1 | 堆场场区排序 | 1=按行正序  2=按列正序  3=按行倒序  4=按列倒序 |
| **yardOrderTypeCustom** | stirng |  |  | 定义每个场区的排序 | cyNo:排序  如：01:1,02:2 |
| **weightDeckLevel** | int |  | 1 | 舱上重量平分 | 越大，重量适配范围越窄 |
| **weightHoldLevel** | int |  | 1 | 舱下重量平分 | 越大，重量适配范围越窄 |
| **stowReq** | int |  | 0 | 配载需求 | 0：效率优先  1：翻倒优先  2：机械优先 |
| **yardMultiCrane** | boolean |  | false | 堆场对应多岸桥 | 开启：堆场对多岸桥同时出箱  关闭：堆场相近箱对应同一岸桥。 |
| **algoCategory** | int |  | 0 | 选择算法 | 0：贝内优化  1：贪婪算法（大学老师v1）  2：全局优化（遗传算法）  3：贪婪算法（大学老师v2） |
| **fieldsValues** | json\_array[string] |  |  | 分组条件 | 箱和预配匹配的条件：默认 ["unitSizeCode", "unitTypeCode", "emptyFullMark", "podCode"]。  可选值：[unitSizeCode, unitTypeCode, emptyFullMark, podCode, undgNo, polCode, finalPodCode, unitCategory, isHazard, imdgClassCode, isReefer, isBundled]（尺寸,箱型,空重,卸货港,危规编号,装货港,目的港,流向类别,危品,危险等级,冷藏,折叠） |
| **populationSize** | int |  | 1000 | 规模 | 全局优化下启用的参数 |
| **limitSeconds** | int |  | 30 | 运算时长 | 全局优化下启用的参数：默认30秒 |
| **weightFactor** | int |  | 1 | 上轻下重 | 全局优化下启用的参数 |
| **turnOverFactor** | int |  | 1 | 翻倒次数 | 全局优化下启用的参数 |
| **bayCenterFactor** | int |  | 20 | 贝集中 | 全局优化下启用的参数 |
| **bayLRWgtFactor** | int |  | 1 | 贝左右平衡 | 全局优化下启用的参数 |
| **unitContinuousFactor** | int |  | 30 | 出箱连续 | 全局优化下启用的参数 |

**stowReq：配载需求详细介绍：**

效率优先：在不考虑翻捣的情况下，堆场同一贝内连续出箱，保证船上贝内尽量连续放箱，岸桥作业效率最高。翻捣量的多少跟收箱堆放有关，如果翻捣量大的情况下也会影响效率，所以就是一个相对效率最优的方案。

翻捣优先：主要从避免堆场翻捣的情况下出箱，当一个场内箱子出现翻捣时，就跳到下一个符合配载要求的场地出箱，当无法避免翻捣时，就从最小翻捣量的贝出箱。弊端，会造成机械频繁移动（收箱分散堆放的情况下）

机械优先：配载开始时认为每一个场地都配有一个机械，且机械位置在最先出箱的贝（默认从小贝到大贝），尽量集中出完一个贝的箱子，再移动到最近的下一贝出箱，避免机械长距离移动或来回移动，同样受限于堆场堆放情况和预配要求，有些情况下会造成岸桥等箱，效率不保证。

配载结果对象：**unitBeanList属性**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **长度** | **含义** | **备注** |
| **unitNo** | string |  | 箱号 |  |
| **unitVisitId** | string |  | 箱ID |  |
| **shipManifestId** | string |  | 清单ID |  |
| **position** | string |  | 配载位置 |  |
| **orderNo** | string |  | 整船序号 | 对应每个岸桥的装箱顺序，每个岸桥从序号1开始。一个岸桥的双吊箱的序号相同。 |
| **orderNoBay** | string |  | 贝内序号 | 每个贝内的配载位置的先后顺序。注意：双吊箱的配载位置在两个贝上，不在同一贝内，不要混淆 |
| **bridgeOrder** | string |  | 岸桥序号 | 对应的岸桥序号 |
| **oppStowLoc** | string |  | 搭对贝号 |  |
| **oppStowUnitNo** | string |  | 搭对箱号 |  |

请求成功：

|  |  |  |
| --- | --- | --- |
| **代码** | 说明 | 备注 |
| **unitBeanList** |  |  |
| **code** | 0表示成功。-1表示失败 |  |
| **data** | 请求成功为null |  |
| **detailMsg** | 算法具体分析 |  |
| **versionInfo** | 版本信息 |  |
| **message** | 概括信息，请求失败为失败信息 |  |

示例：

请求地址：

<http://ip:port/api/v4/transfer/getAlgoResult?facilityId=NICT&shipVisitId=78607>

请求成功返回结果示例如下：

|  |
| --- |
| {  "unitBeanList": [  {  "unitNo": "HJLU1079164",  "unitVisitId": "7277541",  "shipManifestId": "A0F600D02AE30304E0530AC500C9E47A",  "position": "0130102",  "orderNo": "0",  "orderNoBay": "1",  "bridgeOrder": "1",  "oppStowLoc": "0150102",  "oppStowUnitNo": "TEMU0553911",  "queueNo": "013H"  }  ],  "code": "0",  "data": null,  "detailMsg": " 预配数量:2062 。。。。。。}。",  "versionInfo": "name:algo-server 。。。。。。",  "message": " 预配数量:2062。。。。。。"  } |

# 测试服务器地址

测试地址：172.16.50.2:20040

http:// 172.16.50.2: 20040/api/v4/transfer/getAlgoResult

# 变更日志

### V4.1

1 自动配载conf。添加language参数，改变返回结果的语种

2 获取单号：url参数添加shipVisitId

### V4.0

0 api版本：v4

1获取单号后，后续url不在传递facilityId，shipVisitId等信息，只传递orderInfoId唯一标志

2 各实体删除shipVisitId字段

3 删除其他非必要实体：如船结构，堆场结构，由服务器自动生成

4 自动配载-参数配置：新增堆场排序和贝排序的配置