物业自助终端及服务平台体系结构设计文档

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 草稿 | Zrz |  | 2016-6-15 | 开始合并所有的设计文档 |
| 变更 | Zrz |  | 2016-6-16 | 版本1.1A、B，需求相关变更 |
| 变更 | Zrz |  | 2016-6-21 | 版本1.2A  与产品部门确定采用“自助终端+互联网服务平台方案”（方案A）；  取消物管系统电子票开票状态回写接口；  取消物管系统开票接口，物管系统（如极致软件）中不再集成电子票功能；  服务平台中暂不开发缴费/交费网页界面，统一由自助终端承接之；  服务平台一期暂不处理报修业务 |
| 变更 | Zrz |  | 2016-6-22 | 版本1.3A、B  修改服务平台核心实体关系图 |
| 变更 | Zrz |  | 2016-7-26 | 版本1.4：  1.修改UI多语言支持方案，提供.NET平台实现思路；  2.更新缓存解决方案，提供设计思路及示例代码；  3 增加分布式会话及单点登录解决方案，提供设计思路及示例代码； |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 引言

## 目的

系统用于支持自助终端系统中，【服务缴费】这个主要的业务过程，系统属于带有行业性质的O2O服务性电子商务系统，包括O2O电子商务中常见的集束服务，在线支付，会员营销等内容，同时还包括物业相关的业务内容及税务相关的防伪税控工作流。本文档主要用于描述业务系统的组成，功能细节，以及实现细节需要调用的相关接口。接口定义和接口实现的逻辑部分需要参考 系统的接口设计文档。

## 定义、首字母缩写词和缩略语

|  |  |
| --- | --- |
| **定义、首字母缩写词和缩略语** | **描 述** |
| DEMO、原型 | 系统在设计之前有静态页面描述的原型系统，页面的展示效果和交互效果请以原型为准 |
| 规格说明 | 指代《航天物业电子发票\_产品需求文档》 |
| 智慧云社区 / 服务平台 | 规格说明中称“PC端后台系统” |
|  |  |

## 阅读对象

|  |  |
| --- | --- |
| **阅读对象** | **阅读建议** |
| 需求 | 概要设计：请重点阅读第5章节内容，核实与外部系统对接的前期准备及约定是否已完成  详细设计：请重点阅读第6,7,8章节内容，核实是否覆盖需求 |
| 编码 | 除了需要开发的功能点，请重点阅读第5章节中相关的内容 |
|  |  |
|  |  |

## 参考资料

|  |  |
| --- | --- |
| **名 称** | **描 述** |
| 《航天物业电子发票\_产品需求文档\_20160615更新》 |  |
|  |  |
|  |  |

## 文档概述

第2章节主要描述系统的整体结构

第3,4章节主要对需求进行分析，并得出设计的相关约束和主要的设计策略

第5章节主要描述系统架构相关的设计内容

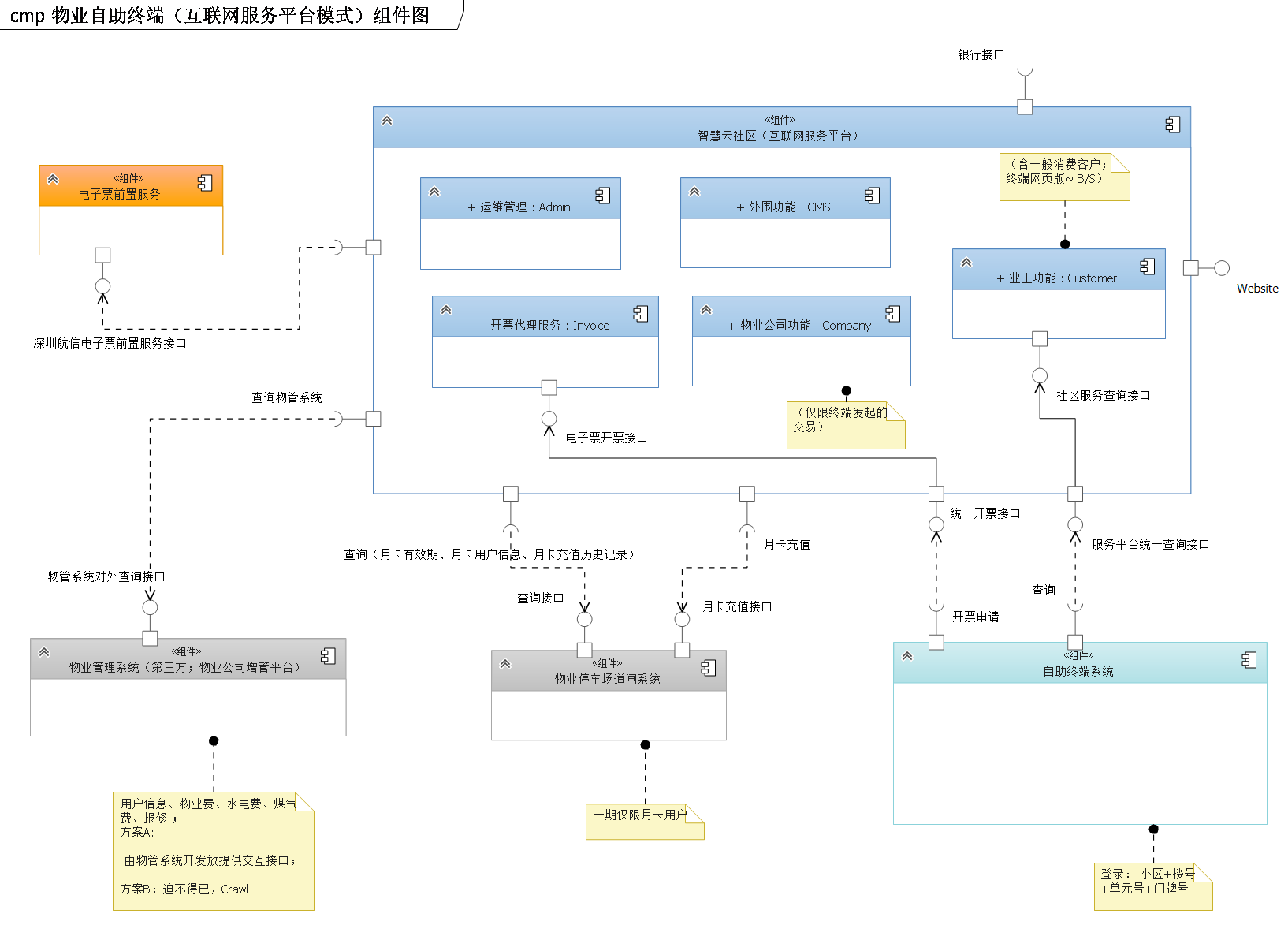
~~第6,7,8章节描述系统的各个组成部分的详细设计内容~~

~~第9章节描述用于支持系统的后台子系统部分的详细设计内容~~

~~第10章节描述系统的运行环境~~

# 系统概述

## 功能模块划分



1. 系统总共划分为五个组件：

A自助终端功能；

B社区服务云平台功能；

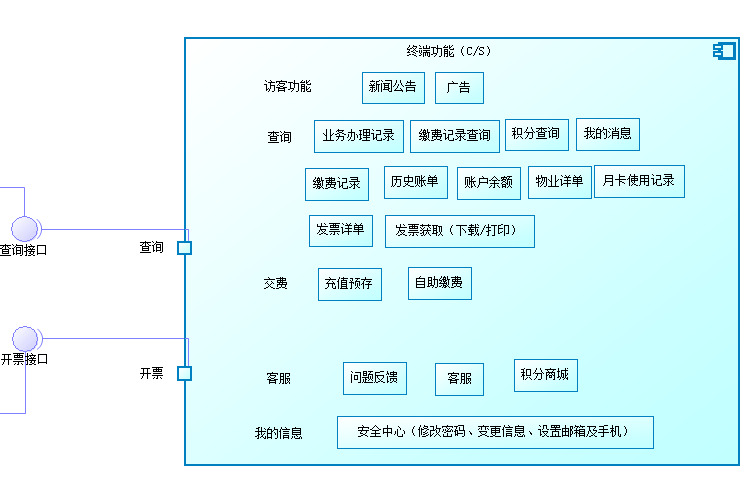
C电子票前置服务；

D物业管理软件；

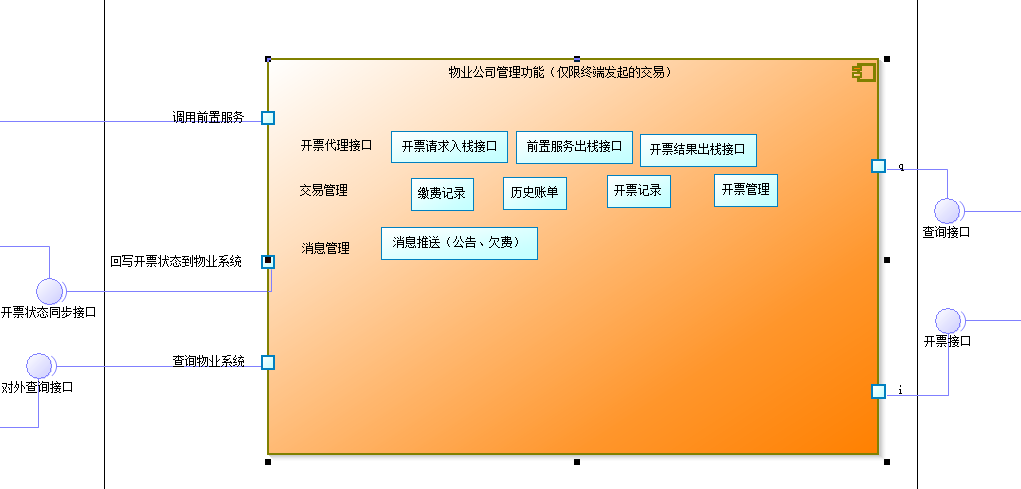
E物业停车场道闸管理软件。

其中D和E目前采用第三方实现；C 已由深圳航信电子票项目组实现。

1. 自助终端功能主要包括：
2. 访客功能
3. 查询
4. 交费
5. 客服
6. 用户中心

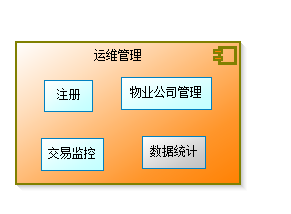


1. 社区服务云平台功能功能主要包括：
2. 开票代理接口
3. 交易管理
4. 消息管理

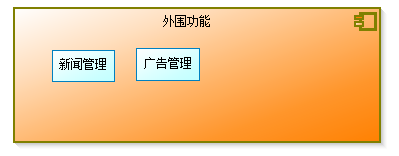


1. 运维管理功能主要包括支持系统运行相关的管理的监控功能。
2. 所有物业公司的基础字典，除了物业公司自己维护之外，客服需要权限进行操作。
3. 运维控制所有的用户注册和公司注册功能。
4. 运维还包括交易数据的监控以及异常处理功能。

系统上线阶段，可能的情况是，基础数据的初始化工作是由物业公司提供，或者从其他系统获取，通过运维的功能初始化到系统中。物业公司仅处理数据的核对和变更工作。



1. 外围功能包括的是网站结构中带有的功能，主要包括
2. 新闻动态，~~这部分内容预计使用深圳航天信息CMS，不再重新开发~~。
3. 论坛，暂不实现
4. 广告管理



1. 接口部分。
2. 物业公司需要维护的信息需要提供导入的接口，以方便物业公司对信息的维护工作；
3. 物业公司物管系统需提供常规业务查询接口，以便后台管理系统/自助终端获取业务信息；
4. 物业停车场道闸管理系统需提供查询和缴费接口；

# 设计约束

## 3.1需求约束

|  |  |
| --- | --- |
| **约束** | **描 述** |
| **功能约束** | 需要提供菜单、功能、按钮级权限控制 |
| **功能约束** | 需要支持对企业管理员授权、收权、停用；企业管理员对企业操作人员的注册、授权； |
| **功能约束** | 需要支持对数据行按指定字段值范围，进行授权使用，包括查看、修改、删除、审批、确认等，常用的如公司信息、物业费明细管理、折扣率或优惠额等 |
| **功能约束** | 平台管理员只能管理、授权用户企业管理员、平台操作员权限，无业务操作权限 |
| **功能约束** | 系统必须提供完善的审计功能，对系统关键数据的每一次增加、修改和删除都能记录相应的修改时间、操作人和修改前的数据记录 |
| **功能约束** | 需要英语、汉语（简体 、繁体）支持，包括界面、内容、提示预警等所有信息；一期不支持除英、汉(简体、繁体)之外的语言，但需要能方便扩展更多语言的支持 |
| **功能约束** | 需要跨浏览器支持，包括IE8及以上，Chrome，360和腾讯浏览器；一期不支持火狐浏览器、Opera浏览器，但需方便扩展更多浏览器支持 |
| **功能约束** | 平台支持超级管理员与平台管理员角色，平台超极管理员只能管理、授权平台管理员，无业务操作权限 |
| **~~功能约束~~** | ~~支持根据不同语言使用不同的日期格式~~ |
| **用户界面约束** | 系统发现用户提交有误信息，必须以弹出窗口的形式明确提示用户错误的原因，并把界面控制焦点置于发生错误的控件对象上。 |
| **用户界面约束** | 系统支持操作员登录后，不超过三次鼠标点击，即可访问业务所需功能 |
| **安全性约束** | 系统在解决跨站攻击、注入攻击等系统性安全问题上，具体标准参照《OWASP Top10–2013》 |
| **安全性约束** | 用户名全系统不重复，字母数字组合，不区分大小写，长度不小于5位；  密码由字母数字组合，区分大小写，长度不小于5位，采用MD5或更高级别加密； |
| **安全性约束** | 3次密码输入错误，要求输入验证码；10次以上错误，锁定用户名； |
| **可扩展性约束** | 需要支持原有逻辑扩展，在接口不变的情况，能方便增加新的逻辑实现。  需要支持新增逻辑在不影响原有业务、代码情况下，能增加新业务、新逻辑的处理分支  需要支持原逻辑再封装，支持在原逻辑封装基础上，进行再封装、不同逻辑组合。 |
| **稳定性约束** | 系统可以通过运行更多的实例或者采用分布式处理来支持更多的用户。可以通过线性的增加硬件设备、实例个数或者分布式处理点来处理更多的数据量，能更好的在不增加响应时间的前提下支持更多的业务处理。 |

## 3.2隐含约束

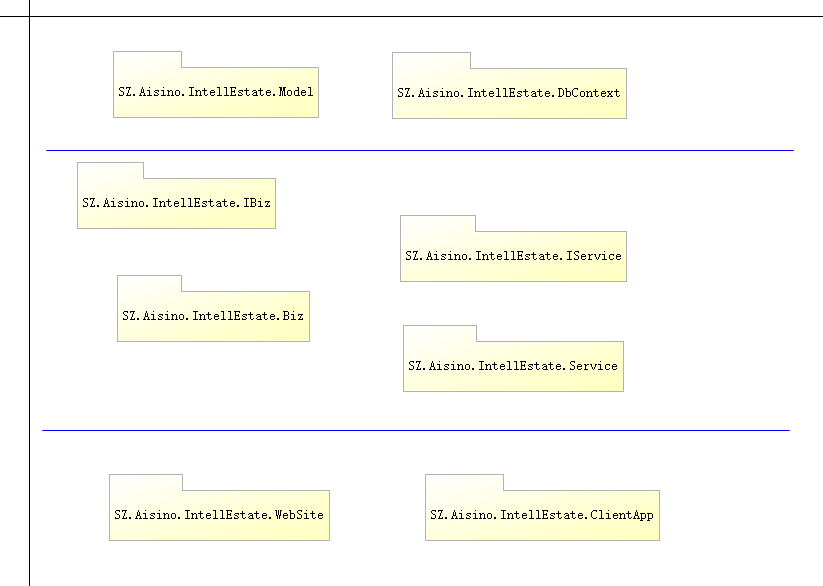
|  |  |
| --- | --- |
| **约束** | **描 述** |
| **功能约束** | 需求中提出的价格涉及多种货币，系统需要支持多币种（人民币、港币）的相关功能。 |
| **功能约束** | 需求中提出的报表部分，要求用户可自定义和扩展，因此系统需要支持自定义报表功能 |
| **功能约束** | 基础字典的管理需要同时支持通用字典和物业公司专用字典，物业公司在可以使用通用字典，专用字典，或者是专用字典+通用字典（专用字典中未定义的） |
| **功能约束** | 需求中未涉及运维相关的功能，但在某些逻辑中描述了运维的操作。因此系统需要有运维相关的功能。 |
| **性能约束** | 系统中使用的频度很高的数据需要有缓存机制，以增加页面的响应速度 |
|  |  |
|  |  |

# 设计策略

|  |  |
| --- | --- |
| **策 略** | **描 述** |
| **扩展策略** | 逻辑封装，面向接口编程  预计经常变化的内容采用XML配置文档将可变内容外置，扩展时不需要修改代码  Business层细化为两层，领域服务层（Biz）和服务facade层（Service）。其中Biz层基于领域模型划分为各个业务领域，封装具体的业务实现；Service层将Biz层的逻辑组织为大粒度的API，并有针对性地以 SOAP 或 Restful 的方式对外发布。 |
| **复用策略** | 与第三方系统对接信息采用XML配置，需要定义数据转换的处理接口，对接信息内容的生成要求可方便的进行扩展，适用于不同物业管理系统等其他需要交互的系统，可重用。 |
| **折衷策略** | 业务数据在遇到性能和功能冲突时，采用“分时”策略处理，将复杂操作拆分成多个简单操作  报表功能在遇到性能和功能冲突时，优先于功能，并通过SQL优化处理 |
| **数据关联策略** | 用户使用账户进行关联，不区分大小写  业务数据统一使用主键（Sql Server）或序列（Oracle）关联 |

# 总体设计

## 封装规范



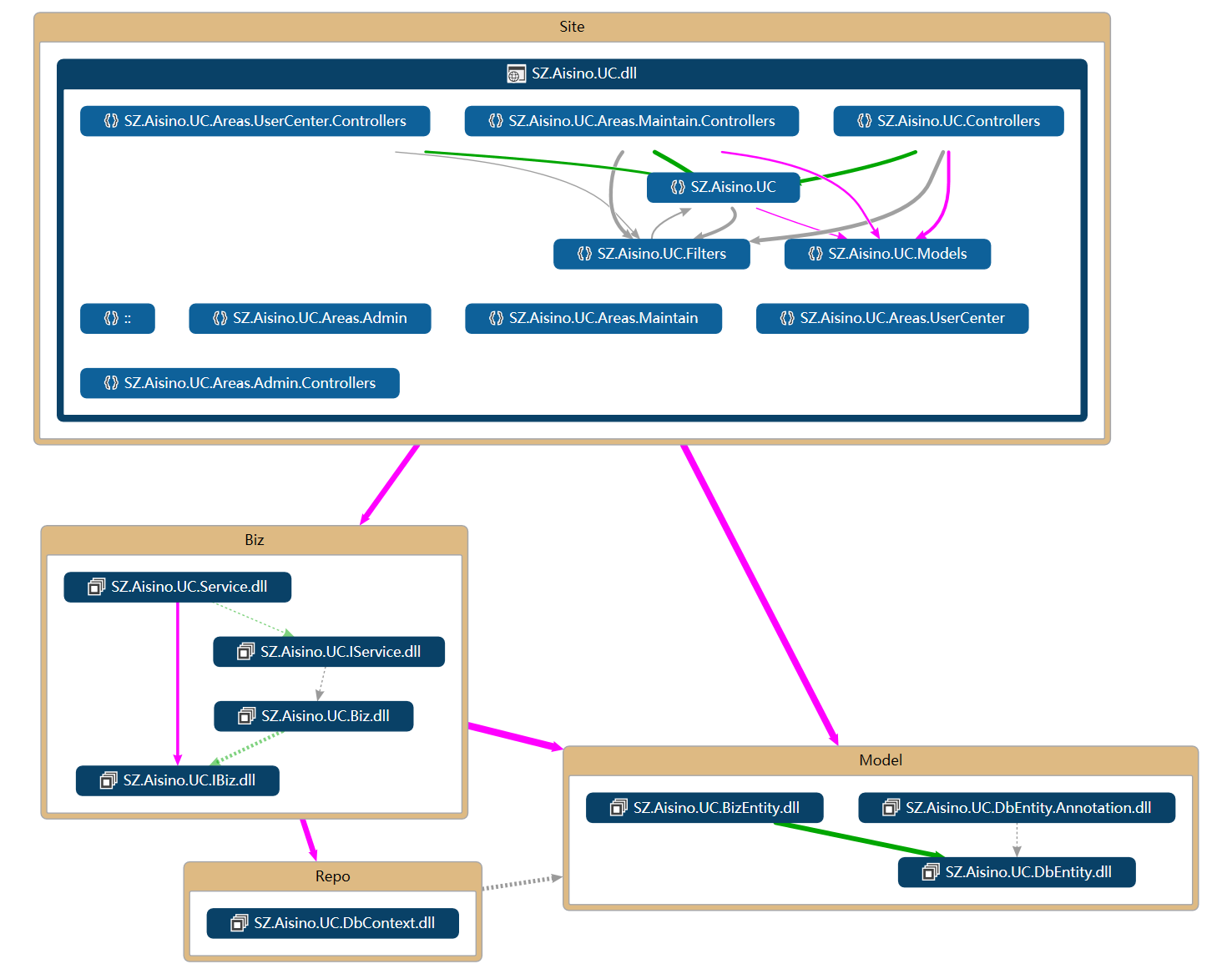
封装方案主体参考重构逻辑，三层架构。Model层和DbContext处理数据的映射和增删改查操作；Biz层和Service实现领域模型，Biz层封装可重用的逻辑代码，根据逻辑实现的不同分别封装不同的DLL，Service层对应大粒度的业务过程的封装。最终的应用层UI和逻辑分离。

### 接口定义

|  |  |  |
| --- | --- | --- |
| 所属系统 | 接口 | 说明 |
| 智慧云社区（互联网服务平台） | SZ.Aisino.IntellEstate.IQueryService. | 智慧云社区查询接口：互联网服务平台对外部系统有限开放的查询接口 |
| SZ.Aisino.IntellEstate.IEinvoiceService. | 统一开票接口：互联网服务平台对外部系统有限开放的电子票开票代理服务接口 |
| 电子票前置服务 |  | 深圳航信电子票前置服务接口 |
| 物业停车场道闸系统（第三方 ） | 三方待协商 | 查询接口（月卡有效期、月卡用户信息、月卡充值历史记录） |
| 三方待协商 | 月卡充值缴费接口  （月卡基准费率配置在“公司基础信息维护模块”中实现） |
|  |  |
| 物业管理系统（第三方 ） | 三方待协商 | 物管系统对外查询接口 |
| ~~三方待协商~~ | ~~开票状态回写接口~~ |

### 逻辑层级关系

#### SZ.Aisino.IntellEstate的层级关系



此处以服务平台-权限子系统为例说明：

1. 在Model层增加业务实体（BizEntity）定义，将业务实体作为数据迁移对象（DTO）来封装前台接口，以利于前台接口的易用性和扩展性。
2. Repo（仓储层）和Model层（表字段映射）继续保持分离。
3. 业务逻辑分开打包，并用接口支持扩展。
4. 高层组件所有针对Biz的依赖一律采用倒置原则设计； 通过依赖注入，插件式加载所需的操作实例。

### 类库命名规范

1. 类库命名格式定义为“SzAisino.<项目代码>.<类库分层>.<类库子分类>”。最大层级不超5 层。

项目代码暂定：IntellEstate

1. 程序集/包名称采用“驼峰”格式。
2. 类名称，方法名称和属性名称采用“驼峰”格式，类名称，属性、方法名称首字大写，字段名称首字小写。
3. 接口名称固定大写“I”字头；但如过领域模型的核心类首字母为I，则接口前缀“I\_”,如 I\_InformationSyncService
4. 函数/方法命名时采用“动词+名词”组合方式。

### 数据库设计规范

Oracle：

1. 数据表名称和字段名称采用全大写，“名词+(动词)”组合，单词之间使用下划线分割。例如订单主表可命名为“ORDER\_HEAD”，对应的跟踪表可命名为“ORDER\_HEAD\_TRACE”.
2. 禁止使用中文汉语拼音或缩写来命名表、字段、主外键等一切数据库对象。
3. 数据表命名不要带有 HIS，TEMP，TMP，BAK，BACK等容易混淆的关键字。
4. 所有的数据表都需要带有唯一的SEQ关键字，并作为主键处理，除非特殊需要。SEQ的命名规则为“表名\_SEQ”。
5. 所有的数据表都需要带有时间戳字段和操作人字段。
6. 不允许使用触发器。
7. 一个数据表不要使用超过一个的BOLB（CLOB）类型字段。
8. 字符型的字段统一使用“VARCHAR2”格式，长度设置为预估长度的2-5倍。尽量不超过2000长度。
9. 所有的公司名称，街道名称等涉及公司名称的字段，统一长度为VARCHAR2(2000)

SqlServer：

1. 数据表名称和字段名称采用驼峰命名，并根据表所对应的领域（Domain），尽量保证以领域名开始；名称一律采用复数形式，例如订单主表可命名为“Orders”，对应的明细表可命名为“OrderDetails”.
2. 禁止使用中文汉语拼音或缩写来命名表、字段、主外键等一切数据库对象。
3. 数据表命名不要带有关键字。
4. 所有的数据表都需要带有唯一的标识字段，并作为主键处理，除非特殊需要。SEQ的命名规则为“<表名单数形式>ID”，如：Orders表的主键字段命名为：OrderID
5. 所有的数据表都需要带有时间戳字段和操作人字段,关键表时间戳需分化为创建和修改两个字段，如：CreateDate和ModifyDate 。
6. 不允许使用触发器。
7. 字符型的字段统一使用“nvarchar(n)”格式，长度设置为预估长度的2~5倍。尽量不超过2000字符长度, 超过4000字符请使用ntext 。
8. 所有的公司名称，街道名称等涉及公司名称的字段，统一长度为nvarchar (2000)

### 数据库核心实体关系图



## 多语种支持解决方案

### 需求说明

需求提出需要对系统支持多语种方案，包括UI界面，提示信息和内容三个部分。

### 解决方案

备注：解决方案提供了.NET平台的具体实现策略，其他平台可根据自身特点参考或独立实现：

#### UI界面的多语种支持

UI界面的多语种支持有多个解决方案：1）采用.net本身的语言包功能，在编码时绑定资源文件； 2）采用分站点开发，不同的语言单独部署不同的站点； 3）采用数据库的编码对照方式处理。

三种方案都能实现UI界面的多语种支持，相比较而言，使用语言包的方案和本地文件有关，在程序发布时需配合自动发布实现一致的集群部署。而采用不同站点的方案,UI维护的工作量会成倍增加，不利于功能的扩展和维护。采用数据库编码方式灵活性不佳，亦不予考虑。

如果团队基于ASP.NET MVC中间件开发，推荐基于元数据（Metadata）配合资源字典来解决多语言问题，可以自定义元数据提供程序来具体实现之：

public class ResDataAnnotationsModelMetadataProvider : DataAnnotationsModelMetadataProvider {

       private ResourceManager ResMgr;

       public ResDataAnnotationsModelMetadataProvider(ResourceManager resMgr) {

           if (resMgr == null)

               throw new ArgumentNullException("resMgr");

           this.ResMgr = resMgr;

       }

       private string GetString(Type containerType, string propertyName) {

           var key = string.Format(**"{0}{1}\_{2}\_DisplayName", containerType.Namespace.Replace(".", ""), containerType.Name, propertyName**);

           return this.ResMgr.GetString(key);

       }

       /// <summary>

       /// 重写生成元数据的方式

       /// 只有在页面上来通过Lambda表达式表述的字段，在渲染时才会使用元数据，才能支持多语言转换

       /// (该过滤为了减少元数据的中英文转换次数)

       /// </summary>

       /// <param name="attributes"></param>

       /// <param name="containerType"></param>

       /// <param name="modelAccessor"></param>

       /// <param name="modelType"></param>

       /// <param name="propertyName"></param>

       /// <returns></returns>

       protected override ModelMetadata CreateMetadata(IEnumerable<Attribute> attributes, Type containerType, Func<object> modelAccessor, Type modelType, string propertyName) {

           if (containerType != null) {

    //动态类型/代理类 EF

               var cType = containerType.Assembly.IsDynamic ? containerType.BaseType : containerType;

               var value = this.GetString(cType, propertyName);

               if (!string.IsNullOrWhiteSpace(v)) {

                   var display = new DisplayAttribute() {

                       Name = value

                   };

                   var attrs = attributes.ToList();

**attrs.RemoveAll(a => a is DisplayAttribute);**

**attrs.Add(display);**

                   return base.CreateMetadata(attrs, containerType, modelAccessor, modelType, propertyName);

               }

           }

           return base.CreateMetadata(attributes, containerType, modelAccessor, modelType, propertyName);

       }

   }

以下编码体现了多语言接口的设计：

public interface ILang

{

        ModelMetadataProvider GetMetadataProvider();

        string GetByKey(string key);

}

public class Lang : ILang

   {

       public ModelMetadataProvider GetMetadataProvider()

       {

           return new ResDataAnnotationsModelMetadataProvider(EntityRes.ResourceManager);

       }

       public string GetByKey(string key)

       {

           return EntityRes.ResourceManager.GetString(key);

       }

   }

多语言模块的初始化工作在Global.asax.cs中完成：

public class MvcApplication : System.Web.HttpApplication

   {

       protected void Application\_Start()

       {

          ……

ILang lang = DependencyResolver.Current.GetService<ILang>();

          ModelMetadataProviders.Current = lang.GetMetadataProvider();

……

       }

   }

此后，只要在资源字典中存在相应的键名，使用HtmlHelper的DisplayFor即自动支持多语言功能。

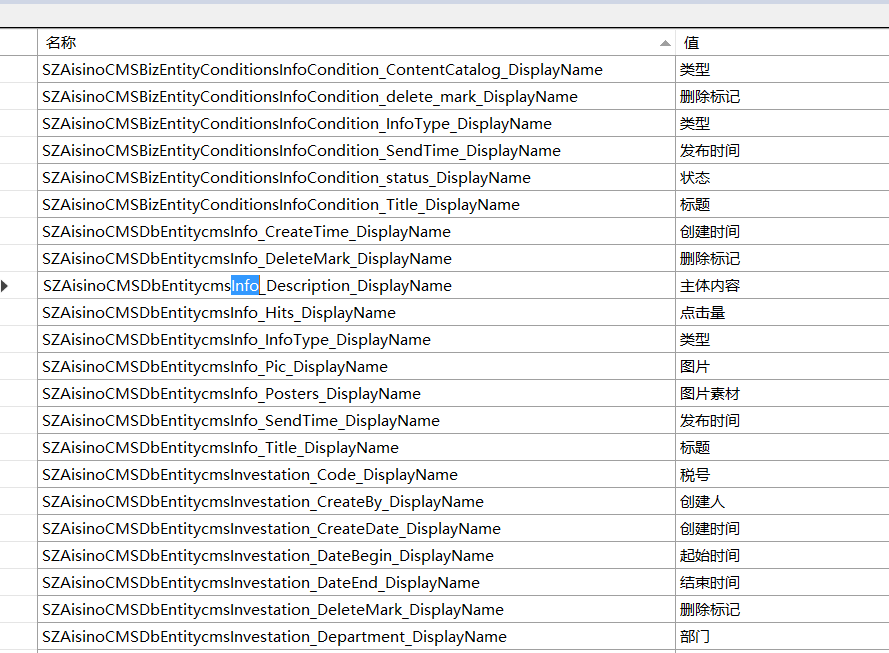
#### 相关问题

多语种的页面布局有一定的影响，开发界面时需要测试多个语种的显示效果，避免语种切换导致的版面问题。

### 资源字典设计

以下是资源字典的一个示例，请结合自定义元数据提供程序理解键的命名原则：

EntityRes.resx：



## 多币种支持解决方案

### 需求说明

系统需要支持币种处理，包括本位币和统计报表时的币种转换。

### 方案说明

1. 在所有使用价格的位置增加币种参数，“币种 + 价格”合并体现完整的价格信息。
2. 币种来源自每家物业公司的“本位币”全局参数配置。
3. 所有显示价格，金额的页面同时显示金额和币种，格式为“200USD”。
4. 每家物业公司独立维护自己的汇率转换关系，在需要使用的环节根据每家物业公司的汇率和设定进行货币转换。

### 数据库设计 （以ORACLE为例）



1. “CURRENCY\_RATE<汇率转换表>”用于保存货币转换的汇率信息，每家物业公司维护自己使用的汇率转换关系。
2. 汇率带有有效起至时间控制，在一个时间点应该只存在一套转换关系（输入数据时控制）。实际使用汇率时根据业务的时间查找汇率表获取对应汇率进行处理。
3. 汇率数据采用标记删除，避免已经使用的汇率找不到数据根源。

## 多浏览器支持解决方案

### 需求说明

系统要求支持浏览器兼容，支持IE8-10，Chrome，360和QQ浏览器。浏览器兼容问题主要集中在CSS样式的处理和JS脚本的跨浏览器兼容。

### 方案说明

1. HTML原型在制作时需要注意处理样式相关内容的浏览器兼容问题。
2. 页面的样式修饰全部要求使用CSS样式表控制，不要在元素上直接修饰。以方便在出现问题时统一修改。
3. JS脚本的兼容问题，需要注意对以下JS脚本的浏览器兼容：
4. 弹出窗口，系统的弹出窗口需要使用层弹出模式，不使用原有的Show Module方式。主要目的是避免浏览器对弹出窗口的拦截而导致功能不可用。因此建议在系统框架搭建的过程中，优先编写或者改造开源的弹出窗口控件。
5. 日期控件，系统需要使用到日期的录入控件，建议修改原有的日期控件或者测试开源的日期控件，在框架搭建阶段确定下能支持浏览器兼容的日期控件。
6. Message窗口，系统的Message消息不在使用JS的Alert函数处理，需要使用弹出层的方式，建议在框架阶段预先封装好消息提示的相关控件。
7. 菜单的显示效果，系统的菜单定义为下拉层的方式显示，相关的JS脚本需要预先进行浏览器兼容处理。建议在框架阶段确定一套系统使用的JS程序包，其他组件可JS脚本的编写基于这个包进行开发，避免不同的同时使用不同的程序包带来的浏览器兼容问题。

特别需要使用到的其他JS脚本和控件，在实际编码过程中需要进行浏览器兼容检查。

## 权限控制方案

### 方案说明

* 数据权限控制体系，以“User\_Data\_Func<用户数据权限>”数据表为核心，所有的权限控制都最终绑定到一个用户账户上。通过SQL语法实现控制逻辑。
* 某些设置在生成语法时，需要同时生成对多个功能点，多个用户账户的配置。这部分内容通过模板配置的方式实现批量生成。
* 权限控制分为三类，查询类，操作类和录入控制类
* 所有的查询操作，在查询数据时，都需要询问“User\_Data\_Func <用户数据权限>”获取到附加的SQL语法，以实现各种控制逻辑需要的效果。
* 所有的操作（Action）在执行时需要将操作的业务数据送到接口，与“User\_Data\_Func <用户数据权限>”中的条件进行验证，以确定是否允许操作。
* 录入控制的权限，设置内容保存在“User\_Data\_Func <用户数据权限>”中，实现部分由于涉及到的响应方式不同，需要在各个具体的功能中编码实现。
* 权限控制的重点在于所有的功能点都接受“User\_Data\_Func <用户数据权限>”的控制，无论现阶段是否有对应的权限设置对应到这个功能点。之后如果有新的权限控制需求，或者是原有的权限控制需求有改变的，仅需要变更规则写入“User\_Data\_Func <用户数据权限>”的方式，即可扩展对应的功能。

## 异常和日志方案

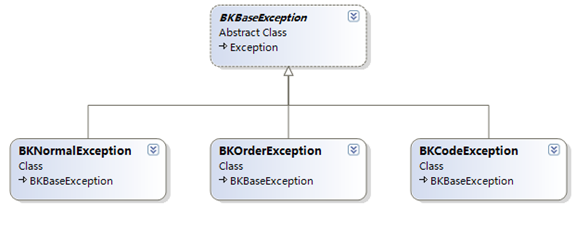
### 功能说明

异常和日志接口用于异常捕获，以及常规的业务日子记录。

### 数据库变更

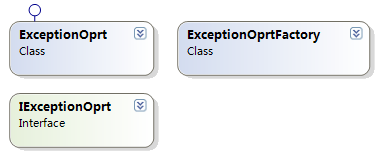
1. “OPERATION\_TYPE<操作类型>”日志记录的简略标题，用于区分一组相关的日志或者是为订阅日志提供分类。
2. “LEVELS<信息类型>”信息的分类，定义为4级“1-DEBUG 2-INFO 3-WARN 4-ERROR”。必要字段。
3. “INFORMATION<信息描述>”消息的主体，具体的内容在功能点的详细设计文档中进行描述。
4. “OBJECT\_TABLE<操作表名>，OBJECT\_SEQ<操作记录号>”用于特定数据的跟踪，目前仅在对“BUSINESS\_ORDER<订单表>”相关的日志中使用。后续可能扩展到其他需要关注的表。
5. “ORDER\_NO”在记录订单相关的日志时的必要信息，用于用户在追踪某一个订单时，作为查询条件使用。PS：此信息可能转换成Booking No，用于追踪订单的后续状态。
6. “OWNER\_COMPANY\_SEQ<物业公司>，OPRT\_CLIENT\_SEQ<操作终端>，PAY\_ CUSTOMER \_SEQ<业主>”用于登记日志的相关对象，这部分信息影响限定用户查询日志时的数据范围，后续根据实际的需求，可能作为索引使用。
7. “STATUS<数据状态>”取值为 “10-有效数据 -10-无效数据 -20-删除数据”。初始插入时为“10-有效数据”，其他两个状态值预留，目前暂不使用。

### 接口定义



在Exception对象基础上扩展了一层对象：

* “BKBaseException”类继承自“Exception”对象，抽象类，作为系统异常的基类。
* “BKNormalException”常规异常类，用于登记“非特定”的日志和异常。
* “BKOrderException”订单异常类，用于登记“订单”相关的日志和异常。和常规异常类的区别在于登记时会增加订单相关信息的采集。
* “BKCodeException”代码异常类，用于登记代码错误，常规用于Catch捕获中使用。



ExceprionOprt对象用于控制具体的日志写入操作，系统内部的异常或者是Exception对象都可以经过此类进行异常处理。

### 日志说明格式

在设计文档中的日志说明定义为以下格式

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标题 | LEVEL | 内容 | 接口 | 备注 |
| 订单撤销 | INFO  ERROR  DEBUG  WARNING |  | 常规异常  订单异常 | OWNER\_SEQ填写当前物业公司；  objName = “HTWY” |

## 缓存方案

### 功能说明

本章节用于描述缓存模块在系统中缓存的具体设计及使用场景。

### 缓存方案概要

备注：解决方案提供了.NET平台的具体实现策略，其他平台可根据自身特点参考或独立实现：

1. 缓存的主体可以是dataTable，也可以是实体集合，请根据自身仓储机制的具体实现选择，
2. 缓存推荐基于微软企业库AppFabric实现。

以下编码体现了缓存的接口设计：

public class AppFabricCaching : ICaching {

      private static DataCacheFactory Factory = null;

      private static DataCache Cache = null;

      static AppFabricCaching() {

          var cacheName = ConfigurationManager.AppSettings.Get("AppFabricCaching.Name");

          Factory = new DataCacheFactory();

          Cache = string.IsNullOrWhiteSpace(cacheName) ? Factory.GetDefaultCache() : Factory.GetCache(cacheName);

      }

      public T Get<T>(string key) where T : class {

          return (T)Cache.Get(key);

      }

      public void Set<T>(string key, T value) where T : class {

          Cache.Put(key, value);

      }

      public void Set<T>(CacheSet config, string key, T value) where T : class {

          if (config == null)

              throw new ArgumentNullException("missing config");

          if (config.Timeout <= TimeSpan.Zero)

              config.Timeout = TimeSpan.FromMinutes(10);

          // AppFabric Caching 没有相关API指定某项的优先级,只能通过加大过期时间来实现.

          if (config.NotRemoveable)

              config.Timeout = TimeSpan.FromDays(360);

          if (!string.IsNullOrEmpty(config.Region)) {

              Cache.Put(key, value, config.Timeout, config.Region);

          } else {

              Cache.Put(key, value, config.Timeout);

          }

      }

      public void Remove(string key) {

       Cache.Remove(key);

      }

  }

1. 缓存的数据采用“首次使用加载”机制，即在缓存首次被请求时方从数据库初始化。
2. AppFabric分布式缓存采用区域策略，这里使用缺省区域：default

<appSettings>

  <add key="AppFabricCaching.Name" value="default" />

……

</appSettings>

1. 对于使用Oracle作为存储单元的方案，缓存的刷新推荐使用Oracle的通知机制实现，需要使用“ODP.NET”的指定版本配合。
2. 需要缓存的数据定义

|  |  |  |
| --- | --- | --- |
| 缓存数据表 | KEY值 | 说明 |
| DICT\_CODE | DICT\_CODE\_{OWNER\_SEQ} | 字典代码表 |
| SYSTEM\_UI\_LANGUAGE | SYSTEM\_UI\_LANGUAGE | 语言包 |
| DICT\_LOCATION\_AREA | DICT\_ LOCATION\_AREA | 地区代码表 |
| DICT\_ESTATE\_STREET | DICT\_ESTATE\_STREET | 社区街道代码表 |

## 分布式会话

### 功能说明

服务平台各个子系统间需要支持单点登录，结合该系统集群部署及负载均衡的方案，推荐采用分布式会话实现同一顶级域名下个子站点的会话共享。

### 解决方案概要

备注：解决方案提供了.NET平台的具体实现策略，其他平台可根据自身特点参考或独立实现：

* + - 1. 推荐基于微软企业库AppFabric实现。
      2. 以下编码定义了一个非常简明的会话工具类，注意示例中采用了静态类实现：

public static class SessionHelper

{

    public static void Set<TSessionKey>(this TSessionKey key, object value) where TSessionKey : struct

    {

        var session = HttpContextHelper.Current.Session;

        session[key.ToString()] = value;

    }

    public static void Set(string key, object value)

    {

        var session = HttpContextHelper.Current.Session;

        session[key] = value;

    }

    public static TValue Get<TSessionKey, TValue>(this TSessionKey key) where TSessionKey : struct

    {

        var session = HttpContextHelper.Current.Session;

        object value = session[key.ToString()];

        return (TValue)value;

    }

    public static TValue Get<TValue>(string key)

    {

        var session = HttpContextHelper.Current.Session;

        object value = session[key];

        return (TValue)value;

    }

    public static void Remove<TSessionKey>(this TSessionKey key) where TSessionKey : struct

    {

        var session = HttpContextHelper.Current.Session;

        session.Remove(key.ToString());

    }

    public static void Remove(string key)

    {

        var session = HttpContextHelper.Current.Session;

        session.Remove(key);

    }

    public static void Clear()

    {

        var session = HttpContextHelper.Current.Session;

        session.Clear();

    }

}

以下是消费此工具类的示例：

var menus = SessionHelper.Get<Dictionary<string, IEnumerable<MenuItem>>>(SessionKeys.Menus.ToString());

……

menus.Set(appCode, appMenus);

SessionKeys.Menus.Set(menus);

以下是AppFabric的参考配置：

<configuration>

  <configSections>

    <section name="dataCacheClient" type="Microsoft.ApplicationServer.Caching.DataCacheClientSection,Microsoft.ApplicationServer.Caching.Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" allowLocation="true" allowDefinition="Everywhere" />

 <dataCacheClient>

    <hosts>

      <host name="WIN12-STATION.szhx.com" cachePort="22233" />

    </hosts>

   </dataCacheClient>

  <system.web>

    <sessionState sessionIDManagerType=" SZAisino.Common.SessionIDManager, SZAisino.Common" timeout="30" mode="Custom" cookieName="ASP.NET\_SessionId\_Global" customProvider="AppFabricCacheSessionStoreProvider" cookieless="false">

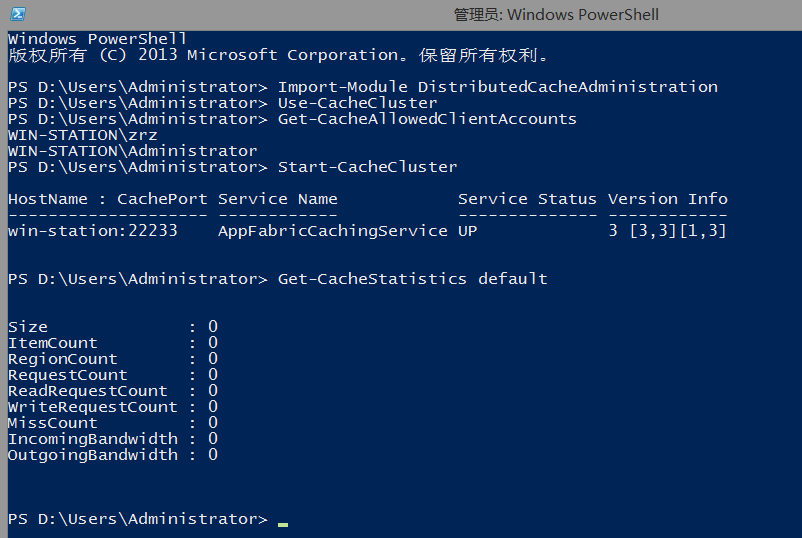
      <providers>

        <add name="AppFabricCacheSessionStoreProvider" type="Microsoft.ApplicationServer.Caching.DataCacheSessionStoreProvider, Microsoft.ApplicationServer.Caching.Client, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" cacheName="default" sharedId="OnlineSiteSession" />

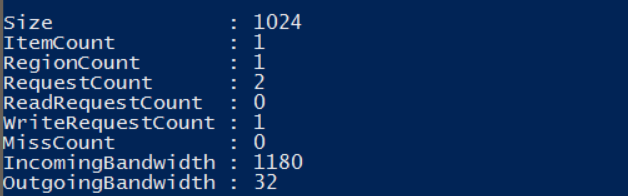
      </providers>

    </sessionState>

……  
以下为缓存服务器启动命令及运行状态：



有会话后：



# 服务平台详细设计

## 菜单定义

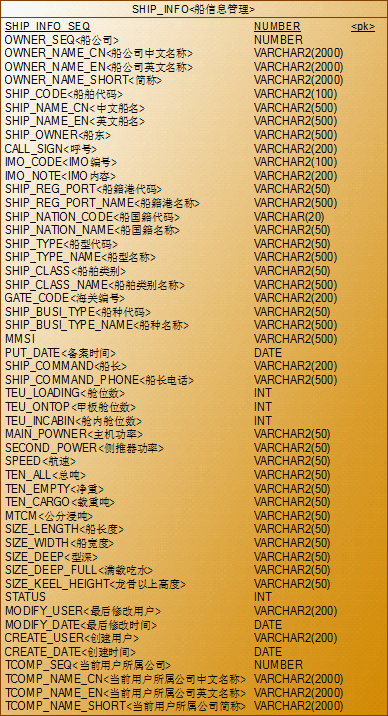
|  |  |  |
| --- | --- | --- |
| **物业公司功能** |  |  |
|  | 船期管理 |  |
|  |  | 船舶管理 |
|  |  | 航线管理 |
|  |  | 船期管理 |
|  |  | 舱位管理 |
|  | 订舱管理 |  |
|  |  | 放舱管理 |
|  |  | 议价管理 |
|  |  | 改船审批 |
|  |  | 批量改船 |
|  |  | 装船确认 |
|  |  | 用箱期审核 |
|  |  | 订舱确认备注设置 |
|  |  | 备案申请审批 |
|  | 价格管理 |  |
|  |  | FAK运价管理 |
|  |  | VIP合约管理 |
|  |  | 驳船价格管理 |
|  |  | 签约货代返点 |
|  |  | 签约货代返佣 |
|  |  | 一般货代返佣 |
|  |  | 中转运价管理 |
|  |  | 超限费用管理 |
|  |  | 附加费用管理 |
|  | 自定义报表 |  |
|  |  | 自定义报表配置 |
|  |  | 自定义报表查询 |
|  | 结算管理 |  |
|  |  | 佣金账套 |
|  |  | 佣金管理 |
|  |  | 查看调佣历史 |
|  |  | 漏计佣金记录 |
|  | 基础信息 |  |
|  |  | 客户管理 |
|  |  | 航线区域管理 |
|  | 设置管理 |  |
|  |  | 账户管理 |
|  |  | 订舱审核权限 |
|  |  | 议价审核权限 |
|  | 远期订舱 |  |
|  |  | 远期运价管理 |
|  |  | 远期放舱管理 |
|  |  | 远期订舱改期审批 |
| **货主货代功能** |  |  |
|  | 订舱管理 |  |
|  |  | 订舱管理 |
|  |  | 订舱申请 |
|  |  | 询价议价 |
|  |  | 改船申请 |
|  |  | 用箱期申请 |
|  |  | 备案申请 |
|  | 价格管理 |  |
|  |  | 签约货代返佣 |
|  |  | 一般货代返佣 |
|  |  | FAK运价查询 |
|  |  | 驳船运价查询 |
|  | 结算管理 |  |
|  |  | 佣金账单 |
|  | 基础信息 |  |
|  |  | 订舱人客户备案 |
|  | 业务查询 |  |
|  |  | 船期查询 |
|  |  | 航线查询 |
|  | 远期订舱 |  |
|  |  | 远期舱位预定 |
|  |  | 远期订舱管理 |
|  |  | 远期订舱改期 |

## 船舶管理

### 需求说明

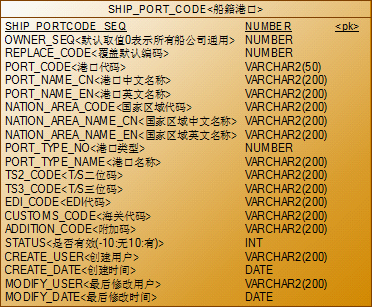
船舶信息是业务开展的基本数据，该功能是船公司用户进入系统根据搜索条件查询已录入船舶信息。

### 数据表



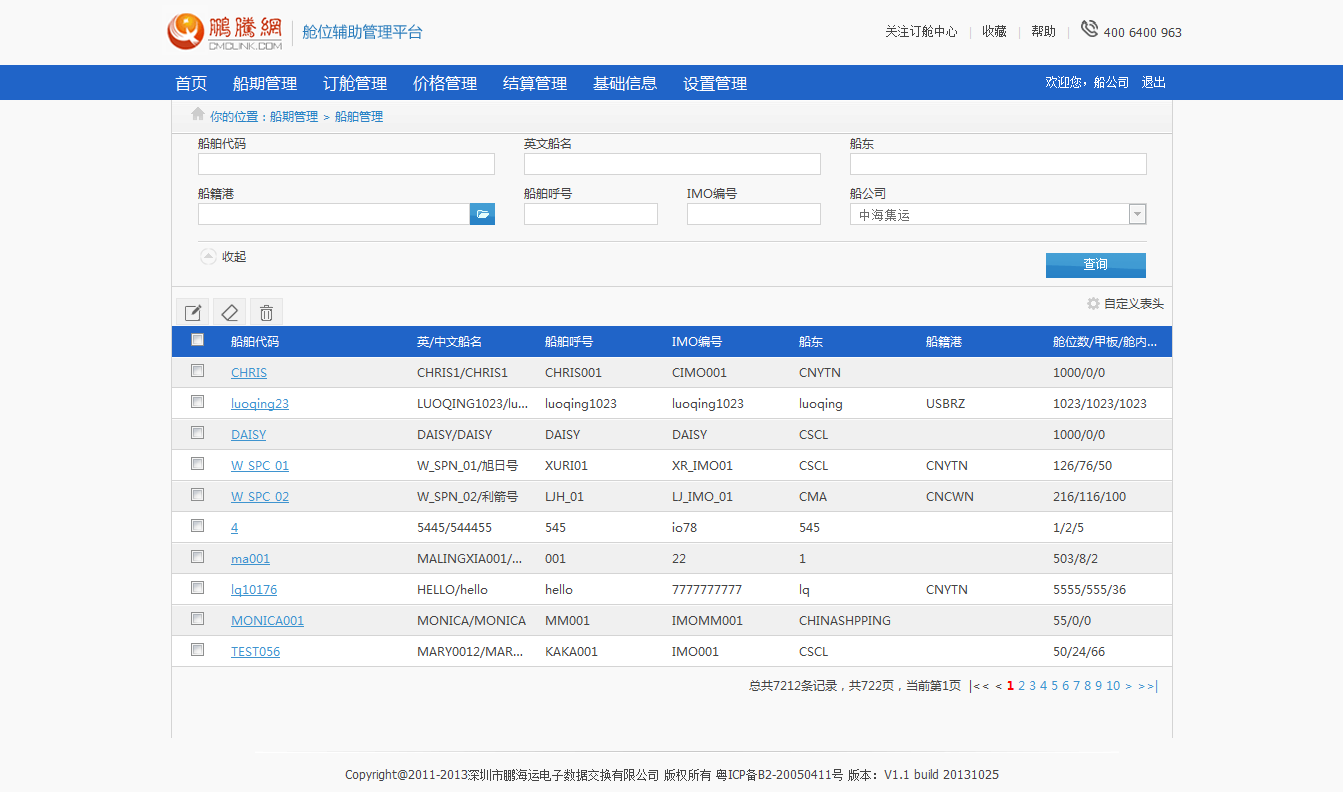


* “SHIP\_INFO<船信息管理表>”用于保存船相关的信息，“SHIP\_ATTR<船附件>”用于保存上传的船旗信息。
* “SHIP\_INFO\_SEQ”为关联字段，其他业务数据表在关联船信息时使用此字段进行关联。
* “OWNER\_SEQ<船公司>”字段用于标记不同船公司的船信息。
* “TCOMP\_SEQ<当前用户所属公司>”字段查询船舶信息用于区别同一船公司的不同下属操作公司。
* “STATUS”取值为“-10 删除 10有效”。船信息数据的删除采用标记删除，删除之前需要判断是否在使用中，使用中的船信息不允许删除。



1. “OWNER\_SEQ”默认取值“0”，取值为0的编码表示所有船公司通用，而船公司特定的编码标记船公司自己的SEQ，不支持对不同的操作公司不同的编码体系，对一家船公司仅能存在一套编码体系。
2. “STATUS”取值为“-10 删除 10有效”，采用标记删除方式处理。
3. “REPLACE\_CODE”参见《BKOnline系统详细设计文档LEVEL2.doc》编码表体系解决方案。
4. 港口类型、名称取值来自DICTIONARY\_CODE<字典代码表>。

### 列表界面



船舶列表（图1）

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 船舶代码、英文船名、船东、  舶呼号、IMO编号 | 支持模糊查询，默认为空，非必要条件 |
| 船公司 | 1. 普通用户不显示此查询条件。   查询条件参考权限接口，根据登录用户对应的公司属性，实现以下两种角色使用的区别：  **运维角色**：登录才能使用下拉条件“船公司”：下拉 精确 默认ALL。可查看所有船公司及下属操作公司航线代码/名称，即条件可选择。  **船公司及下属公司角色**：航线以船公司为归属当前登录的用户可查自己所属船公司下所有的航线代码/名称（即同一船公司的多家下属操作公司可互相查看对方已录入的航线信息），但不能查其他船公司的航线代码/名称。即条件初始化为用户对应的**船公司，**不允许编辑（PS：用户登录账户对应的公司可能是船公司下属的操作公司，并不是船公司，数据来源“TCOMP<公司资料表>TCOMP\_SEQ与TCOMP\_GROUP<公司组织>::COMP\_SEQ关联”表获取到对应的船公司，这部分有封装接口，参考IOMTUser对象）如果账户没有对应的船公司信息的，船公司条件默认为-1，显示为空，即无船公司对应的用户账户在此页面查询不到任何信息。 |
| 船籍港 | 参见“船籍港”选择列表数据后返回船籍港代码到文本框,不需要显示港口名称，默认空，非必要。 |
| 列表项 |  |
| 列表字段 | 字段的对应关系请参考数据库定义的同名字段。 |
| 英/中船名 | OWNER\_NAME\_EN<船公司中文名称>/  OWNER\_NAME\_CN<船公司中文名称>  如果中文名称或者英文名称任意一个为空时，不显示分割符号 |
| 船籍港 | SHIP\_REG\_PORT<船籍港代码>  仅显示代码，不需要显示名称 |
| 舱位数/甲板/舱内 | TEU\_LOADING<舱位数>/  TEU\_ONTOP<甲板舱位数>/  TEU\_INCABIN<舱内舱位数>  数据为空时显示为0 |
| 交互 |  |
| 查询 | * 列表默认不查询数据，用户点击“查询”之后再加载数据。 * 仅列出当前登录用户公司所属的船公司状态为10（owner\_seq =当前用户所属船公司SEQ and status = 10）的记录，每条船舶数据判断“SHIP\_INFO<船信息管理表>.TCOMP\_SEQ<当前用户所属公司>”是否等于登录用户“TCMOP\_SEQ<当前用户所属公司>”，相等在“操作”列需要给出“删除”“修改”“详情”连接，否则只给予“详情”连接，**并且数据前的复选框不可选择**。**若当前登录用户属于运维角色则给出所有操作功能的权限。** * 列表按创建时间倒序排列，即最近创建的排在最前面。 * 列表中的数据长度不允许超出一行的默认宽度，如出现数据换行的情况必须做内容截取，超出部分用“...”三个省略号替换，并增加浮动提示。 * 列表中的数据若不存在使用“-”代替。 |
| 新增 | 弹出编辑界面 |
| 详情 | 弹出一条船的详细信息页面，展示字段和布局方式参见编辑/新增船舶界面，字段的内容以不可编辑的文本框展示 |
| 导入 | 隐藏，暂不实现 |
| 删除 | * 提示“是否确认删除 (船舶代码 英/中文船名)？”，未选择任何数据提示“请选择船舶” * 查询“SHIP\_ROUTE\_SHIPDETAIL<航线信息船明细表>”判断要删除船舶数据的SHIP\_INFO\_SEQ是否存在于表中，已存在提示用户：[“航线代码/名称”已使用该船，不允许删除]，如有多条航线引用该船则提示[[“航线代码/名称1”,“航线代码/名称2”,“航线代码/名称3”]已使用该船，请将船舶从对应航线中移除之后再进行删除。] * 对记录做标记删除。即将STATUS修改为“-10删除数据”，并修改记录的时间戳   调用日志接口，记录删除操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 删除船舶 | INFO | 用户{用户账户}删除了船公司{船公司SEQ}-下属操作公司{下属操作公司SEQ}的船舶信息， 船名{船舶代码}-{船舶英文名称}，相关SEQ{ SHIP\_INFO\_SEQ } | 常规异常 | 无 |  * 批量删除时，采用单条循环操作模式，选中的记录顺序处理，事物独立。当出现不能删除的情况时，终止删除操作，已经删除的数据不回滚。   注：如果船舶数据已存在“航线信息船明细”表但又非要删除，必须在航线中先删除该船舶的引用。   * 参考航线管理。 |

### 编辑页面



船舶信息（图2）

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 船舶国籍 | 使用[船舶国籍区域选择器列表](#_船舶国籍区域选择器列表)，选择列表数据返回船舶国籍代码到文本框不需要显示国家名称需要保存国籍名称。 |
| 船籍港 | 使用[港口选择器](#_港口选择器)，选择列表数据返回船籍港代码到文本框不需要显示港口名称需要保存港口名称。 |
| IMO编号 | 文本框输入，必填项。 |
| 船公司 | 如果当前登录用户属于船公司角色，默认为对应的船公司，禁止编辑；如果是运维角色，船公司显示为下拉，数据来源“TCOMP<公司资料表>TCOMP\_SEQ与TCOMP\_GROUP<公司组织>::COMP\_SEQ关联”。填写方式参考“**船舶管理-列表页面-查询条件-船公司**”说明。 |
| 船型 | 下拉必填项，取DICT\_CODE表DICT\_TYPE=**SHIP\_TYPE**数据，根据当前系统语言环境显示中文或英文名称。 |
| 船舶种类 | 下拉非必填项，取DICT\_CODE表DICT\_TYPE=SHIP\_CLASS数据，根据当前系统语言  环境显示中文或英文名称。 |
| ~~备案日期~~  建造时间 | 日期控件，不需要时间。  明细页面中的列标题“备案时间”，文字修改为“建造时间”。数据库不变更，依旧对应“PUT\_DATE”字段；格式验证和非空检查不变更。包括“新增、修改”的编辑界面和“详情”的查看页面 |
| 船旗信息 | 从“SHIP\_ATTR<船附件>”表获取，参考Eir中校验码的处理方式。 |
| 状态栏 | “有效”对应“STATUS 10”，“无效”对应“STATUS -10”。 |
| 操作公司 | 名称：默认为操作公司中文名称，用户设置英文语言环境那么需要显示英文版的“操作公司中文名称”。**如果是运维人员操作可以下拉显示船公司和下属操作公司名称，操作公司需和船公司关联。** |
| 交互 |  |
| 新增 | 1. 数据校验： 2. 检查必填字段，对应列表   SHIP\_INFO<船信息管理>  船舶代码 SHIP\_CODE<船舶代码>  英文船名 SHIP\_NAME\_EN<英文船名>  中文船名 SHIP\_NAME\_CN<中文船名>  船舶呼号 CALL\_SIGN<船舶呼号>  IMO编码 IMO\_CODE<IMO编号>  船舶国籍 SHIP\_COUNTRY<船国籍>  /SHIP\_COUNTRY\_CODE<船国籍代码>  船东 SHIP\_OWNER  船公司 OWNER\_SEQ<船公司>默认用户所属船公司  船型 SHIP\_TYPE<船型>/SHIP\_TYPE\_NAME<船型>  舱位数 TEU\_LOADING<舱位数>  PS：原型中的船旗不在定义为必填字段   1. 查重处理：在“SHIP\_INFO<船信息管理>”表中分别检查“船舶代码，英文船名”是否重复，仅检查有效的数据。存在重复时提示用户“{船舶代码/英文船名}已经存在，请检查。” 2. 去除空格处理：船舶代码和英文船名需要排除字符串前后的空格在进行数据检查。 3. “舱位数，甲板数，舱内数”为数字类型，如果录入的内容非数字的，提示用户录入错误。（JS在录入时进行第一次检查，为了避免数据异常在保存之前还需要再次检查） 4. 将录入的船舶字段数据正确的对应到船信息表，然后写入船舶信息表。主表和子表的操作需要同一事物。 5. 如有船旗附件则参考“图片上传逻辑”，将船旗文件的二进制流保存到“SHIP\_ATTR<船附件>”表中。 |
| 编辑 | 1. 进入编辑船舶页面前查询“航线信息船明细”表判断船舶数据的【SHIP\_INFO\_SEQ、STATUS=10（有效）】是否存在于表中，已存在页面上不允许更改“船舶代码”“英文船名”，否则允许更改。 2. 向后台提交船公司用户修改过字段数据，再根据SHIP\_INFO\_SEQ将修改过字段数据更新到船信息表。   3.编辑数据保存时，如果用户没有重新上传文件的，不需要更新“SHIP\_ATTR<船附件>”表。  4.如船公司用户重新上传“船旗”图片则根据SHIP\_INFO\_SEQ在船附件表找到附件数据，将新上传图片的文件名称、文件大小、文件类型、文件内容覆盖原有附件数据。  5. 编辑船舶信息时隐藏“连续发布”按钮。 |
| 图片上传逻辑 | 用户点击船旗“上传”弹出文件选择框，只支持\*.jpg、\*.png两种格式，只支持1M以内大小图片做保存。  2.预览：用户选择了文件之后需要支持预览功能，即用户选择了船旗图片在图片展示区域显示缩小版的船旗图片，其固定大小，不做缩放。  3.默认图片：“船旗”字段改为不是必填项，如果船旗数据为空的，需要显示为 “无图片”的默认图片（白底红字即可，作为图形资源加载使用） |
| 连续发布 | 用户点击“连续发布”提示用户“发布（船舶代码 英文船名）成功”，重置编辑页面所有数据并继续停留在编辑页面可供用户二次输入。 |
| 发布 | 用户点击“发布”提示用户“发布（船舶代码 英文船名）成功”，关闭弹出的编辑界面跳转至船舶查询界面。 |
|  | 数据保存变更  查重处理：在“SHIP\_INFO<船信息管理>”表中分别检查“船舶代码，英文船名”是否重复，仅检查有效的数据。存在重复时提示用户。  BKM100257 录入的船舶信息中已经存在相同船舶代码的船信息，当前录入的信息不能保存，请检查录入的数据。  BKM100258 录入的英文船名中已经存在相同船名的信息，当前录入的信息不能保存，请检查录入的数据。  录入控制  船舶英文名称在录入时，自动转成大写字母。业务含义是船舶名称仅支持大写。  舱位数(TEU)，在录入时必须检验是大于0的整数。后面的逻辑中有需要根据这个数字计算剩余舱位数。 |

### 港口选择器



|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 港口代码、港口名称 | 文本框支持模糊查询非必填可为空。不区分查询条件内容的大小写。 |
| 交互 |  |
| 加载过程 | 1. 参考以下默认语句：  |  | | --- | | select a.port\_code, a.port\_name\_cn, a.port\_name\_en, b.customs\_id, b. customs\_name, a.nation\_code, a.nation\_name\_cn, a.nation\_name\_en, a.area\_code, a.area\_name\_cn, a.area\_name\_en from dict\_port\_code a  left join dict\_customs\_area b on a.port\_code=b.port\_id  where (a.owner\_seq = 0 or a.owner\_seq = 船公司OWNER\_SEQ) and a.status= 10 *--and a.PORT\_CODE ='CNYTN' --查询条件*  and a.cover\_codes not like '%,船公司OWNER\_SEQ,%'  order by a.create\_date |  1. 列表按创建时间倒序排列，即最近创建的排在最前面。   3.列表中的数据长度不允许超出一行的默认宽度，如出现数据换行的情况必须做内容截取，超出部分用“...”三个省略号替换，并增加浮动提示。  4.列表中的数据若不存在使用“-”代替。  5.需要分页，每页显示10条数据。 |
| 数据返回 | 点击选择列，返回港口信息。 |

## 远期订舱管理

### 需求说明

订舱人为了锁定和减少运输成本，根据运输计划预订船公司的远期舱位。包括远期订舱保存、修改、删除、提交，待审批单据的收回、已审批单据的撤销，复制申请和详情查看功能。

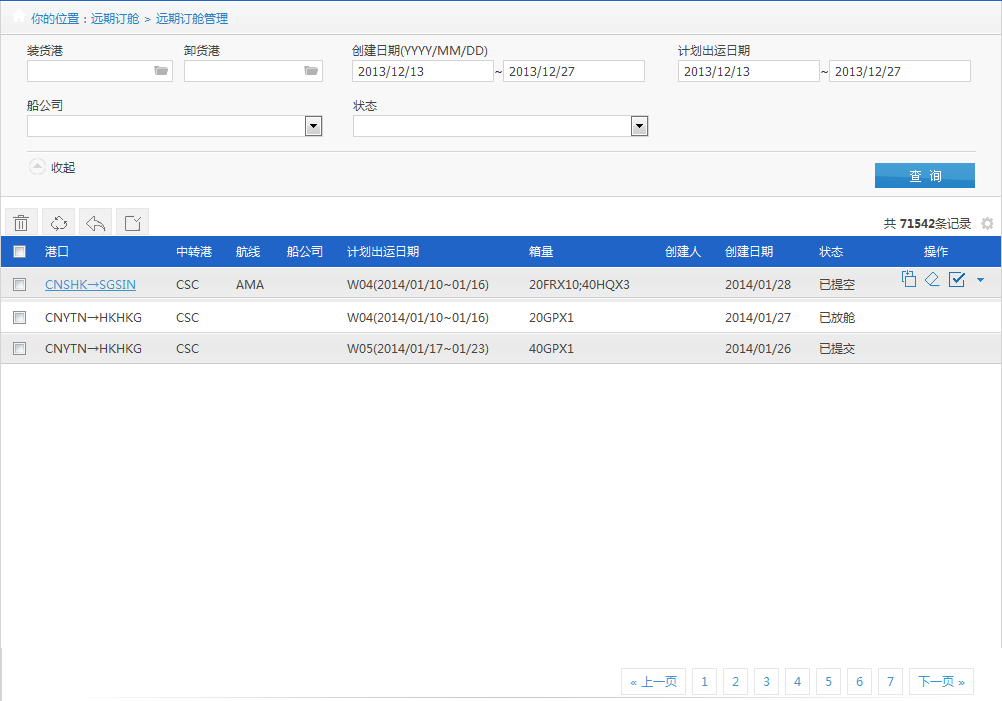
### 数据表

* 新增表结构“FUTURE\_PRESALE\_HEAD<远期预售主表>，FUTURE\_PRESALE\_DETAIL<远期预售明细>，FUTURE\_PRESALE\_LOG<远期预售操作日志>”
* FUTURE\_PRESALE\_HEAD<远期预售主表>，带有自动生成的订单编号，状态控制和航线绑定信息
* FUTURE\_PRESALE\_DETAIL<远期预售明细>，远期预售的明细表，子表，保存箱型，箱量和价格。
* FUTURE\_PRESALE\_LOG<远期预售操作日志>，远期预售操作日志。
* FUTURE\_PRESALE\_ORDERNO\_SEQ 用于生成远期预售单号

### 菜单

订舱人-》远期订舱-》远期订舱管理

### 列表页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 查询条件 |  |
| 船公司 | 下拉，非必要条件。可参考订舱人的订舱管理页面。 |
| 装货港 | 选择器+自动完成，非必要条件。可参考订舱人的订舱管理页面。 |
| 卸货港 | 选择器+自动完成，非必要条件。可参考订舱人的订舱管理页面。 |
| 创建日期 | 时间范围，日期选择控件，非必要条件。  结束时间，默认当天；  开始日期，默认当天 – 14 day |
| 计划出运日期 | 时间范围，日期选择控件，非必要条件。 |
| 状态 | 下拉，非必要条件。  数据加载：ALL=不使用 0草稿 10待审批 -20已删除 -10已撤单 100已放舱 90已拒绝 110交割中 200已交割  已提交(包括以下状态：待审批、已审批、已拒绝、交割中、已交割)，默认已提交状态 |
| 列表项 |  |
| 港口 | LOADING\_PORT<装货港> + “→” + DESTINATION\_PORT<卸货港> |
| 中转港 | TRANSFER\_PORT<中转港> |
| 航线 | ROUTE\_CODE<航线代码> |
| 船公司 | OWNER\_CODE<船公司代码> |
| 计划出运日期 | WEEK\_NO<出运周数> + "(" + WEEK\_BEGIN<格式化yyyy/MM/dd> + "~" + WEEK\_END<格式化MM/dd> + ")" |
| 箱量 | CONTA\_SIZE\_SUM<订单箱量汇总> |
| 创建人 | CREATE\_USER<创建用户> |
| 创建日期 | CREATE\_DATE<创建时间> |
| 状态 | STATUS，0草稿 10待审批 -20已删除 -10已撤单 100已放舱 90已拒绝 110交割中 200已交割 |
| 操作 |  |
| 页面加载 | 默认不查询数据  增加处理“交割”按钮的可用性：  调用【查询】方法后，对查询返回的结果，增加调用以下逻辑：  [1]按照返回的结果data.船公司进行分组，调用以下接口  循环调用data.船公司  {   * OMTOrder.IsAllowOrderForFPT   参数赋值：  tcompSEQ：当前订舱人所属公司SEQ  ownerSEQ：当前记录的船公司SEQ  当结果返回失败后，则表示该船公司禁止该用户进行“交割”，交割按钮不可用；  }  [2] 对于[1]中允许“交割”的船公司记录data.FUTURE\_PRESALE\_HEAD\_SEQ加入到List<Decimal>中，调用接口  - OMTOrder.GetAllowOrderForFPT  接口返回的allowOrderSEQList，表示允许“交割”；  页面加载时，只有allowOrderSEQList的FUTURE\_PRESALE\_HEAD\_SEQ，“交割”按钮才可用； |
| 点击“港口” | 弹出详情页面，传递参数FUTURE\_PRESALE\_HEAD\_SEQ |
| 查询 | 调用接口IOMTFuturePresale：：QueryForListToCustomerByPage，参数  FuturePresalListQueryParm ownerSeq = 页面输入的船公司查询条件  FuturePresalListQueryParm loadingPort = 页面录入的装货港条件  FuturePresalListQueryParm destinationPort = 页面录入的装货港条件  FuturePresalListQueryParm beginDate = 页面录入的创建时间范围条件起始  FuturePresalListQueryParm endDate =页面录入的创建时间范围条件终止  FuturePresalListQueryParm status = 页面录入的条件  FuturePresalListQueryParm weekBegin = 页面录入的起运时间范围条件起始  FuturePresalListQueryParm weekEnd =页面录入的起运时间范围条件终止  FuturePresalListQueryParm tCompSeq=当前用户所属公司SEQ  orderBy = WEEK\_BEGIN ASC,CREATE\_DATE ASC,  FUTURE\_PRESALE\_HEAD\_SEQ DESC  pageIndex = 1  pageRows = 根据原型确定  返回值处理  rowCount = 总记录数，用于分页控件  data = 加载到页面的数据  调用失败时  提示errorMsg，按编码规范处理多语言问题 |
| 复制 | 弹出编辑页面，页面建议增加参数指定需要加载复制后的数据。参考编辑页面的元素“页面加载” |
| 修改 | 弹出编辑页面  （PS：业务逻辑中，只有草稿才能编辑） |
| 删除 | 只有草稿状态，才允许删除  调用接口IOMTFuturePresale：DeleteOrder  参数赋值：  headSEQ = 当前记录FUTURE\_PRESALE\_HEAD\_SEQ |
| 回收 | 只有10待审批 才允许回收  调用接口IOMTFuturePresale：RetrieveOrder  参数赋值：  headSEQ = 当前记录FUTURE\_PRESALE\_HEAD\_SEQ |
| 撤销 | 只有100已放舱 才允许撤销  调用接口IOMTFuturePresale：CancelOrder  参数赋值：  headSEQ = 当前记录FUTURE\_PRESALE\_HEAD\_SEQ |
| 批量删除 | 调用接口IOMTFuturePresale：Cancel  参数赋值：  HeadSeqList = 选中的SEQ（List） |
| 批量提交 | 调用接口IOMTFuturePresale：BatchSubmit  参数赋值：  HeadSeqList = 选中的SEQ（List） |
| 批量回收 | 调用接口IOMTFuturePresale：BatchRetrieve  参数赋值：  HeadSeqList = 选中的SEQ（List） |
| 批量撤销 | 调用接口IOMTFuturePresale：BatchCancel  参数赋值：  HeadSeqList = 选中的SEQ（List） |
| 提交 | 调用接口IOMTFuturePresale：SubmitOrder |
| 交割 | * 调用OMTOrder.IsAllowOrderForFPT接口， * 调用OMTOrder.GetAllowOrderForFPT接口   校验当前时间，是否允许交割（调用逻辑如上）  校验通过后，跳转到页面“远期交割申请页面”  传参：  preSaleHeadSeq = 交割记录预售订单主表seq |
|  |  |

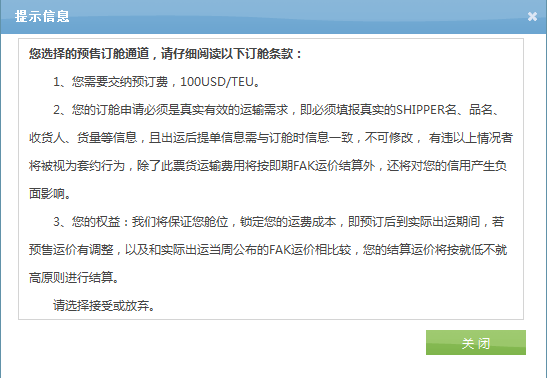
### 申请页面



（箱型，固定加载5行）

|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 |  |
| 以下字段对应FUTURE\_PRESALE\_HEAD<远期预售主表> | |
| 根据远期运价主表SEQ自动带出(只读)，对应对象FuturePriceEntity.futurePriceData第一条记录 | |
| 船公司 | OWNER\_SEQ<船公司>  OWNER\_CODE  OWNER\_NAME |
| 口岸公司 | OPRT\_COMP\_SEQ<口岸公司>  OPRT\_COMP\_NAME<口岸公司名称>  OPRT\_COMP\_NAME\_EN<口岸公司英文名称> |
| 航线 | SHIP\_ROUTE\_HEAD\_SEQ<航线关联>  ROUTE\_CODE<航线代码>  ROUTE\_NAME<航线名称>  BERTH\_WEEK<班期> |
| 状态 | STATUS，默认为草稿 |
| 装货港 | LOADING\_PORT<装货港>  LOADING\_PORT\_NAME<装货港中文名称>  LOADING\_PORT\_NAME\_EN<装货港英文名称> |
| 卸货港 | DESTINATIONPORT<装货港>  DESTINATION\_PORT\_NAME<装货港中文名称>  DESTINATION\_PORT\_NAME\_EN<装货港英文名称> |
| 中转港 | TRANSFER\_PORT<中转港>  TRANSPORT\_NAME\_EN<中转港英文>  TRANSPORT\_NAME\_CN<中转港中文> |
| 计划出运日期 | 对应对象FuturePriceEntity.futurePricePeriodMuti[0]  WEEK\_NO<出运周数>  WEEK\_BEGIN  WEEK\_END |
| 运费条款 | RULE\_PRICE<运费条款代码>  RULE\_PRICE\_NAME<运费条款中文名称>  RULE\_PRICE\_NAME\_EN<运费条款英文名称> |
| 运\输条款 | TRANS\_TYPE<运输条款代码>  TRANS\_TYPE\_NAME<运输条款中文名称>  TRANS\_TYPE\_NAME\_EN<运输条款英文名称> |
| 提前交割时限 | CROSS\_BEFOREDAY<允许交割提前天数> |
| 延后交割时限 | CROSS\_DELAYDAY<允许交割延后天数> |
| 运价SEQ | FUTURE\_PRICE\_HEAD\_SEQ |
| 运价编号 | FUTURE\_PRICE\_NO<运价编号> |
| 以下字段是页面录入(\*为必填字段) | |
| 发货人\* | COMP\_SEND<托运人>  COMP\_SEND\_SEQ<托运人SEQ>  可以从选择器上选择也可以手工录入，参考FAK订舱申请，要检查不每行不能超过35个字符最多6行，如果从选择器带过来的话要记录SEQ，保存时比较发货人的内容是否和选择器带过来一样（即未做任何修改），此时要保存SEQ，否则将SEQ为0存库。  PS：只能录入英文和数字，赋值给接口时，需要转为大写  新增模式，默认加载当前订舱公司的英文名称  IOMTUser: userCompNameEn |
| 收货人 | 同上  COMP\_GET<收货人>  COMP\_GET\_SEQ<收货人SEQ> |
| 通知人 | 页面加载，调用接口IOMTCustomer.GetOrderDesc<获取提单内容>，赋值后，允许手工编辑；  COMP\_TRANS<转运人>  COMP\_TRANS\_SEQ<转运人SEQ> |
| 订舱人联系地址\* | COMP\_SEND\_ADDR<订舱公司地址> |
| 联系人姓名\* | LINKER<联系人姓名> |
| 电子邮箱\* | LINKER\_EMAIL<电子邮件> |
| 联系电话\* | LINKER\_PHONE<电话> |
| 传真\* | LINKER\_FAX<传真> |
| 付款地 | PAY\_LOCATION<付款地点代码>  下拉框，调用现有FAK订舱页面的“付款地”数据源  混合（默认） |
| 货物描述\* | CARGO\_DESC<货物描述>，必须全英文大写，需要报文输出；最长50个字节，中间不允许有换行。 |
| 订舱备注 | ORDER\_REMARK<订舱备注> |
| 同意[船公司业务办理协议] | SIGN\_AGREEMENT<同意协议>  -1否 1是  新增模式，默认不勾选 |
| 隐藏字段保存 | * PUBLISH\_PRICE\_SUM<发布价格概要> * FUTURE\_PRICE\_HEAD\_SEQ * FUTURE\_PRICE\_NO<运价编号> |
| 以下字段对应FUTURE\_PRESALE\_DETAIL<远期预售明细>  对应对象FuturePriceEntity.futurePriceDetail（此为List） | |
| 箱型 | 根据远期运价编号,带出所有箱型列表及价格.  CONTA\_SIZE<标准箱型>  以下用隐藏字段保存  CONTA\_SIZE\_OWNER<箱型>  CONTA\_SIZE\_REPORT<统计箱型>  CONTA\_SZIE\_DISP<显示箱型>  CONTA\_SIZE\_DESC<箱型描述>  JS控制，同一个箱型不允许录入到多行记录中(重要) |
| 数量 | UNIT<箱数量>，手工录入 |
| 远期运价 | FINAL\_PRICE\_TOTAL<最终成交价>  以下用隐藏字段保存  PRICE\_UNIT<币种>  FUTURE\_PRICE\_CONTADETAIL\_SEQ |
| 操作 |  |
| 页面加载 | * 新增模式   [获取远期运价信息]  调用PHY.BKONLINE.Facade.FuturePrice.IOMTFuturePrice：QueryByHeadSeq  参数HeadSeq = 传参远期运价主表SEQ  输出参数FuturePriceEntity为远期运价所有表信息  [订舱人信息赋值] – FAK订舱页面已做  订舱人联系地址：取当前用户的所属公司的名称TCOMP .COMP\_ADDRESS  联系人姓名：默认取TUSER<用户表>.USER\_NAME<用户账户>  电子邮件：默认取TUSERATTR<用户附件信息表>EMAIL<邮箱>  联系电话：默认取TUSERATTR<用户附件信息表> TELEPHONE<电话>  传真：默认取TUSERATTR<用户附件信息表> FAX<传真>     * 编辑模式   调用IOMTFuturePresale：QueryByHeadSeq  参数 HeadSeq = 当前主表SEQ   * 不管新增还是编辑，都需要重置[信用等级]   [获取信用级别]  调用PHY.BKONLINE.Facade.Customers.IOMTCustomers.GetCustomer接口  参数 tCompSEQ = 当前用户所属公司SEQ  ownerSEQ = 当前业务所属船公司SEQ  输出结果中CUSTOMERS.CREDIT\_LEVEL<信用等级> ==  NULL?NULL:CUSTOMERS.CREDIT\_LEVEL<信用等级>(隐藏字段保留，用于保存时赋值)   * 复制模式   调用IOMTFuturePresale：Clone  参数HeadSeq = 页面传递参数HeadSeq |
| 选择“箱型” | JS刷新当行箱型的“远期运价” |
| 暂存 | * 必填字段校验 * 发货人、收货人、通知人录入有效性校验 * SIGN\_AGREEMENT<同意协议>必须勾选，否则弹出提示信息   “BKM100609|操作失败，原因：请先阅读《船公司业务协议》，并同意，才能申请远期订舱！”   * 调用接口IOMTFuturePresale.Save   参数赋值：  Parm.FuturePresaleModel，根据页面对应字段进行赋值（页面能看到的都需要保存）  箱型，要进行统计（同一箱型的，页面保存为一条箱型明细）  新增模式：  LAST\_CHANGE\_SEQ<最近改期的SEQ>=0  不管新增还是编辑，都需要重置如下内容：  CONTA\_SIZE\_SUM<订单箱量汇总>=箱型1x数量；箱型2x数量；... |
| 发布 | STATUS= 0草稿，10已发布时可见。  调用接口 IOMTFuturePresale：SaveAndSubmit保存并提交数据，参数赋值与“暂存”一致 |
| 点击“船公司业务办理协议” | 弹出静态页面“协议页面” |
| 关闭 | 关闭弹出窗口，重新刷新列表数据 |

### 协议页面



静态页面，直接Copy demo页

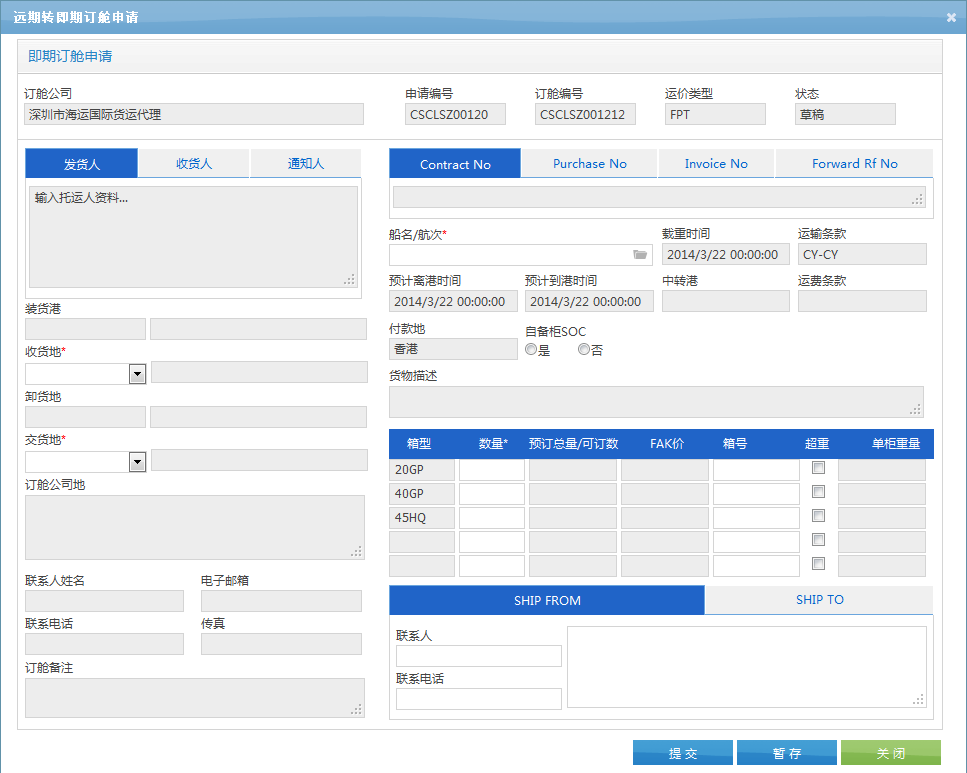
### 详细页面



页面元素大部分与“申请页面”一致，申请人和审批信息，请参照数据库字段。

页面元素中（删除，审批人）

### 远期交割申请页面



* 船名/航次，调用“远期交割航线选择器”，同步更新页面的“截重时间、预计离港时间、预计到港时间”；
* “收货地”和“交货地”，取对应FAK的“收货地”和“交货地”，可能有多个，所以是下拉框；
* 箱型列表：

增加“预订总量/可订数”的统计数据；

箱型来源于“远期预售订舱单”的订舱箱型（不会有重复的箱型，预售已合并）；

* Contract No 取远期预售订舱单号；

提交逻辑：

* 船名/航次的有效性校验；
* 收货地和交货地的有效性校验
* 可交割箱数量的校验；
* 校验FAK是否有对应的箱型？如果没有，禁止提交。

暂存逻辑：

* 不需要任何业务校验；

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| \*为必填 | 无特别说明，默认标签显示，  新增模式：页面取值于初始化接口返回的FuturePresaleModel  FuturePresaleModel.FuturePresaleData只有一条记录，定义为head  FuturePresaleModel.FuturePresaleDetail 有>=1条记录，定义为detail  编辑模式：页面取值于初始化接口返回的OrderData  OrderData.orderHead只有一条记录，定义为head  OrderData.orderDetail有>=1条记录，定义为detail |
| 订舱人 | CUSTOMER\_NAME\_PY |
| 申请编号 | 进入页面给出默认编号，只读，调用OMTOrder：CreateNewOrderNo取值。 |
| 订舱编号 | 草稿状态（0）为NULL，该编号在审批时生成。 |
| 运价类型 | 固定FPT |
| 状态信息 | STATUS，根据字典转换 |
| 托运人 | head.COMP\_SEND<托运人>  隐藏head. COMP\_SEND\_SEQ<托运人SEQ> |
| 收货人 | head.COMP\_GET<收货人>  隐藏head. COMP\_GET\_SEQ<收货人SEQ> |
| 转运人 | head.COMP\_TRANS<转运人>  隐藏head.COMP\_TRANS\_SEQ<转运人SEQ> |
| CONTRACT\_NO | Head.ORDER\_NO<申请编号> |
| PURCHASE\_NO | <商业合同号>手工输入，非必填项。 |
| INVOICE\_NO | <商业发票号>手工输入，非必填项。 |
| FORWARD\_RF\_NO | <货代参考号>手工输入，非必填项。 |
| 船名/航次\* | 选择器，弹出“远期交割航线选择器”，传递参数preSaleHeadSEQ |
| 截重时间 | 选择器选择后，动态加载 |
| 预计到港时间 | 选择器选择后，动态加载 |
| 预计离港时间 | 选择器选择后，动态加载 |
| 运输条款 | Head.TRANS\_TYPE |
| 中转港 | Head.TRANSFER\_PORT |
| 运费条款 | Head. RULE\_PRICE |
| 付款地点 | Head. PAY\_LOCATION，根据字典转换 |
| SOC | 默认为“否”； |
| 装货港 | Head.LOADING\_PORT  Head. LOADING\_NAME\_EN（根据语系来选择显示）  Head. LOADING\_NAME\_CN |
| 卸货港 | Head.DESTINATION\_PORT  Head. DESTINATION\_PORT\_NAME  Head. DESTINATION\_PORT\_NAME\_EN |
| 收货地 交货地 | 根据FAK\_PRICE\_NO动态加载 |
| 货物描述 | Head. CARGO\_DESC，对应字段CARGO\_DESC |
| 联系人、电子邮件、电话、 传真 | Head对应字段：LINKER<联系人姓名>、LINKER\_EMAIL<电子邮件>、LINKER\_PHONE<电话>、LINKER\_FAX<传真> |
| 订舱公司地址 | Head. COMP\_SEND\_ADDR<订舱公司地址>，对应字段COMP\_SEND\_ADDR |
| SHIP FROM | 当运输条款为Door/\*时“SF公司名称和地址/联系人/联系电话”必填;  SHIP\_FROM\_LINKER<联系人>  SHIP\_FROM\_LINKPHONE<联系人电话>  SHIP\_FROM\_ADDR<名称和地址> |
| SHIP TO | 当运输条款为\*/Door时“ST公司名称和地址/联系人/联系电话”必填;  SHIP\_TO\_LINKER<联系人>  SHIP\_TO\_LINKPHONE<联系人电话>  SHIP\_TO\_ADDR<名称和地址> |
| 【箱型列表】 | |
| 箱型\* | 固定加载detail.CONTA\_SIZE（CONTA\_SIZE顺序加载）。 |
| 数量\* | 文本框，只允许输入整形数字，不能为空。  选择【集装箱箱型】但未填写数量，需提示“集装箱箱型的数量不能为空”。消息号：ContaNumberIsEmpty   * 编辑模式   按箱型分组显示（例如：20GP\*2，订舱明细表有2个一样记录（SEQ除外），页面只能显示一个20GP，数量2） |
| 预计总量/可订数 | 匹配infoList中的CONTA\_SIZE  预计总量 = totalUnit  可订数 = remindUnit  如果没有匹配到，两者都显示“-” |
| FAK价 | 选择器选择后，动态加载 |
| 箱号 | 文本框 |
| 超重 | 默认不选择 |
| 当柜重量 | “超重”选中后，文本框录入 |
| 交互 |  |
| 界面初始化 | * 新增模式   命名空间PHY.BKONLINE.Facade.FuturePresale  获取传过来的preSaleHeadSeq，调用接口IOMTFuturePresale.QueryByHeadSeq  返回实体：FuturePresaleModel   * 编辑模式   命名空间PHY.BKONLINE.Facade.Order  调用接口IOMTOrder.QueryOrderBySeq  返回实体：OrderData (订舱主表和订舱子表)  不管新增、编辑，都执行以下：  获取传过来的preSaleHeadSeq，调用接口IOMTFuturePresale.GetTradInfo  返回实体：infoList = List<FPTTradInfo>  获取【箱型列表】中的“预计总量/可订数”，每次都即时计算； |
| 页面交互 | 1. 当“远期交割航线选择器”选中结果，返回后，   [1]动态加载：（来源于返回值）  截重时间= PLAN\_LEAVE  预计离港时间= PLAN\_LEAVE  预计到港时间=PLAN\_ARRIVE  隐藏字段FAK\_PRICE\_NO = FAK\_PRICE\_NO  [2]当返回的FAK\_PRICE\_NO不为NULL，则JS调用后台接口IOMTPrice.GetListBySql  加载收货地、交货地和FAK价格等信息  参数赋值：  sqlWhere = “PRICE\_NO = 参数FAK\_PRICE\_NO AND OWNER\_SEQ = 当前预售订舱单的OWNER\_SEQ AND LOADING\_PORT = {页面装货港} AND DESTINATION\_PORT = {页面卸货港} + RULE\_PRICE={页面运费条款} AND TRANS\_TYPE={页面运输方式}”  orderString = “ORDER BY CONTA\_SIZE DESC”  pageRows = 1000  获得FAK信息（即PriceData），   * 收货地加载：LOADING\_LOACTION（按半角逗号为分隔符拆分）   （PS：需要排重）  LOADING\_LOACTION\_NAME,LOADING\_LOACTION\_NAME\_EN（按半角<\_>为分隔符拆分）  默认选中与【装货港】一致的值   * 交货地加载：DESTINATION\_LOCATION（同收货地）,   DESTINATION\_LOCATION\_NAME,DESTINATION\_LOCATION\_NAME\_EN  默认选中与【卸货港】一致的值  匹配不到，默认选中第一个   * 箱型列表FAK价（对应字段PRICE\_TOTAL）加载：   根据箱型列表中的箱型，匹配PriceData.CONTA\_SIZE，匹配不到，加载“-”  ~~PS：当船期选择器返回NULL数据，则清空上述联动字段（FAK价格恢复为“-”）~~   1. 当用户选中SOC为“是”，则箱号\*必填，箱数量默认加载为1，不允许编辑； 2. 当用户选中某个箱的“超重”，则该箱的单柜重量为文本框，允许录入； |
| 保存逻辑 | * 非空校验 * 校验每个箱型的“数量”与“可订数”，前者>后者（只要一个不满足停止，提示信息），则提示   BKM100614 操作失败，原因：本次交割的箱型{0}数量超过可交割的数量！   * 对订单信息进行赋值（远期运价转即期，有很多信息来源与挂靠的FAK）   调用OMTPrice.GetListBySql，根据页面交互获取FAK的逻辑，  获得FAK运价后，对下面字段进行重新赋值：  [1]订舱主表（FAK主表）  SHIP\_CODE<船舶代码>  SHIP\_NAME\_CN<中文船名>  SHIP\_NAME\_EN<英文船名>  VOYAGE<航次>  EXPIRY\_DATETIME<截重时间>  ETD\_OUT\_DATE<预计开船日期>  ETA\_IN\_DATE<预计到港日期>  EFFECTIVE\_DATETIME<开仓时间>  SHIP\_VOYAGE\_HEAD\_SEQ<船期信息关联>  IS\_TRANSFER<是否中转>  TRANSFER\_PORT<中转港>  TRANSPORT\_NAME\_EN<中转港英文>  TRANSPORT\_NAME\_CN<中转港中文>  [2]订舱子表（FAK主表）  BUSINESS\_ORDER\_SEQ<参考FAK关联SEQ>  FAK\_BOOKING\_PRICE\_NO<FAK运价编号关联>   * 调用IOMTOrder：SaveOrder方法   参数赋值：  OrderData data：传递的是订舱表和订舱子表，请根据数据测试案例进行赋值  oprtType：新增模式 = ADD，编辑模式 = UPD  PS：编辑模式下，SaveOrder并不是通过先删后加的方式操作的，它是存在则更新，不存在则新增。所以，编辑保存，订舱明细表的SEQ必须赋原来的值。   * 加箱情况，多加的箱的明细记录SEQ = 0（接口会作为新增来处理）； * 加箱情况，对应该箱型的明细记录SEQ（随便一个就行，接口会把没有的标志位备份状态）； * 不加箱不减箱，对应箱型的明细记录SEQ；   例如：20GP\*2（SEQ：201,202），变更保存  201，20GP  202，20GP  （可参考订舱申请页面）  操作成功提示“已保存（申请编号）订舱信息！”，失败提示“保存（申请编号）失败，errorMsg。”刷新列表页面。（取FAK订舱页面一样的消息号） |
| 提交逻辑 | 数据处理逻辑与“保存”一致；  调用OMTOrder：SubmitAndUpdOrder方法  参数赋值：  OrderData data：传递的是订舱表和订舱子表，请根据数据测试案例进行赋值  oprtType：新增模式 = ADD，编辑模式 = UPD  操作成功提示“已提交（申请编号）订舱信息！”，失败提示“保存（申请编号）失败，errorMsg。”刷新列表页面。（取FAK订舱页面一样的消息号） |
| 关闭逻辑 | 关闭当前录入页面。 |

### 数据测试用例

|  |  |  |
| --- | --- | --- |
| 功能名称 | 测试过程 | 期望结算 |
| 暂存远期预售订舱单 | 远期订舱-》远期舱位预订  选中以下远期运价  CNSHK→SGSIN  中转港：CSC  出运日期：  W04(2014/01/19~01/25)  的远期预售订舱单  （一个20GP的箱，100USD）  -》订舱 | \* FUTURE\_PRESALE\_HEAD<远期预售主表>新增一条记录  FUTURE\_PRESALE\_HEAD\_SEQ=数据库生成  ORDER\_NO<申请编号>=OWNER\_CODE+YYMMDD+ FUTURE\_PRESALE\_ORDERNO\_SEQ  CUSTOMER\_SEQ<订舱公司>=当前用户所属公司SEQ  CUSTOMER\_NAME<订舱公司名称>=来源于公司表  CUSTOMER\_NAME\_PY<订舱公司简称>=来源于公司表  CUSTOMER\_NAME\_EN<订舱公司英文名称>=来源与公司表  OPRT\_COMP\_SEQ<口岸公司>=选中的远期运价  OPRT\_COMP\_NAME<口岸公司名称>=选中的远期运价  OPRT\_COMP\_NAME\_EN<口岸公司英文名称>=选中的远期运价  OWNER\_SEQ<船公司>=选中的远期运价船公司SEQ  OWNER\_CODE=选中的远期运价船公司CODE  OWNER\_NAME=选中的远期运价船公司NAME  COMP\_SEND<托运人>=页面录入  COMP\_GET<收货人>=页面录入  COMP\_TRANS<转运人>=页面录入  COMP\_SEND\_SEQ<托运人SEQ>=来源于Customer表  COMP\_GET\_SEQ<收货人SEQ>=来源于Customer表  COMP\_TRANS\_SEQ<转运人SEQ>=来源于Customer表  COMP\_SEND\_ADDR<订舱公司地址>=页面录入  LINKER<联系人姓名>=页面录入  LINKER\_EMAIL<电子邮件>=页面录入  LINKER\_PHONE<电话>=页面录入  LINKER\_FAX<传真>=页面录入  ORDER\_REMARK<订舱备注>=页面录入  SHIP\_ROUTE\_HEAD\_SEQ<航线关联>=来源于远期运价表  ROUTE\_CODE<航线代码>=来源于远期运价表  ROUTE\_NAME<航线名称>=来源于远期运价表  BERTH\_WEEK<班期>=来源于远期运价表  LOADING\_PORT<装货港>=来源于远期运价表  LOADING\_NAME\_EN<装货港英文>=来源于远期运价表  LOADING\_NAME\_CN<装货港中文>=来源于远期运价表  DESTINATION\_PORT<卸货港>=来源于远期运价表  DESTINATION\_PORT\_NAME<卸货港中文名称>=来源于远期运价表  DESTINATION\_PORT\_NAME\_EN<卸货港英文名称>=  来源于远期运价表  RULE\_PRICE<运费条款代码>=来源于远期运价表  RULE\_PRICE\_NAME<运费条款中文名称>=来源于远期运价表  RULE\_PRICE\_NAME\_EN<运费条款英文名称>=来源于远期运价表  TRANS\_TYPE<运输条款代码>=来源于远期运价表  TRANS\_TYPE\_NAME<运输条款中文名称>=来源于远期运价表  TRANS\_TYPE\_NAME\_EN<运输条款英文名称>=来源于远期运价表  TRANSFER\_PORT<中转港>=来源于远期运价表  TRANSPORT\_NAME\_EN<中转港英文>=来源于远期运价表  TRANSPORT\_NAME\_CN<中转港中文>=来源于远期运价表  CROSS\_BEFOREDAY<允许交割提前天数>=来源于远期运价表  CROSS\_DELAYDAY<允许交割延后天数>=来源于远期运价表  WEEK\_NO<出运周数>=W04  WEEK\_BEGIN=2014/01/19  WEEK\_END=2014/01/25  PUBLISH\_PRICE\_SUM<发布价格概要>=来源于远期运价表  FUTURE\_PRICE\_HEAD\_SEQ=来源于远期运价表  FUTURE\_PRICE\_NO<运价编号>=来源于远期运价表  PAY\_LOCATION<付款地点代码>=页面录入  CARGO\_DESC<货物描述>=页面录入  SALE\_SPACE<可售舱位>= 来源于远期运价表MAX\_TEU  SALE\_F\_SPACE<已售舱位>= NULL  SALE\_R\_SPACE<剩余舱位>= NULL  ORDER\_SPACE<本次订舱(TEU)>= NULL  SALE\_W\_SPACE<本周已售舱位>= NULL  CREDIT\_LEVEL<信用等级>=来源于Customer表  LAST\_CHANGE\_SEQ<最近改期的SEQ>=0  CHANGE\_COUNT<改期次数>=0  SUBMITUSER<提交人>= NULL  SUBMIT\_DATE<提交时间>= NULL  APPROVER<审批人>= NULL  APPROVED\_DATE<审批时间>= NULL  OP\_CHECKRESON<审批意见>=NULL  CONTA\_SIZE\_SUM<订单箱量汇总>=箱型x数量，多个用半角分号串联，例如：箱型1x数量；箱型2x数量；...  SIGN\_AGREEMENT<同意协议>= 1是  STATUS=0 草稿  时间戳  \* FUTURE\_PRESALE\_DETAIL<远期预售明细表>新增记录到表  FUTURE\_PRESALE\_DETAIL\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  FUTURE\_PRICE\_CONTADETAIL\_SEQ=远期运价箱表SEQ  IS\_APPLYNEWPRICE<是否应用新价格>=0  ORDER\_NO<申请编号>=主表ORDER\_NO  CONTA\_SIZE<标准箱型>=选中的远期运价标准箱型  CONTA\_SIZE\_OWNER<箱型>=来源于远期运价  CONTA\_SIZE\_REPORT<统计箱型>=来源于远期运价  CONTA\_SZIE\_DISP<显示箱型>=来源于远期运价  CONTA\_SIZE\_DESC<箱型描述>=来源于远期运价  PRICE\_UNIT<币种>=来源于远期运价  UNIT<箱数量>=页面录入  FINAL\_PRICE\_TOTAL<最终成交价>=来源于远期运价  时间戳  \* FUTURE\_PRESALE\_LOG<远期预售订舱申请操作日志>新增一条记录  FUTURE\_PRESALE\_LOG\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=D  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS= 10 有效  时间戳 |
| 提交远期预售订舱单 | 远期订舱-》远期订舱管理-》选中记录-》提交 | \* FUTURE\_PRESALE\_HEAD<远期预售主表>更新  STATUS=10  SUBMITUSER<提交人>=当前用户  SUBMIT\_DATE<提交时间>=当前时间  时间戳  \* FUTURE\_PRESALE\_LOG<远期预售订舱申请操作日志>新增一条记录  FUTURE\_PRESALE\_LOG\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=S  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS= 10 有效  时间戳 |
| 删除 | 远期订舱-》远期订舱管理-》选中记录-》删除 | \* FUTURE\_PRESALE\_HEAD<远期预售主表>更新  STATUS=-20已删除  时间戳  \* FUTURE\_PRESALE\_LOG<远期预售订舱申请操作日志>新增一条记录  FUTURE\_PRESALE\_LOG\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION= I: 删除草稿  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS= 10 有效  时间戳 |
| 回收（已提交的订舱单） | 远期订舱-》远期订舱管理-》选中记录-》回收 | \* FUTURE\_PRESALE\_HEAD<远期预售主表>更新  STATUS=0草稿  时间戳  \* FUTURE\_PRESALE\_LOG<远期预售订舱申请操作日志>新增一条记录  FUTURE\_PRESALE\_LOG\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION= C:撤回提交的申请  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS= 10 有效  时间戳 |
| 撤销（已审批通过的订舱单） | 远期订舱-》远期订舱管理-》选中记录-》撤销 | \* FUTURE\_PRESALE\_HEAD<远期预售主表>更新  STATUS=-10已撤单  时间戳  \* FUTURE\_PRESALE\_LOG<远期预售订舱申请操作日志>新增一条记录  FUTURE\_PRESALE\_LOG\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=预售主表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION= CA:撤销已审批通过  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS= 10 有效  时间戳 |
| 交割>>保存 | 订舱人>>远期订舱>>远期订舱管理>>列表>>交割  （注意：交割箱数量，保存到  “BUSINESS\_ORDER\_DETAIL”是按一个箱一条记录） | \* BUSINESS\_ORDER<订单表>新增一条记录  自于“FUTURE\_PRESALE\_HEAD<远期预售主表>”一样字段，标示为Presale  FAK\_BOOKING\_PRICE\_NO<FAK运价编号关联> = 选中船期返回的FAK\_PRICE\_NO  CUSTOMER\_SEQ<订舱公司>=Presale  CUSTOMER\_NAME<订舱公司名称>=Presale  CUSTOMER\_NAME\_PY<订舱公司简称>=Presale  CUSTOMER\_NAME\_EN<订舱公司英文名称>=Presale  OPRT\_COMP\_SEQ<口岸公司>=Presale  OPRT\_COMP\_NAME<口岸公司名称>=Presale OPRT\_COMP\_NAME\_EN<口岸公司英文名称>=Presale  ORDER\_NO<订单号码>=自动生成，唯一，格式正确  BOOKING\_NO<订舱编号> =NULL  BUSINESS\_TYPE<订舱方式>= FPT  STATUS<状态>= 0草稿  COMP\_SEND<托运人>=Presale  COMP\_GET<收货人>=Presale  COMP\_TRANS<转运人>=Presale  CONTRACT\_NO<协议号>= Presale.ORDER\_NO  PURCHASE\_NO<商业合同号>=录入，可空  INVOICE\_NO<商业发票号>=录入，可空  FORWARD\_RF\_NO<货代参考号>=录入，可空  OWNER\_SEQ<船公司>=Presale  OWNER\_CODE=Presale  OWNER\_NAME=Presale  SHIP\_CODE<船舶代码>=页面选中FAK对应的船期  SHIP\_NAME\_CN<中文船名>=页面选中FAK对应的船期  SHIP\_NAME\_EN<英文船名>=页面选中FAK对应的船期  VOYAGE<航次>=页面选中FAK对应的船期  CALL\_SIGN<呼号>= -  ETD\_OUT\_DATE<预计开船日期>=页面预计离港时间  ETA\_IN\_DATE<预计到港日期>=页面预计到港时间  EXPIRY\_DATETIME<截重时间>=页面截重时间  EFFECTIVE\_DATETIME<开仓时间>=取挂靠FAK的信息  TRANS\_TYPE\_CODE<运输方式代码>= PreSale  TRANS\_TYPE\_NAME<运输方式名称>= PreSale  CTRANS\_TYPE\_CODE<托运方式代码>= PreSale  CTRANS\_TYPE\_NAME<托运方式名称>= PreSale  CARGO\_DESC<货物描述>= PreSale  LOADING\_PORT<装货港>= PreSale  LOADING\_PORT\_NAME<装货港名称>= PreSale  LOADING\_PORT\_NAME\_EN<装货港英文名称>= PreSale  DESTINATION\_PORT<卸货港>= PreSale  DESTINATION\_PORT\_NAME<卸货港名称>= PreSale  DESTINATION\_PORT\_NAME\_EN<卸货港英文名称>= PreSale  LOADING\_LOACTION<收货地>=页面选中  LOADING\_LOACTION\_NAME<收货地名称>=  页面选中收货地中文名称  LOADING\_LOACTION\_NAME\_EN<收货地英文名称>=页面选中收货地英文名称  DESTINATION\_LOCATION<交货地>=页面选中  DESTINATION\_LOCATION\_NAME<交货地名称>=页面选中交货地名称  DESTINATION\_LOCATION\_NAME\_EN<交货地英文名称>=页面选中交货地英语名称  LINKER<联系人姓名>= PreSale  LINKER\_EMAIL<电子邮件>= PreSale  LINKER\_PHONE<电话>= PreSale  LINKER\_FAX<传真>= PreSale  WEIGHT<重量>=0  WEIGHT\_UNIT<重量单位>= -  PICS<件数>=0  PICS\_UNIT<包装方式>= -  SHIP\_FROM\_LINKER<联系人>=页面录入  SHIP\_FROM\_LINKPHONE<联系人电话>=页面录入  SHIP\_FROM\_ADDR<名称和地址>=页面录入  SHIP\_TO\_LINKER<联系人>=页面录入  SHIP\_TO\_LINKPHONE<联系人电话>=页面录入  SHIP\_TO\_ADDR<名称和地址>= 页面录入  ORDER\_REMARK<订舱备注>= PreSale  OP\_CHECKRESON<审核原因>=NULL  SALE\_SPACE<可售舱位>= NULL  SALE\_F\_SPACE<已售舱位>= NULL  SALE\_R\_SPACE<剩余舱位>= NULL  ORDER\_SPACE<本次订舱(TEU)>= NULL  SPECIAL\_LOAD<特殊装载>=NULL  SPECIAL\_REMARK<特殊备注>= NULL  INQUIRE\_PRICE\_SEQ<议价单号>=0  SHIP\_VOYAGE\_HEAD\_SEQ<船期信息关联>=取挂靠FAK的信息  SHIP\_ROUTE\_HEAD\_SEQ<航线关联>= PreSale  ROUTE\_CODE<航线代码>= PreSale  ROUTE\_NAME<航线名称>= PreSale  BOOKING\_PRICE\_SEQ<参考FAK关联SEQ>=废弃字段，不用关注  SALE\_USER\_ACCOUNT<航线销售员>=NULL  PAY\_LOCATION<付款地点>= PreSale  SPLIT\_PARENT\_SEQ<拆单关联字段>=NULL  APPROVER<放舱人>=NULL  APPROVED\_DATE<放舱时间>=NULL  DISCHARGE\_TRAN\_WAY<二程运输方式>=NULL  COMP\_SEND\_SEQ<托运人SEQ>=PreSale  COMP\_GET\_SEQ<收货人SEQ>= PreSale  COMP\_TRANS\_SEQ<转运人SEQ>= PreSale  IS\_TRANSFER<是否中转>=取挂靠FAK的信息  TRANSFER\_PORT<中转港>=取挂靠FAK的信息  TRANSPORT\_NAME\_EN<中转港英文>=取挂靠FAK的信息  TRANSPORT\_NAME\_CN<中转港中文>=取挂靠FAK的信息  备用字段为NULL  时间戳  \*BUSINESS\_ORDER\_DETAIL<订单明细表>新增记录  自于“FUTURE\_PRESALE\_DETAIL<远期预售明细表>”一样字段，标示为PresaleDetail  BUSINESS\_ORDER\_SEQ=主表SEQ  ORDER\_NO<订单号码（申请编号）>=  BUSINESS\_ORDER.ORDER\_NO  BOOKING\_NO<订舱编号>=NULL  BOOKING\_PRICE\_SEQ=  BUSINESS\_ORDER. BOOKING\_PRICE\_SEQ  FAK\_BOOKING\_PRICE\_NO<FAK运价编号关联>=  BOOKING\_ ORDER.FAK\_BOOKING\_PRICE\_NO  CONTA\_NO<箱号>=页面录入，只有SOC为必填  CONTA\_SIZE<标准箱型>=PresaleDetail  CONTA\_SIZE\_OWNER<箱型>=PresaleDetail  CONTA\_SIZE\_REPORT<统计箱型>=PresaleDetail  CONTA\_SZIE\_DISP<显示箱型>=PresaleDetail  CONTA\_SIZE\_DESC<箱型描述>=PresaleDetail  PRICE\_UNIT<币种>=PresaleDetail  PRICE\_TOTAL<FAK报价>=页面显示的FAK价  PRICE\_TRANS<申请价>=  PresaleDetail.FINAL\_PRICE\_TOTAL  PRICE\_FEE<佣金>=NULL  PRICE\_FEE\_RATIO<佣金比例>=NULL  FINAL\_PRICE\_TOTAL<成交价>=  PresaleDetail.FINAL\_PRICE\_TOTAL  FINAL\_PRICE\_FEE<结算佣金>=NULL  FINAL\_PRICE\_FEE\_LOGGER<佣金计算说明>=NULL  STATUS<状态>= 0 - 草稿  CONTA\_MOVEMENT\_TIME<箱动态变化时间>=NULL  IS\_SOC<自用柜标记>=页面选中  IS\_OVER\_WEIGHT<是否超重>=页面勾选  CONTA\_WEIGHT<单柜重量>=页面录入  时间戳 |
| 交割>>提交 | 提交 | 更新页面修改的内容外  \* BUSINESS\_ORDER<订单表>更新一条记录  STATUS = 10 –已提交  时间戳  \*BUSINESS\_ORDER\_DETAIL<订单明细表>更新记录  STATUS = 10 –已提交  时间戳 |
|  |  |  |

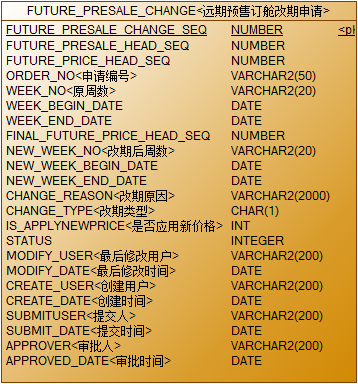
## 远期订舱改期

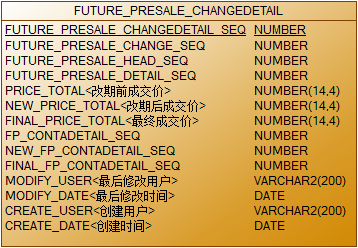
* 批量操作（提交、删除）
* 管理页面
* 触发远期订舱改期的查询页面（与订舱人远期订舱的管理列表不一样，所以要新做）
* 申请/修改页面
* 详情页面

### 功能描述

提供给订舱人，对远期预售订舱进行改期申请的一系列操作。

### 数据表





FUTURE\_PRESALE\_CHANGE<远期预售订舱改期申请> 1：N FUTURE\_PRESALE\_CHANGEDETAIL

### 菜单

订舱人 >> 远期订舱 >> 远期订舱改期

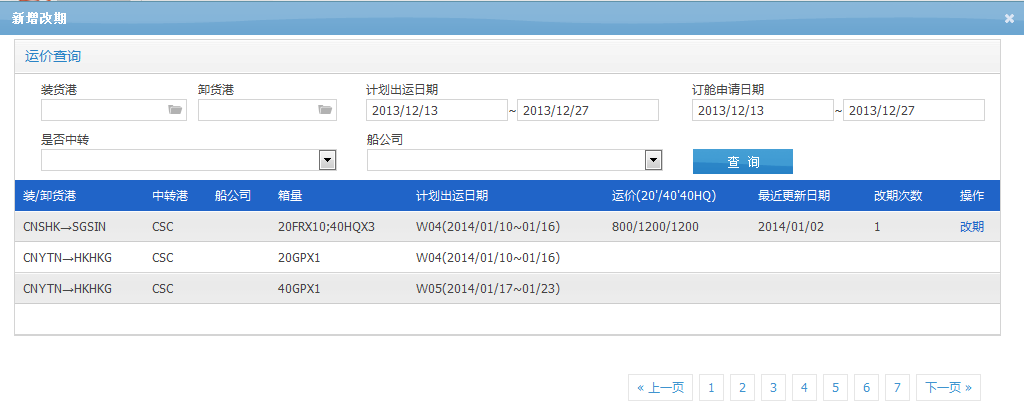
### 列表界面



(支持分页)

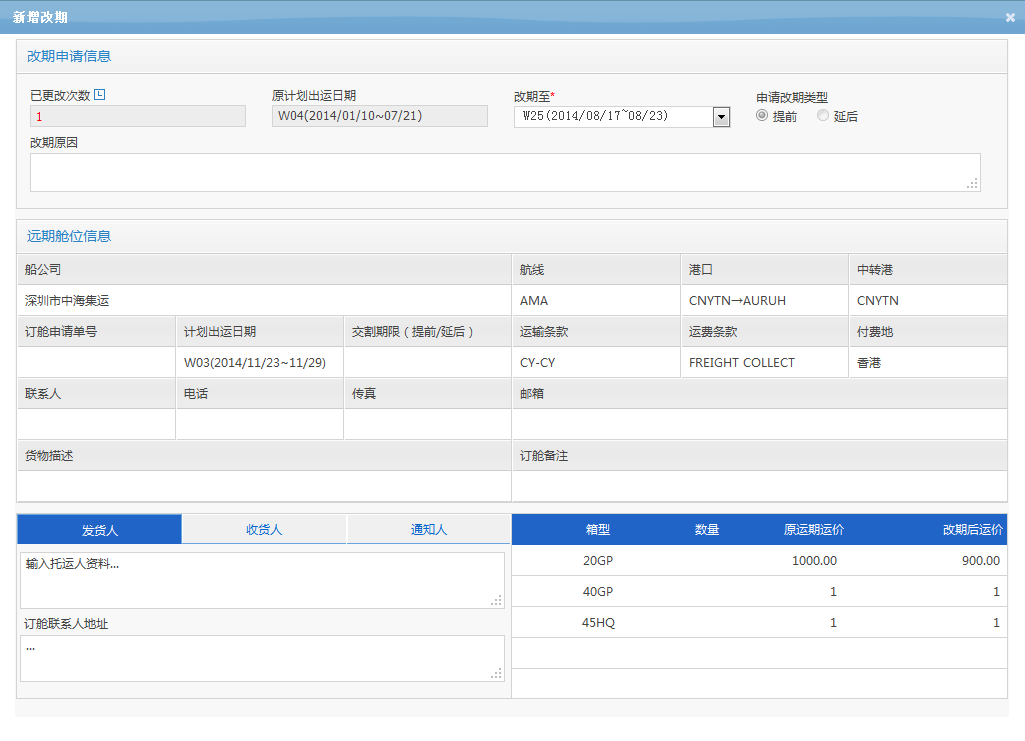
|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 查询条件 |  |
| 装货港 | 使用港口选择器选择，数据来源于公共港口代码  调用 订舱人-》订舱申请-》“起运港”选择器  支持单选 |
| 卸货港 | 支持选择器，调用 订舱人-》订舱申请-》“目的港”选择器  支持单选 |
| 是否中转 | 下拉框，加载：  NULL（默认）  直航  中转 |
| 计划出运日期 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间  默认结束时间 = 当前时间+2周  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 船公司 | 下拉框，加载所有船公司  NULL（默认）  与“订舱申请”页面的船公司一致 |
| 状态 | 下拉框，字典表加入，NULL（默认）  0草稿,10待审, 90已拒绝,100已审核,-20已删除,已提交(10,90,100)  PS：已提交，是个虚拟状态，表示包括10待审,90已拒绝,100已审核  PS：选择NULL时，查询需要把非ALL的传递进去 |
| 申请时间 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间 – 2周  默认结束时间 = 当前时间  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 操作列 | * 删除（0草稿） * 修改（0 草稿） * 提交（0草稿、90 拒绝） * 回收（10待审） |
| 列表项 |  |
| 港口 | LOADING\_PORT - > DESTINATION\_PORT |
| 中转港 | TRANSFER\_PORT |
| 船公司 | OWNER\_CODE |
| 箱量 | CONTA\_SIZE\_SUM |
| 计划出运日期 | WEEK\_NO（WEEK\_BEGIN~WEEK\_END） |
| 类型 | CHANGE\_TYPE<改期类型>，根据字典转换显示 |
| 申请更改出运日期 | NEW\_WEEK\_NO<改期后周数>( NEW\_WEEK\_BEGIN\_DATE~  NEW\_WEEK\_END\_DATE) |
| 创建日期 | CREATE\_DATETIME |
| 状态 | STATUS，根据字典转换 |
| 操作 |  |
| 查询 | 调用接口IOMTFutureChange.QueryForListToCustomerByPage  参数赋值parm（匹配页面的查询条件）  orderBy = P.WEEK\_BEGIN\_DATE ASC,  FUTURE\_PRESALE\_CHANGE\_SEQ DESC  按出运日期的顺序排列 |
| 新增 | 弹出窗口，链接到“查询远期舱位页面” |
| 修改 | 弹出窗口，链接到“查询远期舱位页面”，传参为改期主表SEQ  （PS：业务逻辑中，只有草稿、作废才能编辑） |
| 提交 | 调用接口IOMTFutureChange.Submit（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 删除 | 调用接口IOMTFutureChange.Delete（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 回收 | 调用接口IOMTFutureChange.RecallSubmit（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 点击“港口” | 弹出窗口，链接到“详情页面” |
| 批量提交 | 调用接口IOMTFutureChange.BatchSubmit  参数changeSEQList = FUTURE\_PRESALE\_CHANGE\_SEQ（选中） |
| 批量删除 | 调用接口IOMTFutureChange.BatchDelete  参数changeSEQList = FUTURE\_PRESALE\_CHANGE\_SEQ（选中） |

### 查询远期舱位页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 查询条件 |  |
| 装货港 | 使用港口选择器选择，数据来源于公共港口代码  调用 订舱人-》订舱申请-》“起运港”选择器 |
| 卸货港 | 支持选择器，调用 订舱人-》订舱申请-》“目的港”选择器 |
| 计划出运日期 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间  默认结束时间 = 当前时间+2周  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 订舱申请日期 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间 – 2周  默认结束时间 = 当前时间  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 是否中转 | 下拉框，加载：  NULL（默认）  直航  中转 |
| 船公司 | 下拉框，加载所有船公司  NULL（默认）  与“订舱申请”页面的船公司一致 |
| 列表项 |  |
| 装/卸货港 | LOADING\_PORT<装货港>/DESTINATION\_PORT<卸货港> |
| 中转港 | TRANSFER\_PORT<中转港> |
| 船公司 | OWNER\_CODE |
| 箱量 | CONTA\_SIZE\_SUM<订单箱量汇总> |
| 计划出运日期 | WEEK\_NO（WEEK\_BEGIN~WEEK\_END） |
| 运价 | PUBLISH\_PRICE\_SUM |
| 最近更新日期 | MODIFY\_DATE |
| 改期次数 | CHANGE\_COUNT |
| 操作 | 改期（查询出来的记录一定是允许改期的） |
| 操作 |  |
| 查询 | 调用IOMTFutureChange.QueryPreSaleForChange接口  参数QueryPreSaleForChangeParm 匹配页面查询条件  orderBy = WEEK\_BEGIN\_DATE DESC |
| 改期 | 当前弹出窗口刷新，跳转到“新增编辑页面”，传参  FUTURE\_PRESALE\_HEAD\_SEQ |

### 新增编辑页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 | \*为必填（无特别说明，以标签显示） |
| 已更改次数 | CHANGE\_COUNT<改期次数> |
| 原计划出运日期 | WEEK\_NO<出运周数>( WEEK\_BEGIN~WEEK\_END) |
| 改期至\* | 下拉框，对应字段（NEW\_WEEK\_NO/ NEW\_WEEK\_BEGIN\_DATE/  NEW\_WEEK\_END\_DATE)  根据当前出运日期，加载（往前推4周+NULL+往后延4周）的数据。  展示方式 = 原出运计划日期  Value = Text  新增时NULL（默认选中） |
| 申请改期类型 | 单选框，对应字段CHANGE\_TYPE<改期类型>  B：提前  A：延后  不允许编辑 |
| 改期原因 | 文本框录入，对应CHANGE\_REASON<改期原因> |
| 远期舱位信息（无特别说明，以标签显示）对应FuturePresaleModel.FuturePresaleData | |
| 船公司 | OWNER\_CODE |
| 航线 | ROUTE\_CODE<航线代码> |
| 港口 | LOADING\_PORT<装货港> -> DESTINATION\_PORT<卸货港> |
| 中转港 | TRANSFER\_PORT<中转港> |
| 订舱申请单号 | ORDER\_NO<申请编号> |
| 计划出运日期 | WEEK\_NO<出运周数>( WEEK\_BEGIN~WEEK\_END) |
| 交割期限(提前/延后) | CROSS\_BEFOREDAY<允许交割提前天数>  CROSS\_DELAYDAY<允许交割延后天数> |
| 运输条款 | 对应字段，TRANS\_TYPE\_NAME<运输条款中文名称>  英文环境，取TRANS\_TYPE\_NAME\_EN<运输条款英文名称> |
| 运费条款 | 对应字段，RULE\_PRICE\_NAME<运费条款中文名称>  英文环境，取RULE\_PRICE\_NAME\_EN<运费条款英文名称> |
| 付款地 | PAY\_LOCATION<付款地点代码>，请根据DICT\_CODE字段进行转换  DICT\_TYPE = PAY\_LOCATION  NULL的，不转换 |
| 联系人 | LINKER<联系人姓名> |
| 电话 | LINKER\_PHONE<电话> |
| 传真 | LINKER\_FAX<传真> |
| 邮箱 | LINKER\_EMAIL<电子邮件> |
| 货物描述 | CARGO\_DESC<货物描述> |
| 订舱备注 | ORDER\_REMARK<订舱备注> |
| 发货人 | COMP\_SEND<托运人> |
| 收货人 | COMP\_GET<收货人> |
| 通知人 | COMP\_TRANS<转运人> |
| 订舱联系人地址 | COMP\_SEND\_ADDR<订舱公司地址> |
| 箱型列表（无特别说明，以标签显示）循环FuturePresaleModel.FuturePresaleDetail(预售订舱单箱表信息) | |
| 箱型 | CONTA\_SIZE<标准箱型>  （隐藏保留FuturePresaleDetail. FUTURE\_PRICE\_CONTADETAIL\_SEQ） |
| 数量 | UNIT<箱数量> |
| 原运期运价 | FINAL\_PRICE\_TOTAL |
| 改期后运价 | * 草稿下编辑情况（初始化）   查询FutureChangeEntity.FutureChangeDetailMuti，  FUTURE\_PRESALE\_DETAIL\_SEQ = 当前  FuturePresaleDetail.FUTURE\_PRESALE\_DETAIL\_SEQ，  获得记录的“NEW\_PRICE\_TOTAL”输出 |
| 操作 |  |
| 页面加载 | * 新增情况   根据参数FUTURE\_PRESALE\_HEAD\_SEQ，获取远期预售订舱，  调用接口：IOMTFutureChange.QueryPreSaleByHeadSeq  PreSaleEntity – output （预售订舱单信息）   * 草稿下编辑情况   根据FUTURE\_PRESALE\_CHANGE\_SEQ，获取远期改期信息，  调用接口：IOMTFutureChange.QueryChangeByHeadSeq  参数 headSeq = FUTURE\_PRESALE\_CHANGE\_SEQ  Data FutureChangeEntity – output（包括改期信息和远期预售订舱信息） |
| 改期至 | 选择后，JS调用IOMTFutureChange.GetNewFuturePriceData  参数赋值：  futurePriceHeadSeq ：FUTURE\_PRICE\_HEAD\_SEQ  newPeriod ：改期至的记录，周数、开始、结束日期赋值  data – output，返回的就是改期后远期运价的信息，  请判断data是否为Null（调用实体的IsNull方法），返回非空，则表示获取到远期运价，则给“改期后运价”赋值：   * 根据箱型匹配，根据FuturePriceEntity.futurePriceDetail.contaPriceDataSingle * 匹配到，取“PUBLISH\_PRICE\_TOTAL<发布运价>”值   （隐藏保留当前匹配的data.futurePriceDetail.  FUTURE\_PRICE\_CONTADETAIL\_SEQ）  PS：如匹配不到对应箱型的运价，显示为“-”，不允许提交。 |
| 点击“已更改次数” | 弹出窗口，链接到远期订舱改期申请 >>“改期申请历史”  传参 ORDER\_NO<远期预售订舱单> |
| 暂存 | 页面必填字段校验；  调用接口IOMTFutureChange.Save方法  参数FutureChangeEntity赋值：  FutureChangeHeadSingle：主表记录，根据页面赋值  - FUTURE\_PRESALE\_HEAD\_SEQ= 远期预售订舱单主表SEQ  - FUTURE\_PRICE\_HEAD\_SEQ = 远期预售订舱  FUTURE\_PRICE\_HEAD\_SEQ  - NEW\_FUTURE\_PRICE\_HEAD\_SEQ = 新运价的主表SEQ  - IS\_APPLYNEWPRICE<是否应用新价格>=0  - FUTURE\_PRESALE\_CHANGE\_SEQ（新增时，不需要赋值）  其他值，对应页面标签  FutureChangeDetailMuti：子表记录（多条）  - FUTURE\_PRESALE\_CHANGEDETAIL\_SEQ=不赋值  - FUTURE\_PRESALE\_CHANGE\_SEQ=不赋值  - FUTURE\_PRESALE\_HEAD\_SEQ= 远期预售订舱单主表SEQ  - FUTURE\_PRESALE\_DETAIL\_SEQ =远期预售订舱单箱表SEQ  - PRICE\_TOTAL<改期前成交价>= 原运价  - NEW\_PRICE\_TOTAL<改期后成交价>= 新原价  - FINAL\_PRICE\_TOTAL<最终成交价>=NULL  - FP\_CONTADETAIL\_SEQ = 原运价箱表SEQ（从隐藏字段获取）  - NEW\_FP\_CONTADETAIL\_SEQ = 新运价箱表SEQ（从隐藏字段获取）  - FINAL\_FP\_CONTADETAIL\_SEQ = 0 |
| 提交 | 调用接口IOMTFutureChange.SaveAndSubmit方法  参数赋值与“暂存”一致。 |
| 关闭 | 关闭当前窗口 |

### 详情页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 | （与新增页面字段匹配一致） |
| 操作 |  |
| 点击“已更改次数” | 弹出窗口，链接到远期订舱改期申请 >>“改期申请历史”  传参 ORDER\_NO<远期预售订舱单> |
| 关闭 | 关闭当前窗口 |

### 数据测试用例

|  |  |  |
| --- | --- | --- |
| 功能名称 | 测试过程 | 期望结算 |
| 新增改期申请 | 新增-》选择  CNSHK→SGSIN  中转港：CSC  出运日期：  W04(2014/01/19~01/25)  的远期预售订舱单  （一个20GP的箱，100USD）  点击改期  改期至  W05(2014/01/26~02/01)  改期原因：交货延误  (远期运价的SEQ=10000)  (20GP 120USD)  保存 | \*FUTURE\_PRESALE\_CHANGE 新增一条数据  FUTURE\_PRESALE\_CHANGE\_SEQ=数据库生成  FUTURE\_PRESALE\_HEAD\_SEQ=选中改期的远期预售订舱主表SEQ  FUTURE\_PRICE\_HEAD\_SEQ=改期的远期预售订舱使用的远期运价主表SEQ  CHANGE\_ORDER\_NO= OWNER\_CODE+FCA+YYMMDD+FUTURE\_PRESALE\_CHANGE\_ORDER\_NO(5位码)  例如：CSCLFCA14032610000  WEEK\_NO<原周数>=W04  WEEK\_BEGIN\_DATE=2014/01/19  WEEK\_END\_DATE=2014/01/25  NEW\_FUTURE\_PRICE\_HEAD\_SEQ=10000  NEW\_WEEK\_NO=W05  NEW\_WEEK\_BEGIN\_DATE=2014/01/26  NEW\_WEEK\_END\_DATE=2014/02/01  FINAL\_FUTURE\_PRICE\_HEAD\_SEQ=0  CHANGE\_REASON=交货延误  CHANGE\_TYPE= D：延后  IS\_APPLYNEWPRICE=0  STATUS=0  SUBMITUSER= NULL  SUBMIT\_DATE= NULL  APPROVER<审批人> = NULL  APPROVED\_DATE<审批时间>= NULL  APPROVED\_COMMENT<审批意见> = NULL  时间戳  \* FUTURE\_PRESALE\_CHANGEDETAIL新增一条记录  FUTURE\_PRESALE\_CHANGEDETAIL\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ=改期主表SEQ  FUTURE\_PRESALE\_HEAD\_SEQ=选中改期的远期预售订舱主表SEQ  FUTURE\_PRESALE\_DETAIL\_SEQ=选中改期的远期预售订舱箱表SEQ（20GP）  PRICE\_TOTAL<改期前成交价>=100  NEW\_PRICE\_TOTAL<改期后成交价>=120  FINAL\_PRICE\_TOTAL<最终成交价>=0  FP\_CONTADETAIL\_SEQ=对应改期前远期运价箱表20GP的SEQ  NEW\_FP\_CONTADETAIL\_SEQ=改期后远期运价箱表20GP的SEQ  FINAL\_FP\_CONTADETAIL\_SEQ=0  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=D:保存为草稿  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS=10  时间戳 |
| 提交改期申请 | 列表页面》查询》  选中上述草稿记录》提交 | \*更新FUTURE\_PRESALE\_CHANGE  STATUS=10  SUBMITUSER= 当前用户  SUBMIT\_DATE= 当前时间  APPROVER<审批人> = NULL  APPROVED\_DATE<审批时间>= NULL  APPROVED\_COMMENT<审批意见> = NULL  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=S:提交待审批  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS=10  时间戳 |
| 删除改期申请 | 列表页面》查询》  选中记录》删除 | \*更新FUTURE\_PRESALE\_CHANGE  STATUS=-20  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=I：删除  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS=10  时间戳 |
| 回收（已提交的申请） | 列表页面》查询》  选中提交状态的记录》回收 | \*更新FUTURE\_PRESALE\_CHANGE  STATUS=0  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=C:撤回提交的申请  OPRT\_REMARK=NULL  OPRT\_ROLE=S  STATUS=10  时间戳 |

## 远期订舱改期审批

* 针对订舱人的远期订舱，进行审批、拒绝操作
* 管理页面
* 审批界面
* 改期历史页面
* 详情页面（与订舱人的详细页面有区别）

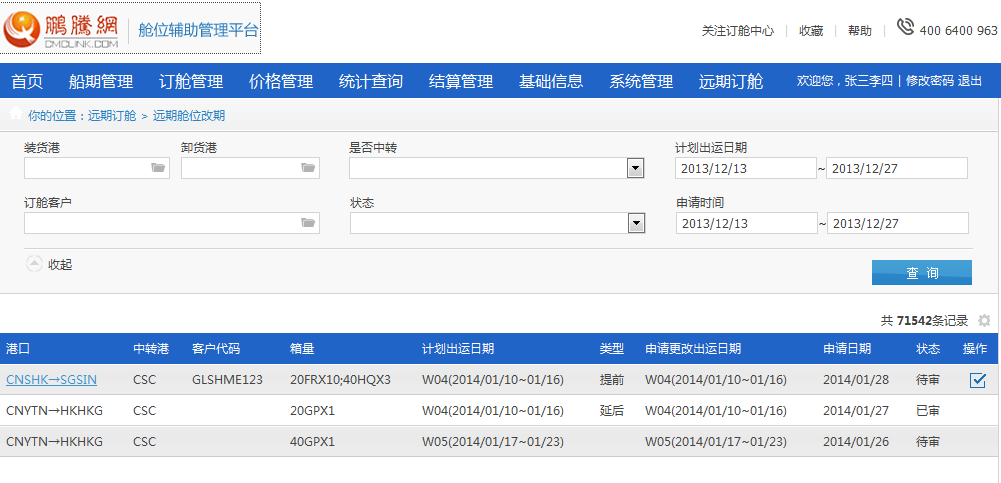
### 功能描述

提供给船公司，对改期申请的审批操作。

### 菜单

船公司 >> 远期订舱 >> 远期订舱改期

### 列表界面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 查询条件 |  |
| 装货港 | 使用港口选择器选择，数据来源于公共港口代码  调用 2.0A版带区域的港口选择器  支持单选 |
| 卸货港 | 支持选择器，调用 2.0A版带区域的港口选择器  支持单选 |
| 是否中转 | 下拉框，加载：  NULL（默认）  直航  中转 |
| 计划出运日期 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间  默认结束时间 = 当前时间+2周  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 订舱客户 | 使用订舱客户选择器，调用放舱管理中的“订舱客户” |
| 状态 | 下拉框，字典表加入，NULL（默认）  0草稿,10待审, 90已拒绝,100已审核,-20已删除,已提交(10,90,100)  PS：已提交，是个虚拟状态，表示包括10待审,90已拒绝,100已审核  PS：选择NULL时，查询需要把非ALL的传递进去 |
| 申请时间 | 日期控件，格式YYYY-MM-DD  默认开始时间 = 当前时间 – 2周  默认结束时间 = 当前时间  允许为NULL，当用户设置为NULL时，开始时间取DateTime.MixValue，结束时间取DateTime.MaxValue，传递给查询接口 |
| 操作列 | * 审批（10待审） |
| 列表项 |  |
| 港口 | LOADING\_PORT - > DESTINATION\_PORT |
| 中转港 | TRANSFER\_PORT |
| 客户代码 | CUSTOMER\_NAME\_PY |
| 箱量 | CONTA\_SIZE\_SUM |
| 计划出运日期 | WEEK\_NO（WEEK\_BEGIN~WEEK\_END） |
| 类型 | CHANGE\_TYPE<改期类型>，根据字典转换显示 |
| 申请更改出运日期 | NEW\_WEEK\_NO<改期后周数>( NEW\_WEEK\_BEGIN\_DATE~  NEW\_WEEK\_END\_DATE) |
| 申请日期 | SUBMIT\_DATE |
| 状态 | STATUS，根据字典转换 |
| 操作 |  |
| 查询 | 调用接口IOMTFutureChange.QueryForListByPage  参数赋值parm（匹配页面的查询条件）  ownerSeq<船公司> = 当前用户所属船公司SEQ  orderCompSEQ = 订舱客户  isTran<是否中转> = 0 忽略  orderBy = P.WEEK\_BEGIN\_DATE ASC, P.CREDIT\_LEVEL DESC, C.SUBMIT\_DATE ASC，P. FUTURE\_PRESALE\_CHANGE\_SEQ DESC  按出运日期顺序、信用级别倒序、申请日期顺序排列 |
| 新增 | 弹出窗口，链接到“查询远期舱位页面” |
| 提交 | 调用接口IOMTFutureChange.Submit（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 删除 | 调用接口IOMTFutureChange.Delete（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 撤销 | 调用接口IOMTFutureChange.RecallSubmit（不带事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ |
| 点击“港口” | 弹出窗口，链接到“详情页面” |
| 批量提交 | 新开事务；循环调用  {  调用接口IOMTFutureChange.Submit（事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ（选中）  }  事务提交； |
| 批量删除 | 新开事务；循环调用  {  调用接口IOMTFutureChange.Delete（事务）  参数changeSEQ = FUTURE\_PRESALE\_CHANGE\_SEQ（选中）  }  事务提交； |

### 审批页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 | \*为必填（无特别说明，以标签显示） |
| 远期预售订舱信息，对应对象FutureChangeQueryEntity.preSaleModel.FuturePresaleData | |
| 船公司 | OWNER\_CODE |
| 发货人 | COMP\_SEND<托运人> |
| 收货人 | COMP\_GET<收货人> |
| 通知人 | COMP\_TRANS<转运人> |
| 订舱联系人地址 | COMP\_SEND\_ADDR<订舱公司地址> |
| 联系人 | LINKER<联系人姓名> |
| 电话 | LINKER\_PHONE<电话> |
| 传真 | LINKER\_FAX<传真> |
| 邮箱 | LINKER\_EMAIL<电子邮件> |
| 装货港 | LOADING\_PORT<装货港> |
| 卸货港 | DESTINATION\_PORT<卸货港> |
| 中转港 | TRANSFER\_PORT<中转港> |
| 运输条款 | 对应字段，TRANS\_TYPE\_NAME<运输条款中文名称>  英文环境，取TRANS\_TYPE\_NAME\_EN<运输条款英文名称> |
| 运费条款 | 对应字段，RULE\_PRICE\_NAME<运费条款中文名称>  英文环境，取RULE\_PRICE\_NAME\_EN<运费条款英文名称> |
| 付款地 | PAY\_LOCATION<付款地点代码>，请根据DICT\_CODE字段进行转换  DICT\_TYPE = PAY\_LOCATION  NULL的，不转换 |
| 计划出运日期 | WEEK\_NO<出运周数>( WEEK\_BEGIN~WEEK\_END) |
| 交割期限(提前/延后) | CROSS\_BEFOREDAY<允许交割提前天数>  CROSS\_DELAYDAY<允许交割延后天数> |
| 货物描述 | CARGO\_DESC<货物描述> |
| 订舱备注 | ORDER\_REMARK<订舱备注>  隐藏保留ORDER\_NO<申请编号> |
| 改期信息(公司名) | CUSTOMER\_NAME<订舱公司名称>  英文环境，CUSTOMER\_NAME\_EN<订舱公司英文名称>  信用级别  CREDIT\_LEVEL<信用等级> |
| 原计划出运日期 | SALE\_SPACE<可售舱位> |
| 申请更改出运日期 | NEW\_WEEK\_NO<改期后周数>( NEW\_WEEK\_BEGIN\_DATE~  NEW\_WEEK\_END\_DATE) |
| 原舱位(总数/已售) | SALE\_SPACE<可售舱位>/ SALE\_F\_SPACE<已售舱位> |
| 新舱位(总数/已售) | FutureChangeQueryEntity.newTotalTeu/ FutureChangeQueryEntity.newSalesTeu |
| 本次订舱数 | FutureChangeQueryEntity.currentOrderTeu |
| 适用运价\* | 单选按钮  原运价  新运价(默认选中) |
| 改期信息FutureChangeQueryEntity.changeModel.FutureChangeHeadSingle | |
| 申请改期原因 | CHANGE\_REASON<改期原因> |
| 审批意见 | 多文本框录入，限制200个汉字以内 |
| 箱型列表（无特别说明，以标签显示）循环FutureChangeQueryEntity.preSaleModel  .FuturePresaleDetail(预售订舱单箱表信息) | |
| 箱型 | CONTA\_SIZE<标准箱型> |
| 数量 | UNIT<箱数量> |
| 原运期运价 | FINAL\_PRICE\_TOTAL |
| 改期后运价 | 查询FutureChangeEntity.FutureChangeDetailMuti，  FUTURE\_PRESALE\_DETAIL\_SEQ = 当前  FuturePresaleDetail.FUTURE\_PRESALE\_DETAIL\_SEQ，  获得记录的“NEW\_PRICE\_TOTAL”输出  （与新增编辑页面逻辑一致，请封私有方法） |
| 操作 |  |
| 页面加载 | 根据FUTURE\_PRESALE\_CHANGE\_SEQ，获取远期改期信息，  调用接口：IOMTFutureChange.QueryChangeByHeadSeq  参数 headSeq = FUTURE\_PRESALE\_CHANGE\_SEQ  Data FutureChangeEntity – output（包括改期信息和远期预售订舱信息） |
| 点击“改期历史” | 弹出窗口，链接到远期订舱改期申请 >>“改期申请历史”  传参 ORDER\_NO<远期预售订舱单> |
| 审批通过 | * 调用接口IOMTFutureChange.Approve   参数赋值  ChangeSEQ = 当前记录SEQ  AppOprtEntity  - OprtUser：当前用户  - OprtAction: 枚举类EnumActionType.A(审批操作)  - OprtRemark：审批意见  - OprtRole : 枚举类EnumAppActionRole.A（审批人） |
| 拒绝 | * 审批意见必填，否则弹出提示信息 BKM100280|审批意见 * 调用接口IOMTFutureChange.Reject   参数赋值  ChangeSEQ = 当前记录SEQ  AppOprtEntity  - OprtUser：当前用户  - OprtAction: 枚举类EnumActionType.R(拒绝操作)  - OprtRemark：审批意见  - OprtRole : 枚举类EnumAppActionRole.A（审批人） |
| 关闭 | 关闭当前窗口 |

### 改期申请历史



（不需要分页显示）

|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 | （无特别说明，以标签显示） |
| 次数 | 根据记录自动生成 |
| 申请时间 | SUBMIT\_DATE |
| 原计划出运日期 | WEEK\_NO（WEEK\_BEGIN\_DATE~WEEK\_END\_DATE） |
| 申请更改出运日期 | NEW\_WEEK\_NO（NEW\_WEEK\_BEGIN\_DATE~NEW\_WEEK\_END\_DATE） |
| 改期原因 | CHANGE\_REASON |
| 审批人 | APPROVER |
| 审批时间 | APPROVED\_DATE |
| 审批意见 | APPROVED\_COMMENT<审批意见> |
| 操作 |  |
| 页面初始化 | 调用接口IOMTFutureChange.QueryChangeHis  参数OrderNo = 页面传递参数 |
| 关闭 | 关闭当前窗口 |

### 详情页面



|  |  |
| --- | --- |
| 页面元素 | 元素说明 |
| 页面显示 | （与审批页面字段匹配一致） |
| 操作 |  |
| 页面初始化 | 调用接口IOMTFutureChange.QueryChangeByHeadSeq  参数HeadSeq = 改期主表SEQ |
| 关闭 | 关闭当前窗口 |

### 数据测试用例

|  |  |  |
| --- | --- | --- |
| 功能名称 | 测试过程 | 期望结算 |
| 审批通过 | 列表页面》查询》  选中记录》  CNSHK→SGSIN  中转港：CSC  出运日期：  W04(2014/01/19~01/25)  的远期预售订舱单  （一个20GP的箱，100USD）  点击改期  改期至  W05(2014/01/26~02/01)  改期原因：交货延误  (远期运价的SEQ=10000)  (20GP 120USD)  应用新价钱  审批通过 | \*更新FUTURE\_PRESALE\_CHANGE  FINAL\_FUTURE\_PRICE\_HEAD\_SEQ=10000  IS\_APPLYNEWPRICE=1  STATUS=100  APPROVER<审批人> = 当前用户  APPROVED\_DATE<审批时间>=当前时间  APPROVED\_COMMENT<审批意见> = 页面录入  时间戳  \*更新 FUTURE\_PRESALE\_CHANGEDETAIL  FINAL\_PRICE\_TOTAL<最终成交价>=120  FINAL\_FP\_CONTADETAIL\_SEQ=NEW\_FP\_CONTADETAIL\_SEQ  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=A:审批通过  OPRT\_REMARK= APPROVED\_COMMENT  OPRT\_ROLE=A  STATUS=10  时间戳  \*更新 FUTURE\_PRESALE\_HEAD<远期预售主表>  (改期的预售订舱单)  STATUS = -40  时间戳  \*新增记录 “FUTURE\_PRESALE\_HEAD<远期预售主表>”  FUTURE\_PRESALE\_HEAD\_SEQ=数据库生成  PUBLISH\_PRICE\_SUM<发布价格概要>=新远期运价的发布价格概要  WEEK\_NO<出运周数>=W05  WEEK\_BEGIN=2014/01/26  WEEK\_END=2014/02/01  FUTURE\_PRICE\_HEAD\_SEQ=10000  FUTURE\_PRICE\_NO<运价编号>=新远期运价主表FUTURE\_PRICE\_NO<运价编号>  LAST\_CHANGE\_SEQ<最近改期的SEQ>=当前FUTURE\_PRESALE\_CHANGE的SEQ  CHANGE\_COUNT<改期次数>=CHANGE\_COUNT<改期次数>+1  STATUS=100  时间戳  （其他字段与改期前的远期预售订舱单一致）  （PS：如果应用旧价格，PUBLISH\_PRICE\_SUM 、FUTURE\_PRICE\_HEAD\_SEQ、FUTURE\_PRICE\_NO不变）  \*新增记录到“FUTURE\_PRESALE\_DETAIL”  FUTURE\_PRESALE\_DETAIL\_SEQ=数据库自动生成  FUTURE\_PRESALE\_HEAD\_SEQ= FUTURE\_PRESALE\_HEAD\_SEQ  FUTURE\_PRICE\_CONTADETAIL\_SEQ=20GP 120USD的远期运价箱表SEQ  IS\_APPLYNEWPRICE<是否应用新价格>=1  FINAL\_PRICE\_TOTAL<最终成交价>=120  时间戳  （其他字段与改期前的远期预售订舱单一致） |
| 拒绝改期申请 |  | \*更新FUTURE\_PRESALE\_CHANGE  STATUS=90  APPROVER<审批人> = 当前用户  APPROVED\_DATE<审批时间>=当前时间  APPROVED\_COMMENT<审批意见> = 页面录入  时间戳  \*更新FUTURE\_PRESALE\_HEAD  CHANGE\_COUNT = CHANGE\_COUNT+1  时间戳  \* FUTURE\_PRESALE\_CHANGELOG<改期申请操作日志>新增一条记录  FUTURE\_PRESALE\_CHANGELOG\_SEQ=数据库生成  FUTURE\_PRESALE\_CHANGE\_SEQ= FUTURE\_PRESALE\_CHANGE表SEQ  OPRT\_USER=当前用户  OPRT\_DATE=当前时间  OPRT\_ACTION=R:拒绝  OPRT\_REMARK= APPROVED\_COMMENT  OPRT\_ROLE=A  STATUS=10  时间戳 |

# 运维站点详细设计

## 菜单定义

|  |  |  |
| --- | --- | --- |
| **运维功能** |  |  |
|  | 船期管理 |  |
|  |  | 船舶管理 |
|  |  | 航线管理 |
|  |  | 船期管理 |
|  |  | 舱位管理 |
|  | 设置管理 |  |
|  |  | 基础资料维护 |
|  |  | 账户信息管理 |
|  |  | 公司组织架构维护 |
|  |  | 公司权限设置 |
|  |  | 箱型管理 |
|  |  | 提示消息与字符集维护 |
|  |  | 日志查询 |

## 基础资料维护（运维）

### 字典维护

#### 需求说明

船舶、航线等基础数据存在要下拉选择类型的数据字段，如船舶有：“船型”“船舶类别” “船船种类”；航线有：“航线周期”；港口管理：“港口类型”等等一些字段，考虑到这些类型字段的数据量不多特集合到一张表管理。DEMO中的系统管理包括了大量字典类型的数据管理，这里总结了一期使用到的数据表，未使用的数据表待以后与需求确认再建立。对于已使用的功能都已定义出表这期需要实现管理。

|  |  |  |
| --- | --- | --- |
| 功能名称 | 表名称（DICT\_TYPE） | 是否使用 |
| 免费用箱期 | - | 未使用 |
| 结算费率 | - | 未使用 |
| 币种管理 | DICT\_CODE(MONEY\_TYPE) | **已使用** |
| 费用种类管理 | DICT\_CODE(COST\_TYPE) | **已使用** |
| 运费条款 | DICT\_CODE(TRAN\_COST\_CLAUSE) | **已使用** |
| 运输条款管理 | DICT\_CODE(TRAN\_CLAUSE) | **已使用** |
| 航线/包装提示 | - | 未使用 |
| 新增航线区域 | - | 未使用 |
| 港口管理 | DICT\_PORT\_CODE | **已使用** |
| 国家/区域代码 | DICT\_NATION\_AREA | **已使用** |
| 集装箱动态类型 | - | 未使用 |
| 箱号箱型 | - | 未使用 |
| 卸货港与交货地  装货港与收货地 | DICT\_PORT\_CODE | **已使用** |
| SCAC对照管理 | - | 未使用 |
| 集装箱管理 | - | 未使用 |
| 危品分类管理 | - | 未使用 |
| 货物类型管理 | DICT\_CODE(GOODS\_TYPE) | **已使用** |
| 包装类型管理 | DICT\_CODE(WRAP\_TYPE) | **已使用** |
| 装运条款管理 | - | 未使用 |
| 进口默认委托 | - | 未使用 |
| 美线S/C之S/C | - | 未使用 |
| 美线S/C之VIP | - | 未使用 |

#### 数据表



1. “OWNER\_SEQ”默认取值“0”，取值为0的编码表示所有船公司通用，而船公司特定的编码标记船公司自己的SEQ，不支持对不同的操作公司不同的编码体系，对一家船公司仅能存在一套编码体系。
2. “STATUS”取值为“-10 删除 10有效”，采用标记删除方式处理。
3. “COVER\_CODES”记录覆盖该条数据被哪家船公司覆盖的作用，数据格式：“,1,2,3,”。
4. “REPLACE\_CODE”参见《BKOnline系统详细设计文档LEVEL2.doc》编码表体系解决方案。
5. 存入表的数据如果是“币种”那么扩展字段1用于存:[币种符号]，扩展字段2用于存 [币种：是本位币（10:是; -10:不是）]。其他扩展字段用于备用。
6. “DICT\_TYPE”字典类型按如下编码对应不同类型字段。

|  |  |  |
| --- | --- | --- |
| 字典类型代码(DICT\_TYPE) | 代表字段 | 内容 |
| SHIP\_TYPE | 船型 | 已定，见表初始化数据 |
| SHIP\_CLASS | 船舶种类 | 已定，见表初始化数据 |
| SHIP\_KIND | 船舶类别 | 已定，见表初始化数据 |
| PORT\_TYPE | 港口类型 | 已定，见表初始化数据 |
| SHIP\_WAY | 航向 | 已定，见表初始化数据 |
| WEIGHT\_UNIT | 重量单位 | 已定，见表初始化数据 |
| TRAN\_COST\_CLAUSE | 运费条款 | 已定，见表初始化数据。可界面维护 |
| TRAN\_CLAUSE | 运输条款 | 已定，见表初始化数据。可界面维护 |
| MONEY\_TYPE | 币种 | 已定，见表初始化数据。可界面维护 |
| COST\_TYPE | 费用种类 | 已定，见表初始化数据。可界面维护 |
| GOODS\_TYPE | 货物类型 | 已定，见表初始化数据。可界面维护 |
| WRAP\_TYPE | 包装类型 | 已定，见表初始化数据。可界面维护 |
| LINE\_BUSITYPE | 航线运营类型 | 已定，见表初始化数据。可界面维护 |
| DISCHARGE\_TRAN\_WAY | 二程运输方式 | 已定，见表初始化数据。可界面维护 |

DICT\_TYPE”字典类型加入两个订舱时使用的类型字段。

|  |  |  |
| --- | --- | --- |
| 字典类型代码(DICT\_TYPE) | 代表字段 | 内容 |
| ORDER\_CHECK\_WAY | 托运方式 | CODE NAME  **27 Door to Door**  **28 Door to Pier**  **29 Pier to Door**  **30 Pier to Pier** |
| ORDER\_TRAN\_WAY | 运输方式 | CODE NAME  **2 LCL/LCL**  **3 FCL/FCL**  **4 FCL/LCL**  **5 LCL/FCL** |

|  |  |  |  |
| --- | --- | --- | --- |
| DICT\_TYPE | DICT\_CODE | DICT\_NAME\_CN | DICT\_CODE\_DESC |
| USER\_CLASS | ROUTE | 航线 | 用户分类 |
| USER\_CLASS | SALES | 销售 | 用户分类 |
| USER\_CLASS | SERVICE | 客服 | 用户分类 |
| USER\_CLASS | ADMIN | 管理员 | 用户分类 |

查询语句样例：

|  |
| --- |
| select \* from dict\_code  where (owner\_seq = 0 or owner\_seq = 船公司OWNER\_SEQ)  and status= 10  and dict\_type='数据类型'  and cover\_codes not like '%,船公司OWNER\_SEQ,%' order by create\_date |

#### 数据字典的使用方式

* 数据字典专门用于维护 “代码 + 名称”格式的数据，数据只需要维护代码名称的可将数据放置到此表中，如果字段不满足要求的字典需要单独设计专用的字典表。
* 字典表中支持中英文和繁体，在使用时通过字典“字典数据接口”处理语种问题，不同的语种环境下返回不同的字段组合。默认返回“中文”，即在语种环境不清的情况下，使用中文
* 字典支持通用定义和船公司特定数据两种方式。即系统上线时初始化的数据为“OWNER\_SEQ = 0”，船公司用户可以在此基础上增加或者修改编码和内容，修改之后的数据将另存为“OWNER\_SEQ = 船公司SEQ”的数据。
* 数据字典使用字段“DICT\_TYPE”标记是何种用途的字典，具体定义参考数据库部分的对照表。

#### 相关功能

公共字典功能用于支持菜单中的以下页面：

* 费用种类维护
* 运费条款
* 运输条款管理
* 包装类型管理
* 船型
* 船舶种类
* 船舶类别
* 港口类型
* 航向
* 重量单位
* 币种
* 货物类型

### 日志查询

#### 需求说明

为更好的监控系统，需要经常对业务类型的日志查询。常用的可能按船公司查一段时间内的一票订单跟踪日志，查系统代码出现的程序异常日志，按操作类型区分常规日志等等。

#### 数据表

参见[异常和日志接口;](#_异常和日志接口)

#### 列表页面

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询条件：<船公司><操作公司><订舱公司><操作人><订单号><操作表名><信息类型><消息时间> [查询](#_日志查询) | | | | | | | | | | | |
| **流水号** | **操作类型** | **操作表名** | **操作记录号** | **信息类型** | **订单号码** | **信息描述** | **船公司** | **操作公司** | **订舱公司** | **操作人** | **消息时间** |

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 船公司 | 下拉加载船公司列表。操作公司下拉，需与船公司联动。 |
| 订舱公司、操作人、订单号 | 支持模糊查询，前后去空格，内容不区分大小写。 |
| 操作表名 | 下拉加载出所有数据库表名称。 |
| 消息类型 | 下拉。内容限定为：1-DEBUG 2-INFO 3-WARN 4-ERROR。 |
| 数据状态 | 下拉。内容限定为：0-有效数据 -10-无效数据 -20-删除数据。 |
| 消息时间 | 分开始结束时间，精确到时分。【年月日】用时间控件选择，不可输入。【时分】文本框显示可输入小时限制“00-23”，分钟限制“00-59”的数字，否则提示格式错误。例如默认格式，正确格式，错误格式。 |
| 交互 |  |
| 数据加载 | 列表默认不查询数据，用户点击“查询”之后再加载数据。  字段的对应关系请参考数据库定义的同名字段。  按消息时间倒序排列，即最新日志排在最前。 |

## 币种转化汇率

### 功能说明

由于系统需要支持币种，需要使用汇率进行货币转换。汇率由船公司维护，每期汇率变动时都需要维护到系统中。

### 数据库变更



### 列表页面

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 查询条件：<船公司> <外币> <有效时间> 查询 新增 | | | | | |
| 原币种 | 转换币种 | 汇率 | 有效时间起 | 有效时间止 | 操作 |
| USD | RMB | 1.2568999 | 2011-04-01 | 2011-05-01 | 编辑 删除 |

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 船公司 | 登录用户是船公司的，默认，锁定，不允许编辑。登录是运维的，可下拉。 |
| 外币 | 录入，精确，拼接查询条件语法时，自动转换成大写。 |
| 有效时间 | 时间范围控件，默认起始时间为当天-7，默认结束时间当天。最大查询时间范围限制12个月。 |
| 交互 |  |
| 数据加载 | 列表默认不查询数据，用户点击“查询”之后再加载数据。  字段的对应关系请参考数据库定义的同名字段。  按消息时间倒序排列，即最新日志排在最前。 |

### 编辑界面

|  |  |
| --- | --- |
| 本位币 | USD |
| 转换货币 |  |
| 汇率 |  |
| 生效时间 |  |
| 失效时间 |  |

保存 关闭

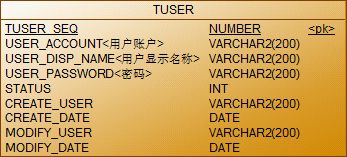
|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 交互 |  |
| 界面加载 | 1. 本位币从全局参数获取，不允许编辑。KEY = 101 2. 转换货币加载货币下拉列表 |
| 保存 | 保存之前检查，汇率，装货货币，生效时间，失效时间不能为空。  生效时间不能大于失效时间  唯一性：任何时间范围之内，本位币与指定的转换货币之间，只能存在一个转换汇率 |

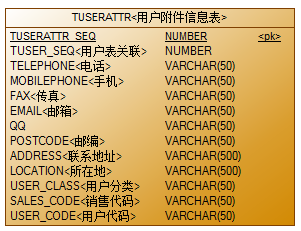
## 账户信息管理（运维）

### 需求说明

用户账户信息采用卡片式管理，即新建用户信息不关联任何公司，视为独立的信息。需要区分公司级管理员与运维级管理员查看数据的权限，公司级管理员能查看和更改自己公司用户信息，运维级管理员能查看和更改所有公司用户信息。

### 数据表





1.TUSERATTR<用户附件信息表>是TUSER<用户表>的扩展表。

2.STATUS：10正常 -10停用 -20删除 。

追加用户分类标记维护，数据来源自字典表，初始化为“航线，销售，客服”三类，必填；管理员默认显示管理员，不能修改

如果是航线销售的，销售员代码必填

用户代码即"EDI代码"，发送报文会要用到

### 列表页面

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询条件：<账户> <显示名称> [查 询](#_列表页面_7)  [新 增](#_列表页面_7) | | | | | | | | |
| 序号 | 账 户 | 显示名称 | 电 话 | 手 机 | QQ | EMAIL | 状 态 | 操 作 |
|  |  |  |  |  |  |  |  | 查看 更多↓  删除  冻结  解冻  用户权限 |

**查询条件：**

1. 账户、显示名称支持模糊查询，前后去空格，内容不区分大小写。

2.普通公司用户不给出“用户权限”连接，公司级管理员和运维级管理员给出链接，可以分配普通用户权限，进入[用户菜单分配页面](#_用户菜单分配页面)。

**加载过程：**

1.假设“鹏海运”公司为**运维级管理员**所属公司（SEQ:1）：

|  |
| --- |
| select \* from tuser a  left join tuserattr b on a.tuser\_seq = b.tuser\_seq  left join tusercomp c on a.tuser\_seq = c.tuser\_seq |

~~假设“中海”公司为~~**~~公司级管理员~~**~~所属公司（SEQ:2）：~~

|  |
| --- |
| ~~select \* from tuser a~~  ~~left join tuserattr b on a.tuser\_seq = b.tuser\_seq~~  ~~left join tusercomp c on a.tuser\_seq = c.tuser\_seq~~  ~~where c.tcomp\_seq = 2~~ |

2列表默认不查询数据，用户点击“查询”之后再加载数据。

3.字段的对应关系请参考数据库定义的同名字段。

4.按消息时间倒序排列，即最新日志排在最前。

**界面交互：**

1. 用户点击“新增”“编辑”时，弹出编辑界面。
2. 用户点击“删除”时，提示“是否确认删除（账户名称）？”，确认的话，对记录做标记删除。即将STATUS修改为~~“-10无效数据”~~“-20 删除”，并修改记录的时间戳。

调用日志接口，记录操作日志：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标题 | LEVEL | 内容 | 接口 | 备注 |
| 删除用户 | INFO | 用户{用户账户}删除用户SEQ{TUSER\_SEQ},用户账户{user\_account},用户显示名称{USER\_DISP\_NAME}。 | 常规异常 | 无 |

1. 用户点击“查询”时，根据查询条件加载数据。
2. 用户点击“查看”弹出用户详细信息页面，展示字段和布局方式参见编辑界面，字段的内容直接用文字展示，不用文本框。

### 编辑页面

|  |  |  |  |
| --- | --- | --- | --- |
| 归属公司\* | 下拉 | | |
| 用户账户\* |  | 用户显示名称\* |  |
| 密 码\* |  | 用户分类\* |  |
| 用户代码 |  | 销售员代码 |  |
| 电话 |  | 手机 |  |
| 邮箱\* |  | 邮编 |  |
| QQ |  |  |  |
| 联系地址 |  | | |

[冻 结](#_编辑页面_5) [解 冻](#_编辑页面_5) [保 存](#_编辑页面_5) [取 消](#_编辑页面_5)

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
|  | 带“\*”为必填字段 |
| 运维级管理员 | 归属公司，下拉选择，必填字段。 |
| 公司级管理员 | 归属公司，默认当前公司，不可编辑。 |
| 用户分类 | 新增页面：  用户下拉选择，默认为空，必填。  数据来源DICT\_CODE<字典代码表>.USER\_CLASS。  修改页面：  如果用户数据是管理员（TUSERCOMP<账户公司关联>.IS\_ADMIN == 1）  默认显示“管理员”不能修改，不再显示下拉。 |
| 销售员代码 | 如果用户分类是销售，销售员代码必填  BKM100280 {0}不能为空 |
| 用户代码 | 用户代码非必填 |
| 交互 |  |
| 保 存 | 开启事物  校验用户账户、用户显示名称、~~密码、（删除）~~邮箱必填字段。  用户账户校验规则参考 [IUserOprt:AccountFormatCheck账户的规则使用通用规则](#_IUserOprt：AccountFormatCheck)。  ~~密码校验规则参考~~ [~~IUserOprt:PasswordFormatCheck密码的校验规则。~~](#_IUserOprt：PasswordFormatCheck)  页面需要用JS做判断，提交后台还需再判断一次。  检查“用户账户”在TUSER<用户表>是否重复，如重复提示“用户账户”已存在。  保存之前，如果录入了邮箱，检查“邮箱”在TUSERATTR<用户附件信息表>是否重复，如重复提示“邮箱”已存在。  调用IUserOprt：InitPassWord 生成新的初始化密码，作为参数（parm.userPsw）调用IUserOprt：PasswordFormatCheck校验密码强度，如生成的密码强度为中或者低，重新生成，直至生成强度高的密码  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 新增用户 | INFO | 用户{用户账户}新增用户SEQ{TUSER\_SEQ},用户账户{USER\_NAME},用户显示名称{USER\_DISP\_NAME}。 | 常规异常 | 无 |   将录入的字段数据正确的对应到TUSER<用户表>写入表，和TUSERATTR<用户附件信息表>的操作需要同一事物。如果操作用户是公司级管理员那么新增用户默认为该公司的用户（还需往TUSERCOMP<账户公司关联表>新增关联数据，同一事物），如果是运维级管理员新增的用户则不与任何公司关联。  调用IUserOprt：InitResetUrl 获取 resetURL  如果录入了邮箱，调用OMTMail：Send发送邮件信息（仅新增账户时发送邮件）  **MailProperty：**  Sendmail：发件人地址 = “webmaster@pub.szedi.com.cn”（可配置）  revMail：收件人地址 = TUSERATTR.EMAIL  ssMail：抄送地址 = 空  ssmMail：密送地址 = 空  title：邮件标题 = “【网上订舱在线交易平台】初始化密码邮件，请勿回复”  text：邮件正文，仅支持文本格式，不支持HTML格式，将 “resetURL” 替换 {URL}   |  | | --- | | 海运舱位交易中心用户，您好！  感谢您对系统的支持，请使用以下链接进行初始化密码：  [{URL}](http://.....//...../.....?ret=%7b%7d)  此链接只能使用一次。  如果点击以上链接无法使用，请您尝试以下方法：  第一步：将上述链接拷贝到您计算机浏览器(例如IE)的地址栏中；  第二步：按回车（Enter）键。  如果仍无法使用，请您尝试通过以下方法解决：  （1）检查您电子邮件信箱中的垃圾箱或广告箱，邮件有可能被误认为垃圾或广告邮件。  （2）联系我司客服人员，联系电话：4006400963。  请勿直接回复本邮件。 |   attrBlob：空  记录日志   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 用户密码初始化信息 | INFO | 向用户{TUSERModel.USER\_NAME}的{TUSERATTR.EMAIL}发送了初始化邮件。 | 常规异常 |  |   事物结束。  事物采用批量策略，全部成功，全部失败。 |
| 编辑 | 进入编辑页面前查询 TUSER\_SEQ 是否存在于TUSERCOMP<账户公司关联>，如存在则不允许更改“用户账户”，否则允许更改。  其他同保存逻辑，但不需记录操作日志。 |
| 冻结 | 用户点击“停用”提示“您确认冻结（用户账户名称）？”，确认的话，更改STATUS为“-10”。  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 冻结用户 | INFO | 用户{用户账户}冻结用户SEQ{TUSER\_SEQ},用户账户{user\_account},用户显示名称{USER\_DISP\_NAME}。 | 常规异常 | 无 | |
| 解冻 | 用户点击“恢复”提示“您确认解冻（用户账户名称）？”，确认的话，更改STATUS为“10”。  要把登录用户userAccount在LOGIN\_ERROR<登录错误信息表>的所有记录状态置为“-10无效”  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 解冻用户 | INFO | 用户{用户账户}解冻用户SEQ{TUSER\_SEQ},用户账户{user\_account},用户显示名称{USER\_DISP\_NAME}。 | 常规异常 | 无 | |

## 公司组织架构维护

### 需求说明

权限设置中需要根据公司架构设置权限，运维功能中需要有维护页面架构公司。

### 组织机构设置

|  |  |
| --- | --- |
| 查询条件：<船公司> 查询 | |
| 公司架构 | 操作 |
| 船公司 | 增加下级公司 |
| 操作公司 | 增加下级公司 删除节点 |
| 子公司A | 增加下级公司 删除节点 |
| 操作公司 | 增加下级公司 删除节点 |

#### 界面元素

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
| 船公司 | 查询条件，下拉。数据来源自TCOMP\_GROUP<公司组织>中“GROUP\_COM\_SEQ”为0的数据。默认显示空。 |
| 列表数据 | 默认不加载数据，点击查询之后加载指定船公司的数据，数据需要根据层级关系显示为树状结构。  调用接口OMTCompany：QueryCompGroupByOwner获取到需要显示的数据。  ownerSeq = 选择的船公司SEQ |
| 公司架构栏位 | 船公司信息显示公司中文名称，对应字段“COMP\_NAMECN\_SHORT<公司中文简称>” |
| 操作按钮栏位 | 根节点“船公司”不支持删除，仅支持增加下级公司 |
| 查询 | 用户点击查询时检查查询条件是否为空，为空的提示“请选择船公司。”。由于此页面时运维使用的页面，不需要支持双语，不使用SYSTEM\_UI\_LANGUAGE中的数据，直接代码写死即可  调用接口IOMTCompany：QueryCompGroupByOwner  ownerSeq = 查询条件中的船公司SEQ |
| 增加下级公司 | 用户点击之后弹出公司选择器，由用户选择需要增加为下级的公司。  调用接口IOMTCompany：AddCompGroupAsChild  currentCompSeq = 当前选择的公司SEQ  parentCompSeq = 点击按钮时对应的公司SEQ，父节点的SEQ |
| 删除节点 | 删除当前公司和下属所有子公司的组织机构关联信息。  调用接口IOMTCompany：DelCompGroup  currentCompSeq = 当前需要删除的船公司SEQ |

#### 数据初始化

根节点的数据直接使用语法初始化到数据库中，不在页面中维护。

## 公司权限设置（运维）

### 需求说明

* 所有公司都需要后台运维人员操作的权限分配页面。
* 公司权限需要区别公司业务属性控制，公司的业务属性由运维人员设置。一家公司允许多个业务属性，例如可以同时具有货主和船公司业务属性，具体属性在绑定到公司下属账户时明确一一对应。
* 所有公司都需要有菜单授权功能
* 所有公司都需要“公司管理员”绑定的功能
* 船公司额外的需要数据权限的设置

### 公司列表页面

公司权限列表页面

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询条件：<公司中文名称><组织机构代码><营业执照号><是否认证><状态> [查 询](#_列表页面_6) 新 增 | | | | | | | | | | |
| 序号 | 公司中文名称 | 组织机构代码 | 营业执照号 | 税务登记证号码 | 公司地址 | 是否认证 | 状态 | 创建人 | 创建时间 | 操作 |
|  |  |  |  |  |  |  |  |  |  | 查看 公司权限↓  编辑 |

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 公司中文名称、  组织机构代码、  营业执照号 | 支持模糊查询，前后去空格，内容不区分大小写。 |
| 是否认证 | 下拉。内容限定为：V 已认证； N 未认证。 |
| 状态 | 下拉。内容限定为：-20已归并 -10 无效 10有效。 |
| 交互 |  |
| 加载过程 | 假设“鹏海运”公司为运维级管理员所属公司（SEQ:1）：   |  | | --- | | select comp\_namecn, comp\_orgno, comp\_licence, comp\_cmccode, comp\_address, identify\_mark, status, create\_user, create\_date from tcomp |   列表默认不查询数据，用户点击“查询”之后再加载数据。  字段的对应关系请参考数据库定义的同名字段。  按消息时间倒序排列，即最新日志排在最前。 |
| 删除 | 1. 运维管理员用户点击“删除”时，提示“是否确认删除（公司中文名称）？”，确认的话，对记录做标记删除。即将STATUS修改为“-10无效数据”，并修改记录的时间戳。   调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 删除公司 | INFO | 用户{用户账户}删除公司SEQ{TCOMP\_SEQ},公司英文名称{COMP\_NAMEEN},公司中文全称{COMP\_NAMECN}。 | 常规异常 | 无 |  1. 用户点击“查询”时，根据查询条件加载数据。 2. 用户点击“查看”弹出公司详细信息页面，展示字段和布局方式参见编辑界面，字段的内容直接用文字展示，不用文本框。   “审核”暂不实现。 |

### 公司新增（编辑）页面

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 公司英文名称\* |  | 公司中文全称\* | | |  |
| 简称拼音 |  | 公司中文简称 | | |  |
| 组织机构代码\* |  | 营业执照号\* | | |  |
| 税务登记证号码\* |  | 公司地址\* | | |  |
| 传真 |  | 公司前台电话 | | |  |
| 邮政编码 |  | 开票公司名称 | | |  |
| 财务联系人 |  | 财务联系人电话 | | |  |
| 开户行 |  | 账号 | | |  |
| 账单投递地址 |  | | | | |
| 合并公司 | 下拉 | |  |  | |
| 营业执照 | 上传  选择  图片预览 | | 公司LOGO | 上传  选择  图片预览 | |
|  |  |

[认 证](#_编辑页面_4) [撤销认证](#_编辑页面_4) [保 存](#_编辑页面_4) [取 消](#_编辑页面_4)

|  |  |
| --- | --- |
| **页面元素** | **描述** |
|  | 带“\*”为必填字段 |
| 合并公司 | 下拉。内容为“公司中文名称”，TCOMP<公司资料表>. COMP\_NAMECN<公司中文全称>。 |
| 营业执照、公司LOGO | “TCOMPATTR<公司附件信息表>”表获取，参考Eir中校验码的处理方式。 |
| 交互 |  |
| 图片上传逻辑 | 用户点击“上传”弹出文件选择框，只支持\*.jpg、\*.png两种格式，只支持10M以内大小图片做保存。  预览：用户选择了文件之后需要支持预览功能，即在图片展示区域显示缩小版的图片，其固定大小，不做缩放。  默认图片：如果营业执照、公司LOGO数据为空的，需要显示为 “无图片”的默认图片（白底红字即可，作为图形资源加载使用）。 |
| **保存逻辑** | 校验所有必填字段。  检查“公司英文名称”“公司中文全称”在TCOMP<公司资料表>是否重复，如重复提示“公司英文名称/公司中文全称”已存在。  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 新增公司 | INFO | 用户{用户账户}新增公司SEQ{TCOMP\_SEQ},公司英文名称{COMP\_NAMEEN},公司中文全称{COMP\_NAMECN}。 | 常规异常 | 无 |   将录入的字段数据正确的对应到TCOMP<公司资料表>写入表，和TCOMPATTR<公司附件信息表>的操作需要同一事物。 |
| 编辑逻辑 | 向后台提交修改过字段数据，再根据TCOMP\_SEQ将修改过字段数据更新到公司信息表。  编辑数据保存时，如果用户没有重新上传文件的，不需要更新“TCOMPATTR<公司附件信息表>”表。  如船公司用户重新上传“营业执照”、“公司LOGO”图片则根据TCOMP\_SEQ在公司附件表找到附件数据，将新上传图片的文件名称、文件大小、文件类型、文件内容覆盖原有附件数据。 |
| 认证逻辑 | 用户点击“认证”提示“您确认认证（公司中文名称）？”，确认的话，更改IDENTIFY\_MARK<PHY认证标记>为“V”已认证。  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 认证公司 | INFO | 用户{用户账户}认证了公司SEQ{TCOMP\_SEQ},公司英文名称{COMP\_NAMEEN},公司中文全称{COMP\_NAMECN}。 | 常规异常 | 无 | |
| 撤销认证逻辑 | 用户点击“撤销认证”提示“您确认撤销认证（公司中文名称）？”，确认的话，更改IDENTIFY\_MARK<PHY认证标记>为“N”未认证。  调用日志接口，记录操作日志：   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 标题 | LEVEL | 内容 | 接口 | 备注 | | 撤销认证公司 | INFO | 用户{用户账户}撤销认证了公司SEQ{TCOMP\_SEQ},公司英文名称{COMP\_NAMEEN},公司中文全称{COMP\_NAMECN}。 | 常规异常 | 无 | |

### 公司权限页面

公司权限设置页面由一组Tab页面组成，通过“下一步”操作导航设置。设置的步骤根据上一步选择内容的不同，存在逻辑变化。规则为：

* 如果当前公司在组织架构中的，需要设置组织架构级别的业务权限。
* 如果当前公司有“船公司”角色属性的，需要设置业务级别的权限。

### 第一页：业务属性设置

|  |
| --- |
| 公司名称：XXXX公司 |
| 【】CA班轮公司 【】TA货代 |
| 下一步 |

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
| 业务属性 | 调用接口IOMTCompany：GetFuncGroupNameList获取显示的数据。根据返回的code集合和name集合顺序加载，返回值中IList的顺序是对应的。显示为Code + 空格 + Name, 加载数据时，选择项的值定义为Code |
| 数据加载 | 调用接口OMTCompany ：GetFuncGroupNameList获取列表数据，并显示  <代码><空格><名称>  调用接口OMTCompany：QueryCompType获取当前公司设置的业务属性，并对应将显示中的亚吴属性显示为勾选效果。 |
| 下一步 | 如果用户选择了“CA 班轮公司”的（使用选择项的值CA进行判断）。  如果当前公司在公司层级关系中有下级公司的，进第二页，否则直接进第三页。  如果用户没有选择“CA班轮公司”的，进第四页  \*调用接口IOMTCompany：IsInGroup判断是否在公司层级中 |

### 第二页：公司层级权限设定

|  |  |
| --- | --- |
| 公司名称：XXXX公司 | |
|  | 当前公司具有船公司业务属性，并且在组织机构中定义了下属公司，需要确认当前公司在组织架构中的数据权限  【】本公司  【】本公司及下级公司 |
| 上一步 下一步 | |

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
| 公司组织机构树 | 调用接口IOMTCompany：QueryCompGroupByComp获取公司架构数据，并做成树状展示。  默认树展开，并对当前的公司名称末尾增加“\*”标示。 |
| 权限加载 | 调用接口IOMTCompany：QueryCompRuleConfig获取当前公司的设置  compSeq = 当前公司SEQ |
| 下一步 | 跳转第三页 |
| 上一步 | 回到第一页，页面内容需要缓存 |

### 第三页：业务权限设定

|  |  |
| --- | --- |
| 公司名称：XXXX公司 | |
|  | 当前公司具有船公司业务属性，需要确认当前公司在系统中的业务范围  装货港范围  【】所有港口 【】指定港口 选择港口  驳船起运港权限  【】所有港口 【】指定港口 选择港口 |
| 上一步 下一步 | |

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
| 公司组织机构树 | 调用接口IOMTCompany：QueryCompGroupByComp获取公司架构数据，并做成树状展示。  默认树展开，并对当前的公司名称末尾增加“\*”标示。 |
| 装货港范围 | 指定港口时允许指定多个港口，允许手工录入。选择器选择的港口，作为追加内容追加到指定港口。多个港口之间采用半角逗号间隔 |
| 驳船起运港范围 | 指定港口时允许指定多个港口，允许手工录入。选择器选择的港口，作为追加内容追加到指定港口。多个港口之间采用半角逗号间隔 |
| 权限加载 | 调用接口IOMTCompany：QueryCompRuleConfig获取当前公司的设置  compSeq = 当前公司SEQ |
| 下一步 | 跳转第四页 |
| 上一步 | 需要根据进入时的情况判断上一页是第几页，跳转后的页面内容需要缓存 |

### 第四页：菜单权限和管理员绑定

|  |
| --- |
| 公司名称：XXXX公司 |
| 公司管理员账户： 张三 检查 |
| √ 船期管理  √ 船舶管理  √ 查询 √ 新增 √ 修改 √ 删除  √ 航线管理  √ 查询 √ 新增 √ 修改 √ 删除  √ 船期管理  √ 查询 √ 新增 √ 修改 √ 删除  √ 舱位管理  √ 查询 √ 新增 √ 修改 √ 删除  √ 订舱管理  √ 订舱审批  √ 查询 |
| 上一步 保存 |

|  |  |
| --- | --- |
| **界面元素** | **描 述** |
| 公司管理员 | 调用接口IOMTCompany：GetCompAdmin获取公司对应的管理员账户。  允许手工修改，空值时提示操作员“尚未绑定账户”，并阻拦数据保存。 |
| 检查 | 调用接口IOMTCompany：CheckCompAdmin检查公司管理员账户是否正常，不正常的提示用户。 |
| 权限树 | 调用接口 IOMTCompany：GetFuncList获取FUNC\_TBL<菜单注册表>，并按样式部署。  树状建议展示在DIV中，超出高度时显示竖向滚动条。  调用接口IOMTCompany：GetRule获取管理员对应的权限，并将对应的功能点设置为勾选。  如果没有管理员的，或者是获取到管理员对应的权限为空的，调用接口 IOMTCompany：GetGroupsRule获取在第一页设置业务属性对应的权限列表，接口返回的列表数据已经在接口内部进行过滤了重复数据，并将对应的功能点设置为勾选。 |
| 上一步 | 需要根据进入时的情况判断上一页是第几页，跳转后的页面内容需要缓存 |
| 保存 | 调用接口 IOMTCompany：SaveRuleConfig。  参数赋值：  compSeq decimal= 当前的公司SEQ  useOwnerRule int = 如果进入了第三页，设置为1，否则0  useGroupRule int = 如果进入了第二页，设置为1，否则0  compType IList<String> = 业务属性页面选择的业务属性  groupRuleCfg int = 公司层级权限设置中，选择“本公司”的传入1.“本公司及下级公司”的传入2.  ownerRuleLoadCfg int = 业务权限设置中，装货港范围为全部的传入1，否则传入2.传入2时ownerRuleLoadingPort不能为空。  ownerRuleLoadingPort IList<String> = 装货港范围，代码，需要根据半角逗号分拆之后传入  ownerRuleBoatLoadCfg int = 业务权限设置中，驳船起运港范围为全部的传入1，否则传入2.传入2时ownerRuleBoatLoadingPort不能为空。  ownerRuleBoatLoadingPort IList<String>= 装货港范围代码，需要根据半角逗号分拆之后传入  compAdmingAccount String = 公司管理员账户，不能为空  funcForAdmin IList<String> =公司管理员权限，传入页面勾选内容的FUNC\_ID。功能层级关系在接口内部有检查和补充逻辑，可传入完整的功能结构或者是仅传入最后一级的功能FUNC\_ID。 |

## 提示消息与字符集维护

### 需求说明

UI界面，提示信息为支持中、英文显示，采用数据库的编码映射对照方式处理双语。该功能只对运维人员使用。

### 数据表

参见[多语种接口](#_多语种接口);

### 列表页面

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询条件：<Path(路径)><中文><英文><状态> [查 询](#_列表页面_5) [新 增](#_列表页面_5) [导 入](#_列表页面_5) [导 出](#_列表页面_5) | | | | | | | | | | |
| **序号** | **Path（路径）** | **中文** | **英文** | **繁体** | **状态** | **创建人** | **创建时间** | **修改人** | **修改时间** | **操 作** |
|  |  |  |  |  |  |  |  |  |  | [编辑](#_列表页面_5) [删除](#_列表页面_5) |

**查询条件：**

1.Path(路径)、中文、英文支持模糊查询，前后去空格，内容不区分大小写。

2.状态，下拉。内容限定为：-10无效 10有效。

**加载过程：**

1. 列表默认不查询数据，用户点击“查询”之后再加载数据。

2.字段的对应关系请参考数据库定义的同名字段。

3.按创建时间倒序排列，即最新日志排在最前。

**界面交互：**

1. 用户点击“新增”“编辑”时，弹出编辑界面。
2. 用户点击“删除”时，提示“是否确认删除（中文/英文）？”，确认的话，对记录做标记删除。即将STATUS修改为“-10删除数据”，并修改记录的时间戳。

3.用户点击“查询”时，根据查询条件加载数据。

4.用户点击“导出”根据查询条件导出SYSTEM\_UI\_LANGUAGE<界面语言对照>表数据写入Excel

文件头数据格式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **Path（路径）** | **中文** | **英文** | **繁体** |

* 用户点击“导入”弹出导入页面。

**导入逻辑：**

|  |
| --- |
| 选择文件  [导 入](#_列表页面_5) |

1. 用户点击选择“选择文件”，选择Excel文件（以导出功能导出的文件作为模板），点击导入按钮。
2. 需要校验Excel文件头是否正确，提示用户“文件格式不正确”。
3. “序号”可以为空，“Path路径”不能为空，“中文”不能为空，“英文”不能为空。
4. 文件内不允许重复数据，规则为序号不能重复，提示用户“N行与M行存在重复数据请修改”。
5. 如有多行错误数据则需要一次提示多条错误信息，只有文件内容和格式正常情况下才做数据录入。
6. 序号不为空，根据序号可以找到已存在表的数据则用文件数据覆盖，如未找到则要新增一条数据。
7. 序号为空，新增一条数据。

### 新增/编辑页面

|  |  |  |  |
| --- | --- | --- | --- |
| Path(路径) |  | Text\_Code |  |
| 中文名 |  | | |
| 英文名 |  | | |
| 繁体名 |  | | |

保存 取消

**保存逻辑**：

* + - 1. 将填写数据写入SYSTEM\_UI\_LANGUAGE<界面语言对照>即可，对应字段参照表结构，默认状态Status=10。

## 日志查询

### 需求说明

为更好的监控系统，需要经常对业务类型的日志查询。常用的可能按船公司查一段时间内的一票订单跟踪日志，查系统代码出现的程序异常日志，按操作类型区分常规日志等等。

### 数据表

参见[异常和日志接口;](#_异常和日志接口)

### 列表页面

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 查询条件：<船公司><操作公司><订舱公司><操作人><订单号><操作表名><信息类型><消息时间> [查询](#_日志查询) | | | | | | | | | | | |
| **流水号** | **操作类型** | **操作表名** | **操作记录号** | **信息类型** | **订单号码** | **信息描述** | **船公司** | **操作公司** | **订舱公司** | **操作人** | **消息时间** |

**查询条件：**

1. 船公司，下拉加载船公司列表。操作公司下拉，需与船公司联动。
2. 订舱公司、操作人、订单号支持模糊查询，前后去空格，内容不区分大小写。
3. 操作表名，下拉加载出所有数据库表名称。
4. 消息类型，下拉。内容限定为：1-DEBUG 2-INFO 3-WARN 4-ERROR。
5. 数据状态，下拉。内容限定为：0-有效数据 -10-无效数据 -20-删除数据。
6. 消息时间，分开始结束时间，精确到时分。【年月日】用时间控件选择，不可输入。【时分】文本框显示可输入小时限制“00-23”，分钟限制“00-59”的数字，否则提示格式错误。例如默认格式，正确格式，错误格式。

**数据加载：**

1.列表默认不查询数据，用户点击“查询”之后再加载数据。

2.字段的对应关系请参考数据库定义的同名字段。

3.按消息时间倒序排列，即最新日志排在最前。

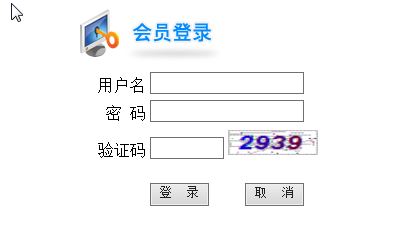
# 网站详细设计

## 登录页面

### 需求说明

为船公司用户、下属操作公司用户、货主货代用户、运维（运营）用户使用，WEB页面提供统一的登录入口调用后台登录接口完成登录。

### 登录界面



**登录规则：**

1. 用户名、密码、验证码不能为空，都为必填项。
2. 用户名输入正确密码输入错误，提示“用户名或密码错误”。
3. 勾选记住密码，用户下次打开登录页面默认填好用户名和密码。

**处理逻辑：**

1.向后台调用户接口[OMTUser：Login](#_OMTUser：Login)。

2.登录成功后在主功能页面加载出用户对应的所有菜单权限。

## 首页登录

### 功能说明

实现首页的登录功能。

### 界面



根据原型实现用户的登录操作。

1. 原型界面中没有验证码，实际实现时也不需要增加验证码。但请预留相关的位置，可能会增加
2. 调用OMTUser的方法Login实现登录操作，登录之后将OMTUser对象保存到Session中，Key请开发时自行定义
3. 登录失败的，直接提示“登录不成功，请检查用户名，密码组合是否输入正确。”不需要提示具体的错误是什么。
4. 登录之后，需要将页头中的“登录 | 注册”替换成登录的用户名称。





## 订舱中心

### 功能说明

根据原型实现订舱中心页面的相关功能。

### 界面说明



区域1：查询条件

* 查询条件按照原型实现，Tab切换，支持按“航线 + 起止港”或者按“船名航次” 查询。
* 航线代码，支持自动完成，数据来源自“SHIP\_ROUTE\_HEAD<航线信息表头>”，获取“ROUTE\_CODE<航线代码>”，不区分船公司，Distinct处理。需要支持手工录入，支持[航线选择器](#_航线选择器)。
* 起始港，目的港加载港口自动完成，支持手工录入，支持[港口选择器](#_港口选择器)。
* 船名，航次：手工录入
* 调用“[OMTPrice：GetListOPageByParm](#_OMTPrice：GetListOPageByParm)”接口实现查询，相关细节请参考接口的说明。
* Tab切换时，保留原有录入的查询条件。

区域1：快捷查询

* 快捷查询中的条件与录入的查询条件之间是and关系，即两个条件是同时起作用的。
* 船公司：加载船公司列表，数据库顺序。（TCOMP\_GROUP<公司组织>）
* 时间段：根据原型固定加载
* 箱型：根据原型固定加载

区域2：查询结果

* 页面打开时，根据配置加载列表数据。调用接口“[OMTPrice：GetListByCfg](#_OMTPrice：GetListByCfg)”，配置的语法暂定为“20尺价格最低，并且最新发布的排在前面”。
* 输入条件之后的查询调用接口“[OMTPrice：GetListOPageByParm](#_OMTPrice：GetListOPageByParm)”实现。

区域3：登录

* 用户已经登录的，显示用户名称。 样式



* 用户点击登录之后，弹出登录窗口“[登录页面](#_登录页面)”

区域4：常见问题

保持静态，本次不实现。

### 驳船装货港查询



1. “按航线/起止港”查询部分增加“驳船起运港”查询条件。“起始港”标签文字修改为“大船起运港”，“目的港”标签，文字修改为“大船目的港”； “航线”保持不变。
2. “按航线/起止港”查询时增加新的查询条件限制条件：如果输入了“大船起运港”或者“大船目的港”条件中的一个时，必须输入另外一个，两个条件绑定，同时使用。
3. 查询的列表数据有变更，排序参考下图（PS：图中的查询条件少了“航次”，请认为有此条件）

- “起止港口”定义显示为：（驳船起运港-）大船起运港-大船目的港。括号中的内容在查询输入了“驳船起运港”条件时显示，否则隐藏；显示时不带括号。

- “起止港口”显示为港口的编码“CNYTN-UNSTG”。

- “船公司”显示船公司代码，对应字段“BOOKING\_PRICE ：OWNER\_CODE<船公司代码>”

- “截重日期”对应字段“EXPIRY\_DATETIME<截重时间>”（PS：新增字段，参考“FAK价格维护变更”），日期的格式化为“YYYY-MM-DD WW”。（日期的格式化请调用格式化接口，参考“日期格式化”）

- “ETD”预计离港时间，对应字段“OUT\_DATE<离港时间>”；（日期的格式化请调用格式化接口，参考“日期格式化”）

- “大船运价”对应原有的“USD（20'/40'/40HQ/45')”，将运价的货币单位从列标题中去除，改为在显示价格之前增加货币单位，显示为“USD 1880/3850/3750/-”.

- “驳船运价”的显示与大船价格的显示类似，具体的查询方式参考“驳船价格查询逻辑”。

- 在“操作”的预订按钮之前，增加“详细”图标，由于原型尚未变更，显示的图形暂定为

大小确定为“16\*16”。原型确定后，请UI变更此图形。

1. 点击“详细”之后弹出“价格详细页面”。
2. 查询页面中的“20'/40'/40HQ/45'”与标准箱型之间的对应关系

|  |  |
| --- | --- |
| 20' | 20GP |
| 40' | 40GP |
| 40HQ | 40HQ |
| 45 | 45GP |

## 磁贴页面效果

### 需求说明

1. 磁贴主要在三个位置应用，首页的磁贴分为两个部分，左边的认为是快捷菜单，右边的定义为快捷页面；还有在用户登录后的首页也有磁贴的应用。

### 首页磁贴左



1. 调用接口 IOMTMenuQuickLink：GetHomeQuickMenu获取需要显示的菜单项，按“从左到右，从上到下”的顺序**对应位置**加载，位置定义分别是“1 – 9 ”。位置配置中允许存在空配置，即加载时需要按位置定义处理，如果遇到无定义的位置，跳过直接加载下一菜单项，当前的位置仅显示地图或者底色。
2. 用户点击图形对应的链接之后，如果无权限的需要停留在首页，这部分逻辑依赖页面自身的权限控制功能实现，不需要有增加的逻辑，当请测试和开发的同事重点验证一下。
3. UI在处理中加载时还需要注意位置和图形的大小关系，1号位置的图形使用大图，其他位置使用小图。
4. UI层中的元素需要将底图或者底色，与菜单的图标分离，HTML的样式参考下面的例程代码。重点关注： 1）底图和底色使用样式控制，静态写到页面中。2）链接和加载图形根据配置信息加载。

|  |
| --- |
| <table>  <tr>  <td rowspan="2" style="background-image:url('Image/yjfb.jpg');width:266px; height:191px;">  <a href="#">  <img style=" width:100%; height:100%" src="Image/teij01.png" />  </a>  </td>  <td style="background-color:#12a2de; width:130px; height:93px;">  <a href="#">  <img style=" width:100%; height:100%" src="Image/teij01.png" />  </a>  </td>  </table> |

### 首页磁贴右



1. 调用接口 IOMTMenuQuickLink：GetHomeQuickTool获取需要显示的推荐工具项，按“从左到右，从上到下”的顺序**对应位置**加载，位置定义分别是“1 – 4 ”，其中的1, 2对应大图，3,4对应小图。
2. 页面的UI处理参考“首页磁贴左”处理。

### 登录后的首页磁贴



1. 调用接口 IOMTMenuQuickLink：GetUserHomeQuickMenu获取需要显示的推荐工具项，按“从左到右，从上到下”的顺序**对应位置**加载，位置定义分别是“1 – 9 ”，其中的1对应大图，其他位置对应小图。
2. 页面的UI处理参考“首页磁贴左”处理。
3. 无如果遇到无定义的位置，默认加载图形（请开发直接从原型获取图形文件），链接设置为跳转到“自定义快捷设置”与“编辑”按钮相同。



1. 用户点击编辑之后弹出快捷编辑页面，参考功能点“自定义快捷设置”

### 登录后的产品推荐



1. 调用接口 IOMTMenuQuickLink：GetUserProduct获取需要显示的推荐工具项，按“MENU\_LOCATION<定位标记>”排序之后加载对应的小图和链接。
2. 左上角的“编辑”按钮取消，暂不实现。 即本期仅实现固定的产品推介，暂不支持用户自定义的产品链接。

### 菜单的图形定义

菜单的图形定义保存到“FUNC\_TBL<菜单注册表>”中的“PIC\_URL”，考虑图形的格式可能存在扩展，对大图和小图不单独字段维护，采用格式化字符串维护在同一个字段中。格式化的字符串采用JSON格式存储，样式定义和样例

|  |
| --- |
| "L": "image/asd.jpg", "M": "image/asd222.jpg"   1. 属性L表示大图，M表示小图。 |

1. 菜单图形快捷图形的定义直接初始化到数据库中，不需要配置页面。

### 数据初始化样例

1. 通用的配置需要直接使用语法初始化到数据库中，本期不计划实现配置页面。使用的配置数据单独文档提供。
2. 用户自定义的快捷菜单，本期需要实现配置页面，参考功能点“自定义快捷设置”。
3. 通用配置的初始数据参考下表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PAGE  \_LOCATION | MENU  \_GROUP | USER  \_ACCOUNT | MENU  \_LOCATION | L\_IMAGE | M\_IMAGE | FUNC\_ID | FUNC\_NAME | LINK\_URL |
| HomePage | PushFunc | \* | 1 | a.jpg | b.jpg | Fn01000 | 发布运价 |  |
| HomePage | PushFunc | \* | 2 |  |  |  | 用箱期申请 |  |
| HomePage | PushFunc | \* | 3 |  |  |  | 做柜查询 |  |
| HomePage | PushFunc | \* | 4 |  |  |  | FAK订舱 |  |
| HomePage | PushFunc | \* | 5 |  |  |  | 特价订舱 |  |
| HomePage | PushFunc | \* | 6 |  |  |  | 船期 |  |
| HomePage | PushFunc | \* | 7 |  |  |  | 订舱审批 |  |
| HomePage | PushFunc | \* | 8 |  |  |  | 货物状态跟踪 |  |
| HomePage | PushFunc | \* | 9 |  |  |  |  |  |
| HomePage | PushTool | \* | 1 |  |  |  | 常用工具 |  |
| HomePage | PushTool | \* | 2 |  |  |  | 联系客服 |  |
| HomePage | PushTool | \* | 3 |  |  |  | 支持我们 |  |
| HomePage | PushTool | \* | 4 |  |  |  | 关注我们 |  |

### 应用范围

磁贴的效果需要在两个位置应用，一个是登录之前的的网站主页，一个是登录之后的用户主页。两个主页都需要实现磁贴的效果。

## 箱动态查询

### 需求说明

迁移原有的<http://cmclink.com/Search/Default.aspx>页面中的箱信息查询功能。

### 查询条件



1. 原型中的货物追踪只有一个查询条件，在编码时需要调整为“箱号”。即在主页中只有一个查询条件框的，仅支持箱号条件查询，不支持BK NO号查询。
2. 点击查询之后 **跳转到新页面**“查询结果页面”，这个页面暂时没有原型，请参考鹏腾网的查询结果页面编码，不需要复杂样式的修饰，白板页面即可。编码完成之后交给UI修饰。



## 用户密码找回（网站）

### 需求说明

用户在忘记密码时，通过网站找回密码页面三种方式“用户名找回”、“EMAIL找回”、“手机号找回”找回自己密码，目前需求提出一期实现“用户名找回”、“EMAIL找回”功能。

【非功能需求】除用户自身可修改密码外，授权用户只能初始化其它用户密码并发到预留邮箱或手机上

### 功能说明

向用户填写的Email地址发送重置密码邮件，邮件内容包含了重置密码地址，用户通过该地址进入修改密码页面完成操作。

### 菜单

网站首页 >> 登录框 >> 忘记密码?

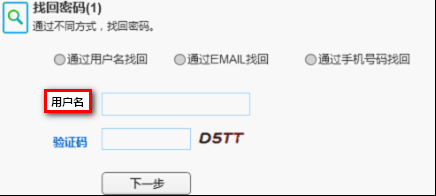
### 数据库变更

TUSER<用户账户表>

TUSERCOMP<账户公司关联>

### 功能实现

#### 用户名找回



PS:需求文档中的操作页面

注：验证码功能实现参考EIR系统

选择“通过用户名找回”方式

传入用户输入的“账户”= userName调用IOMTUser：ResetPasswordByUserName

成功则跳转至“操作提示页面”提示：

|  |
| --- |
| 设置新密码链接已发送至您的邮箱:{email}  请您在24小时内登录邮箱，按邮件中的知识完成设置新密码操作。  另：如有返回的goMailUrl参数不为空则显示  “**去邮箱收信**”按钮，通过该按钮可跳转至邮箱网址  否则不显示。 |

失败则在用户名文本框后面提示：发送密码重置邮件失败，errorMsg

#### EMAIL找回



PS:需求文档中的操作页面

注：验证码功能实现参考EIR系统

选择“通过EMAIL找回”方式

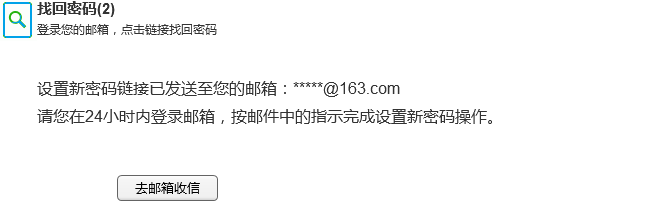
传入用户输入的“EMail”= email调用IOMTUser：ResetPasswordByEmail

成功则跳转至“操作提示页面”提示：

|  |
| --- |
| 设置新密码链接已发送至您的邮箱:{fmtEmai}  请您在24小时内登录邮箱，按邮件中的知识完成设置新密码操作。  另：如有返回的goMailUrl参数不为空则显示  “**去邮箱收信**”按钮，通过该按钮可跳转至邮箱网址  否则不显示。 |

失败则在用户名文本框后面提示：发送密码重置邮件失败，errorMsg

#### 操作提示页面



## 用户密码重置（网站）

### 功能说明

用户收到密码重置邮件，通过邮件中的URL地址跳转到修改密码页面做密码修改功能。

### 菜单

“用户邮件” 跳转至 “站首页 >> 修改密码页面”

### 数据库变更

TUSER<用户账户表>

TUSERCOMP<账户公司关联>

### 功能实现

**逻辑说明：**

用户在自己的邮件内，点击密码重置邮件中的URL请求：

[https://bookingonline.com/ResetPassword.aspx?ret=SAGHSDHSDFHERWE2344632DFSHE](https://epay.cmclink.com/epay/ResetPassword.aspx?ret=SAGHSDHSDFHERWE2344632DFSHE)

1. 在该页面将ret作为参数调用IOMTUser：InitResetPasswordPage方法
2. 获取当前session中是否有“USER\_NAME”的登录用户对象，如果有需要清空以下对象

userDispName：用户的显示名称

userCompSeq：用户账户对应的公司SEQ

userCompNameXXX：用户账户对应公司的名称

userOwnerSeq：用户所属公司对应的船公司

userOwnerCompNameCn userOwnerCompNameEn

userOwnerCODE：“COMP\_NAMESHORT\_PY<简称拼音>”。

userRules：权限对象。

userDataRules：数据权限对象。

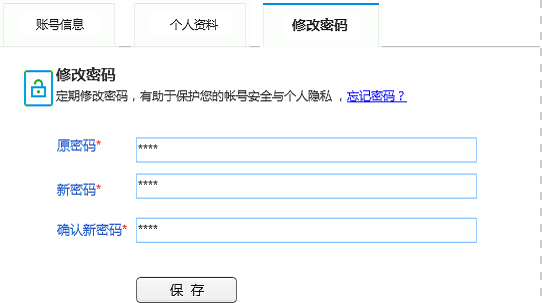
1. 取IOMTUser：InitResetPasswordPage方法返回的userAccount、passWord作为参数调用

OMTUser：Login 登录接口

判断返回的errorMsg

为空

登陆成功，跳转到修改密码页面；



PS:需求文档中的操作页面

1、获取返回的原密码“passWord”作为“原密码”的默认值，该密码框不可修改。

2、用户输入的“新密码”和“确认新密码”点击保存，判断密码是否一致

3、oldPassword =原密码，newPassword=确认新密码。调用OMTUser：ChangePsw修改密码。

不为空

登陆失败，跳转登陆页面

## 用户登录保护

### 需求说明

* 录入3次密码输入错误，要求输入验证码；10次以上错误，锁定(停用)用户名；无无时间限制。用户登录系统失败的错误信息记入表保存。每当用户登录成功将该表中与用户相关的数据置为无效。被停用的用户需要通过运维人员在后台“解冻”账户。

### 登陆页面修改

* 在用户输入账户名称之后，通过Ajax调用接口OMTUser：NeedVerifyCode

userAccount = 当前用户录入的账户

* 接口返回true的需要显示验证码，并在后台校验验证码

# 其他子系统设计

## 船公司参数配置

### 功能说明

船公司参数用于配制系统相关的全局参数功能，对应于船公司。功能参考公共部分的全局参数表，增加了部分显示字段。

### 数据库变更



### 接口

接口请参考EIR重构中全局参数获取的方式编写，要点提示

1. 需要分船公司维护全局参数
2. 全局参数的变更需要登记日志

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标题 | LEVEL | 内容 | 接口 | 备注 |
| 全局参数 | INFO | {UserAccount}修改了全局参数{key\_word},Key\_Value由{old\_value}修改为{new\_value}, Key\_Value2由{old\_value}修改为{new\_value} | 常规异常 | OWNER\_SEQ填写当前船公司；  OBJECT\_SEQ= SEQ  OBJECT\_TABLE=BKONLINE\_CONFIG  ORDER\_NO=KEY\_WORD |

### 全局参数初始化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OWNER\_SEQ | KEY | KEY\_VALUE | KEY\_VALUE2 | REMARK |
| 37 | 101 | USD |  | 本位币设置 |

## 国家区域选择器

### 需求说明

作为船信息基本数据，用到的地方有新增船舶-船舶国籍字段用到该选择器取值。

### 数据表



1. “OWNER\_SEQ”默认取值“0”，取值为0的编码表示所有船公司通用，而船公司特定的编码标记船公司自己的SEQ，不支持对不同的操作公司不同的编码体系，对一家船公司仅能存在一套编码体系。
2. “REPLACE\_CODE”参见《BKOnline系统详细设计文档LEVEL2.doc》编码表体系解决方案。用于替换“SHIP\_NATION\_SEQ”。
3. “COVER\_CODES”记录覆盖该条数据被哪家船公司覆盖的作用，数据格式：“,1,2,3,”。
4. “STATUS”取值为“-10 删除 10有效”，采用标记删除方式处理。
5. “PARENT\_SEQ”默认为“0”表示为国家，录入区域需要指定为国家的“SHIP\_NATION\_SEQ”。

例如：

|  |  |  |  |
| --- | --- | --- | --- |
| SHIP\_NATION\_SEQ | NATION\_AREA\_CODE | NATION\_AREA\_NAME | PARENT\_SEQ |
| 1 | CN | 中国 | 0 |
| 2 | CN\_SZ | 深圳 | 1 |
| 3 | CN\_SH | 上海 | 1 |

### 界面



船舶国籍区域选择器(图1)

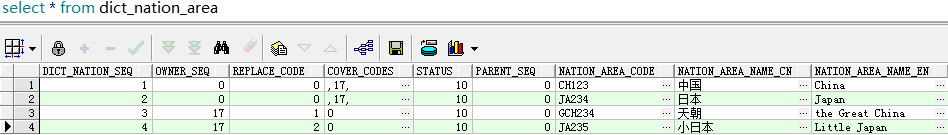
**查询条件：**

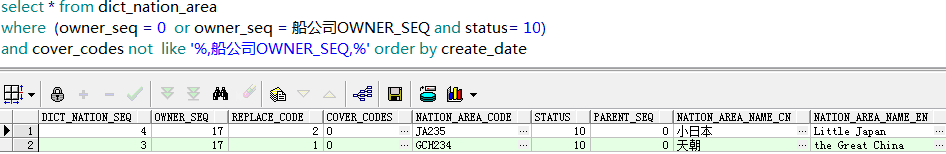
1. 【国籍】文本框需要支持对“中文名称”“英文名称”“国籍代码”三个字段的模糊查询非必填可为空。

不区分查询条件内容的大小写。

**加载过程：**

1. 在“中文名称”前加入“选择”列。
2. 参照下图基本查询语句。





1. 列表按创建时间倒序排列，即最近创建的排在最前面。

4.列表中的数据长度不允许超出一行的默认宽度，如出现数据换行的情况必须做内容截取，超出部分用“...”三个省略号替换，并增加浮动提示。

5.列表中的数据若不存在使用“-”代替。

6.需要分页，每页显示10条数据。

**数据返回：**

点击选择列，返回国家信息。

## 箱动态采集分发数据变更(后台EXE 9903)

### 说明

原有的分发后台仅分发了从中海采集上来的箱动态数据。需求变更后，箱动态数据增加了从堆场分布式系统采集上来的箱动态数据，这部分数据同样需要分发下去。且堆场分布式系统的箱动态数据和中海的箱动态数据间存在大量重复数据，需要去掉重复的数据后再统一分发

### 功能实现

后台运行参数(未变更)

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 样例 |
| P1 | 船公司SEQ | 37 |
| P2 | 采集开始时间 | 2013-01-01 |
| P3 | 每次获取的记录条数 | 300 |
|  | 运行周期 | 周期任务 |
|  | 运行间隔 | 3分钟 |
| P6 | 重复数据校验的时间范围 | 90 单位：天 |

实现步骤：

* 标识两个数据源需要处理的数据，锁定需要处理的数据范围，避免在处理过程中有新数据采集上来干扰后续处理。参考SQL：

1. 锁定中海箱动态未处理数据。参考SQL：

UPDATE CONTA\_MOVEMENT SET ISCOLLECT = -99 WHERE ISCOLLECT=0 AND OWNER\_SEQ = [P1]

1. 锁定堆场分布式箱动态未处理数据。参考SQL：

UPDATE PRD\_CONTA\_MOVEMENT SET ISCOLLECT=-99 WHERE ISCOLLECT=0 AND OWNER\_SEQ = [P1]

说明：

-99为临时标识

* 查询两个箱动态数据源(表CONTA\_MOVEMENT和表PRD\_CONTA\_MOVEMENT数据)的最后采集时间，确定过滤掉哪个数据源的重复数据

a．查询中海的箱动态数据最后采集时间。参考SQL：

SELECT MAX(CREATE\_DATETIME) [P4] FROM CONTA\_MOVEMENT WHERE ISCOLLECT=-99

b.查询堆场分布式系统的箱动态数据最后采集时间。参考SQL：

SELECT MAX(CREATE\_DATETIME) [P5] FROM PRD\_CONTA\_MOVEMENT WHERE ISCOLLECT=-99

* 过滤掉两个数据源的重复数据后将记录归集到箱动态汇总表CONTA\_MOVEMENT\_ALL

1. [P4] > =[P5]，以堆场分布式的箱动态数据为准，过滤掉中海重复的箱动态数据
2. 丢弃堆场分布式中重复的箱动态数据，参考SQL

UPDATE PRD\_CONTA\_MOVEMENT

SET ISCOLLECT = 20

WHERE ISCOLLECT = -99

AND EXISTS

(SELECT CONTA\_MOVEMENT\_SEQ

FROM CONTA\_MOVEMENT\_ALL

WHERE OWNER\_SEQ = PRD\_CONTA\_MOVEMENT.OWNER\_SEQ *--船公司相同*

AND BOOKING\_NO = PRD\_CONTA\_MOVEMENT.BOOKING\_NO *--订舱单号相同*

AND CONTAINERNO = PRD\_CONTA\_MOVEMENT.CONTAINERNO *--箱号相同*

AND MOVEMENT\_CODE = PRD\_CONTA\_MOVEMENT.MOVEMENT\_CODE *--箱动态类型相同*

AND CREATE\_DATETIME >= SYSDATE - [P6]) *–在限定的时间范围内比较*

1. 将堆场分布式的箱动态数据写入到汇总表。参考SQL：

INSERT INTO CONTA\_MOVEMENT\_ALL(CONTA\_MOVEMENT\_SEQ,…, ISCOLLECT, DATASOURCE) SELECT CONTA\_MOVEMENT\_SEQ,…,0,1 FROM PRD\_CONTA\_MOVEMENT WHERE ISCOLLECT=-99

注：

CONTA\_MOVEMENT\_ALL.ISCOLLECT = 0 (未处理)

CONTA\_MOVEMENT\_ALL.DATASOURCE=1 (堆场分布式系统)

其他同名字段取值

1. 更新堆场分布式的箱动态数据状态。参考SQL：

UPDATE PRD\_CONTA\_MOVEMENT SET ISCOLLECT=10 WHERE ISCOLLECT=-99

1. 丢弃中海箱动态数据中重复的数据。

需求对于重复箱动态的定义：

公共平台数据库的原始箱动态已存在相同**船公司、订舱单号、箱号、箱动态类型**的数据，以先取到的数据为准

参考SQL：

UPDATE CONTA\_MOVEMENT

SET ISCOLLECT = 20

WHERE ISCOLLECT = -99

AND EXISTS

(SELECT CONTA\_MOVEMENT\_SEQ

FROM CONTA\_MOVEMENT\_ALL

WHERE OWNER\_SEQ = CONTA\_MOVEMENT.OWNER\_SEQ *--船公司相同*

AND BOOKING\_NO = CONTA\_MOVEMENT.BOOKING\_NO *--订舱单号相同*

AND CONTAINERNO = CONTA\_MOVEMENT.CONTAINERNO *--箱号相同*

AND MOVEMENT\_CODE = CONTA\_MOVEMENT.MOVEMENT\_CODE *--箱动态类型相同*

AND CREATE\_DATETIME >= SYSDATE - [P6]) *–在限定的时间范围内比较*

1. 将过滤后的箱动态数据写入汇总表。参考SQL：

INSERT INTO CONTA\_MOVEMENT\_ALL(CONTA\_MOVEMENT\_SEQ,…, ISCOLLECT, DATASOURCE) SELECT CONTA\_MOVEMENT\_SEQ,…,0,1 FROM CONTA\_MOVEMENT WHERE ISCOLLECT=-99

注：

CONTA\_MOVEMENT\_ALL.ISCOLLECT = 0 (未处理)

CONTA\_MOVEMENT\_ALL.DATASOURCE=0 (中海)

其他同名字段取值

1. 更新采集到汇总表的中海箱动态数据状态。参考SQL：、

UPDATE CONTA\_MOVEMENT SET ISCOLLECT=10 WHERE ISCOLLECT=-99

1. [P4] <[P5]，以中海箱的动态数据为准，过滤掉堆场分布式重复的箱动态数据

* 丢弃中海箱动态数据中重复数据。参考SQL：

UPDATE CONTA\_MOVEMENT

SET ISCOLLECT = 20

WHERE ISCOLLECT = -99

AND EXISTS

(SELECT CONTA\_MOVEMENT\_SEQ

FROM CONTA\_MOVEMENT\_ALL

WHERE OWNER\_SEQ = PRD\_CONTA\_MOVEMENT.OWNER\_SEQ *--船公司相同*

AND BOOKING\_NO = PRD\_CONTA\_MOVEMENT.BOOKING\_NO *--订舱单号相同*

AND CONTAINERNO = PRD\_CONTA\_MOVEMENT.CONTAINERNO *--箱号相同*

AND MOVEMENT\_CODE = PRD\_CONTA\_MOVEMENT.MOVEMENT\_CODE *--箱动态类型相同*

AND CREATE\_DATETIME >= SYSDATE - [P6]) *–在限定的时间范围内比较*

* 将中海的箱动态数据写入到汇总表。参考SQL：

INSERT INTO CONTA\_MOVEMENT\_ALL(CONTA\_MOVEMENT\_SEQ,…, ISCOLLECT, DATASOURCE) SELECT CONTA\_MOVEMENT\_SEQ,…,0,1 FROM CONTA\_MOVEMENT WHERE ISCOLLECT=-99

注：

CONTA\_MOVEMENT\_ALL.ISCOLLECT = 0 (未处理)

CONTA\_MOVEMENT\_ALL.DATASOURCE=0 (中海)

其他同名字段取值

* 更新中海的箱动态数据状态。参考SQL：

UPDATE CONTA\_MOVEMENT SET ISCOLLECT=10 WHERE ISCOLLECT=-99

* 丢弃堆场分布式系统箱动态数据中重复的数据。

需求对于重复箱动态的定义：

公共平台数据库的原始箱动态已存在相同**船公司、订舱单号、箱号、箱动态类型**的数据，以先取到的数据为准

参考SQL：

UPDATE PRD\_CONTA\_MOVEMENT

SET ISCOLLECT = 20

WHERE ISCOLLECT = -99

AND EXISTS

(SELECT CONTA\_MOVEMENT\_SEQ

FROM CONTA\_MOVEMENT\_ALL

WHERE OWNER\_SEQ = PRD\_CONTA\_MOVEMENT.OWNER\_SEQ *--船公司相同*

AND BOOKING\_NO = PRD\_CONTA\_MOVEMENT.BOOKING\_NO *--订舱单号相同*

AND CONTAINERNO = PRD\_CONTA\_MOVEMENT.CONTAINERNO *--箱号相同*

AND MOVEMENT\_CODE = PRD\_CONTA\_MOVEMENT.MOVEMENT\_CODE *--箱动态类型相同*

AND CREATE\_DATETIME >= SYSDATE - [P6]) *–在限定的时间范围内比较*

* 将过滤后的箱动态数据写入汇总表。参考SQL：

INSERT INTO CONTA\_MOVEMENT\_ALL(CONTA\_MOVEMENT\_SEQ,…, ISCOLLECT, DATASOURCE) SELECT CONTA\_MOVEMENT\_SEQ,…,0,1 FROM PRD\_CONTA\_MOVEMENT WHERE ISCOLLECT=-99

注：

CONTA\_MOVEMENT\_ALL.ISCOLLECT = 0 (未处理)

CONTA\_MOVEMENT\_ALL.DATASOURCE=1 (堆场分布式系统)

其他同名字段取值

* 更新采集到汇总表的中海箱动态数据状态。参考SQL：、

UPDATE PRD\_CONTA\_MOVEMENT SET ISCOLLECT=10 WHERE ISCOLLECT=-99

* 步骤1、2、3事务提交，异常，发送异常邮件，回滚，退出后台程序；提交成功继续后续步骤
* 后续逻辑沿用原有的后台逻辑，分发的箱动态数据由表CONTA\_MOVEMENT改为表CONTA\_MOVEMENT\_ALL

说明：

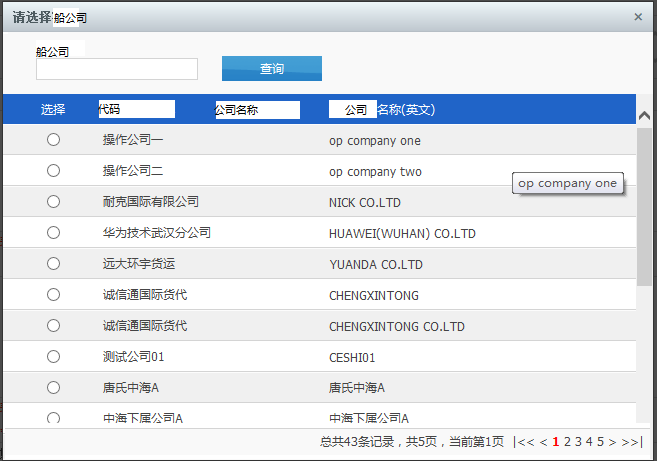
1. 归集、去重逻辑性能较差，注意建索引
2. 注意数据库事务的控制

### 数据测试用例

|  |  |
| --- | --- |
| 用例 | 预期结果 |
| 汇总数据的完整性 | 统计中海箱动态数据记录数[P1](表CONTA\_MOVEMENT WHERE ISCOLLECT=10 AND OWNER\_SEQ = {后台指定的船公司SEQ}) 统计堆场分布式箱动态记录数[P2](表PRD\_CONTA\_MOVEMENT WHERE ISCOLLECT=10 AND OWNER\_SEQ = {后台指定的船公司SEQ}) 统计汇总表记录数[P3](CONTA\_MOVEMENT\_ALL WHERE OWNER\_SEQ = {后台指定的船公司SEQ}) 预期结果：[P3]=[P1]+[P2] |
| 重复数据校验 | 表CONTA\_MOVEMENT中标记为丢弃的(ISCOLLECT=20)记录在表CONTA\_MOVEMENT\_ALL中能找到(WHERE OWNER\_SEQ = CONTA\_MOVEMENT.OWNER\_SEQ AND BOOKING\_NO = CONTA\_MOVEMENT.BOOKING\_NO AND CONTAINERNO = CONTA\_MOVEMENT.CONTAINERNO AND MOVEMENT\_CODE = CONTA\_MOVEMENT.MOVEMENT\_CODE); 表PRD\_CONTA\_MOVEMENT中标记为丢弃的(ISCOLLECT=20)记录在表CONTA\_MOVEMENT\_ALL中能找到(WHERE OWNER\_SEQ = PRD\_CONTA\_MOVEMENT.OWNER\_SEQ AND BOOKING\_NO = PRD\_CONTA\_MOVEMENT.BOOKING\_NO AND CONTAINERNO = PRD\_CONTA\_MOVEMENT.CONTAINERNO AND MOVEMENT\_CODE =PRD\_CONTA\_MOVEMENT.MOVEMENT\_CODE); |
| 汇总、去重执行完成 | 表CONTA\_MOVEMENT和表PRD\_CONTA\_MOVEMENT均不存在ISCOLLECT=0(未处理)记录 |
| 异常校验 | 程序运行过程中出现中断或出现异常，表CONTA\_MOVEMENT和表PRD\_CONTA\_MOVEMENT不存在ISCOLLECT=-99的记录 |

分发部分的测试用例沿用原有测试用例

## 船公司选择器

这个选择器的逻辑同时要实现一个自动联想的方法供其它页面使用。

|  |  |
| --- | --- |
| **页面元素** | **描述** |
| 查询条件 |  |
| 船公司 | 文本框支持模糊查询非必填可为空。不区分查询条件内容的大小写。 |
| 交互 |  |
| 加载过程 | 1. 参考以下默认语句：  |  | | --- | | SELECT C.\* FROM TCOMP\_GROUP T, TCOMP C WHERE T.GROUP\_COMP\_SEQ=0 AND T.COMP\_SEQ=C.TCOMP\_SEQ  AND (C.COMP\_NAMECN LIKE 'xxx' OR C.COMP\_NAMEEN LIKE 'xxx' OR C.COMP\_NAMESHORT\_PY = 'XXX')--查询条件  ORDER BY C.COMP\_NAMESHORT\_PY ASC, C.COMP\_NAMEEN ASC, C.TCOMP\_SEQ ASC |  1. 列表中的数据长度不允许超出一行的默认宽度，如出现数据换行的情况必须做内容截取，超出部分用“...”三个省略号替换，并增加浮动提示。   使用Table样式为“ class="mytable"”，每个单元格设定宽度并加上Title, 会自动据据单元格宽度出现“...”   1. 需要分页，每页显示10条数据。 |
| 数据返回 | 点击选择列，返回“船公司SEQ<TCOMP\_SEQ>，船公司代码<COMP\_NAMESHORT\_PY>，公司中文名称<COMP\_NAMECN>，公司英文名称<COMP\_NAMEEN>”。 |

## 航线港口选择器



|  |  |
| --- | --- |
| 页面元素 | 说明 |
| URL参数 |  |
| MODE | 取值为SINGLE=单选 MUTI=多选  参数为空，没有参数或者参数不合法的，默认为SINGLE  MODE=SINGLE时需隐藏页面素：选择返回按钮，已选择港口列表。 |
| SELECTED | 已选择的备案客户的SEQ列表，字符串格式，使用半角逗号间隔。仅用于在MUTI模式，在MUTI模式下用于在选择区加载已经选择的备案客户信息。  SELECT B.\*, A.ROUTE\_AREA\_NAME FROM ROUTE\_AREA A INNER JOIN ROUTE\_AREA\_PORT B ON A.ROUTE\_AREA = B.ROUTE\_AREA WHERE A.OWNER\_SEQ = 37 AND B.PORT\_CODE IN({SELECTED}) |
| 查询条件 |  |
| 航线区域 | 精确查询，下拉初始化数据为当前船公司的所有有效航线区域。[同航线区域管理的查询条件](#_航线区域查询) |
| 港口代码 | 模糊查询，将输入的内容转大写去数据库查询 |
| 港口名称 | 模糊查询，将输入的内容转大写去数据库查询 |
| 列表 |  |
| 航线区域代码 | ROUTE\_AREA\_CODE<航线区域代码> |
| 航线区域 | ROUTE\_AREA\_NAME<航线区域名称> |
| 港口代码 | PORT\_CODE<港口编码> |
| 港口名称 | PORT\_NAME\_EN<港口英文名称> |
| 交互 |  |
| 选择结果 | 如果参数MODE = SINGLE，隐藏“选择结果”  选择的记录后面增加“删除”链接，点击后将对应内容从选择结果中移除  如果参数SELECTED中有值的，按半角逗号拆分后加载到选择结果中，需要过滤重复数据 |
| 查询 | SELECT B.\*, A.ROUTE\_AREA\_NAME FROM ROUTE\_AREA A INNER JOIN ROUTE\_AREA\_PORT B ON A.ROUTE\_AREA = B.ROUTE\_AREA WHERE A.OWNER\_SEQ = 37 AND A.ROUTE\_AREA\_CODE = '航线区域' AND B.PORT\_CODE = '港口代码' AND (B.PORT\_NAME\_CN = '港口名称' OR B.PORT\_NAME\_EN = '港口名称')--查询条件  ORDER BY PORT\_CODE ASC, ROUTE\_AREA\_PORT\_SEQ  注意：  该结果集要使用缓存。 |
| 选择 | MODE=SINGLE时：  选择后将记录带回父页面，目前的方法是：  setPort(code, nameCn, nameEn)  MODE=MUTI时：  将该记录加入SELECTED变量，刷新已选择列表（SELECT B.\*, A.ROUTE\_AREA\_NAME FROM ROUTE\_AREA A INNER JOIN ROUTE\_AREA\_PORT B ON A.ROUTE\_AREA = B.ROUTE\_AREA WHERE A.OWNER\_SEQ = 37 AND B.PORT\_CODE IN({SELECTED})）。 |
| 选择返回 | 调用父页面的clearCompName方法清空已选择的值。  再遍历“已选择”列表循环调用  setPort(code, nameCn, nameEn)  将“已选择”列表的值全部带回 |
|  |  |

实现时可参考FAK新增页面的“收货地”选择器。（app/PortMulti?selectedPorts=）

# 运行环境配置

## 报文模板配置样例

|  |
| --- |
| 报文模板 汤文波 根据报文规范整理，已失效  UNA:+.? '  UNB+UNOC:2+SZEDI:ZZZ+{Recipient}:ZZZ++{YYMMDD}:{HHMI}+{MsgId}'  UNH+{MsgId}+IFTMBF:D:99B:UN'  BGM+335+{BookNo}+{FunctionCode}'  CTA+IC+{ContactID}:{ContactInfo}'  COM+{Tel}:TE'  COM+{MAIL}:EM'  DTM+137:{MSGDate\_203}:203'  TSR+{Condition\_coded}+2'  FTX+AAI+++{Remark}'  RFF+CT:{CT\_ContractNo}'  RFF+SI:{SI\_ContractNo}'  {TCC}  TDT+20+{Voyage}+1++{Carrier}:172+++:::{Vessel}'  LOC+9+{LoadPort}:139:6:{LoadPortLocation}'  LOC+88+{ReceiptPlace}:139:6:{ReceiptPlaceLocation}'  LOC+11+{UnLoadPort}:139:6:{UnLoadPortLocation}'  LOC+7+{Delivery}:139:6:{DeliveryLocation}'  NAD+CA+{Carrier}:160:86++{CarrierName}'  NAD+FW+{Freight}:160:86++{FreightName}'  CTA+IC+{SalesID}:{SalesName}'  COM+{SalesTel}:TE'  COM+{SalesMAIL}:EM'  NAD+ZZZ+{Freight}:160:86++{FreightName}'  CTA+IC+:{ContactInfo}'  COM+{Tel}:TE'  COM+{MAIL}:EM'  NAD+CZ+{BuyID}:160:86++{Consignor}'  CPI+1+{C\_PayCode}+{C\_PayType}'  NAD+CN+++{Consignee}'  GID+{GoodsSeq}+{GoodsCount}::::PCS'  FTX+AAA+++{GoodsName}'  MEA+AAE+WT+KGM:{GoodsWeight}'  {EQD}  UNT+{LINES}+{MsgId}'  UNZ+1+{ROWS}+{MsgId}'  TCC+{TCC\_SIZETYPE}'  PRI+{Currency\_Type}:{PRICE}:{Price\_Type}'  EQN+{TCC\_ContaQuantity}'  MOA+{Currency\_Type}:{AmountValue}'  EQD+CN++{SizeType}'  EQN+{ContaQuantity}' |

|  |
| --- |
| CSCL订舱报文模板样例 吴华提供，1.3 发布版本    UNA:+.? '  UNB+UNOC:2+SZEDI:ZZZ+{Recipient}:ZZZ+{YYMMDD}:{HHMI}+{MsgId}'  UNH+{MsgId}+IFTMBF:D:99B:UN'  BGM+335+{MsgId}+{FunctionCode}'  CTA+IC+{ContactID}:{ContactInfo}'  COM+{Tel}:TE'  COM+{MAIL}:EM'  DTM+137:{YYMMDD}{HHMI}:201'  DTM+36:{Expiry\_DateTime201}:201'  TSR+{Condition\_coded}+{DISCHARGE\_TRAN\_WAY}'  FTX+AAI+++{Remark}'  RFF+CT:{CT\_ContractNo}'  RFF+BN:{BookNo}'  MOA+{Currency\_Type}:{AmountValue}'  {TCC}  TDT+20+{Voyage}+1++{Carrier}:172+++{SHIP\_CODE}:::{Vessel}'  LOC+9+{LoadPort}:139:6:{LoadPortLocation}'  LOC+88+{ReceiptPlace}:139:6:{ReceiptPlaceLocation}'  LOC+11+{UnLoadPort}:139:6:{UnLoadPortLocation}'  LOC+13+{TransPort}:139:6:{TransPortLocation}'  LOC+7+{Delivery}:139:6:{DeliveryLocation}'  NAD+CA+{SHIP\_OWNER}:160:86++{SHIP\_OWNER}'  NAD+FW+{BuyID}:160:86++{FreightName}'  CTA+IC+{SalesID}:{SalesName}'  COM+{SalesTel}:TE'  COM+{SalesMAIL}:EM'  NAD+CZ+++{Consignor}'  NAD+CN+++{Consignee}'  CPI+707+{C\_PayCode}+{C\_PayType}'  GID+1'  FTX+AAA+++{GoodsName}'  {HAN}  {TMD}'  {EQD}'  UNT+{LINES}+{MsgId}'  UNZ+{ROWS}+{MsgId}'    {TCC}部分模版  TCC+{TCC\_SIZETYPE}'  PRI+{Currency\_Type}:{PRICE}'  EQN+{TCC\_ContaQuantity}'  MOA+{Currency\_Type}:{AmountValue}:{Price\_Type}'    {EQD}部分模版  EQD+CN+{CONTA\_NO}+{SizeType}+{IS\_SOC}+E+F'  EQN+{ContaQuantity}' |

## 报文模板输出占位符定义

|  |
| --- |
| <?xml version="1.0" encoding="utf-8" ?>  <ELEMENTS>  <OUTITEM KEY="{BookNo}" NAME="订舱号" COL="BOOKING\_NO" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{MSGDate\_101}" NAME="报文时间" COL="CREATE\_DATE" FORMAT="OrderExportFormatDate101" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{MSGDate\_203}" NAME="报文时间" COL="CREATE\_DATE" FORMAT="OrderExportFormatDate203" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Condition\_coded}" NAME="运输要求" COL="TRANS\_TYPE\_CODE" FORMAT="OrderExportFormatTransType" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{CT\_ContractNo}" NAME="CT合约号" COL="CONTRACT\_NO" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{SI\_ContractNo}" NAME="SI合约号" COL="CONTRACT\_NO" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Vessel}" NAME="船名" COL="SHIP\_NAME\_EN" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Voyage}" NAME="航次" COL="VOYAGE" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{LoadPort}" NAME="装货港代码" COL="LOADING\_PORT" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{LoadPortLocation}" NAME="装货港地址" COL="LOADING\_PORT\_NAME" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{ReceiptPlace}" NAME="收货地代码" COL="LOADING\_LOACTION" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{ReceiptPlaceLocation}" NAME="收货地地址" COL="LOADING\_LOACTION\_NAME" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{UnLoadPort}" NAME="卸货港代码" COL="DESTINATION\_PORT" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{UnLoadPortLocation}" NAME="卸货港地址" COL="DESTINATION\_PORT\_NAME" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Delivery}" NAME="交货地代码" COL="DESTINATION\_LOCATION" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{DeliveryLocation}" NAME="交货地地址" COL="DESTINATION\_LOCATION\_NAME" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Carrier}" NAME="承运人" COL="OWNER\_CODE" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{CarrierName}" NAME="承运人名称" COL="OWNER\_CODE" FORMAT="" NULL="" DATASOURCE ="head"/>    <OUTITEM KEY="{Freight}" NAME="代运人简称" COL=" CUSTOMER\_SEQ" FORMAT="" NULL="" DATASOURCE ="head"/>    <OUTITEM KEY="{Consignor}" NAME="货主" COL="COMP\_SEND" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{GoodsSeq}" NAME="货物序号" COL="CARGO\_DESC" FORMAT="OrderExportFormatGId" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{GoodsCount}" NAME="货物数量" COL="PICS" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{GoodsName}" NAME="货物名称" COL="CARGO\_DESC" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{GoodsWeight}" NAME="货物重量" COL="WEIGHT" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{SizeType}" NAME="箱型尺寸" COL="CONTA\_SIZE" FORMAT="" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{ContaQuantity}" NAME="箱数量" COL="CONTA\_SIZE" FORMAT="OrderExportFormatContaNoIsSoc" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{TCC\_SIZETYPE}" NAME="计价的箱型尺寸" COL="CONTA\_SIZE" FORMAT="" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{Currency\_Type}" NAME="计价的币种" COL="PRICE\_UNIT" FORMAT="" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{PRICE}" NAME="单价" COL="PRICE\_TRANS" FORMAT="OrderExportFormatMoney" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{Price\_Type}" NAME="计价类型" COL="BUSINESS\_TYPE" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{ContaQuantity}" NAME="箱数量" COL="CONTA\_SIZE" FORMAT="OrderExportFormatContaNoIsSoc" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{TCC\_ContaQuantity}" NAME="计价箱数量" COL="CONTA\_SIZE" FORMAT="OrderExportFormatConta" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{AmountValue}" NAME="TCC段计费箱型总价" COL="CONTA\_SIZE" FORMAT="OrderExportFormatContaAmount, OrderExportFormatMoney" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{C\_PayType}" NAME="付款方式" COL="TRANS\_TYPE\_CODE" FORMAT="" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{C\_PayCode}" NAME="付款地" COL="PAY\_LOCATION" FORMAT="" NULL="" DATASOURCE ="head"/>  <!--1.3C需求版本中修改的配置-->  <OUTITEM KEY="{Remark}" NAME="备注" COL="ORDER\_REMARK" FORMAT="OrderExportCutChar20,OrderExportSpecialChar" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{BuyID}" NAME="订舱人代码" COL="CUSTOMER\_SEQ" FORMAT="OrderExportFormaCustomer" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{FreightName}" NAME="代运人名称" COL="CUSTOMER\_NAME\_EN" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{ContactID}" NAME="联系人ID" COL="APPROVER" FORMAT="OrderExportFormatCheckUser" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{ContactInfo}" NAME="联系人或部门" COL="APPROVER" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Tel}" NAME="联系电话" COL="APPROVER" FORMAT="OrderExportFormatCheckUserTel" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{MAIL}" NAME="联系邮箱" COL="APPROVER" FORMAT="OrderExportFormatCheckUserMail" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{SalesID}" NAME="销售员代码" COL="SALE\_USER\_ACCOUNT" FORMAT="OrderExportFormatSalesEDICode" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{SalesName}" NAME="销售员名称" COL="SALE\_USER\_ACCOUNT" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{SalesTel}" NAME="销售员电话" COL="SALE\_USER\_ACCOUNT" FORMAT="OrderExportFormatSalesPhone" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{SalesMAIL}" NAME="销售员邮箱" COL="SALE\_USER\_ACCOUNT" FORMAT="OrderExportFormatSalesEmail" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Consignee}" NAME="货代" COL="COMP\_GET" FORMAT="" NULL="" DATASOURCE ="head"/>  <!--1.3C需求版本中新增的配置-->  <OUTITEM KEY="{Expiry\_DateTime101}" NAME="截重时间" COL="EXPIRY\_DATETIME" FORMAT="OrderExportFormatDate101" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Expiry\_DateTime201}" NAME="截重时间" COL="EXPIRY\_DATETIME" FORMAT="OrderExportFormatDate201" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{Expiry\_DateTime203}" NAME="截重时间" COL="EXPIRY\_DATETIME" FORMAT="OrderExportFormatDate203" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{DISCHARGE\_TRAN\_WAY}" NAME="二程运输方式" COL="DISCHARGE\_TRAN\_WAY" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{SHIP\_CODE}" NAME="船舶代码" COL="SHIP\_CODE" FORMAT="" NULL="" DATASOURCE ="head"/>  <OUTITEM KEY="{TransPort}" NAME="中转港代码" COL="TRANSFER\_PORT" FORMAT="OrderExportGetTransPort" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{TransPortLocation}" NAME="中转港名称" COL="TRANSPORT\_NAME\_EN" FORMAT="OrderExportGetTransPortLocation" NULL="" DATASOURCE ="head"/>    <OUTITEM KEY="{CONTA\_NO}" NAME="箱号" COL="CONTA\_NO" FORMAT="" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{IS\_SOC}" NAME="自用柜标记" COL="IS\_SOC" FORMAT="OrderExportFormatSOC" NULL="" DATASOURCE ="body"/>  <OUTITEM KEY="{HAN}" NAME="特殊装载" COL="SPECIAL\_LOAD" FORMAT="OrderExportFormatHAN130" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{TMD}" NAME="特殊备注" COL="SPECIAL\_REMARK" FORMAT="OrderExportFormatTMD" NULL="TRUE" DATASOURCE ="head"/>  <OUTITEM KEY="{SHIP\_OWNER}" NAME="船东代码" COL="SHIP\_CODE" FORMAT="OrderExportFormatShipOwner" NULL="" DATASOURCE ="head"/>    <!--  Format 报文输出内容处理接口，对应实现类功能概要说明  OrderExportFormatDate203 将参数data转换成日期，并输出成报文的203标准格式  OrderExportFormatDate101 将参数data转换成日期，并输出成报文的101标准格式  OrderExportRemarkCSCL 获取到的值在结尾追加?:FAX NO + LINKER\_FAX（从parm中获取）  OrderExportChar512 传入的字符串按512长度切开，用半角的“：”间隔。  OrderExportFormatMoney 格式化金额输出，格式化为0.00格式  OrderExportFormatContaAmount 传入箱型，计算对应箱型的 金额= 箱数量\*金额  OrderExportSpecialChar 通配符转换，报文中的保留字按照报文规范参考原有代码输出  OrderExportFormatContaNoIsSoc 传入箱型，计算对应详细的箱数量,忽略IS\_DOC的箱  OrderExportFormatGId 固定返回1即可，业务含义是本次仅实现一货多箱的订单  OrderExportFormatConta 传入箱型，统计箱数量。根据data参数在sourceParm：orderDetail中统计对应箱型的数量，对应“CONTA\_SIZE<标准箱型>”  OrderExportFormatTransType 中海“运输要求”代码转换，select t.dict\_ext\_field3, t.\* from dict\_code t where t.owner\_seq = 37 and t.status = '10' and t.dict\_type ='TRAN\_CLAUSE' and dict\_code = '13'    （2013-11-26 P9新增）  OrderExportCutChar20 20英文长度自动截断的处理，不考虑单词的完整性。  OrderExportFormatDate201 将参数data转换成日期，并输出成报文的201标准格式  OrderExportGetTransPort 判定是否中转“IS\_TRANSFER<是否中转>”，是的话输出传入的港口代码  OrderExportGetTransPortLocation 判定是否中转“IS\_TRANSFER<是否中转>”，是的话输出传入的港口名称  OrderExportFormatSOC 转换输出SOC标记，1=SCO 其他=CSC  OrderExportFormatHAN130 拆分多个标记，输出“HAN+OD:130:ZZZ'<换行>”格式  OrderExportFormatTMD 拆分多个标记，输出“TMD+C'<换行>”格式  OrderExportFormatCheckUserTel 获取审核人的电话，根据传入的“BUSINESS\_ORDER：APPROVER<放舱人>”从“TUSERATTR<用户附件信息表>”中获取“TELEPHONE<电话>”  OrderExportFormatCheckUserMail 获取审核人的邮箱，根据传入的“BUSINESS\_ORDER：APPROVER<放舱人>”从“TUSERATTR<用户附件信息表>”中获取“EMAIL<邮箱>”  OrderExportFormatSalesEDICode 根据销售的账户获取对应的EDI编码，根据传入的“SALE\_USER\_ACCOUNT<航线销售员>”从“TUSERATTR<用户附件信息表>”中获取“USER\_CODE<用户代码>”  OrderExportFormatShipOwner 根据船信息获取船东，读取“SHIP\_INFO<船信息管理>”中的“SHIP\_OWNER<船东>”字段返回    （2013-11-26 P9逻辑重构）  OrderExportFormaCustomer 获取订舱公司的编码，根据订舱公司的SEQ从CUSTOMERS<客户备案>获取“EDI\_CODE”。  OrderExportFormatCheckUser 获取审核人的EDI编码，根据传入的“BUSINESS\_ORDER：APPROVER<放舱人>”从“TUSERATTR<用户附件信息表>”中获取“USER\_CODE<用户代码>”  OrderExportFormatSalesPhone 根据销售的账户获取对应的电话，根据传入的“SALE\_USER\_ACCOUNT<航线销售员>”从“TUSERATTR<用户附件信息表>”中获取“TELEPHONE<电话>”  OrderExportFormatSalesEmail 根据销售的账户获取对应的邮箱，根据传入的“SALE\_USER\_ACCOUNT<航线销售员>”从“TUSERATTR<用户附件信息表>”中获取“EMAIL<邮箱>”  -->    <!--  以下参数使用指令表本身的数据替换，在EXE循环内部，不在接口内部处理/>  -->  <!--OUTITEM KEY="{YYMMDD}" NAME="当前日期" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{HHMI}" NAME="当前时间" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{FunctionCode}" NAME="取指令表MSG\_COMMAND.COMMAND\_TYPE（1-撤销 4-修改9-新增）" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{MsgId}" NAME="文件参考号" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{LINES}" NAME="报文行数" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{ROWS}" NAME="单数" COL="" FORMAT="" NULL=""/>-->  <!--OUTITEM KEY="{Recipient}" NAME="接收方" COL="" FORMAT="" NULL=""/>-->    <!--  CSCL订舱报文模板样例    UNA:+.? '  UNB+UNOC:2+SZEDI:ZZZ+{Recipient}:ZZZ+{YYMMDD}:{HHMI}+{MsgId}'  UNH+{MsgId}+IFTMBF:D:99B:UN'  BGM+335+{MsgId}+{FunctionCode}'  CTA+IC+{ContactID}:{ContactInfo}'  COM+{Tel}:TE'  COM+{MAIL}:EM'  DTM+137:{YYMMDD}{HHMI}:201'  DTM+36:{Expiry\_DateTime201}:201'  TSR+{Condition\_coded}+{DISCHARGE\_TRAN\_WAY}'  FTX+AAI+++{Remark}'  RFF+CT:{CT\_ContractNo}'  RFF+BN:{BookNo}'  MOA+{Currency\_Type}:{AmountValue}'  {TCC}  TDT+20+{Voyage}+1++{Carrier}:172+++{SHIP\_CODE}:::{Vessel}'  LOC+9+{LoadPort}:139:6:{LoadPortLocation}'  LOC+88+{ReceiptPlace}:139:6:{ReceiptPlaceLocation}'  LOC+11+{UnLoadPort}:139:6:{UnLoadPortLocation}'  LOC+13+{TransPort}:139:6:{TransPortLocation}'  LOC+7+{Delivery}:139:6:{DeliveryLocation}'  NAD+CA+{SHIP\_OWNER}:160:86++{SHIP\_OWNER}'  NAD+FW+{BuyID}:160:86++{FreightName}'  CTA+IC+{SalesID}:{SalesName}'  COM+{SalesTel}:TE'  COM+{SalesMAIL}:EM'  NAD+CZ+++{Consignor}'  NAD+CN+++{Consignee}'  CPI+707+{C\_PayCode}+{C\_PayType}'  GID+1'  FTX+AAA+++{GoodsName}'  {HAN}  {TMD}'  {EQD}'  UNT+{LINES}+{MsgId}'  UNZ+{ROWS}+{MsgId}'    {TCC}部分模版  TCC+{TCC\_SIZETYPE}'  PRI+{Currency\_Type}:{PRICE}'  EQN+{TCC\_ContaQuantity}'  MOA+{Currency\_Type}:{AmountValue}:{Price\_Type}'    {EQD}部分模版  EQD+CN+{CONTA\_NO}+{SizeType}+{IS\_SOC}+E+F'  EQN+{ContaQuantity}'  -->  </ELEMENTS> |

## 数据权限配置文件