**HDI一期详细设计说明书V1.0**

**广东以大供应链管理有限公司**

**编写人：罗荣熙**

**2019.04.22**

修改记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本号 | 内容 | 修改人 | 审批人 | 发布日期 |
| 1 | V1.0 | 创建：添加供应商信息设计 | 罗荣熙 | 罗然熹 | 2019.04.22 |
| 2 | V1.1 | 修改：添加厂商信息设计 | 罗荣熙 | 罗然熹 | 2019.04.24 |
| 3 | V1.2 | 修改：添加字典查询设计 | 罗荣熙 | 罗然熹 | 2019.04.26 |
| 4 | V1.3 | 修改：添加系统消息设计 | 罗荣熙 | 罗然熹 | 2019.04.28 |
| 5 | V1.4 | 修改：添加代理商信息设计 | 罗荣熙 | 罗然熹 | 2019.04.29 |
| 6 | V1.5 | 修改：添加平台药品、供应商药品、医院药品设计 | 罗荣熙 | 罗然熹 | 2019.05.09 |
| 7 | V1.6 | 修改：添加平台试剂、供应商试剂、医院试剂设计 | 罗荣熙 | 罗然熹 | 2019.05.13 |
| 8 | V1.7 | 修改：添加平台耗材、供应商耗材、医院耗材设计 | 罗荣熙 | 罗然熹 | 2019.05.14 |
| 9 | V1.8 | 修改：添加证照管理（供应商、商品、厂商、代理商、预警）设计 | 罗荣熙 | 罗然熹 | 2019.05.24 |
| 10 | V1.9 | 修改：添加退货管理（退货申请单、退货单）设计 | 罗荣熙 | 罗然熹 | 2019.06.06 |

目录

[1. 引言](#_Toc31456)

[1.1编写的目的](#_Toc3569)

[1.2术语表](#_Toc22688)

[1.3参考资料](#_Toc23204)

[2. 工具与规范](#_Toc9557)

[2.1开发工具](#_Toc11835)

[2.2代码规范](#_Toc1805)

[3. 平台结构图](#_Toc22198)

[4. 数据结构说明](#_Toc31563)

[4.1机构管理](#_Toc11530)

[4.1.1表清单](#_Toc27334)

[4.1.2表结构](#_Toc25690)

[4.2统一编码管理](#_Toc7215)

[4.2.1表清单](#_Toc4143)

[4.2.2表结构](#_Toc1572)

[4.3商品管理](#_Toc280)

[4.3.1表清单](#_Toc16822)

[4.3.2表结构](#_Toc21399)

[4.4证照管理](#_Toc10803)

[4.4.1表清单](#_Toc11090)

[4.4.2表结构](#_Toc23562)

[4.5采购管理](#_Toc9432)

[4.5.1表清单](#_Toc5121)

[4.5.2表结构](#_Toc6778)

[4.6供货管理](#_Toc14630)

[4.6.1表清单](#_Toc32010)

[4.6.2表结构](#_Toc28437)

[4.7退货管理](#_Toc18306)

[4.7.1表清单](#_Toc15235)

[4.7.2表结构](#_Toc3288)

[4.8系统消息](#_Toc24893)

[4.8.1表清单](#_Toc22931)

[4.8.2表结构](#_Toc25588)

[4.9视图表](#_Toc11990)

[4.9.1表清单](#_Toc909)

[4.9.2表结构](#_Toc19201)

[5. 模块设计](#_Toc13006)

[5.1机构管理](#_Toc21526)

[5.1.1供应商信息](#_Toc27812)

[5.1.2医院信息](#_Toc23167)

[5.1.3厂商信息（平台）](#_Toc9293)

[5.1.4代理商信息](#_Toc7926)

[5.2统一编码管理](#_Toc12337)

[5.2.1用例图](#_Toc7311)

[5.2.1功能设计说明](#_Toc19507)

[5.3商品管理](#_Toc8251)

[5.3.1平台药品信息](#_Toc30646)

[5.3.2供应商药品信息](#_Toc16701)

[5.3.3医院药品信息](#_Toc21682)

[5.3.4平台试剂信息](#_Toc27236)

[5.3.5供应商试剂信息](#_Toc6284)

[5.3.6医院试剂信息](#_Toc25101)

[5.3.7平台耗材信息](#_Toc25740)

[5.3.8供应商耗材信息](#_Toc28472)

[5.3.9医院耗材信息](#_Toc19240)

[5.4证照管理](#_Toc27698)

[5.4.1证照分类信息](#_Toc12409)

[5.4.2供应商证照信息](#_Toc23458)

[5.4.3商品证照信息](#_Toc26524)

[5.4.4厂商证照信息](#_Toc31488)

[5.4.5代理商证照信息](#_Toc15305)

[5.4.6证照预警信息](#_Toc26046)

[5.5采购管理](#_Toc19296)

[5.5.1用例图](#_Toc13988)

[5.5.2功能设计说明](#_Toc3146)

[5.6供货管理](#_Toc16692)

[5.6.1用例图](#_Toc19539)

[5.6.2功能设计说明](#_Toc2835)

[5.7退货管理](#_Toc21868)

[5.7.1退货申请单管理](#_Toc11717)

[5.7.2退货单管理](#_Toc23805)

[5.8公共模块](#_Toc19264)

[5.8.1字典查询设计](#_Toc24227)

[5.8.2枚举查询设计](#_Toc21539)

[5.8.3导入功能设计](#_Toc17943)

[5.8.4导出功能设计](#_Toc5266)

[5.9系统消息](#_Toc10509)

[5.9.1系统消息管理](#_Toc25862)

[5.9.2用户消息管理](#_Toc6477)

# 引言

## 1.1编写的目的

本详细设计说明书编写的目的是说明程序模块的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 1.2术语表

|  |  |  |
| --- | --- | --- |
| 序号 | 术语或缩略语 | 说明行定义 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 1.3参考资料

|  |  |  |  |
| --- | --- | --- | --- |
| 资料名称 | 作者 | 文件编号、版本 | 资料存放地址 |
| HDI总体设计方案 | 罗荣熙 | V1.0 | svn://192.168.1.12/develop/hscm/hdi/trunk/hdi/hdi-document/技术方案 |
| HDI一期技术设计方案 | 罗荣熙 | V1.1 | svn://192.168.1.12/develop/hscm/hdi/trunk/hdi/hdi-document/技术方案 |
| HDI一期解决方案 | 郑丽君 | V1.3 | svn://192.168.1.12/develop/hscm/hdi/trunk/hdi/hdi-document/需求分析 |
| HDI一期研发工作计划 | 罗荣熙 | V1.0 | svn://192.168.1.12/develop/hscm/hdi/trunk/hdi/hdi-document/工作计划 |
| HDI库表设计 | 罗荣熙 |  | svn://192.168.1.12/develop/hscm/hdi/trunk/hdi/hdi-document/库表设计 |

# 工具与规范

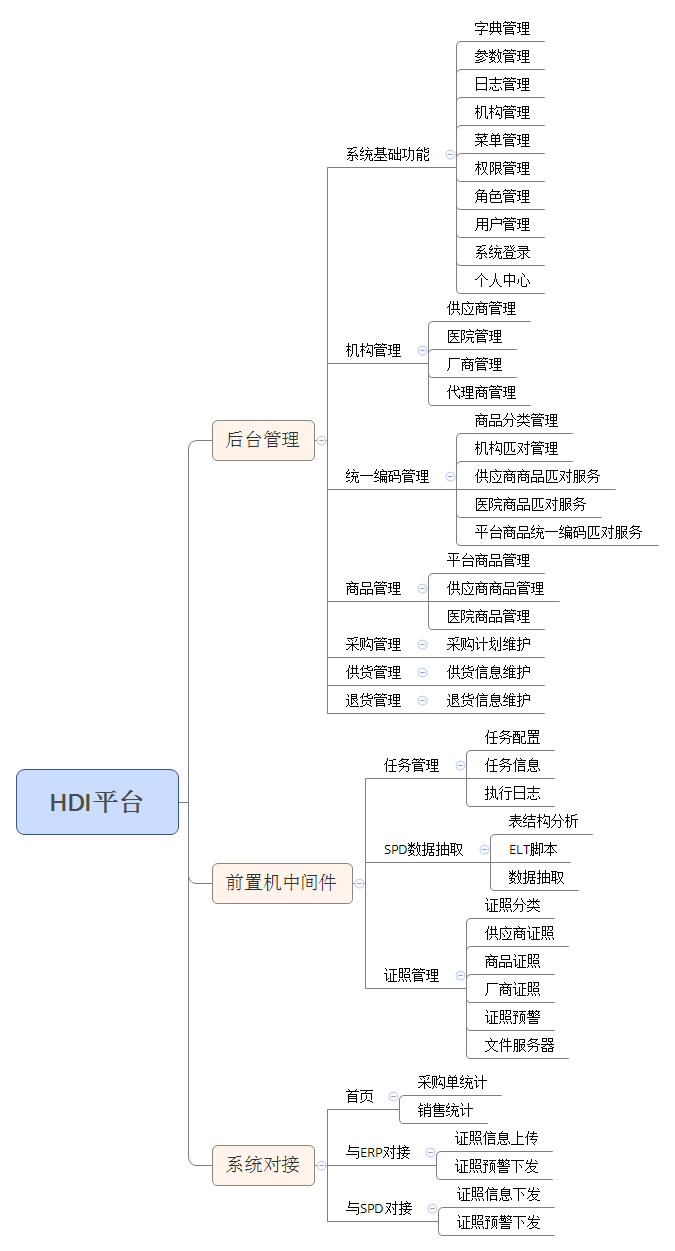
## 2.1开发工具

|  |  |  |
| --- | --- | --- |
| 工具 | 版本 | 集成插件 |
| jdk | 1.8以上 |  |
| eclipse | 4.6.0以上 | Lombok |
| spring tool suite |
| mysql | 5.5.38以上 |  |
| maven | 3.3.3以上 |  |
| powerdesigner | 16以上 |  |

## 2.2代码规范

|  |  |  |  |
| --- | --- | --- | --- |
| 规范 | 内容 | 样例 | 说明 |
| 库表设计 | 表名 | hdi\_org\_supplier\_info | hdi为产品名；org为功能模块名；supplier\_info为具表存储内容；中间用下滑杠隔开，目的是提高可读性。 |
| 字段 | supplier\_code | 中间用下滑杠隔开，目的是提高可读性。 |
| 编码规范 | 代码编写 | supplierCode | 采用驼峰型，目的是提高可读性。 |
| 插入 | insert | 单条记录保存操作 |
| 批量插入 | insertBatch | 批量保存操作 |
| 删除 | deleteBy... | 根据条件进行删除 |
| 更新 | updateBy... | 根据条件进行更新 |
| 条件查询 | selectBy... | 更加条件进行查询 |
| 分页查询 | queryPage | 分页条件进行查询 |
| 状态码 | 系统异常 | -1 |  |
| 请求成功 | 0 |  |
| 请求失败 | 500 |  |

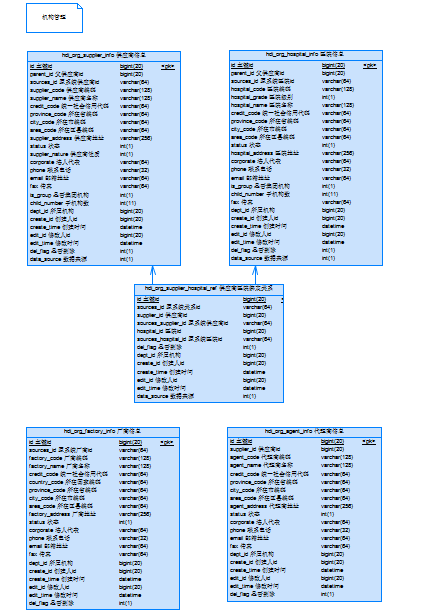
# 平台结构图



# 数据结构说明

包括数据结构名称，功能说明，具体数据结构说明（定义、注释、取值）等。

## 4.1机构管理



### 4.1.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| hdi\_org\_supplier\_info | 供应商信息 |
| hdi\_org\_hospital\_info | 医院信息 |
| hdi\_org\_supplier\_hospital\_ref | 供应商医院绑定关系 |
| hdi\_org\_factory\_info | 厂商信息（平台） |
| hdi\_org\_agent\_info | 代理商信息 |

### 4.1.2表结构

#### 4.1.2.1表hdi\_org\_supplier\_info供应商信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| parent\_id 父供应商id | parent\_id | 父供应商id | bigint(20) |
| sources\_id 原系统供应商id | sources\_id | 原系统供应商id | varchar(64) |
| supplier\_code 供应商编码 | supplier\_code | 供应商编码 | varchar(128) |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 | varchar(128) |
| credit\_code 统一社会信用代码 | credit\_code | 统一社会信用代码 | varchar(64) |
| province\_code 所在省编码 | province\_code | 所在省编码 | varchar(64) |
| city\_code 所在市编码 | city\_code | 所在市编码 | varchar(64) |
| area\_code 所在区县编码 | area\_code | 所在区县编码 | varchar(64) |
| supplier\_address 供应商地址 | supplier\_address | 供应商地址 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| supplier\_nature 供应商性质 | supplier\_nature | 供应商性质(0:国营企业;1:民营企业;2:中外合资企业;3:股份制企业;4:个体企业) | int(1) |
| corporate 法人代表 | corporate | 法人代表 | varchar(64) |
| phone 联系电话 | phone | 联系电话 | varchar(32) |
| email 邮箱地址 | email | 邮箱地址 | varchar(64) |
| fax 传真 | fax | 传真 | varchar(64) |
| is\_group 是否集团机构 | is\_group | 是否集团机构(0:否;1:是) | int(1) |
| child\_number 子机构数 | child\_number | 子机构数(集团供应商允许创建的子供应商数) | int(11) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.1.2.2表hdi\_org\_hospital\_info医院信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| parent\_id 父供应商id | parent\_id | 父供应商id | bigint(20) |
| sources\_id 原系统医院id | sources\_id | 原系统医院id | varchar(64) |
| hospital\_code 医院编码 | hospital\_code | 医院编码 | varchar(128) |
| hospital\_grade 医院级别 | hospital\_grade | 医院级别(0:三级特等;1:三级甲等;2:三级乙等;3:三级丙等;4:二级甲等;5:二级乙等;6:二级丙等;7:一级甲等;8:一级乙等;9:一级丙等) | int(1) |
| hospital\_name 医院名称 | hospital\_name | 医院名称 | varchar(128) |
| credit\_code 统一社会信用代码 | credit\_code | 统一社会信用代码 | varchar(64) |
| province\_code 所在省编码 | province\_code | 所在省编码 | varchar(64) |
| city\_code 所在市编码 | city\_code | 所在市编码 | varchar(64) |
| area\_code 所在区县编码 | area\_code | 所在区县编码 | varchar(64) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| hospital\_address 医院地址 | hospital\_address | 医院地址 | varchar(256) |
| corporate 法人代表 | corporate | 法人代表 | varchar(64) |
| phone 联系电话 | phone | 联系电话 | varchar(32) |
| email 邮箱地址 | email | 邮箱地址 | varchar(64) |
| is\_group 是否集团机构 | is\_group | 是否集团机构(0:否;1:是) | int(1) |
| child\_number 子机构数 | child\_number | 子机构数(集团医院允许创建的子医院数) | int(11) |
| fax 传真 | fax | 传真 | varchar(64) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.1.2.3表hdi\_org\_supplier\_hospital\_ref供应商医院绑定关系

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统关系id | sources\_id | 原系统关系id | varchar(64) |
| supplier\_id 供应商id | supplier\_code | 供应商id | bigint(20) |
| sources\_supplier\_id 原系统供应商id | sources\_supplier\_id | 原系统供应商id | varchar(64) |
| hospital\_id 医院id | hospital\_code | 医院id | bigint(20) |
| sources\_hospital\_id 原系统医院id | sources\_hospital\_id | 原系统医院id | varchar(64) |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

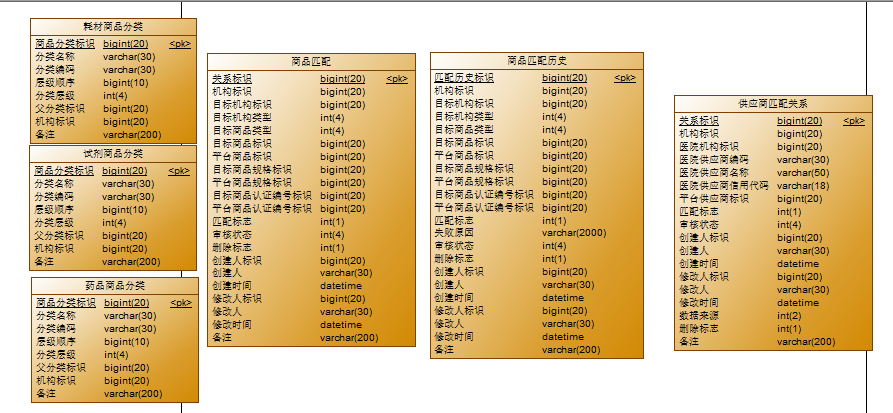
#### 4.1.2.4表hdi\_org\_factory\_info厂商信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统厂商id | sources\_id | 原系统厂商id | varchar(64) |
| factory\_code 厂商编码 | factory\_code | 厂商编码 | varchar(128) |
| factory\_name 厂商名称 | factory\_name | 厂商名称 | varchar(128) |
| credit\_code 统一社会信用代码 | credit\_code | 统一社会信用代码 | varchar(64) |
| province\_code 所在省编码 | province\_code | 所在省编码 | varchar(64) |
| city\_code 所在市编码 | city\_code | 所在市编码 | varchar(64) |
| area\_code 所在区县编码 | area\_code | 所在区县编码 | varchar(64) |
| factory\_address 厂商地址 | factory\_address | 厂商地址 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| corporate 法人代表 | corporate | 法人代表 | varchar(64) |
| phone 联系电话 | phone | 联系电话 | varchar(32) |
| email 邮箱地址 | email | 邮箱地址 | varchar(64) |
| fax 传真 | fax | 传真 | varchar(64) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.1.2.5表hdi\_org\_agent\_info代理商信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| agent\_code 代理商编码 | agent\_code | 代理商编码 | varchar(128) |
| agent\_name 代理商名称 | agent\_name | 代理商名称 | varchar(128) |
| credit\_code 统一社会信用代码 | credit\_code | 统一社会信用代码 | varchar(64) |
| province\_code 所在省编码 | province\_code | 所在省编码 | varchar(64) |
| city\_code 所在市编码 | city\_code | 所在市编码 | varchar(64) |
| area\_code 所在区县编码 | area\_code | 所在区县编码 | varchar(64) |
| agent\_address 代理商地址 | agent\_address | 代理商地址 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| corporate 法人代表 | corporate | 法人代表 | varchar(64) |
| phone 联系电话 | phone | 联系电话 | varchar(32) |
| email 邮箱地址 | email | 邮箱地址 | varchar(64) |
| fax 传真 | fax | 传真 | varchar(64) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

## 4.2统一编码管理



### 4.2.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| hdi\_unicode\_supply\_ship | 供应商匹配关系 |
| hdi\_unicode\_goods\_ship | 商品匹配 |
| hdi\_unicode\_consumables\_cate | 耗材商品分类 |
| hdi\_unicode\_drugs\_cate | 药品商品分类 |
| hdi\_unicode\_reagent\_cate | 试剂商品分类 |

### 4.2.2表结构

#### 4.2.2.1表hdi\_unicode\_supply\_ship 供应商匹配关系

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 关系标识 | ship\_id | 关系标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 医院机构标识 | horg\_id | 医院机构标识 | bigint(20) |
| 医院供应商编码 | supply\_no | 医院供应商编码 | varchar(30) |
| 医院供应商名称 | supply\_name | 医院供应商名称 | varchar(50) |
| 医院供应商信用代码 | supply\_code | 医院供应商信用代码 | varchar(18) |
| 平台供应商标识 | porg\_id | 平台供应商标识 | bigint(20) |
| 匹配标志 | ship\_flag | 0 未匹配 1已匹配 | int(1) |
| 创建人标识 | cremanid | 创建人标识 | bigint(20) |
| 创建人 | cremanname | 创建人 | varchar(30) |
| 创建时间 | credate | 创建时间 | datetime |
| 修改人标识 | editmanid | 修改人标识 | bigint(20) |
| 修改人 | editmanname | 修改人 | varchar(30) |
| 修改时间 | editdate | 修改时间 | datetime |
| 数据来源 | datasource | 1接口 2手工 | int(2) |
| 删除标志 | del\_flag | 0 否 1是 | int(1) |
| 备注 | memo | 备注 | varchar(200) |

#### 4.2.2.2表hdi\_unicode\_goods\_ship 商品匹配

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 关系标识 | ship\_id | 关系标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 目标机构标识 | torg\_id | 目标机构标识 | bigint(20) |
| 目标机构类型 | torg\_type | 目标机构类型 | int(4) |
| 目标商品类型 | tgoods\_type | 1 药品 2 试剂 3 耗材 | int(4) |
| 目标商品标识 | tgoods\_id | 目标商品标识 | bigint(20) |
| 平台商品标识 | pgoods\_id | 平台商品标识 | bigint(20) |
| 目标商品规格标识 | tspecs\_id | 目标商品规格标识 | bigint(20) |
| 平台商品规格标识 | pspecs\_id | 平台商品规格标识 | bigint(20) |
| 目标商品认证编号标识 | tapproval\_id | 目标商品认证编号标识 | bigint(20) |
| 平台商品认证编号标识 | papproval\_id | 平台商品认证编号标识 | bigint(20) |
| 匹配标志 | ship\_flag | 0 未匹配 1已匹配 | int(1) |
| 审核状态 | check\_status | 审核状态 | int(4) |
| 删除标志 | del\_flag | 0否 1是 | int(1) |
| 创建人标识 | cremanid | 创建人标识 | bigint(20) |
| 创建人 | cremanname | 创建人 | varchar(30) |
| 创建时间 | credate | 创建时间 | datetime |
| 修改人标识 | editmanid | 修改人标识 | bigint(20) |
| 修改人 | editmanname | 修改人 | varchar(30) |
| 修改时间 | editdate | 修改时间 | datetime |
| 备注 | memo | 备注 | varchar(200) |

#### 4.2.2.3表hdi\_unicode\_goods\_ship\_hist 商品匹配历史

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 匹配历史标识 | shiphist\_id | 关系标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 目标机构标识 | torg\_id | 目标机构标识 | bigint(20) |
| 目标机构类型 | torg\_type | 目标机构类型 | int(4) |
| 目标商品类型 | tgoods\_type | 1 药品 2 试剂 3 耗材 | int(4) |
| 目标商品标识 | tgoods\_id | 目标商品标识 | bigint(20) |
| 平台商品标识 | pgoods\_id | 平台商品标识 | bigint(20) |
| 目标商品规格标识 | tspecs\_id | 目标商品规格标识 | bigint(20) |
| 平台商品规格标识 | pspecs\_id | 平台商品规格标识 | bigint(20) |
| 目标商品认证编号标识 | tapproval\_id | 目标商品认证编号标识 | bigint(20) |
| 平台商品认证编号标识 | papproval\_id | 平台商品认证编号标识 | bigint(20) |
| 匹配标志 | ship\_flag | 0 未匹配 1已匹配 | int(1) |
| 失败原因 | fail\_reason | 失败原因 | varchar(2000) |
| 审核状态 | check\_status | 审核状态 | int(4) |
| 删除标志 | del\_flag | 0否 1是 | int(1) |
| 创建人标识 | cremanid | 创建人标识 | bigint(20) |
| 创建人 | cremanname | 创建人 | varchar(30) |
| 创建时间 | credate | 创建时间 | datetime |
| 修改人标识 | editmanid | 修改人标识 | bigint(20) |
| 修改人 | editmanname | 修改人 | varchar(30) |
| 修改时间 | editdate | 修改时间 | datetime |
| 备注 | memo | 备注 | varchar(200) |

#### 4.2.2.4表hdi\_unicode\_consumables\_cate 耗材商品分类

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 商品分类标识 | cate\_id | 商品分类标识 | bigint(20) |
| 分类名称 | cate\_name | 分类名称 | varchar(30) |
| 分类编码 | cate\_no | 分类编码 | varchar(30) |
| 分类层级 | cate\_level | 根节点为1，第一层，依次往下 | int(4) |
| 父分类标识 | pcate\_id | 父分类标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 备注 | memo | 备注 | varchar(200) |

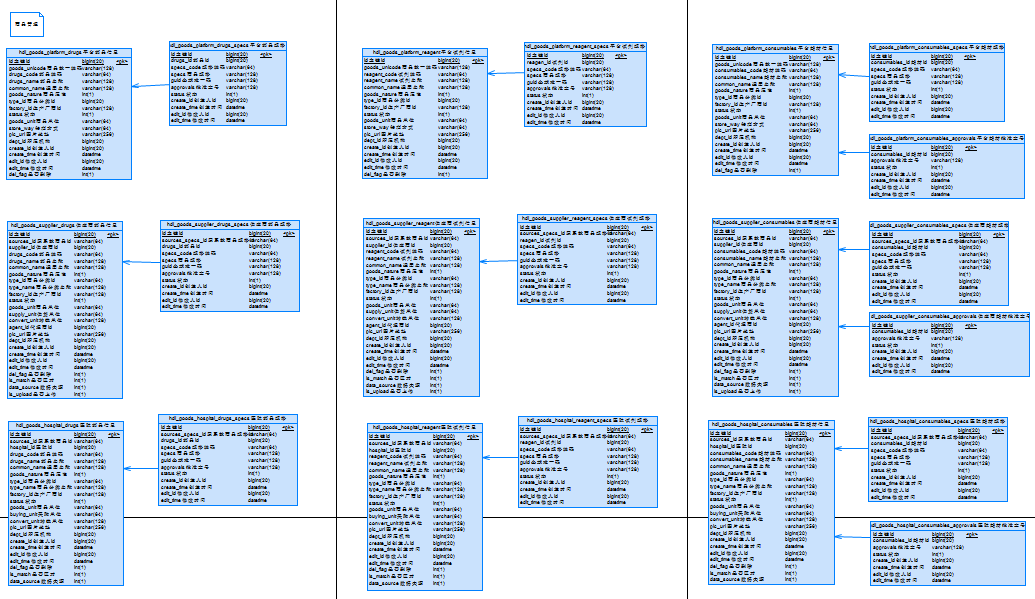
#### 4.2.2.5表hdi\_unicode\_reagent\_cate 试剂商品分类

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 商品分类标识 | cate\_id | 商品分类标识 | bigint(20) |
| 分类名称 | cate\_name | 分类名称 | varchar(30) |
| 分类编码 | cate\_no | 分类编码 | varchar(30) |
| 分类层级 | cate\_level | 根节点为1，第一层，依次往下 | int(4) |
| 父分类标识 | pcate\_id | 父分类标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 备注 | memo | 备注 | varchar(200) |

#### 4.2.2.6表hdi\_unicode\_drugs\_cate 药品商品分类

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| 商品分类标识 | cate\_id | 商品分类标识 | bigint(20) |
| 分类名称 | cate\_name | 分类名称 | varchar(30) |
| 分类编码 | cate\_no | 分类编码 | varchar(30) |
| 分类层级 | cate\_level | 根节点为1，第一层，依次往下 | int(4) |
| 父分类标识 | pcate\_id | 父分类标识 | bigint(20) |
| 机构标识 | dept\_id | 机构标识 | bigint(20) |
| 备注 | memo | 备注 | varchar(200) |

## 4.3商品管理



### 4.3.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| hdi\_goods\_platform\_drugs | 平台药品信息 |
| hdi\_goods\_platform\_drugs\_specs | 平台药品规格 |
| hdi\_goods\_supplier\_drugs | 供应商药品信息 |
| hdi\_goods\_supplier\_drugs\_specs | 供应商药品规格 |
| hdi\_goods\_hospital\_drugs | 医院药品信息 |
| hdi\_goods\_hospital\_drugs\_specs | 医院药品规格 |
| hdi\_goods\_platform\_reagent | 平台试剂信息 |
| hdi\_goods\_platform\_reagent\_specs | 平台试剂规格 |
| hdi\_goods\_supplier\_reagent | 供应商试剂信息 |
| hdi\_goods\_supplier\_reagent\_specs | 供应商试剂规格 |
| hdi\_goods\_hospital\_reagent | 医院试剂信息 |
| hdi\_goods\_hospital\_reagent\_specs | 医院试剂规格 |
| hdi\_goods\_platform\_consumables | 平台耗材信息 |
| hdi\_goods\_platform\_consumables\_specs | 平台耗材规格 |
| hdi\_goods\_platform\_consumables\_approvals | 平台耗材批准文号 |
| hdi\_goods\_supplier\_consumables | 供应商耗材信息 |
| hdi\_goods\_supplier\_consumables\_specs | 供应商耗材规格 |
| hdi\_goods\_supplier\_consumables\_approvals | 供应商耗材批准文号 |
| hdi\_goods\_hospital\_consumables | 医院耗材信息 |
| hdi\_goods\_hospital\_consumables\_specs | 医院耗材规格 |
| hdi\_goods\_hospital\_consumables\_approvals | 医院耗材批准文号 |

### 4.3.2表结构

#### 4.3.2.1表hdi\_goods\_platform\_drugs 平台药品信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| goods\_unicode 商品统一编码 | goods\_unicode | 商品统一编码 | varchar(128) |
| drugs\_code 药品编码 | drugs\_code | 药品编码 | varchar(64) |
| drugs\_name 药品名称 | drugs\_name | 药品名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | bigint(20) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| store\_way 储存方式 | store\_way | 储存方式 | varchar(64) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

#### 4.3.2.2表hdi\_goods\_platform\_drugs\_specs 平台药品规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| drugs\_id 药品id | drugs\_id | 药品id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.3表hdi\_goods\_supplier\_drugs 供应商药品信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| drugs\_code 药品编码 | drugs\_code | 药品编码 | varchar(64) |
| drugs\_name 药品名称 | drugs\_name | 药品名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| supply\_unit 供货单位 | supply\_unit | 供货单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| agent\_id 代理商id | agent\_id | 代理商id | bigint(20) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:供应商ERP) | int(1) |
| is\_upload 是否上传 | is\_upload | 是否上传(0:未上传;1:已上传) | int(1) |

#### 4.3.2.4表hdi\_goods\_supplier\_drugs\_specs 供应商药品规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| drugs\_id 药品id | drugs\_id | 药品id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.5表hdi\_goods\_hospital\_drugs 医院药品信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| drugs\_code 药品编码 | drugs\_code | 药品编码 | varchar(64) |
| drugs\_name 药品名称 | drugs\_name | 药品名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| buying\_unit 采购单位 | buying\_unit | 采购单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.3.2.6表hdi\_goods\_hospital\_drugs\_specs 医院药品规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| drugs\_id 药品id | drugs\_id | 药品id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.7表hdi\_goods\_platform\_reagent 平台试剂信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| goods\_unicode 商品统一编码 | goods\_unicode | 商品统一编码 | varchar(128) |
| reagent\_code 试剂编码 | reagent\_code | 试剂编码 | varchar(64) |
| reagent\_name 试剂名称 | reagent\_name | 试剂名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | bigint(20) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| store\_way 储存方式 | store\_way | 储存方式 | varchar(64) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

#### 4.3.2.8表hdi\_goods\_platform\_reagent\_specs 平台试剂规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| reagen\_id 试剂id | reagen\_id | 试剂id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.9表hdi\_goods\_supplier\_reagent 供应商试剂信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| reagent\_code 试剂编码 | reagent\_code | 试剂编码 | varchar(64) |
| reagent\_name 试剂名称 | reagent\_name | 试剂名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| supply\_unit 供货单位 | supply\_unit | 供货单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| agent\_id 代理商id | agent\_id | 代理商id | bigint(20) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:供应商ERP) | int(1) |
| is\_upload 是否上传 | is\_upload | 是否上传(0:未上传;1:已上传) | int(1) |

#### 4.3.2.10表hdi\_goods\_supplier\_reagent\_specs 供应商试剂规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| reagen\_id 试剂id | reagen\_id | 试剂id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.11表hdi\_goods\_hospital\_reagent 医院试剂信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| reagent\_code 试剂编码 | reagent\_code | 试剂编码 | varchar(64) |
| reagent\_name 试剂名称 | reagent\_name | 试剂名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| buying\_unit 采购单位 | buying\_unit | 采购单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.3.2.12表hdi\_goods\_hospital\_reagent\_specs 医院试剂规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| reagen\_id 试剂id | reagen\_id | 试剂id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.13表hdi\_goods\_platform\_consumables 平台耗材信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| goods\_unicode 商品统一编码 | goods\_unicode | 商品统一编码 | varchar(128) |
| consumables\_code 耗材编码 | consumables\_code | 耗材编码 | varchar(64) |
| consumables\_name 耗材名称 | consumables\_name | 耗材名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | bigint(20) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| store\_way 储存方式 | store\_way | 储存方式 | varchar(64) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

#### 4.3.2.14表hdi\_goods\_platform\_consumables\_specs 平台耗材规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.16表hdi\_goods\_platform\_consumables\_approvals 平台耗材批准文号

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.17表hdi\_goods\_supplier\_consumables 供应商耗材信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| consumables\_code 耗材编码 | consumables\_code | 耗材编码 | varchar(64) |
| consumables\_name 耗材名称 | consumables\_name | 耗材名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| supply\_unit 供货单位 | supply\_unit | 供货单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| agent\_id 代理商id | agent\_id | 代理商id | bigint(20) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:供应商ERP) | int(1) |
| is\_upload 是否上传 | is\_upload | 是否上传(0:未上传;1:已上传) | int(1) |

#### 4.3.2.18表hdi\_goods\_supplier\_consumables\_specs 供应商耗材规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.19表hdi\_goods\_supplier\_consumables\_approvals 供应商耗材批准文号

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.20表hdi\_goods\_hospital\_consumables 医院耗材信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_id 原系统商品id | sources\_id | 原系统商品id | varchar(64) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| consumables\_code 耗材编码 | consumables\_code | 耗材编码 | varchar(64) |
| consumables\_name 耗材名称 | consumables\_name | 耗材名称 | varchar(128) |
| common\_name 通用名称 | common\_name | 通用名称 | varchar(128) |
| goods\_nature 商品属性 | goods\_nature | 商品属性(0:国产;1:进口) | int(1) |
| type\_id 商品分类id | type\_id | 商品分类id | varchar(64) |
| type\_name 商品分类名称 | type\_name | 商品分类名称 | varchar(128) |
| factory\_id 生产厂商id | factory\_id | 生产厂商id | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| goods\_unit 商品单位 | goods\_unit | 商品单位 | varchar(64) |
| buying\_unit 采购单位 | buying\_unit | 采购单位 | varchar(64) |
| convert\_unit 转换单位 | convert\_unit | 转换单位 | varchar(128) |
| pic\_url 图片地址 | pic\_url | 图片地址 | varchar(256) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| is\_match 是否匹对 | is\_match | 是否匹对(0:未匹对;1:已匹对) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

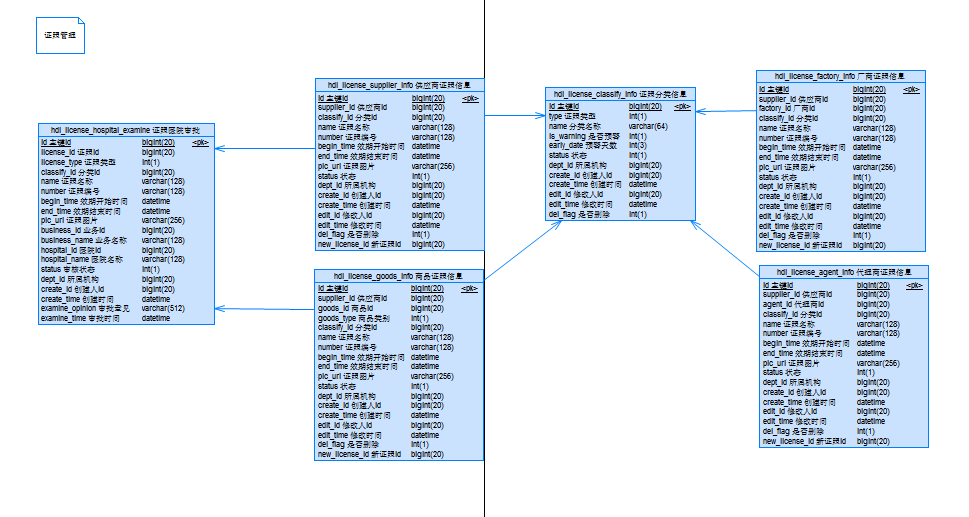
#### 4.3.2.21表hdi\_goods\_hospital\_consumables\_specs 医院耗材规格

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id | varchar(64) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| specs\_code 规格编码 | specs\_code | 规格编码 | varchar(64) |
| specs 商品规格 | specs | 商品规格 | varchar(128) |
| guid 全球唯一码 | guid | 全球唯一码 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

#### 4.3.2.22表hdi\_goods\_hospital\_consumables\_approvals 医院耗材批准文号

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| consumables\_id 耗材id | consumables\_id | 耗材id | bigint(20) |
| approvals 批准文号 | approvals | 批准文号 | varchar(128) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |

## 4.4证照管理



### 4.4.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| hdi\_license\_classify\_info | 证照分类信息 |
| hdi\_license\_supplier\_info | 供应商证照信息 |
| hdi\_license\_goods\_info | 商品证照信息 |
| hdi\_license\_factory\_info | 厂商证照信息 |
| hdi\_license\_agent\_info | 代理商证照信息 |
| hdi\_license\_hospital\_examine | 证照医院审批 |

### 4.4.2表结构

#### 4.4.2.1表hdi\_license\_classify\_info 证照分类信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| type 证照类型 | type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) | int(1) |
| name 分类名称 | name | 分类名称 | varchar(20) |
| is\_warning 是否预警 | is\_warning | 是否预警(0:否;1:是) | int(1) |
| early\_date 预警天数 | early\_date | 预警天数 | int(3) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

#### 4.4.2.2表hdi\_license\_supplier\_info 供应商证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| classify\_id 分类id | classify\_id | 分类id | bigint(20) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | bigint(20) |

#### 4.4.2.3表hdi\_license\_goods\_info 商品证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| goods\_id 商品id | goods\_id | 商品id | bigint(20) |
| goods\_type 商品类别 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) | int(1) |
| classify\_id 分类id | classify\_id | 分类id | bigint(20) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | bigint(20) |

#### 4.4.2.4表hdi\_license\_factory\_info 厂商证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| factory\_id 厂商商id | factory\_id | 厂商商id | bigint(20) |
| classify\_id 分类id | classify\_id | 分类id | bigint(20) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | bigint(20) |

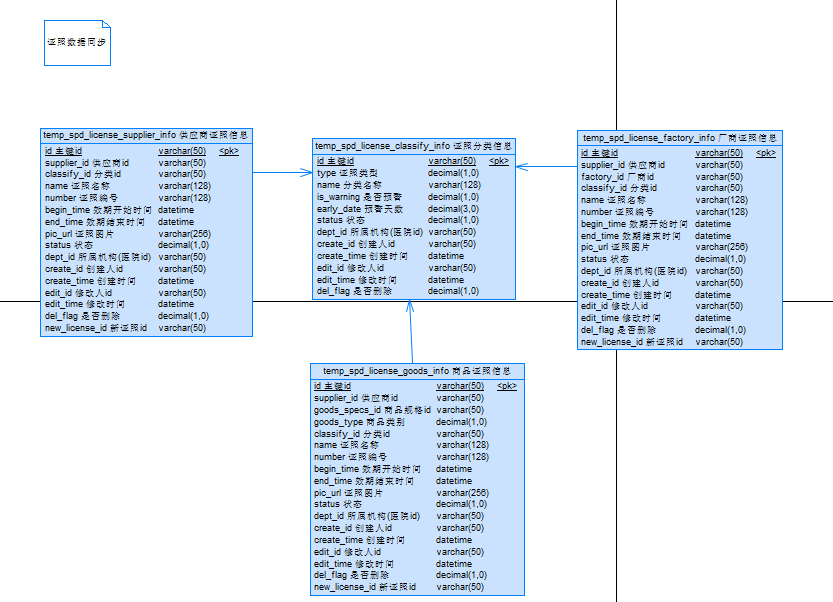
#### 4.4.2.5表hdi\_license\_agent\_info 代理商证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| agent\_id 代理商id | agent\_id | 代理商id | bigint(20) |
| classify\_id 分类id | classify\_id | 分类id | bigint(20) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | bigint(20) |

#### 4.4.2.6表hdi\_license\_hospital\_examine 证照医院审批

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| license\_id 证照id | license\_id | 证照id | bigint(20) |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) | int(1) |
| classify\_id 分类id | classify\_id | 分类id | bigint(20) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| business\_id 业务id | business\_id | 业务id(商品id或供应商id) | bigint(20) |
| business\_name 业务名称 | business\_name | 业务名称(商品名称或供应商名称) | varchar(128) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| hospital\_name 医院名称 | hospital\_name | 医院名称 | varchar(128) |
| status 审核状态 | status | 审核状态(0:待审批;1:审核通过;2:审核不通过) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| examine\_opinion 审批意见 | examine\_opinion | 审批意见 | varchar(512) |
| examine\_time 审批时间 | examine\_time | 审批时间 | datetime |

## 4.5证照数据同步



### 4.5.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| temp\_spd\_license\_classify\_info | 证照分类信息 |
| temp\_spd\_license\_supplier\_info | 供应商证照信息 |
| temp\_spd\_license\_goods\_info | 商品证照信息 |
| temp\_spd\_license\_factory\_info | 厂商证照信息 |

### 4.5.2表结构

#### 4.5.2.1表temp\_spd\_license\_classify\_info 证照分类信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | varchar(50) |
| type 证照类型 | type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) | decimal(1,0) |
| name 分类名称 | name | 分类名称 | varchar(128) |
| is\_warning 是否预警 | is\_warning | 是否预警(0:否;1:是) | decimal(1,0) |
| early\_date 预警天数 | early\_date | 预警天数 | decimal(3,0) |
| status 状态 | status | 状态(0:停用;1:启用) | decimal(1,0) |
| dept\_id 所属机构(医院id) | dept\_id | 所属机构(医院id) | varchar(50) |
| create\_id 创建人id | create\_id | 创建人id | varchar(50) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | varchar(50) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | decimal(1,0) |

#### 4.5.2.2表temp\_spd\_license\_factory\_info 厂商证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | varchar(50) |
| supplier\_id 供应商id | supplier\_id | 供应商id | varchar(50) |
| factory\_id 厂商id | factory\_id | 厂商id | varchar(50) |
| classify\_id 分类id | classify\_id | 分类id | varchar(50) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | decimal(1,0) |
| dept\_id 所属机构(医院id) | dept\_id | 所属机构(医院id) | varchar(50) |
| create\_id 创建人id | create\_id | 创建人id | varchar(50) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | varchar(50) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | decimal(1,0) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | varchar(50) |

#### 4.5.2.3表temp\_spd\_license\_goods\_info 商品证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | varchar(50) |
| supplier\_id 供应商id | supplier\_id | 供应商id | varchar(50) |
| goods\_specs\_id 商品规格id | goods\_specs\_id | 商品规格id | varchar(50) |
| goods\_type 商品类别 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) | decimal(1,0) |
| classify\_id 分类id | classify\_id | 分类id | varchar(50) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | decimal(1,0) |
| dept\_id 所属机构(医院id) | dept\_id | 所属机构(医院id) | varchar(50) |
| create\_id 创建人id | create\_id | 创建人id | varchar(50) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | varchar(50) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | decimal(1,0) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | varchar(50) |

#### 4.5.2.4表temp\_spd\_license\_supplier\_info 供应商证照信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | varchar(50) |
| supplier\_id 供应商id | supplier\_id | 供应商id | varchar(50) |
| classify\_id 分类id | classify\_id | 分类id | varchar(50) |
| name 证照名称 | name | 证照名称 | varchar(128) |
| number 证照编号 | number | 证照编号 | varchar(128) |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 | datetime |
| end\_time 效期结束时间 | end\_time | 效期结束时间 | datetime |
| pic\_url 证照图片 | pic\_url | 证照图片 | varchar(256) |
| status 状态 | status | 状态(0:停用;1:启用) | decimal(1,0) |
| dept\_id 所属机构(医院id) | dept\_id | 所属机构(医院id) | varchar(50) |
| create\_id 创建人id | create\_id | 创建人id | varchar(50) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | varchar(50) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | decimal(1,0) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) | varchar(50) |

## 4.6采购管理

### 4.6.1表清单

### 4.6.2表结构

## 4.7供货管理

### 4.7.1表清单

### 4.7.2表结构

## 4.8退货管理



### 4.8.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| hdi\_refunds\_apply\_master | 退货申请单信息 |
| hdi\_refunds\_apply\_detail | 退货申请单明细信息 |
| hdi\_refunds\_master | 退货单信息 |
| hdi\_refunds\_detail | 退货单明细信息 |

### 4.8.2表结构

#### 4.8.2.1表hdi\_refunds\_apply\_master 退货申请单信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| sources\_supplier\_id 原系统医院供应商id | sources\_supplier\_id | 原系统医院供应商id | varchar(64) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| sources\_hospital\_id 原系统医院id | sources\_hospital\_id | 原系统医院id | varchar(64) |
| store\_house\_id 医院库房id | store\_house\_id | 医院库房id | bigint(20) |
| sources\_store\_house\_id 原系统医院库房id | sources\_store\_house\_id | 原系统医院库房id | varchar(64) |
| refunds\_apply\_no 医院退货申请单号 | refunds\_apply\_no | 医院退货申请单号 | varchar(64) |
| apply\_time 申请时间 | apply\_time | 申请时间 | datetime |
| status 状态 | status | 状态(0:未确认;1:已作废;2:已确认;3:已退货;4:部分退货) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.8.2.2表hdi\_refunds\_apply\_detail 退货申请单明细信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| apply\_master\_id 退货申请主单id | apply\_master\_id | 退货申请主单id | bigint(20) |
| accept\_master\_id 验证主单id | accept\_master\_id | 验证主单id | bigint(20) |
| accept\_no 验收单编号 | accept\_no | 验收单编号 | varchar(64) |
| accept\_detail\_id 验收明细单id | accept\_detail\_id | 验收明细单id | bigint(20) |
| goods\_type 商品类型 | goods\_type | 商品类型(1:药品;2:试剂;3:耗材) | int(1) |
| goods\_id 医院商品id | goods\_id | 医院商品id | bigint(20) |
| sources\_goods\_id 原医院商品id | sources\_goods\_id | 原医院商品id | varchar(64) |
| specs\_id 医院商品规格id | specs\_id | 医院商品规格id | bigint(20) |
| sources\_specs\_id 原医院商品规格id | sources\_specs\_id | 原医院商品规格id | varchar(64) |
| lot\_id 生产批号id | lot\_id | 生产批号id | bigint(20) |
| goods\_unit\_code 商品单位编码 | goods\_unit\_code | 商品单位编码 | varchar(64) |
| apply\_refunds\_number 申请退货数量 | apply\_refunds\_number | 申请退货数量 | int(11) |
| reality\_refunds\_number 实际退货数量 | reality\_refunds\_number | 实际退货数量 | int(11) |
| supply\_price 供货单价 | supply\_price | 供货单价 | decimal(11,2) |
| refunds\_remark 退货原因 | refunds\_remark | 退货原因 | varchar(512) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

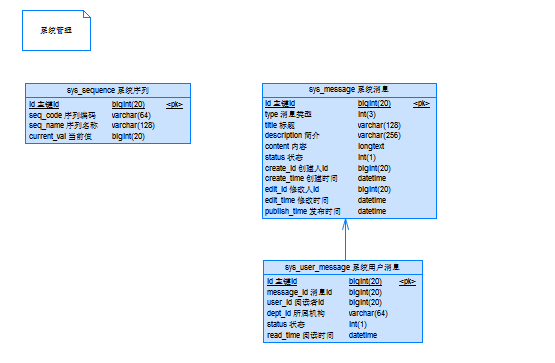
#### 4.8.2.3表hdi\_refunds\_master 退货单信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| refunds\_no 退货单编号 | refunds\_no | 退货单编号 | varchar(64) |
| supplier\_id 供应商id | supplier\_id | 供应商id | bigint(20) |
| sources\_supplier\_id 原系统医院供应商id | sources\_supplier\_id | 原系统医院供应商id | varchar(64) |
| hospital\_id 医院id | hospital\_id | 医院id | bigint(20) |
| sources\_hospital\_id 原系统医院id | sources\_hospital\_id | 原系统医院id | varchar(64) |
| store\_house\_id 医院库房id | store\_house\_id | 医院库房id | bigint(20) |
| sources\_store\_house\_id 原系统医院库房id | sources\_store\_house\_id | 原系统医院库房id | varchar(64) |
| refunds\_apply\_no 医院退货申请单号 | refunds\_apply\_no | 医院退货申请单号 | varchar(64) |
| regression\_number 消退单号 | regression\_number | 消退单号 | varchar(64) |
| refunds\_time 退货时间 | refunds\_time | 退货时间 | datetime |
| status 状态 | status | 状态(0:未确认;1:已提交;2:已完成) | int(1) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |
| data\_source 数据来源 | data\_source | 数据来源(0:系统录入;1:医院SPD) | int(1) |

#### 4.8.2.4表hdi\_refunds\_detail 退货单明细信息

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| master\_id 退货主单id | master\_id | 退货主单id | bigint(20) |
| sources\_type 来源类型 | sources\_type | 来源类型(0:退货申请单;1:验收单;医院退货申请单号不为空则为0,否则为1) | int(1) |
| sources\_master\_id 来源主单id | sources\_master\_id | 来源主单id(根据来源类型,对应退货申请单明细或验收单明细id) | bigint(20) |
| sources\_detail\_id 来源明细单id | sources\_detail\_id | 来源明细单id(根据来源类型,对应退货申请主单或验收主单id) | bigint(20) |
| accept\_no 验收单编号 | accept\_no | 验收单编号 | varchar(64) |
| goods\_type 商品类型 | goods\_type | 商品类型(1:药品;2:试剂;3:耗材) | int(1) |
| goods\_id 医院商品id | goods\_id | 医院商品id | bigint(20) |
| sources\_goods\_id 原医院商品id | sources\_goods\_id | 原医院商品id | varchar(64) |
| specs\_id 医院商品规格id | specs\_id | 医院商品规格id | bigint(20) |
| sources\_specs\_id 原医院商品规格id | sources\_specs\_id | 原医院商品规格id | varchar(64) |
| lot\_id 生产批号id | lot\_id | 生产批号id | bigint(20) |
| goods\_unit\_code 商品单位编码 | goods\_unit\_code | 商品单位编码 | varchar(64) |
| apply\_refunds\_number 申请退货数量 | apply\_refunds\_number | 申请退货数量 | int(11) |
| reality\_refunds\_number 实际退货数量 | reality\_refunds\_number | 实际退货数量 | int(11) |
| refunds\_price 退货单价 | refunds\_price | 退货单价 | decimal(11,2) |
| refunds\_remark 退货原因 | refunds\_remark | 退货原因 | varchar(512) |
| dept\_id 所属机构 | dept\_id | 所属机构 | bigint(20) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) | int(1) |

## 4.9系统消息



### 4.9.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| sys\_message | 系统消息 |
| sys\_user\_message | 系统用户消息 |

### 4.9.2表结构

#### 4.9.2.1表sys\_message 系统消息的栏的清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| type 消息类型 | type | 消息类型(0:系统通知;1:证照提醒) | int(3) |
| title 标题 | title | 标题 | varchar(128) |
| description 简介 | description | 简介 | varchar(256) |
| content 内容 | content | 内容 | longtext |
| status 状态 | status | 状态(0:草稿;1:已发布;2:已删除) | int(1) |
| create\_id 创建人id | create\_id | 创建人id | bigint(20) |
| create\_time 创建时间 | create\_time | 创建时间 | datetime |
| edit\_id 修改人id | edit\_id | 修改人id | bigint(20) |
| edit\_time 修改时间 | edit\_time | 修改时间 | datetime |
| publish\_time 发布时间 | publish\_time | 发布时间 | datetime |

#### 4.9.2.2表sys\_user\_message 系统用户消息的栏的清单

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 代码 | 注释 | 数据类型 |
| id 主键id | id | 主键id | bigint(20) |
| message\_id 消息id | message\_id | 消息id | bigint(20) |
| user\_id 阅读者id | user\_id | 阅读者id | bigint(20) |
| dept\_id 所属机构 | dept\_id | 所属机构 | varchar(64) |
| status 状态 | status | 状态(0:未读;1:已读;2:删除) | int(1) |
| read\_time 阅读时间 | read\_time | 阅读时间 | datetime |

## 4.10视图表

### 4.10.1表清单

|  |  |
| --- | --- |
| 名称 | 说明 |
| view\_hdi\_license\_supplier | 供应商证照视图 |
| view\_hdi\_license\_goods | 供应商商品证照视图 |
| view\_hdi\_license\_factory | 厂商证照视图 |
| view\_hdi\_license\_agent | 代理商证照视图 |
| view\_hdi\_license\_warning | 证照预警视图 |
| view\_hdi\_goods\_supplier | 供应商商品视图 |
| view\_hdi\_goods\_supplier\_hospital | 供应商合作医院商品视图 |
| view\_hdi\_goods\_specs\_platform | 平台商品规格视图 |
| view\_hdi\_goods\_specs\_supplier | 供应商商品规格视图 |
| view\_hdi\_goods\_specs\_hospital | 医院商品规格视图 |
| view\_hdi\_goods\_specs\_match | 平台供应商医院商品匹对视图 |
| view\_hdi\_core\_accept\_detail | 验收单明细视图 |

### 4.10.2表结构

#### 4.10.2.1表view\_hdi\_license\_supplier供应商证照视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 主键id | id | 主键id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| classify\_id 分类id | classify\_id | 分类id |
| name 证照名称 | name | 证照名称 |
| number 证照编号 | number | 证照编号 |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 |
| end\_time 效期结束时间 | end\_time | 效期结束时间 |
| pic\_url 证照图片 | pic\_url | 证照图片 |
| status 状态 | status | 状态(0:停用;1:启用) |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| create\_id 创建人id | create\_id | 创建人id |
| create\_time 创建时间 | create\_time | 创建时间 |
| edit\_id 修改人id | edit\_id | 修改人id |
| edit\_time 修改时间 | edit\_time | 修改时间 |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) |
| classify\_name 分类名称 | classify\_name | 分类名称 |
| is\_warning 是否预警 | is\_warning | 是否预警 |
| early\_date 预警天数 | early\_date | 预警天数 |
| lose\_time 距离失效天数 | lose\_time | 距离失效天数 |
| license\_status 失效状态 | license\_status | 失效状态（0:正常;1:预警;2:失效;3:换证中） |

#### 4.10.2.2表view\_hdi\_license\_goods供应商商品证照视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 主键id | id | 主键id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| goods\_id 商品id | goods\_id | 商品id |
| goods\_type 商品类别 | goods\_type | 商品类别(0:药品;1:试剂;2:耗材) |
| classify\_id 分类id | classify\_id | 分类id |
| name 证照名称 | name | 证照名称 |
| number 证照编号 | number | 证照编号 |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 |
| end\_time 效期结束时间 | end\_time | 效期结束时间 |
| pic\_url 证照图片 | pic\_url | 证照图片 |
| status 状态 | status | 状态(0:停用;1:启用) |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| create\_id 创建人id | create\_id | 创建人id |
| create\_time 创建时间 | create\_time | 创建时间 |
| edit\_id 修改人id | edit\_id | 修改人id |
| edit\_time 修改时间 | edit\_time | 修改时间 |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name商品通用名称 | common\_name | 商品通用名称 |
| factory\_id 厂商id | factory\_id | 厂商id |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) |
| classify\_name 分类名称 | classify\_name | 分类名称 |
| is\_warning 是否预警 | is\_warning | 是否预警 |
| early\_date 预警天数 | early\_date | 预警天数 |
| lose\_time 距离失效天数 | lose\_time | 距离失效天数 |
| license\_status 失效状态 | license\_status | 失效状态（0:正常;1:预警;2:失效;3:换证中） |

#### 4.10.2.3表view\_hdi\_license\_factory厂商证照视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 主键id | id | 主键id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| classify\_id 分类id | classify\_id | 分类id |
| name 证照名称 | name | 证照名称 |
| number 证照编号 | number | 证照编号 |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 |
| end\_time 效期结束时间 | end\_time | 效期结束时间 |
| pic\_url 证照图片 | pic\_url | 证照图片 |
| status 状态 | status | 状态(0:停用;1:启用) |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| create\_id 创建人id | create\_id | 创建人id |
| create\_time 创建时间 | create\_time | 创建时间 |
| edit\_id 修改人id | edit\_id | 修改人id |
| edit\_time 修改时间 | edit\_time | 修改时间 |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) |
| classify\_name 分类名称 | classify\_name | 分类名称 |
| is\_warning 是否预警 | is\_warning | 是否预警 |
| early\_date 预警天数 | early\_date | 预警天数 |
| lose\_time 距离失效天数 | lose\_time | 距离失效天数 |
| license\_status 失效状态 | license\_status | 失效状态（0:正常;1:预警;2:失效;3:换证中） |

#### 4.10.2.4表view\_hdi\_license\_agent代理商证照视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 主键id | id | 主键id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| agent\_id 代理商id | agent\_id | 代理商id |
| classify\_id 分类id | classify\_id | 分类id |
| name 证照名称 | name | 证照名称 |
| number 证照编号 | number | 证照编号 |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 |
| end\_time 效期结束时间 | end\_time | 效期结束时间 |
| pic\_url 证照图片 | pic\_url | 证照图片 |
| status 状态 | status | 状态(0:停用;1:启用) |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| create\_id 创建人id | create\_id | 创建人id |
| create\_time 创建时间 | create\_time | 创建时间 |
| edit\_id 修改人id | edit\_id | 修改人id |
| edit\_time 修改时间 | edit\_time | 修改时间 |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| agent\_name 代理商名称 | agent\_name | 代理商名称 |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) |
| classify\_name 分类名称 | classify\_name | 分类名称 |
| is\_warning 是否预警 | is\_warning | 是否预警 |
| early\_date 预警天数 | early\_date | 预警天数 |
| lose\_time 距离失效天数 | lose\_time | 距离失效天数 |
| license\_status 失效状态 | license\_status | 失效状态（0:正常;1:预警;2:失效;3:换证中） |

#### 4.10.2.5表view\_hdi\_license\_warning证照预警视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 主键id | id | 主键id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| license\_type 证照类型 | license\_type | 证照类型(0:商品证照;1:供应商证照;2:厂商证照;3:代理商证照) |
| classify\_id 分类id | classify\_id | 分类id |
| classify\_name 分类名称 | classify\_name | 分类名称 |
| name 证照名称 | name | 证照名称 |
| number 证照编号 | number | 证照编号 |
| begin\_time 效期开始时间 | begin\_time | 效期开始时间 |
| end\_time 效期结束时间 | end\_time | 效期结束时间 |
| pic\_url 证照图片 | pic\_url | 证照图片 |
| status 状态 | status | 状态(0:停用;1:启用) |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| create\_id 创建人id | create\_id | 创建人id |
| create\_time 创建时间 | create\_time | 创建时间 |
| edit\_id 修改人id | edit\_id | 修改人id |
| edit\_time 修改时间 | edit\_time | 修改时间 |
| del\_flag 是否删除 | del\_flag | 是否删除(-1:已删除;0:正常) |
| new\_license\_id 新证照id | new\_license\_id | 新证照id(换证) |
| is\_warning 是否预警 | is\_warning | 是否预警 |
| early\_date 预警天数 | early\_date | 预警天数 |
| lose\_time 距离失效天数 | lose\_time | 距离失效天数 |
| license\_status 失效状态 | license\_status | 失效状态（0:正常;1:预警;2:失效;3:换证中） |

#### 4.10.2.6表view\_hdi\_goods\_supplier供应商商品视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| id 供应商商品id | id | 供应商商品id |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name商品通用名称 | common\_name | 商品通用名称 |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |
| dept\_id 所属机构 | dept\_id | 所属机构 |

#### 4.10.2.7表view\_hdi\_goods\_supplier\_hospital供应商合作医院商品视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| supplier\_code 供应商编码 | supplier\_code | 供应商编码 |
| supplier\_name 供应商名称 | supplier\_name | 供应商名称 |
| hospital\_id 医院id | hospital\_id | 医院id |
| hospital\_grade 医院等级 | hospital\_grade | 医院级别(0:三级特等;1:三级甲等;2:三级乙等;3:三级丙等;4:二级甲等;5:二级乙等;6:二级丙等;7:一级甲等;8:一级乙等;9:一级丙等) |
| hospital\_code 医院编码 | hospital\_code | 医院编码 |
| hospital\_name 医院名称 | hospital\_name | 医院名称 |
| goods\_id 商品id | goods\_id | 商品id |
| sources\_goods\_id 原系统商品id | sources\_goods\_id | 原系统商品id |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name 商品通用名称 | common\_name | 商品通用名称 |
| type\_id 商品分类id | type\_id | 商品分类id |
| type\_name 商品分类名称 | type\_name | 商品分类名称 |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| factory\_code 厂商编码 | factory\_code | 厂商编码 |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |

#### 4.10.2.8表view\_hdi\_goods\_specs\_platform平台商品规格视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| goods\_id 商品id | goods\_id | goods\_id |
| goods\_unicode 商品统一编码 | goods\_unicode | 商品统一编码 |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name 商品通用名称 | common\_name | 商品通用名称 |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| goods\_unit 商品单位 | goods\_unit | 商品单位 |
| goods\_specs\_id 商品规格id | goods\_specs\_id | 商品规格id |
| specs\_code 商品规格编码 | specs\_code | 商品规格编码 |
| Specs 商品规格 | specs | 商品规格 |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| factory\_code 厂商编码 | factory\_code | 厂商编码 |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |

#### 4.10.2.9表view\_hdi\_goods\_specs\_supplier供应商商品规格视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| supplier\_sources\_id 原系统供应商id | supplier\_sources\_id | 原系统供应商id |
| goods\_id 商品id | goods\_id | goods\_id |
| sources\_id 原系统商品id | sources\_id | 原系统商品id |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name 商品通用名称 | common\_name | 商品通用名称 |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| goods\_unit 商品单位 | goods\_unit | 商品单位 |
| goods\_specs\_id 商品规格id | goods\_specs\_id | 商品规格id |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id |
| specs\_code 商品规格编码 | specs\_code | 商品规格编码 |
| Specs 商品规格 | specs | 商品规格 |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| factory\_code 厂商编码 | factory\_code | 厂商编码 |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |

#### 4.10.2.10表view\_hdi\_goods\_specs\_hospital医院商品规格视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| hospital\_id 医院id | hospital\_id | 医院id |
| hospital\_sources\_id 原系统医院id | hospital\_sources\_id | 原系统医院id |
| goods\_id 商品id | goods\_id | goods\_id |
| sources\_id 原系统商品id | sources\_id | 原系统商品id |
| goods\_code 商品编码 | goods\_code | 商品编码 |
| goods\_name 商品名称 | goods\_name | 商品名称 |
| common\_name 商品通用名称 | common\_name | 商品通用名称 |
| dept\_id 所属机构 | dept\_id | 所属机构 |
| goods\_unit 商品单位 | goods\_unit | 商品单位 |
| goods\_specs\_id 商品规格id | goods\_specs\_id | 商品规格id |
| sources\_specs\_id 原系统商品规格id | sources\_specs\_id | 原系统商品规格id |
| specs\_code 商品规格编码 | specs\_code | 商品规格编码 |
| Specs 商品规格 | specs | 商品规格 |
| factory\_id 厂商商id | factory\_id | 厂商商id |
| factory\_code 厂商编码 | factory\_code | 厂商编码 |
| factory\_name 厂商名称 | factory\_name | 厂商名称 |

#### 4.10.2.11表view\_hdi\_goods\_specs\_match平台供应商医院商品匹对视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| goods\_unicode 商品统一编码 | goods\_unicode | 商品统一编码 |
| platform\_goods\_id 平台商品id | platform\_goods\_id | 平台商品id |
| platform\_goods\_code 平台商品编码 | platform\_goods\_code | 平台商品编码 |
| platform\_goods\_name 平台商品名称 | platform\_goods\_name | 平台商品名称 |
| platform\_common\_name 平台商品通用名 | platform\_common\_name | 平台商品通用名 |
| platform\_dept\_id 平台商品所属机构 | platform\_dept\_id | 平台商品所属机构 |
| platform\_goods\_unit 平台商品单位 | platform\_goods\_unit | 平台商品单位 |
| platform\_goods\_specs\_id 平台商品规格id | platform\_goods\_specs\_id | 平台商品规格id |
| platform\_goods\_specs\_code 平台商品规格编码 | platform\_goods\_specs\_code | 平台商品规格编码 |
| platform\_goods\_specs\_name 平台商品规格名称 | platform\_goods\_specs\_name | 平台商品规格名称 |
| platform\_factory\_id 平台商品厂商id | platform\_factory\_id | 平台商品厂商id |
| platform\_factory\_code 平台商品厂商编码 | platform\_factory\_code | 平台商品厂商编码 |
| platform\_factory\_name 平台商品厂商名称 | platform\_factory\_name | 平台商品厂商名称 |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| supplier\_sources\_id 原系统供应商id | supplier\_sources\_id | 原系统供应商id |
| supplier\_goods\_id 供应商商品id | supplier\_goods\_id | 供应商商品id |
| supplier\_sources\_goods\_id 供应商原系统商品id | supplier\_sources\_goods\_id | 供应商原系统商品id |
| supplier\_goods\_code 供应商商品编码 | supplier\_goods\_code | 供应商商品编码 |
| supplier\_goods\_name 供应商商品名称 | supplier\_goods\_name | 供应商商品名称 |
| supplier\_common\_name 供应商商品通用名 | supplier\_common\_name | 供应商商品通用名 |
| supplier\_dept\_id 供应商商品所属机构 | supplier\_dept\_id | 供应商商品所属机构 |
| supplier\_goods\_unit 供应商商品单位 | supplier\_goods\_unit | 供应商商品单位 |
| supplier\_goods\_specs\_id 供应商商品规格id | supplier\_goods\_specs\_id | 供应商商品规格id |
| supplier\_sources\_specs\_id 供应商原系统商品规格id | supplier\_sources\_specs\_id | 供应商原系统商品规格id |
| supplier\_goods\_specs\_code 供应商商品规格编码 | supplier\_goods\_specs\_code | 供应商商品规格编码 |
| supplier\_goods\_specs\_name 供应商商品规格名称 | supplier\_goods\_specs\_name | 供应商商品规格名称 |
| supplier\_factory\_id 供应商商品厂商id | supplier\_factory\_id | 供应商商品厂商id |
| supplier\_factory\_code 供应商商品厂商编码 | supplier\_factory\_code | 供应商商品厂商编码 |
| supplier\_factory\_name 供应商商品厂商名称 | supplier\_factory\_name | 供应商商品厂商名称 |
| hospital\_id 医院id | hospital\_id | 医院id |
| hospital\_sources\_id 原系统医院id | hospital\_sources\_id | 原系统医院id |
| hospital\_goods\_id 医院商品id | hospital\_goods\_id | 医院商品id |
| hospital\_goods\_id 医院原系统商品id | hospital\_goods\_id | 医院原系统商品id |
| hospital\_goods\_code 医院商品编码 | hospital\_goods\_code | 医院商品编码 |
| hospital\_goods\_name 医院商品名称 | hospital\_goods\_name | 医院商品名称 |
| hospital\_common\_name 医院商品通用名称 | hospital\_common\_name | 医院商品通用名称 |
| hospital\_dept\_id 医院商品所属机构 | hospital\_dept\_id | 医院商品所属机构 |
| hospital\_goods\_unit 医院商品单位 | hospital\_goods\_unit | 医院商品单位 |
| hospital\_goods\_specs\_id 医院商品规格id | hospital\_goods\_specs\_id | 医院商品规格id |
| hospital\_sources\_specs\_id 医院原系统商品规格id | hospital\_sources\_specs\_id | 医院原系统商品规格id |
| hospital\_goods\_specs\_code 医院商品规格编码 | hospital\_goods\_specs\_code | 医院商品规格编码 |
| hospital\_goods\_specs\_name 医院商品规格名称 | hospital\_goods\_specs\_name | 医院商品规格名称 |
| hospital\_factory\_id 医院商品厂商id | hospital\_factory\_id | 医院商品厂商id |
| hospital\_factory\_code 医院商品厂商编码 | hospital\_factory\_code | 医院商品厂商编码 |
| hospital\_factory\_name 医院商品厂商名称 | hospital\_factory\_name | 医院商品厂商名称 |

#### 4.10.2.12表view\_hdi\_core\_accept\_detail验收单明细视图

|  |  |  |
| --- | --- | --- |
| 名称 | 代码 | 注释 |
| accept\_master\_id 验收主单id | accept\_master\_id | 验收主单id |
| accept\_no 验收单号 | accept\_no | 验收单号 |
| hospital\_id 医院id | hospital\_id | 医院id |
| sources\_hospital\_id 原系统医院id | sources\_hospital\_id | 原系统医院id |
| supplier\_id 供应商id | supplier\_id | 供应商id |
| sources\_supplier\_id 原系统供应商id | sources\_supplier\_id | 原系统供应商id |
| store\_house\_id 医院库房id | store\_house\_id | 医院库房id |
| sources\_store\_house\_id 原系统医院库房id | sources\_store\_house\_id | 原系统医院库房id |
| accept\_detail\_id 验收明细单id | accept\_detail\_id | 验收明细单id |
| hospital\_goods\_id 医院商品id | hospital\_goods\_id | 医院商品id |
| sources\_hospital\_goods\_id 原系统医院商品id | sources\_hospital\_goods\_id | 原系统医院商品id |
| hospital\_goods\_specs\_id 医院商品规格id | hospital\_goods\_specs\_id | 医院商品规格id |
| sources\_hospital\_goods\_specs\_id 原系统医院商品规格id | sources\_hospital\_goods\_specs\_id | 原系统医院商品规格id |
| accept\_number 验收数量 | accept\_number | 验收数量 |
| lot\_id 批号id | lot\_id | 批号id |
| lot\_no 生产批号 | lot\_no | 生产批号 |
| supply\_master\_id 供货主单id | supply\_master\_id | 供货主单id |
| supply\_detail\_id 供货明细单id | supply\_detail\_id | 供货明细单id |
| supply\_number 供货数量 | supply\_number | 供货数量 |
| supply\_price 供货价格 | supply\_price | 供货价格 |
| goods\_type 商品类型 | goods\_type | 商品类别(1:药品;2:试剂;3:耗材) |
| hospital\_goods\_code 医院商品编码 | hospital\_goods\_code | 医院商品编码 |
| hospital\_goods\_name 医院商品名称 | hospital\_goods\_name | 医院商品名称 |
| hospital\_common\_name 医院商品通用名称 | hospital\_common\_name | 医院商品通用名 |
| hospital\_goods\_unit 医院商品单位 | hospital\_goods\_unit | 医院商品单位 |
| hospital\_goods\_specs\_code 医院商品规格编码 | hospital\_goods\_specs\_code | 医院商品规格编码 |
| hospital\_goods\_specs\_name 医院商品规格名称 | hospital\_goods\_specs\_name | 医院商品规格名称 |
| hospital\_factory\_id 医院商品厂商id | hospital\_factory\_id | 医院商品厂商id |
| hospital\_factory\_code 医院商品厂商编码 | hospital\_factory\_code | 医院商品厂商编码 |
| hospital\_factory\_name 医院商品厂商名称 | hospital\_factory\_name | 医院商品厂商名称 |

# 模块设计

## 5.1机构管理

### 5.1.1供应商信息

#### 5.1.1.1用例图



#### 5.1.1.2功能设计说明

##### 5.1.1.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 机构名称、机构所属区域、状态 |
| 列表显示 | 1. 显示字段：机构编码、机构名称、机构所属区域、是否集团机构、状态 2. 排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "page":"1",  "limit":"10",  "supplierName":"...医药有限公司",  ...  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  String supplierName = (String)params.get("supplierName");  String provinceCode = (String)params.get("provinceCode");  String cityCode = (String)params.get("cityCode");  String areaCode = (String)params.get("areaCode");    Page<OrgSupplierInfoEntity> page = this.selectPage(  new Query<OrgSupplierInfoEntity>(params).getPage(),  new EntityWrapper<OrgSupplierInfoEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(supplierName), "supplier\_name", supplierName)  .eq(StringUtils.isNotBlank(provinceCode), "province\_code", provinceCode)  .eq(StringUtils.isNotBlank(cityCode), "city\_code", cityCode)  .eq(StringUtils.isNotBlank(areaCode), "area\_code", areaCode)  .eq(params.get("status")!=null, "status", params.get("status"))  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierCode": "440980120001",  "supplierName": "广州白云山医药有限公司",  "creditCode": "0209010098734",  "provinceCode": "44",  "cityCode": "4401",  "areaCode": "440106",  ......  },  {  "id": 2,  ......  }  ]  }  } |

##### 5.1.1.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 机构编码（系统按规则自动生成唯一编码） 2. 机构名称（要求是营业执照上的公司名称），唯一性约束 3. 机构信用代码（机构的统一社会信息代码），唯一性约束 4. 保存供应商信息时，系统自动生成一个该供应商对应的管理账号，供应商管理员可通过管理账号登录系统进行权限范围内的管理操作。 |
| 必填字段 | 机构名称、机构信用代码、机构所属区域、状态 |
| 选填字段 | 机构性质，法人、企业地址、联系电话、邮箱、传真 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "supplierName":"机构名称",  "creditCode":"机构信用代码",  "provinceCode": "44",  "cityCode": "4401",  "areaCode": "440106",  "status": "1",  ...  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void insert(OrgSupplierInfoEntity orgSupplierInfoEntity) {  //生成机构编码  Long supplierCode = ......;  orgSupplierInfoEntity.setSupplierCode(supplierCode);  orgSupplierInfoEntity.setCreateId(getUserId());  orgSupplierInfoEntity.setCreateTime(new Date());  this.insert(orgSupplierInfoEntity);    //创建系统机构信息    //创建供应商角色信息    //分配供应商菜单、数据权限    //创建供应商管理员    } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.1.1.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、机构名称（要求是营业执照上的公司名称），唯一性约束  2、机构信用代码（机构的统一社会信息代码），唯一性约束 |
| 必填字段 | 机构名称、机构信用代码、机构所属区域、状态 |
| 选填字段 | 机构性质，法人、企业地址、联系电话、邮箱、传真 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "id":"机构id",  "supplierName":"机构名称",  "creditCode":"机构信用代码",  "provinceCode": "44",  "cityCode": "4401",  "areaCode": "440106",  "status": "1",  ...  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(OrgSupplierInfoEntity orgSupplierInfoEntity) {  //更新供应商信息  orgSupplierInfoEntity.setEditId(getUserId());  orgSupplierInfoEntity.setEditTime(new Date());  orgSupplierInfoService.updateById(orgSupplierInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.1.1.2.4查看详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 显示字段 | 机构名称、机构信用代码、机构所属区域、状态、是否集团机构、机构编码、机构性质、法人、企业地址、联系电话、邮箱、传真 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "id":"机构id"  } |
| 业务代码 | @Override  public OrgSupplierInfoEntity selectById(Long id) {  return this.baseMapper.selectById(id);  } |
| 输出 | {  "msg": "success",  "code": 0,  "orgSupplierInfoEntity ": {  "id": 1,  "supplierCode": "440980120001",  "supplierName": "广州白云山医药有限公司",  "creditCode": "0209010098734",  "provinceCode": "44",  "cityCode": "4401",  "areaCode": "440106",  ......  }  } |

##### 5.1.1.2.5绑定医院

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 需求 | 设置供应商和医院的绑定关系，“绑定医院”界面自动显示平台统一维护可用的医院信息，一个供应商可以绑定一个或者多个医院。 |
| 医院分页查询接口 | 可查询医院信息分页查询设计相关内容 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "supplierCode": "440980120001",  "hospitalCode": "440980120001,440980120002,440980120003"  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void insert(OrgSupplierHospitalRefEntity orgSupplierHospitalRefEntity) {  //保存供应商医院绑定关系  Strring[] hospitalCodeArray = orgSupplierHospitalRefEntity.getHospitalCode();  for(int i=0; i < hospitalCodeArray.length; i++){  orgSupplierHospitalRefEntity.setHospitalCode(hospitalCodeArray[i]);  orgSupplierHospitalRefEntity.setCreateId(getUserId());  orgSupplierHospitalRefEntity.setCreateTime(new Date());  this.insert(orgSupplierHospitalRefEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.1.1.2.6集团授权

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 需求 | 1、平台管理员可以对机构进行集团授权，授权时可限制集团机构可添加的下级机构数量。  2、已授权的集团机构管理员登录系统，可以查看“集团管理”模块并进行权限范围内的操作。 |
| 必填字段 | 是否集团机构、集团机构数据 |
| 选填字段 | 上层父级 |
| 选择上层父级接口 | 可查看供应商集团信息列表查询设计相关内容 |
| 输入 | {  "id": "1",  "isGroup": "1",  "childNumber": "5",  "parentCode": ""  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void insert(Map<String, Object> params) {  //更新供应商信息  OrgSupplierInfoEntity orgSupplierInfoEntity = new OrgSupplierInfoEntity();  orgSupplierInfoEntity.setId(params.get("id"));  orgSupplierInfoEntity.setIsGroup(params.get("isGroup"));  orgSupplierInfoEntity.setChildNumber(params.get("childNumber"));  orgSupplierInfoService.updateById(orgSupplierInfoEntity);  //插入供应商机构信息  OrgSupplierGroupEntity orgSupplierGroupEntity = new OrgSupplierGroupEntity();  orgSupplierGroupEntity.set....  orgSupplierGroupEntity.set....  orgSupplierGroupEntity.set....  ......  orgSupplierGroupEntity.insert(orgSupplierGroupEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.1.2医院信息

#### 5.1.2.1用例图



#### 5.1.2.2功能设计说明

##### 5.1.2.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 机构名称、医疗机构级别、机构所属区域 |
| 列表显示 | 1. 显示字段：机构编码、机构名称、医疗机构级别、机构所属区域、状态   2、排序：按最新变动时间倒序排列 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "page": "1",  "limit": "5",  "hospitalName": "XX医院",  "hospitalGrade": 0,  "provinceCode": "44",  "cityCode": "4403",  "areaCode": "440305"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  String hospitalName = (String)params.get("hospitalName");  String hospitalGrade = (String)params.get("hospitalGrade");  String provinceCode = (String)params.get("provinceCode");  String cityCode = (String)params.get("cityCode");  String areaCode = (String)params.get("areaCode");    Page<OrgHospitalInfoEntity> page = this.selectPage(  new Query<OrgHospitalInfoEntity>(params).getPage(),  new EntityWrapper<OrgHospitalInfoEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(hospitalName), "hospital\_name", hospitalName)  .eq(StringUtils.isNotBlank(hospitalGrade), "hospital\_grade", hospitalGrade)  .eq(StringUtils.isNotBlank(provinceCode), "province\_code", provinceCode)  .eq(StringUtils.isNotBlank(cityCode), "city\_code", cityCode)  .eq(StringUtils.isNotBlank(areaCode), "area\_code", areaCode)  .orderBy("edit\_time", false)  .orderBy("create\_time", false)  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 2,  "pageSize": 5,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 17,  "parentId": 0,  "hospitalCode": "12010226",  "hospitalGrade": 0,  "hospitalName": "XX医院",  "creditCode": "9523yy",  "provinceCode": "44",  "cityCode": "4405",  "areaCode": "440513",  "status": 1,  ……  },  {  "id": 8,  ……  }  ]  }  } |

##### 5.1.2.2.3新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 系统根据录入的“机构信用代码”判断该机构是否已存在，如果已存在，提示“本机构已存在”。 |
| 必填字段 | 机构名称、机构信用代码、状态、机构所属区域、医疗机构级别 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "hospitalGrade": 0,  "hospitalName": "测试医院",  "creditCode": "123",  "provinceCode": "44",  "cityCode": "4403",  "areaCode": "440301",  "status": 1  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void insertValidate(OrgHospitalInfoEntity orgHospitalInfo) {  List<OrgHospitalInfoEntity> hospitalName = baseMapper.selectByHospitalName(orgHospitalInfo.getHospitalName());  if(!StringUtil.isEmpty(hospitalName)){  throw new HdiException(MessageFormat.format("医院名称{0}已存在",orgHospitalInfo.getHospitalName()));  }  List<OrgHospitalInfoEntity> creditCode = baseMapper.selectByCreditCode(orgHospitalInfo.getCreditCode());  if(!StringUtil.isEmpty(creditCode)){  throw new HdiException(MessageFormat.format("医院统一社会信用代码{0}已存在",orgHospitalInfo.getCreditCode()));  }  String hospitalCode = sysSequenceService.selectSeqValueBySeqCode(SequenceEnum.HOSPITAL\_CODE.getKey());  orgHospitalInfo.setParentId(0L);  orgHospitalInfo.setHospitalCode(hospitalCode);  orgHospitalInfo.setIsGroup(IsGroupEnum.NO.getKey());  orgHospitalInfo.setCreateTime(new Date());  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.1.2.2.4编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 系统根据录入的“机构信用代码”判断该机构是否已存在，如果已存在，提示“本机构已存在”。 |
| 必填字段 | 机构名称、机构信用代码、状态、机构所属区域、医疗机构级别 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "hospitalGrade": 0,  "hospitalName": "测试医院",  "creditCode": "123",  "provinceCode": "44",  "cityCode": "4403",  "areaCode": "440301",  "status": 1  } |
| 业务代码 | @Override  public void updateValidate(OrgHospitalInfoEntity orgHospitalInfo) {  List<OrgHospitalInfoEntity> hospitalName = baseMapper.selectByHospitalName(orgHospitalInfo.getHospitalName());  // 如果修改的目标医院名已存在(此时必然有且仅有一个)且与当前修改记录不是同一条记录  if(!StringUtil.isEmpty(hospitalName) && hospitalName.get(0).getId() != orgHospitalInfo.getId()){  throw new HdiException(MessageFormat.format("医院名称{0}已存在",orgHospitalInfo.getHospitalName()));  }  // 统一社会信用代码同上  List<OrgHospitalInfoEntity> creditCode = baseMapper.selectByCreditCode(orgHospitalInfo.getCreditCode());  if(!StringUtil.isEmpty(creditCode) && creditCode.get(0).getId() != orgHospitalInfo.getId()){  throw new HdiException(MessageFormat.format("医院统一社会信用代码{0}已存在",orgHospitalInfo.getCreditCode())) ;  }    OrgHospitalInfoEntity hospital = baseMapper.selectById(orgHospitalInfo.getId());  if(StringUtil.isEmpty(orgHospitalInfo.getHospitalCode())) {  orgHospitalInfo.setHospitalCode(hospital.getHospitalCode());  }  if(StringUtil.isEmpty(orgHospitalInfo.getParentId())) {  orgHospitalInfo.setParentId(hospital.getParentId());  }  orgHospitalInfo.setCreateTime(hospital.getCreateTime());  orgHospitalInfo.setEditTime(new Date());  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.1.2.2.5查看详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 列表显示 | 机构名称、机构信用代码、状态、机构所属区域、医疗机构级别 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | http://localhost:2320/hdi-server-admin/org/orghospitalinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  @RequiresPermissions("org:orghospitalinfo:info")  public Hdi info(@PathVariable("id") Long id){  OrgHospitalInfoEntity orgHospitalInfo = orgHospitalInfoService.selectById(id);  return Hdi.ok().put("orgHospitalInfo", orgHospitalInfo);  } |
| 输出 | {  "msg": "success",  "code": 0,  "orgHospitalInfo": {  "id": 1,  "parentId": 0,  "hospitalCode": "4401111",  "hospitalGrade": 0,  "hospitalName": "中山一院11",  "creditCode": "00001992",  "provinceCode": "44",  "cityCode": "4403",  "areaCode": "440305",  "status": 1,  ……  }  } |

##### 5.1.2.2.6删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.org |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void delete(Long[] ids) {  for(Long id : ids) {  OrgHospitalInfoEntity orgHospitalInfoEntity= new OrgHospitalInfoEntity();  orgHospitalInfoEntity.setId(id);  //设置信息为删除状态  orgHospitalInfoEntity.setDelFlag((DelFlagEnum.DELETE.getKey()));  this.updateById(orgHospitalInfoEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.1.3厂商信息

#### 5.1.3.1用例图



#### 5.1.3.2功能设计说明

##### 5.1.3.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 厂商名称、厂商所属区域、状态 |
| 列表显示 | 厂商编码、厂商名称、厂商信用代码、厂商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "page":"1",  "limit":"10",  "factoryName":"",  "countryCode":"",  "provinceCode":"44",  "cityCode":"",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<OrgFactoryInfoEntity> page = this.selectPage(  new Query<OrgFactoryInfoEntity>(params).getPage(),  new EntityWrapper<OrgFactoryInfoEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("factoryName").toString()), "factory\_name", params.get("factoryName").toString())  .eq(StringUtils.isNotBlank(params.get("countryCode").toString()), "country\_code", params.get("countryCode").toString())  .eq(StringUtils.isNotBlank(params.get("provinceCode").toString()), "province\_code", params.get("provinceCode").toString())  .eq(StringUtils.isNotBlank(params.get("cityCode").toString()), "city\_code", params.get("cityCode").toString())  .eq(StringUtils.isNotBlank(params.get("areaCode").toString()), "area\_code", params.get("areaCode").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": "1",  "factoryCode": 1223,  "factoryName": "广州白云山医药有限公司",  "creditCode": "GZ0009381784",  "countryCode": "11",  "provinceCode": "44",  "cityCode": "44020",  "areaCode": "44020890",  "status": "1",  ......  },  {  "id": "2",  "factoryCode": 3456,  "factoryName": "广州白云山中一药业有限公司",  "creditCode": "GZ0009389834",  "countryCode": "11",  "provinceCode": "44",  "cityCode": "44020",  "areaCode": "44020890",  "status": "1",  ......  }......  ]  }  } |

##### 5.1.3.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、厂商编码（系统按规则自动生成唯一编码）  2、厂商名称，唯一性约束  3、厂商信用代码，唯一性约束 |
| 必填字段 | 厂商名称、厂商信用代码、厂商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 业务代码 | {  "factoryName":"广州白云山制药总厂",  "countryCode":"11",  "provinceCode":"44",  "cityCode":"4409",  "status":"1"  } |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void insert(OrgFactoryInfoEntity orgFactoryInfoEntity) {  //校验厂商名称、厂商信用代码  ......  //生成厂商编码  Long factoryCode= ......;  orgFactoryInfoEntity.setFactoryCode(factoryCode);  orgFactoryInfoEntity.setCreateId(getUserId());  orgFactoryInfoEntity.setCreateTime(new Date());  ......  this.insert(orgSupplierGroupEntity);  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

##### 5.1.3.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、厂商名称，唯一性约束  2、厂商信用代码，唯一性约束 |
| 必填字段 | 厂商名称、厂商信用代码、厂商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 业务代码 | {  "id":"1",  "factoryName":"广州白云山制药总厂",  "countryCode":"11",  "provinceCode":"44",  "cityCode":"4409",  "status":"1"  } |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(OrgFactoryInfoEntity orgFactoryInfoEntity) {  //校验厂商名称、厂商信用代码  ......  orgFactoryInfoEntity.setCreateId(getUserId());  orgFactoryInfoEntity.setCreateTime(new Date());  ......  this.updateById(orgSupplierGroupEntity);  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

##### 5.1.3.2.4查看详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 列表显示 | 厂商编码、厂商名称、厂商信用代码、厂商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | http://localhost:2320/hdi-server-admin/org/orgfactoryinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  OrgFactoryInfoEntity orgFactoryInfo = orgFactoryInfoService.selectById(id);  return Hdi.ok().put("orgFactoryInfo", orgFactoryInfo);  } |
| 输出 | {  "msg": "success",  "code": 0,  "orgFactoryInfo": {  "id": "1",  "factory\_code": 1223,  "factory\_name": "广州白云山医药有限公司",  "creditCode": "GZ0009381784",  "country\_code": "11",  "provinceCode": "44",  "cityCode": "44020",  "areaCode": "44020890",  "status": "1",  ......  },  } |

##### 5.1.3.2.4删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.org |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void delete(Long[] ids) {  for(Long id : ids) {  OrgFactoryInfoEntity orgFactoryInfo = new OrgFactoryInfoEntity();  orgFactoryInfo.setId(id);  orgFactoryInfo.setDelFlag(DelFlagEnum.DELETE.getKey());  this.updateById(orgFactoryInfo);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.1.4代理商信息

#### 5.1.4.1用例图



#### 5.1.4.2功能设计说明

##### 5.1.4.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 机构名称、状态 |
| 列表显示 | 1. 显示字段：机构编码、机构名称、状态、证照状态   2、证照状态（判断该代理商证照是否上传，如果已上传，统计显示代理商的证照数量，点击文字链接直接跳转到“代理商证照管理”页面）   1. 排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | {  "page":1,  "limit":10,  "agentName":"...医药有限公司",  "status":1,  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<OrgSupplierInfoEntity> page = this.selectPage(  new Query<OrgSupplierInfoEntity>(params).getPage(),  new EntityWrapper<OrgSupplierInfoEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("agentName")), "agent\_name", params.get("agentName"))  .eq(params.get("status")!=null, "status", params.get("status"))  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 7,  "agentCode": 44011128,  "agentName": "测试代理商1",  "creditCode": "dls001",  "status": 1,  ......  },  {  "id": 2,  "supplierId": 7,  "agentCode": 44011130,  "agentName": "测试代理商2",  "creditCode": "dls002",  "status": 1,  ......  }  ]  }  } |

##### 5.1.4.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、代理商编码（系统按规则自动生成唯一编码）  2、代理商名称，唯一性约束  3、代理商信用代码，唯一性约束 |
| 必填字段 | 代理商名称、代理商信用代码、代理商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 业务代码 | {  "agentName":"代理商测试",  "creditCode": "dls100",  "countryCode":"11",  "provinceCode":"44",  "cityCode":"4409",  "status":"1"  } |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(OrgAgentInfoEntity orgAgentInfoEntity) {  List<OrgAgentInfoEntity> agentName = baseMapper.selectByAgentName(orgAgentInfoEntity.getAgentName());  if(!StringUtil.isEmpty(agentName)){  throw new HdiException("代理商名称已存在");  }  List<OrgAgentInfoEntity> creditCode = baseMapper.selectByCreditCode(orgAgentInfoEntity.getCreditCode());  if(!StringUtil.isEmpty(creditCode)){  throw new HdiException("代理商信用代码已存在");  }  //获取供应商编码序列号  long seqValue = sysSequenceService.selectSeqValueBySeqCode(SequenceEnum.AGENT\_CODE.getKey());  long agentCode = Long.valueOf(orgAgentInfoEntity.getAreaCode() + String.valueOf(seqValue));  orgAgentInfoEntity.setAgentCode(agentCode);  orgAgentInfoEntity.setCreateTime(new Date());  this.insert(orgAgentInfoEntity);  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

##### 5.1.4.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、代理商名称，唯一性约束  2、代理商信用代码，唯一性约束 |
| 必填字段 | 代理商名称、代理商信用代码、代理商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 业务代码 | {  "id":"1",  "agentName":"代理商测试",  "creditCode": "dls100",  "countryCode":"11",  "provinceCode":"44",  "cityCode":"4409",  "status":"1"  ......  } |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(OrgAgentInfoEntity orgAgentInfoEntity) {  Long id = orgAgentInfoEntity.getId();  List<OrgAgentInfoEntity> agentName = baseMapper.selectByAgentName(orgAgentInfoEntity.getAgentName());  if(!StringUtil.isEmpty(agentName)&&agentName.get(0).getId()!=id){  throw new HdiException("代理商名称已存在");  }  List<OrgAgentInfoEntity> creditCode = baseMapper.selectByCreditCode(orgAgentInfoEntity.getCreditCode());  if(!StringUtil.isEmpty(creditCode)&&creditCode.get(0).getId()!=id){  throw new HdiException("代理商信用代码已存在");  }  orgAgentInfoEntity.setEditTime(new Date());  this.updateById(orgAgentInfoEntity);  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

##### 5.1.4.2.4查看详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 列表显示 | 代理编码、代理名称、代理商信用代码、代理商所属区域、状态 |
| 类包 | com.ebig.hdi.modules.org |
| 输入 | http://localhost:2320/hdi-server-admin/org/orgagentinfo/info/2 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  OrgAgentInfoEntity orgAgentInfo = orgAgentInfoService.selectById(id);  return Hdi.ok().put("orgAgentInfo", orgAgentInfo);  } |
| 输出 | {  "msg": "success",  "code": 0,  "orgFactoryInfo": {  "id": 1,  "supplierId": 7,  "agentCode": 44011128,  "agentName": "测试代理商1",  "creditCode": "dls001",  "status": 1,  ......  }  } |

##### 5.1.4.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.org |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void delete(Long[] ids) {  for(Long id : ids) {  OrgAgentInfoEntity orgAgentInfoEntity= new OrgAgentInfoEntity();  orgAgentInfoEntity.setId(id);  //设置信息为删除状态  orgAgentInfoEntity.setDelFlag((DelFlagEnum.DELETE.getKey()));  this.updateById(orgAgentInfoEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

## 5.2统一编码管理

### 5.2.1商品分类管理

#### 5.2.1.1查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 类别名称 |
| 列表显示 | 1、后台字段：类别ID、类别名称、类别码、上层类别、层级、层级顺序  2、根节点ID默认为0  3、排序：按照后台提供的层级和层级顺序排序 |
| 类包 | com.ebig.hdi.modules.consumables  com.ebig.hdi.modules.drugs  com.ebig.hdi.modules.reagent |
| 输入 | {  “pcateId”:”0”  } |
| 业务代码 | **public** List<TreeNode> queryNode(Long pcateId){  List<Map<String, Object>> queryForList = jdbcTemplate.queryForList("select \* from hdi\_unicode\_consumables\_cate where pcate\_id=?", pcateId);  **for** (Map<String, Object> map : queryForList) {  TreeNode treeNode = **new** TreeNode();  //**TODO** 1.增加其他属性 2.叶子节点判断  treeNode.setChildren(queryNode((Long)map.get("cate\_id")));  nodeList.add(treeNode);  }  **return** nodeList;  } |
| 查询语句 | 查询对应插件的分类表（全查） |
| 输出 | **{**  **"msg"**: "success"**,**  **"code"**: 0**,**  **"data"**: **[**  **{**  **"cateId"**: 1**,**  **"cateName"**: "针剂"**,**  **"cateNo"**: "ISO8808-01"**,**  **"levelOrder"**: 1**,**  **"cateLevel"**: 1**,**  **"pcateId"**: 0**,**  **"deptId"**: 1,  **"leaf"**: false,  **"children":[…]**  **},**  **{**  **"cateId"**: 2**,**  **"cateName"**: "片剂"**,**  **"cateNo"**: "ISO8808-02"**,**  **"levelOrder"**: 2**,**  **"cateLevel"**: 1**,**  **"pcateId"**: 0**,**  **"deptId"**: 1,  **"leaf"**: false,  **"children":[…]**  **}**  **]**  **}** |
|  |  |

#### 5.2.1.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 逻辑说明 | 1、前台传入字段：类别名称、类别码、上层类别ID、层级、层级顺序  2、同一层级，要求类别码不能重复  3、新增成功后返回新增的记录 |
| 类包 | com.ebig.hdi.modules.consumables  com.ebig.hdi.modules.drugs  com.ebig.hdi.modules.reagent |
| 输入 | {  "pcateId":0,  "cateName":"西药",  "cateNo":"ISO8808-01",  "cateLevel":1,  "levelOrder":1  } |
| 业务代码 | SQL如下：  select count(\*) from hdi\_unicode\_consumables\_cate where cate\_level=#{cateLevel} and cate\_no=#{cateNo}  若该值不为0,则不能新增 |
| 输出 | **{**  **"msg"**: "success"**,**  **"code"**: 0  **}** |
|  |  |

#### 5.2.1.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 逻辑说明 | 1、前台传入字段：类别ID、类别名称、类别码、上层类别ID、层级、层级顺序  2、同一层级，要求类别码不能重复 |
| 类包 | com.ebig.hdi.modules.consumables  com.ebig.hdi.modules.drugs  com.ebig.hdi.modules.reagent |
| 输入 | {  "cateId":1,  "pcateId":0,  "cateName":"西药",  "cateNo":"ISO8808-01",  "cateLevel":1,  "levelOrder":1  } |
| 业务代码 | SQL如下：  select count(\*) from hdi\_unicode\_consumables\_cate where cate\_level=#{cateLevel} and cate\_no=#{cateNo}  若该值不为0,则不能编辑 |
| 输出 | **{**  **"msg"**: "success"**,**  **"code"**: 0  **}** |
|  |  |

#### 5.2.1.4 删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 逻辑说明 | 1、前台传入字段：类别ID  2、树结构数据不要多选，删除时选中一节点，前端判断若有孩子，则询问是否级联删除所有孩子，传入类别ID、后台默认级联删除所有孩子。 |
| 类包 | com.ebig.hdi.modules.consumables  com.ebig.hdi.modules.drugs  com.ebig.hdi.modules.reagent |
| 输入 | {  "cateId":1  } |
| 业务代码 | **public** void deleteNode(Long cateId){  List<Map<String, Object>> queryForList = jdbcTemplate.queryForList("select \* from hdi\_unicode\_consumables\_cate where pcate\_id=?", cateId);  **for** (Map<String, Object> map : queryForList) {  //**TODO** 删除语句  deleteNode((Long)map.get("cate\_id")) }  //上面只是级联删除孩子，最后记得把自己也删了  } |
| 输出 | **{**  **"msg"**: "success"**,**  **"code"**: 0  **}** |
|  |  |

### 5.2.2供应商匹配管理

插件：hdi-server-admin

类包：com.ebig.hdi.modules.unicode

#### 5.2.2.1 分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 医院名称、医院供应商名称、匹配状态 |
| 列表显示 | 1. 后台返回字段：关系标识、医院名称、医院供应商编码、医院供应商名称、医院供应商信用代码、平台供应商信用代码、平台供应商名称、匹对状态、审批状态 |
| 输入 | {  "page":1,  "limit":10,  "hospitalName":"医院名称",  "supplyName":”医院供应商名称”,  "shipFlag":1  } |
| 业务代码 | 伪代码：Page.like(“hospitalName”, hospitalName).like(“supplyName”, supplyName).eq(“shipFlag”, shipFlag) |
| 查询语句 | 医院机构标识关联医院机构表、平台机构标识关联平台机构表 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "shipId": 1,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  },  {  "shipId": 2,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  },  {  "shipId": 3,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  }  ]  }  } |

#### 5.2.2.2 新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 输入 | {  "horgId":1,  "supplyNo":”医院供应商编码”,  "supplyName":"医院供应商名称",  "supplyCode":”医院供应商信用代码”,  "porgId":1  } |
| 业务代码 | 伪代码：Page.like(“hospitalName”, hospitalName).like(“supplyName”, supplyName).eq(“shipFlag”, shipFlag) |
| 代码逻辑 | 1. 新增时。若根据医院供应商信用代码和平台供应商信用代码查询医院供应商匹配表可以查出记录，则不能新增。 2. 数据来源手工新增默认为2 3. 删除标志默认为0 4. 匹配标志默认为1（前台必须要求两个信用代码必填） 5. 新增记录默认修改人与创建人信息一致 |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 平台供应商分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 平台供应商名称，平台供应商信用代码 |
| 输入 | {  "page":1,  "limit":10,  "porgName":"平台供应商名称",  "porgCode":”平台供应商信用代码,  } |
| 业务代码 | 伪代码：Page.like(“porgName”, porgName).eq(“porgCode”, porgCode)  只查询平台供应商表未匹配医院供应商的记录 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [{  “porgId”:1  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  },  {  “porgId”:2  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  },  {  “porgId”:3  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  }]  }  } |

#### 5.2.2.4 匹对查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 列表显示 | 后台返回字段：关系标识、医院供应商信用代码、医院供应商名称、平台供应商信用代码、平台供应商名称 |
| 输入 | {  "shipId":1,  "supplyName":”医院供应商名称”  } |
| 业务代码 | 配置匹配规则、注入MatchingRulesExecutor类调用matching方法拿到匹配结果。 |
| 输出 | {  "msg": "success",  "code": 0,  "data": {  "shipId": 1,  "supplyName": "医院供应商名称",  "supplyCode": “医院供应商信用代码”,  “list”:[{  "porgId": 1,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码"  “similarity”:1.00  },  {  "porgId":2,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.99  },  {  "porgId": 3,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.98  },  {  "porgId": 4,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.95  },  {  "porgId": 5,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.90  }]  }  } |

#### 5.2.2.5 编辑保存

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 输入 | {  "shipId":1,  "porgId":2  } |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.2.2.6 删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 输入 | [1,2,3……] |
| 业务代码 | deleteBatchIds、修改删除标志del\_flag为1 |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.2.3商品匹配管理

#### 5.2.3.1 分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 医院名称/供应商名称、医院商品名称/供应商商品名称、生产厂商、匹配状态、审批状态 |
| 列表显示 | 1. 后台返回字段：关系标识、医院名称/供应商名称、医院商品名称/供应商商品名称、医院商品规格/供应商商品规格、生产厂商、批准文号/注册证号、平台商品名称、平台商品规格、匹对状态、审批状态 |
| 输入 | 医院商品匹配：  {  "page":1,  "limit":10,  "hospitalName":"医院名称",  "supplyName":”医院供应商名称”,  "shipFlag":1  }  供应商商品匹配：  {  "page":1,  "limit":10,  "hospitalName":"医院名称",  "supplyName":”医院供应商名称”,  "shipFlag":1  } |
| 业务代码 | 伪代码：Page.like(“hospitalName”, hospitalName).like(“supplyName”, supplyName).eq(“shipFlag”, shipFlag) |
| 查询语句 | 医院机构标识关联医院机构表、平台机构标识关联平台机构表 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "shipId": 1,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  },  {  "shipId": 2,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  },  {  "shipId": 3,  "deptId": 1,  "horgId": 1,  "horgName": 医院名称,  "supplyNo": "药品名称",  "supplyName": "通用名称",  "supplyCode": “医院供应商信用代码”,  "porgId": "平台机构标识",  "porgName": "平台机构名称",  "shipFlag ": "20粒/盒",  "datasource": 2,  "delFlag": 1  }  ]  }  } |

#### 5.2.3.2 新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 输入 | {  "horgId":1,  "supplyNo":”医院供应商编码”,  "supplyName":"医院供应商名称",  "supplyCode":”医院供应商信用代码”,  "porgId":1  } |
| 业务代码 | 伪代码：Page.like(“hospitalName”, hospitalName).like(“supplyName”, supplyName).eq(“shipFlag”, shipFlag) |
| 代码逻辑 | 1. 新增时。若根据医院供应商信用代码和平台供应商信用代码查询医院供应商匹配表可以查出记录，则不能新增。 2. 数据来源手工新增默认为2 3. 删除标志默认为0 4. 匹配标志默认为1（前台必须要求两个信用代码必填） 5. 新增记录默认修改人与创建人信息一致 |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.2.3.3平台供应商分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 平台供应商名称，平台供应商信用代码 |
| 输入 | {  "page":1,  "limit":10,  "porgName":"平台供应商名称",  "porgCode":”平台供应商信用代码,  } |
| 业务代码 | 伪代码：Page.like(“porgName”, porgName).eq(“porgCode”, porgCode)  只查询平台供应商表未匹配医院供应商的记录 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [{  “porgId”:1  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  },  {  “porgId”:2  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  },  {  “porgId”:3  “porgName”:”平台供应商名称”,  “porgCode”:”平台供应商信用代码”  }]  }  } |

#### 5.2.3.4 匹对查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 列表显示 | 后台返回字段：关系标识、医院供应商信用代码、医院供应商名称、平台供应商信用代码、平台供应商名称 |
| 输入 | {  "shipId":1,  "supplyName":”医院供应商名称”  } |
| 业务代码 | 配置匹配规则、注入MatchingRulesExecutor类调用matching方法拿到匹配结果。 |
| 输出 | {  "msg": "success",  "code": 0,  "data": {  "shipId": 1,  "supplyName": "医院供应商名称",  "supplyCode": “医院供应商信用代码”,  “list”:[{  "porgId": 1,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码"  “similarity”:1.00  },  {  "porgId":2,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.99  },  {  "porgId": 3,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.98  },  {  "porgId": 4,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.95  },  {  "porgId": 5,  "porgName": "平台机构名称",  "porgCode": "平台机构信用代码",  “similarity”:0.90  }]  }  } |

#### 5.2.3.5 编辑保存

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 输入 | {  "shipId":1,  "porgId":2  } |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.2.3.6 删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 输入 | [1,2,3……] |
| 业务代码 | deleteBatchIds、修改删除标志del\_flag为1 |
| 输出 | {  "msg": "success",  "code": 0  } |

## 5.3商品管理

### 5.3.1平台药品信息

#### 5.3.1.1用例图



#### 5.3.1.2功能设计说明

##### 5.3.1.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 药品名称、商品属性、商品分类、状态、生产厂家、审批状态 |
| 列表显示 | 1、显示字段：商品统一编码、药品名称、商品属性、商品分类、商品规格、生产厂家、批准文号/注册证号、状态、审批状态  2、商品统一编码（系统自动按照统一编码规则生成）  3、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":1,  "limit":10,  "drugsName":"药品名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsPlatformDrugsEntityVo> page = new Page<GoodsPlatformDrugsEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsPlatformDrugsEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("drugsName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 平台药品表关联药品分类表、厂商信息表、平台药品规格表、平台药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "goodsUnicode": "PG201905060001",  "drugsCode": "PYP000000001",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "goodsUnicode": "PT201905060002",  "drugsCode": "PYP000000002",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.1.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、商品统一编码（系统自动按照统一编码规则生成，参考统一编码规范V4.0） 2、药品编码（系统自动按照统一编码规则生成）  3、药品名称，唯一性约束  4、选择生产厂商，不存在则在平台厂商信息先维护  5、商品下的规格和批准文号，唯一性约束 |
| 必填字段 | 药品名称、商品属性、药品分类、药品规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "status":1,  "goodsUnit":"1",  "specs":"8袋/盒",  "guid":"guid00000001",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsPlatformDrugsEntityVo goodsPlatformDrugsEntityVo) {  List<GoodsPlatformDrugsEntityVo> drugsName = baseMapper.selectByDrugsName(goodsPlatformDrugsEntityVo.getDrugsName());  if(!StringUtil.isEmpty(drugsName)){  throw new HdiException("药品名称已存在");  }  //判断药品规格是否存在（药品下规格唯一）  ....  //判断批准文号是否已经存在（药品下规格唯一）  .....  //生成药品编码  .......  //生成商品统一编码  .......  //上传商品图片到fastDFS  ......  //保存药品信息  goodsPlatformDrugsEntityVo.setCreateTime(new Date());  this.insert(goodsPlatformDrugsEntityVo);  //保存商品规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.1.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、药品名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，商品下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 药品名称、商品属性、药品分类、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "id":3,  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsPlatformDrugsEntityVo goodsPlatformDrugsEntityVo) {  //判断药品名称是否存在  if(存在){  throw new HdiException("药品名称已存在");  }  //判断批准文号是否已经存在（药品下规格唯一）  .....  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //生成商品统一编码  .......  //更新药品信息  goodsPlatformDrugsEntityVo.setEditTime(new Date());  this.updateById(goodsPlatformDrugsEntityVo);  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.1.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 商品统一编码、药品名称、药品通用名称、商品属性、商品分类名称、商品规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | http://localhost:2320/hdi-server-admin/drugs/goodsplatformdrugs/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsPlatformDrugsEntityVo goodsPlatformDrugsEntityVo = goodsPlatformDrugsService.selectPlatformDrugsById(id);  return Hdi.ok().put("goodsPlatformDrugs", goodsPlatformDrugsEntityVo );  } |
| 查询语句 | 平台药品表关联药品分类表、厂商信息表、平台药品规格表、平台药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsPlatformDrugs": {  "id": 3,  "drugsCode": "PYP000000001",  "drugsName": "感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.1.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.drugs |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsPlatformDrugsEntity goodsPlatformDrugsEntity = new GoodsPlatformDrugsEntity ();  goodsPlatformDrugsEntity .setId(id);  //设置信息为删除状态  goodsPlatformDrugsEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsPlatformDrugsEntity );  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.1.2.6更多规格

###### 5.3.1.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 平台药品规格（模糊匹配）、状态 |
| 列表显示 | 平台药品规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":"1",  "limit":"10",  "drugsId":1,  "specs":"8袋/盒",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsPlatformDrugsSpecsEntity> page = this.selectPage(  new Query<GoodsPlatformDrugsSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsPlatformDrugsSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "drugsId": 2,  "specsCode": "gg00002",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.1.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 平台药品ID、平台药品规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 业务代码 | [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "drugsId": null,  "specsCode": "",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsPlatformDrugsSpecsEntity> goodsPlatformDrugsSpecsList) {  for(GoodsPlatformDrugsSpecsEntity goodsPlatformDrugsSpecsEntity ： goodsPlatformDrugsSpecsList){  //判断规格是否存在(药品下规格唯一)  if(存在){  throw new HdiException("药品规格已存在");  }  }  for(GoodsPlatformDrugsSpecsEntity goodsPlatformDrugsSpecsEntity ： goodsPlatformDrugsSpecsList){  if(StringUtil.isEmpty(goodsPlatformDrugsSpecsEntity.getId())){  this.insert(goodsPlatformDrugsSpecsEntity);  }else{  this.updateById(goodsPlatformDrugsSpecsEntity);  }  }  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.2供应商药品信息

#### 5.3.2.1用例图



#### 5.3.2.2功能设计说明

##### 5.3.2.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 药品名称、商品属性、商品分类、状态、生产厂家、供应商名称 |
| 列表显示 | 1、显示字段：供应商名称、药品名称、商品属性、商品分类、商品规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":1,  "limit":10,  "drugsName":"药品名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsSupplierDrugsEntityVo> page = new Page<GoodsSupplierDrugsEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsSupplierDrugsEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("drugsName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 供应商药品表关联药品分类表、厂商信息表、供应商药品规格表、供应商药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId":1,  "drugsCode": "GYP000000001",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "supplierId":1,  "drugsCode": "GYP000000002",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.2.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 当前登陆用户为供应商，则供应商名称默认不能选择；如果是平台管理员，则供应商名称可以选择 2. 药品编码（系统自动按照统一编码规则生成）   3、药品名称，在供应商范围内唯一性约束  4、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  5、商品下的规格和批准文号，唯一性约束 |
| 必填字段 | 供应商名称、药品名称、商品属性、药品分类、药品规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "supplierId":1,  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "specs":"8袋/盒",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsSupplierDrugsEntityVo goodsSupplierDrugsEntityVo) {  List<GoodsSupplierDrugsEntityVo> drugsName = baseMapper.selectBySupplierIdAndDrugsName(goodsSupplierDrugsEntityVo.getSupplierId(), goodsSupplierDrugsEntityVo.getDrugsName());  if(!StringUtil.isEmpty(drugsName)){  throw new HdiException("药品名称已存在");  }  //判断药品规格是否存在（药品下规格唯一）  ....  //判断批准文号是否已经存在（药品下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsSupplierDrugsEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //生成药品编码  .......  //上传商品图片到fastDFS  ......  //保存药品信息  goodsSupplierDrugsEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierDrugsEntityVo.setCreateTime(new Date());  this.insert(goodsPlatformDrugsEntityVo);  //保存商品规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.2.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、药品名称，在供应商范围内唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，商品下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 药品名称、商品属性、药品分类、药品规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "id":3,  "supplierId":1,  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsSupplierDrugsEntityVo goodsSupplierDrugsEntityVo ) {  //判断药品名称是否存在  if(存在){  throw new HdiException("药品名称已存在");  }  //判断批准文号是否已经存在（药品下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsSupplierDrugsEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新药品信息  goodsSupplierDrugsEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierDrugsEntityVo.setEditTime(new Date());  this.updateById(goodsSupplierDrugsEntityVo);  //更新药品规格或保存药品规格  .....  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.1.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 药品名称、药品通用名称、商品属性、商品分类名称、商品规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | http://localhost:2320/hdi-server-admin/drugs/goodssupplierdrugs/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsSupplierDrugsEntityVo goodsSupplierDrugsEntityVo= goodsSupplierDrugsService.selectSupplierDrugsById(id);  return Hdi.ok().put("goodsSupplierDrugs", goodsSupplierDrugsEntityVo);  } |
| 查询语句 | 供应商药品表关联药品分类表、厂商信息表、供应商药品规格表、平台药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsPlatformDrugs": {  "id": 3,  "supplierId":1,  "drugsCode": "GYP000000001",  "drugsName": "感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.2.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.drugs |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsSupplierDrugsEntity goodsSupplierDrugsEntity = new GoodsSupplierDrugsEntity ();  goodsSupplierDrugsEntity.setId(id);  //设置信息为删除状态  goodsSupplierDrugsEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsSupplierDrugsEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.2.2.6更多规格

###### 5.3.2.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 供应商药品规格（模糊匹配）、状态 |
| 列表显示 | 供应商药品规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":"1",  "limit":"10",  "drugsId":1,  "specs":"8袋/盒",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsSupplierDrugsSpecsEntity> page = this.selectPage(  new Query<GoodsSupplierDrugsSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsSupplierDrugsSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "drugsId": 2,  "specsCode": "gg00002",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.2.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 供应商药品ID、供应商药品规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 业务代码 | [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "drugsId": null,  "specsCode": "",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsSupplierDrugsSpecsEntity> goodsSupplierDrugsSpecsList) {  for(GoodsSupplierDrugsSpecsEntity goodsSupplierDrugsSpecsEntity ： goodsSupplierDrugsSpecsList){  //判断规格是否存在(药品下规格唯一)  if(存在){  throw new HdiException("药品规格已存在");  }  }  for(GoodsSupplierDrugsSpecsEntity goodsSupplierDrugsSpecsEntity ： goodsSupplierDrugsSpecsList){  if(StringUtil.isEmpty(goodsSupplierDrugsSpecsEntity.getId())){  this.insert(goodsSupplierDrugsSpecsEntity);  }else{  this.updateById(goodsSupplierDrugsSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.3医院药品信息

#### 5.3.3.1用例图



#### 5.3.3.2功能设计说明

##### 5.3.3.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 药品名称、商品属性、商品分类、状态、生产厂家、供应商名称 |
| 列表显示 | 1、显示字段：医院名称、药品名称、商品属性、商品分类、商品规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":1,  "limit":10,  "drugsName":"药品名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsHospitalDrugsEntityVo> page = new Page<GoodsHospitalDrugsEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsHospitalDrugsEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("drugsName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 医院药品表关联药品分类表、厂商信息表、医院药品规格表、医院药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "hospitalId":1,  "drugsCode": "YYP000000001",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specs": "20粒/盒",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "hospitalId":1,  "drugsCode": "YYP000000002",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.3.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 药品名称，在医院范围内唯一性约束   2、药品编码（系统自动按照统一编码规则生成）  3、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  4、商品下的规格和批准文号，唯一性约束 |
| 必填字段 | 医院名称、药品名称、商品属性、药品分类、药品规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "hospitalId":1,  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "specs":"8袋/盒",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsHospitalDrugsEntityVo goodsHospitalDrugsEntityVo) {  List<GoodsHospitalDrugsEntityVo> drugsName = baseMapper.selectByHospitalIdAndDrugsName(goodsHospitalDrugsEntityVo.getHospitalId(), goodsHospitalDrugsEntityVo.getDrugsName());  if(!StringUtil.isEmpty(drugsName)){  throw new HdiException("药品名称已存在");  }  //判断药品规格是否存在（药品下规格唯一）  ....  //判断批准文号是否已经存在（药品下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalDrugsEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //生成药品编码  .......  //上传商品图片到fastDFS  ......  //保存药品信息  goodsHospitalDrugsEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalDrugsEntityVo.setCreateTime(new Date());  this.insert(goodsHospitalDrugsEntityVo);  //保存商品规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.3.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、药品名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，商品下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 药品名称、商品属性、药品分类、药品规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "id":3,  "hospitalId":1,  "drugsName":"感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsHospitalDrugsEntityVo goodsHospitalDrugsEntityVo) {  //判断药品名称是否存在  if(存在){  throw new HdiException("药品名称已存在");  }  //判断批准文号是否已经存在（药品下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalDrugsEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新药品信息  goodsHospitalDrugsEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalDrugsEntityVo.setEditTime(new Date());  this.updateById(goodsHospitalDrugsEntityVo);  //更新药品规格或保存药品规格  .....  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.3.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 药品名称、药品通用名称、商品属性、商品分类名称、商品规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | http://localhost:2320/hdi-server-admin/drugs/goodshospitaldrugs/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsHospitalDrugsEntityVo goodsHospitalDrugsEntityVo= goodsHospitalDrugsService.selectHospitalDrugsById(id);  return Hdi.ok().put("goodsHospitalDrugs", goodsHospitalDrugsEntityVo);  } |
| 查询语句 | 医院药品表关联药品分类表、厂商信息表、医院药品规格表、平台药品批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsPlatformDrugs": {  "id": 3,  "hospitalId":1,  "drugsCode": "YYP000000001",  "drugsName": "感冒灵颗粒",  "commonName": "感冒灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "8袋/盒",  "status": 1,  ......  },  {  "id": 2,  "specs": "10袋/盒",  "status": 0,  ......  },  {  "id": 3,  "specs": "10袋/盒",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.3.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.drugs |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsHospitalDrugsEntity goodsHospitalDrugsEntity= new GoodsHospitalDrugsEntity ();  goodsHospitalDrugsEntity.setId(id);  //设置信息为删除状态  goodsHospitalDrugsEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsHospitalDrugsEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.3.2.6更多规格

###### 5.3.3.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院药品规格（模糊匹配）、状态 |
| 列表显示 | 医院药品规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 输入 | {  "page":"1",  "limit":"10",  "hospitalId":1,  "specs":"8袋/盒",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsHospitalDrugsSpecsEntity> page = this.selectPage(  new Query<GoodsHospitalDrugsSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsHospitalDrugsSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "drugsId": 2,  "specsCode": "gg00002",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.3.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 医院药品ID、供应商药品规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.drugs |
| 业务代码 | [  {  "id": 1,  "drugsId": 1,  "specsCode": "gg00001",  "specs": "8袋/盒",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "drugsId": null,  "specsCode": "",  "specs": "10袋/盒",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsHospitalDrugsSpecsEntity> goodsHospitalDrugsSpecsList) {  for(GoodsHospitalDrugsSpecsEntity goodsHospitalDrugsSpecsEntity ： goodsHospitalDrugsSpecsList){  //判断规格是否存在(药品下规格唯一)  if(存在){  throw new HdiException("药品规格已存在");  }  }  for(GoodsHospitalDrugsSpecsEntity goodsHospitalDrugsSpecsEntity ： goodsHospitalDrugsSpecsList){  if(StringUtil.isEmpty(goodsHospitalDrugsSpecsEntity.getId())){  this.insert(goodsHospitalDrugsSpecsEntity);  }else{  this.updateById(goodsHospitalDrugsSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.4平台试剂信息

#### 5.3.4.1用例图



#### 5.3.4.2功能设计说明

##### 5.3.4.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 试剂名称、商品属性、试剂分类、状态、生产厂家、审批状态 |
| 列表显示 | 1、显示字段：商品统一编码、试剂名称、商品属性、试剂分类、试剂规格、生产厂家、批准文号/注册证号、状态、审批状态  2、商品统一编码（系统自动按照统一编码规则生成）  3、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":1,  "limit":10,  "reagentName":"试剂名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsPlatformReagentEntityVo> page = new Page<GoodsPlatformReagentEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsPlatformReagentEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("reagentName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 平台试剂表关联试剂分类表、厂商信息表、平台试剂规格表、平台试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "goodsUnicode": "PG201905060001",  "reagentCode": "PYP000000001",  "reagentName": "试剂名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "100ml/袋",  "status": 1,  ......  },  {  "id": 2,  "specs": "250ml/袋",  "status": 0,  ......  },  {  "id": 3,  "specs": "50ml/袋",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "goodsUnicode": "PT201905060002",  "reagentCode": "PYP000000002",  "reagentName": "试剂名称",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "100ml/包",  "status": 1,  ......  },  {  "id": 2,  "specs": "200ml/包",  "status": 0,  ......  },  {  "id": 3,  "specs": "250ml/包",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.4.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、商品统一编码（系统自动按照统一编码规则生成，参考统一编码规范V4.0） 2、试剂编码（系统自动按照统一编码规则生成）  3、试剂名称，唯一性约束  4、选择生产厂商，不存在则在平台厂商信息先维护  5、商品下的规格和批准文号，唯一性约束 |
| 必填字段 | 试剂名称、商品属性、试剂分类、试剂规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "reagentName":"XX试剂",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "status":1,  "goodsUnit":"1",  "specs":"100ml/袋",  "guid":"guid00000001",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsPlatformReagentEntityVo goodsPlatformReagentEntityVo) {  List<GoodsPlatformReagentEntityVo> reagentName = baseMapper.selectByReagentName(goodsPlatformReagentEntityVo.getReagentName());  if(!StringUtil.isEmpty(reagentName)){  throw new HdiException("试剂名称已存在");  }  //判断试剂规格是否存在（试剂下规格唯一）  ....  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //生成试剂编码  .......  //生成商品统一编码  .......  //上传商品图片到fastDFS  ......  //保存试剂信息  goodsPlatformReagentEntityVo.setCreateTime(new Date());  this.insert(goodsPlatformReagentEntityVo);  //保存商品规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.4.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、试剂名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，试剂下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 试剂名称、商品属性、试剂分类、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "id":3,  "reagentName":"XX试剂",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsPlatformReagentEntityVo goodsPlatformReagentEntityVo) {  //判断试剂名称是否存在  if(存在){  throw new HdiException("试剂名称已存在");  }  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //生成商品统一编码  .......  //更新试剂信息  goodsPlatformReagentEntityVo.setEditTime(new Date());  this.updateById(goodsPlatformReagentEntityVo);  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.4.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 商品统一编码、试剂名称、试剂通用名称、商品属性、试剂分类名称、试剂规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | http://localhost:2320/hdi-server-admin/reagent/goodsplatformreagent/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsPlatformReagentEntityVo goodsPlatformReagentEntityVo= goodsPlatformReagentService.selectPlatformReagentById(id);  return Hdi.ok().put("goodsPlatformReagent", goodsPlatformReagentEntityVo);  } |
| 查询语句 | 平台试剂表关联试剂分类表、厂商信息表、平台试剂规格表、平台试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsPlatformReagent": {  "id": 3,  "reagentCode": "PSJ000000001",  "reagentName":"XX试剂",  "commonName": "贝特灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "250ml/包",  "status": 1,  ......  },  {  "id": 2,  "specs": "50ml/包",  "status": 0,  ......  },  {  "id": 3,  "specs": "150ml/包",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.4.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.reagent |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsPlatformReagentEntity goodsPlatformReagentEntity= new GoodsPlatformReagentEntity();  goodsPlatformReagentEntity.setId(id);  //设置信息为删除状态  goodsPlatformReagentEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsPlatformReagentEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.4.2.6更多规格

###### 5.3.4.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 平台试剂规格（模糊匹配）、状态 |
| 列表显示 | 平台试剂规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":"1",  "limit":"10",  "reagentId":1,  "specs":"250ml/包",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsPlatformReagentSpecsEntity> page = this.selectPage(  new Query<GoodsPlatformReagentSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsPlatformReagentSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "PSJ00001",  "specs": "250ml/包",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "PSJ00002",  "specs": "50ml/包",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.4.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 平台试剂ID、平台试剂规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 业务代码 | [  {  "id": 1,  "reagentId": 1,  "specsCode": "PSJ00001",  "specs": "250ml/包",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "reagentId": null,  "specsCode": "",  "specs": "50ml/包",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsPlatformReagentSpecsEntity> goodsPlatformReagentSpecsList) {  for(GoodsPlatformReagentSpecsEntity goodsPlatformReagentSpecsEntity ：goodsPlatformReagentSpecsList){  //判断规格是否存在(试剂下规格唯一)  if(存在){  throw new HdiException("试剂规格已存在");  }  }  for(GoodsPlatformReagentSpecsEntity goodsPlatformReagentSpecsEntity： goodsPlatformReagentSpecsList){  if(StringUtil.isEmpty(goodsPlatformReagentSpecsEntity.getId())){  this.insert(goodsPlatformReagentSpecsEntity);  }else{  this.updateById(goodsPlatformReagentSpecsEntity);  }  }  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.5供应商试剂信息

#### 5.3.5.1用例图



#### 5.3.5.2功能设计说明

##### 5.3.5.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 试剂名称、商品属性、试剂分类、状态、生产厂家、供应商名称、审批状态 |
| 列表显示 | 1、显示字段：供应商名称、试剂名称、商品属性、试剂分类、商品规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":1,  "limit":10,  "reagentName":"试剂名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsSupplierReagentEntityVo> page = new Page<GoodsSupplierReagentEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsSupplierReagentEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("reagentName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 供应商试剂表关联试剂分类表、厂商信息表、供应商试剂规格表、供应商试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId":1,  "reagentCode": "GSJ000000001",  "reagentName": "试剂名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "250ml/包",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/包",  "status": 0,  ......  },  {  "id": 3,  "specs": "50ml/包",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "supplierId":1,  "reagentCode": "GSJ000000002",  "reagentName": "试剂名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "50ml/包",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/包",  "status": 0,  ......  },  {  "id": 3,  "specs": "250ml/包",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.5.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 当前登陆用户为供应商，则供应商名称默认不能选择；如果是平台管理员，则供应商名称可以选择 2. 试剂编码（系统自动按照统一编码规则生成）   3、试剂名称，在供应商范围内唯一性约束  4、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  5、试剂下的规格和批准文号，唯一性约束 |
| 必填字段 | 供应商名称、试剂名称、商品属性、试剂分类、试剂规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "supplierId":1,  "reagentName":"试剂名称",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "specs":"250ml/袋",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsSupplierReagentEntityVo goodsSupplierReagentEntityVo) {  List<GoodsSupplierReagentEntityVo> reagentName = baseMapper.selectBySupplierIdAndReagentName(goodsSupplierReagentEntityVo.getSupplierId(), goodsSupplierReagentEntityVo.getReagentName());  if(!StringUtil.isEmpty(reagentName)){  throw new HdiException("试剂名称已存在");  }  //判断试剂规格是否存在（试剂下规格唯一）  ....  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsSupplierReagentEntityVo.getFactoryId())){  //试剂生产厂商信息，状态为待审批  .......  }  //生成试剂编码  .......  //上传商品图片到fastDFS  ......  //保存试剂信息  goodsSupplierReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierReagentEntityVo.setCreateTime(new Date());  this.insert(goodsSupplierReagentEntityVo);  //保存试剂规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.5.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、试剂名称，在供应商范围内唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，试剂下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 试剂名称、商品属性、试剂分类、试剂规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "id":3,  "supplierId":1,  "reagentName":"试剂名称",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsSupplierReagentEntityVo goodsSupplierReagentEntityVo) {  //判断试剂名称是否存在  if(存在){  throw new HdiException("试剂名称已存在");  }  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsSupplierReagentEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新试剂信息  goodsSupplierReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierReagentEntityVo.setEditTime(new Date());  this.updateById(goodsSupplierReagentEntityVo);  //更新试剂规格或保存试剂规格  .....  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.5.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 试剂名称、试剂通用名称、商品属性、商品分类名称、商品规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | http://localhost:2320/hdi-server-admin/reagent/goodssupplierreagent/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsSupplierReagentEntityVo goodsSupplierReagentEntityVo= goodsSupplierReagentService.selectSupplierReagentById(id);  return Hdi.ok().put("goodsSupplierReagent", goodsSupplierReagentEntityVo);  } |
| 查询语句 | 供应商试剂表关联试剂分类表、厂商信息表、供应商试剂规格表、平台试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsSupplierReagent": {  "id": 3,  "supplierId":1,  "reagentCode": "GSJ000000001",  "reagentName": "试剂名称",  "commonName": "贝特灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "250ml/袋",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/袋",  "status": 0,  ......  },  {  "id": 3,  "specs": "50ml/袋",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.5.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.reagent |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsSupplierReagentEntity goodsSupplierReagentEntity= new GoodsSupplierReagentEntity();  GoodsSupplierReagentEntity.setId(id);  //设置信息为删除状态  goodsSupplierReagentEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsSupplierReagentEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.5.2.6更多规格

###### 5.3.5.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 供应商试剂规格（模糊匹配）、状态 |
| 列表显示 | 供应商试剂规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":"1",  "limit":"10",  "reagentId":1,  "specs":"50ml/袋",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsSupplierReagentSpecsEntity> page = this.selectPage(  new Query<GoodsSupplierReagentSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsSupplierReagentSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": 50ml/袋",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "gg00002",  "specs": "55ml/袋",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.5.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 供应商试剂ID、供应商试剂规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 业务代码 | [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": "50ml/袋",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "reagentId": null,  "specsCode": "",  "specs": "55ml/袋",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsSupplierReagentSpecsEntity> goodsSupplierReagentSpecsList) {  for(GoodsSupplierReagentSpecsEntity goodsSupplierReagentSpecsEntity ： goodsSupplierReagentSpecsList){  //判断规格是否存在(试剂下规格唯一)  if(存在){  throw new HdiException("试剂规格已存在");  }  }  for(GoodsSupplierReagentSpecsEntity goodsSupplierReagentSpecsEntity ： goodsSupplierReagentSpecsList){  if(StringUtil.isEmpty(goodsSupplierReagentSpecsEntity.getId())){  this.insert(goodsSupplierReagentSpecsEntity);  }else{  this.updateById(goodsSupplierReagentSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.6医院试剂信息

#### 5.3.6.1用例图



#### 5.3.6.2功能设计说明

##### 5.3.6.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 试剂名称、商品属性、试剂分类、状态、生产厂家、供应商名称、审核状态 |
| 列表显示 | 1、显示字段：医院名称、试剂名称、商品属性、试剂分类、试剂规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":1,  "limit":10,  "reagentName":"试剂名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsHospitalReagentEntityVo> page = new Page<GoodsHospitalReagentEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsHospitalReagentEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("reagentName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 医院试剂表关联试剂分类表、厂商信息表、医院试剂规格表、医院试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "hospitalId":1,  "reagentCode": "YYSJ000000001",  "reagentName": "试剂名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specs": "20粒/盒",  "specsList": [  {  "id": 1,  "specs": "50ml/袋",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/袋",  "status": 0,  ......  },  {  "id": 3,  "specs": "250ml/袋",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "hospitalId":1,  "reagentCode": "YYSJ000000001",  "reagentName": "试剂名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "50ml/袋",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/袋",  "status": 0,  ......  },  {  "id": 3,  "specs": "250ml/袋",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.6.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 试剂名称，在医院范围内唯一性约束   2、试剂编码（系统自动按照统一编码规则生成）  3、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  4、试剂下的规格和批准文号，唯一性约束 |
| 必填字段 | 医院名称、试剂名称、商品属性、试剂分类、试剂规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "hospitalId":1,  "reagentName":"试剂名称",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "specs":"50ml/袋",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "国药准字H10970001,国药准字H10970002,国药准字H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsHospitalReagentEntityVo goodsHospitalReagentEntityVo) {  List<GoodsHospitalReagentEntityVo> reagentName = baseMapper.selectByHospitalIdAndReagentName(goodsHospitalReagentEntityVo.getHospitalId(), goodsHospitalReagentEntityVo.getDrugsName());  if(!StringUtil.isEmpty(reagentName)){  throw new HdiException("试剂名称已存在");  }  //判断规格是否存在（试剂下规格唯一）  ....  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalReagentEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //生成试剂编码  .......  //上传商品图片到fastDFS  ......  //保存试剂信息  goodsHospitalReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalReagentEntityVo.setCreateTime(new Date());  this.insert(goodsHospitalReagentEntityVo);  //保存试剂规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.6.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、试剂名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，试剂下的批准文号唯一性约束  4、编辑是没有试剂规格和全球唯一码字段 |
| 必填字段 | 试剂名称、商品属性、试剂分类、试剂规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "id":3,  "hospitalId":1,  "reagentName":"试剂名称",  "commonName": "贝特灵",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"国药准字H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"国药准字H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"国药准字H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsHospitalReagentEntityVo goodsHospitalReagentEntityVo ) {  //判断试剂名称是否存在  if(存在){  throw new HdiException("试剂名称已存在");  }  //判断批准文号是否已经存在（试剂下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalReagentEntityVo .getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新试剂信息  goodsHospitalReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalReagentEntityVo .setEditTime(new Date());  this.updateById(goodsHospitalReagentEntityVo );  //更新试剂规格或保存试剂规格  .....  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.6.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 试剂名称、试剂通用名称、商品属性、试剂分类名称、试剂规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | http://localhost:2320/hdi-server-admin/reagent/goodshospitalreagent/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsHospitalReagentEntityVo goodsHospitalReagentEntityVo = goodsHospitallReagentService.selectHospitallReagentById(id);  return Hdi.ok().put("goodsHospitalReagent", goodsHospitalReagentEntityVo);  } |
| 查询语句 | 医院试剂表关联试剂分类表、厂商信息表、医院试剂规格表、平台试剂批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsHospitalReagent": {  "id": 3,  "hospitalId":1,  "reagentCode": "YYSJ000000001",  "reagentName": "试剂名称",  "commonName": "贝特灵",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "50ml/袋",  "status": 1,  ......  },  {  "id": 2,  "specs": "150ml/袋",  "status": 0,  ......  },  {  "id": 3,  "specs": "250ml/袋",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "国药准字H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "国药准字H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "国药准字H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.6.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.reagent |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsHospitalReagentEntity goodsHospitalReagentEntity= new GoodsHospitalReagentEntity();  goodsHospitalReagentEntity.setId(id);  //设置信息为删除状态  goodsHospitalReagentEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsHospitalReagentEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.6.2.6更多规格

###### 5.3.6.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院试剂规格（模糊匹配）、状态 |
| 列表显示 | 医院试剂规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 输入 | {  "page":"1",  "limit":"10",  "hospitalId":1,  "specs":"50ml/袋",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsHospitalReagentSpecsEntity> page = this.selectPage(  new Query<GoodsHospitalReagentSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsHospitalReagentSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": "50ml/袋",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "gg00002",  "specs": "150ml/袋",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.6.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 医院试剂ID、医院试剂规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.reagent |
| 业务代码 | [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": "50ml/袋",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "reagentId": null,  "specsCode": "",  "specs": "150ml/袋",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsHospitalReagentSpecsEntity> goodsHospitalReagentSpecsList) {  for(GoodsHospitalReagentSpecsEntity goodsHospitalReagentSpecsEntity： goodsHospitalReagentSpecsList){  //判断规格是否存在(试剂下规格唯一)  if(存在){  throw new HdiException("试剂规格已存在");  }  }  for(GoodsHospitalReagentSpecsEntity goodsHospitalReagentSpecsEntity ：goodsHospitalReagentSpecsList){  if(StringUtil.isEmpty(goodsHospitalReagentSpecsEntity.getId())){  this.insert(goodsHospitalReagentSpecsEntity);  }else{  this.updateById(goodsHospitalReagentSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.7平台耗材信息

#### 5.3.7.1用例图



#### 5.3.7.2功能设计说明

##### 5.3.7.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 数据权限 | 根据当前用户角色分配的数据权限获取数据 |
| 查询条件 | 耗材名称、商品属性、耗材分类、状态、生产厂家、审批状态 |
| 列表显示 | 1、显示字段：商品统一编码、耗材名称、商品属性、耗材分类、耗材规格、生产厂家、批准文号/注册证号、状态、审批状态  2、商品统一编码（系统自动按照统一编码规则生成）  3、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":1,  "limit":10,  "consumablesName":"耗材名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsPlatformConsumablesEntityVo> page = new Page<GoodsPlatformConsumablesEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsPlatformConsumablesEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("consumablesName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 平台耗材表关联耗材分类表、厂商信息表、平台耗材规格表、平台耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "goodsUnicode": "PG201905060001",  "consumablesCode": "PHC000000001",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "goodsUnicode": "PT201905060002",  "consumablesCode": "PHC000000002",  "consumablesName": "耗材名称",  "drugsName": "药品名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.7.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、商品统一编码（系统自动按照统一编码规则生成，参考统一编码规范V4.0） 2、耗材编码（系统自动按照统一编码规则生成）  3、耗材名称，唯一性约束  4、选择生产厂商，不存在则在平台厂商信息先维护  5、商品下的规格和批准文号，唯一性约束 |
| 必填字段 | 耗材名称、商品属性、耗材分类、耗材规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "consumablesName":"XX耗材",  "commonName": "耗材通用名",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "status":1,  "goodsUnit":"1",  "specs":"5mm",  "guid":"guid00000001",  "approvals": "注册证号H10970001,注册证号H10970002,注册证号H10970003",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsPlatformConsumablesEntityVo goodsPlatformConsumablesEntityVo ) {  List<GoodsPlatformConsumablesEntityVo> consumablesName = baseMapper.selectByReagentName(GoodsPlatformConsumablesEntityVo.getConsumablesName());  if(!StringUtil.isEmpty(consumablesName)){  throw new HdiException("耗材名称已存在");  }  //判断试剂规格是否存在（耗材下规格唯一）  ....  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //生成耗材编码  .......  //生成商品统一编码  .......  //上传商品图片到fastDFS  ......  //保存耗材信息  goodsPlatformConsumablesEntityVo.setCreateTime(new Date());  this.insert(goodsPlatformConsumablesEntityVo);  //保存耗材规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.7.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、耗材名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，耗材下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 耗材名称、商品属性、耗材分类、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "id":3,  "consumablesName":"XX耗材",  "commonName": "耗材通用名称",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"注册证号H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"注册证号H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"注册证号H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsPlatformConsumablesEntityVo goodsPlatformConsumablesEntityVo) {  //判断耗材名称是否存在  if(存在){  throw new HdiException("耗材名称已存在");  }  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //生成商品统一编码  .......  //更新耗材信息  goodsPlatformConsumablesEntityVo.setEditTime(new Date());  this.updateById(goodsPlatformConsumablesEntityVo);  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.7.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 商品统一编码、耗材名称、耗材通用名称、商品属性、耗材分类名称、耗材规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | http://localhost:2320/hdi-server-admin/consumables/goodsplatformconsumables/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsPlatformConsumablesEntityVo goodsPlatformConsumablesEntityVo = goodsPlatformConsumablesService.selectPlatformConsumablesById(id);  return Hdi.ok().put("goodsPlatformConsumables", goodsPlatformConsumablesEntityVo);  } |
| 查询语句 | 平台耗材表关联试剂分类表、厂商信息表、平台耗材规格表、平台耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsPlatformReagent": {  "id": 3,  "consumablesCode": "PHC000000001",  "consumablesName":"XX耗材",  "commonName": "耗材通用名称",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.7.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.consumables |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsPlatformConsumablesEntity goodsPlatformConsumablesEntity= new GoodsPlatformConsumablesEntity();  goodsPlatformConsumablesEntity.setId(id);  //设置信息为删除状态  goodsPlatformConsumablesEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsPlatformConsumablesEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.7.2.6更多规格

###### 5.3.7.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 平台耗材规格（模糊匹配）、状态 |
| 列表显示 | 平台耗材规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":"1",  "limit":"10",  "consumablesId":1,  "specs":"5mm",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsPlatformConsumablesSpecsEntity> page = this.selectPage(  new Query<GoodsPlatformConsumablesSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsPlatformConsumablesSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "PSJ00001",  "specs": "250ml/包",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "PSJ00002",  "specs": "50ml/包",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.7.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 平台耗材ID、平台耗材规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 业务代码 | [  {  "id": 1,  "consumablesId": 1,  "specsCode": "PSJ00001",  "specs": "5mm",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "consumablesId": null,  "specsCode": "",  "specs": "15mm",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsPlatformConsumablesSpecsEntity> goodsPlatformConsumablesSpecsList) {  for(GoodsPlatformConsumablesSpecsEntity goodsPlatformConsumablesSpecsEntity ：goodsPlatformConsumablesSpecsList){  //判断规格是否存在(耗材下规格唯一)  if(存在){  throw new HdiException("耗材规格已存在");  }  }  for(GoodsPlatformConsumablesSpecsEntity goodsPlatformConsumablesSpecsEntity： goodsPlatformConsumablesSpecsList){  if(StringUtil.isEmpty(goodsPlatformConsumablesSpecsEntity.getId())){  this.insert(goodsPlatformConsumablesSpecsEntity);  }else{  this.updateById(goodsPlatformConsumablesSpecsEntity);  }  }  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.8供应商耗材信息

#### 5.3.8.1用例图



#### 5.3.8.2功能设计说明

##### 5.3.8.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 耗材名称、商品属性、耗材分类、状态、生产厂家、供应商名称、审批状态 |
| 列表显示 | 1、显示字段：供应商名称、耗材名称、商品属性、耗材分类、耗材规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":1,  "limit":10,  "consumablesName":"耗材名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsSupplierConsumablesEntityVo> page = new Page<GoodsSupplierConsumablesEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsSupplierConsumablesEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("consumablesName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 供应商耗材表关联耗材分类表、厂商信息表、供应商耗材规格表、供应商耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId":1,  "consumablesCode": "GHC000000001",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "supplierId":1,  "consumablesCode": "GHC000000002",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.8.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 当前登陆用户为供应商，则供应商名称默认不能选择；如果是平台管理员，则供应商名称可以选择 2. 耗材编码（系统自动按照统一编码规则生成）   3、耗材名称，在供应商范围内唯一性约束  4、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  5、耗材下的规格和批准文号，唯一性约束 |
| 必填字段 | 供应商名称、耗材名称、商品属性、耗材分类、耗材规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "supplierId":1,  "consumablesName":"耗材名称",  "commonName": "螺丝钉",  "goodsNature":0,  "typeId":1,  "specs":"5mm",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "注册证号H10970001,注册证号H10970002,注册证号H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsSupplierConsumablesEntityVo goodsSupplierConsumablesEntityVo) {  List<GoodsSupplierConsumablesEntityVo> consumablesName = baseMapper.selectBySupplierIdAndrConsumablesName(goodsSupplierConsumablesEntityVo.getSupplierId(), goodsSupplierConsumablesEntityVo.getConsumablesName());  if(!StringUtil.isEmpty(consumablesName)){  throw new HdiException("耗材名称已存在");  }  //判断耗材规格是否存在（耗材下规格唯一）  ....  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsSupplierReagentEntityVo.getFactoryId())){  //耗材生产厂商信息，状态为待审批  .......  }  //生成耗材编码  .......  //上传商品图片到fastDFS  ......  //保存耗材信息  goodsSupplierReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierReagentEntityVo.setCreateTime(new Date());  this.insert(goodsSupplierReagentEntityVo);  //保存耗材规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.8.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、耗材名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，耗材下的批准文号唯一性约束  4、编辑是没有商品规格和全球唯一码字段 |
| 必填字段 | 耗材名称、商品属性、耗材分类、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "id":3,  "consumablesName":"XX耗材",  "commonName": "通用名称",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"注册证号H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"注册证号H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"注册证号H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsSupplierConsumablesEntityVo goodsSupplierConsumablesEntityVo) {  //判断耗材名称是否存在  if(存在){  throw new HdiException("耗材名称已存在");  }  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新耗材信息  goodsSupplierReagentEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsSupplierConsumablesEntityVo.setEditTime(new Date());  this.updateById(goodsSupplierConsumablesEntityVo);  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.8.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 耗材名称、耗材通用名称、商品属性、商品分类名称、商品规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | http://localhost:2320/hdi-server-admin/consumables/goodssupplierconsumables/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsSupplierConsumablesEntityVo goodsSupplierConsumablesEntityVo= goodsSupplierConsumablesService.selectSupplierConsumablesById(id);  return Hdi.ok().put("goodsSupplierConsumables", goodsSupplierConsumablesEntityVo);  } |
| 查询语句 | 供应商耗材表关联耗材分类表、厂商信息表、供应商耗材规格表、平台耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsSupplierConsumables": {  "id": 3,  "supplierId":1,  "consumablesCode": "GHC000000001",  "consumablesName": "耗材名称",  "commonName": "螺丝钉",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.8.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.consumables |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsSupplierConsumablesEntity goodsSupplierConsumablesEntity= new GoodsSupplierConsumablesEntity ();  goodsSupplierConsumablesEntity.setId(id);  //设置信息为删除状态  goodsSupplierConsumablesEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsSupplierConsumablesEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.8.2.6更多规格

###### 5.3.8.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 供应商耗材规格（模糊匹配）、状态 |
| 列表显示 | 供应商耗材规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":"1",  "limit":"10",  "reagentId":1,  "specs":"5mm",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsSupplierConsumablesSpecsEntity> page = this.selectPage(  new Query<GoodsSupplierConsumablesSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsSupplierConsumablesSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": 5mm",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "gg00002",  "specs": "15mm",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.8.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 供应商耗材ID、供应商耗材规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 业务代码 | [  {  "id": 1,  "consumablesId": 1,  "specsCode": "gg00001",  "specs": "5mm",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "consumablesId": null,  "specsCode": "",  "specs": "15mm",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsSupplierConsumablesSpecsEntity> goodsSupplierConsumablesSpecsList) {  for(GoodsSupplierConsumablesSpecsEntity goodsSupplierConsumablesSpecsEntity： goodsSupplierConsumablesSpecsList){  //判断规格是否存在(耗材下规格唯一)  if(存在){  throw new HdiException("耗材规格已存在");  }  }  for(GoodsSupplierConsumablesSpecsEntity goodsSupplierConsumablesSpecsEntity： goodsSupplierConsumablesSpecsList){  if(StringUtil.isEmpty(goodsSupplierConsumablesSpecsEntity.getId())){  this.insert(goodsSupplierConsumablesSpecsEntity);  }else{  this.updateById(goodsSupplierConsumablesSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

### 5.3.9医院耗材信息

#### 5.3.9.1用例图



#### 5.3.9.2功能设计说明

##### 5.3.9.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 耗材名称、商品属性、耗材分类、状态、生产厂家、供应商名称、审核状态 |
| 列表显示 | 1、显示字段：医院名称、耗材名称、商品属性、耗材分类、耗材规格、生产厂家、批准文号/注册证号、状态、审批状态  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":1,  "limit":10,  "consumablesName":"耗材名称",  "goodsNature":0,  "typeId":1,  "status":1,  "factoryId":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<GoodsHospitalConsumablesEntityVo> page = new Page<GoodsHospitalConsumablesEntityVo>(currPage, pageSize);  //多表关联查询  List<GoodsHospitalConsumablesEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("consumablesName"), params.get("goodsNature"), ......);  page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | 医院耗材表关联耗材分类表、厂商信息表、医院耗材规格表、医院耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "hospitalId":1,  "consumablesCode": "YHC000000001",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specs": "20粒/盒",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  },  {  "id": 2,  "hospitalId":1,  "consumablesCode": "YHC000000002",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeName": "分类名称",  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryName": "广东罗浮山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  ......  }  ]  }  } |

##### 5.3.9.2.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 耗材名称，在医院范围内唯一性约束   2、耗材编码（系统自动按照统一编码规则生成）  3、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  4、耗材下的规格和批准文号，唯一性约束 |
| 必填字段 | 医院名称、耗材名称、商品属性、耗材分类、耗材规格、生产厂商、批准文号、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "hospitalId":1,  "consumablesName":"耗材名称",  "commonName": "通用名称",  "goodsNature":0,  "typeId":1,  "specs":"5mm",  "factoryId":1,  "factoryName": "广州白云山",  "approvals": "注册证号H10970001,注册证号H10970002,注册证号H10970003",  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(GoodsHospitalConsumablesEntityVo goodsHospitalConsumablesEntityVo) {  List<GoodsHospitalConsumablesEntityVo> consumablesName = baseMapper.selectByHospitalIdAndConsumablesName(goodsHospitalConsumablesEntityVo.getHospitalId(), goodsHospitalConsumablesEntityVo.getConsumablesName());  if(!StringUtil.isEmpty(consumablesName)){  throw new HdiException("耗材名称已存在");  }  //判断规格是否存在（耗材下规格唯一）  ....  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalConsumablesEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //生成耗材编码  .......  //上传商品图片到fastDFS  ......  //保存耗材信息  goodsHospitalConsumablesEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalConsumablesEntityVo.setCreateTime(new Date());  this.insert(goodsHospitalConsumablesEntityVo);  //保存耗材规格  ......  //保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.9.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、耗材名称，唯一性约束  2、生产厂商存在（厂商id和厂商名字必填），否则只传厂商名称  3、批准文号只能修改、禁用、不能删除，耗材下的批准文号唯一性约束  4、编辑是没有耗材规格和全球唯一码字段 |
| 必填字段 | 耗材名称、商品属性、耗材分类、耗材规格、生产厂商、批准文号、状态、商品单位 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "id":3,  "hospitalId":1,  "consumablesName":"耗材名称",  "commonName": "通用名称",  "goodsNature":0,  "typeId":1,  "factoryId":1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals":"注册证号H10970001" ,  "status": 1,  ......  },  {  "id": 2,  "approvals":"注册证号H10970002" ,  "status": 0,  ......  },  {  "id": null,  "specs":"注册证号H10970003" ,  "status": null,  ......  }  ],  "status":1,  "guid":"guid00000001",  "goodsUnit":"1",  .....  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(GoodsHospitalConsumablesEntityVo goodsHospitalConsumablesEntityVo ) {  //判断耗材名称是否存在  if(存在){  throw new HdiException("耗材名称已存在");  }  //判断批准文号是否已经存在（耗材下规格唯一）  .....  //生产厂商id为空则生产厂商信息  if(StringUtil.isEmpty(goodsHospitalConsumablesEntityVo.getFactoryId())){  //保存生产厂商信息，状态为待审批  .......  }  //图片路径为空则上传商品图片到fastDFS,否则不上传  ......  //更新耗材信息  goodsHospitalConsumablesEntityVo.setIsMatch(IsMatchEnum.NO.getKey());  goodsHospitalConsumablesEntityVo.setEditTime(new Date());  this.updateById(goodsHospitalConsumablesEntityVo);  //更新耗材规格或保存耗材规格  .....  //更新批准文号或保存批准文号信息  ......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.9.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 耗材名称、耗材通用名称、商品属性、耗材分类名称、耗材规格集合、生产厂商名称、批准文号集合、状态、商品单位、存储方式、商品图片URL |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | http://localhost:2320/hdi-server-admin/consumables/goodshospitalconsumables/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  GoodsHospitalConsumablesEntityVo goodsHospitalConsumablesEntityVo = goodsHospitallConsumablesService.selectHospitallConsumablesById(id);  return Hdi.ok().put("goodsHospitalConsumables", goodsHospitalConsumablesEntityVo);  } |
| 查询语句 | 医院耗材表关联耗材分类表、厂商信息表、医院耗材规格表、平台耗材批准文号表进行查询 |
| 输出 | {  "msg": "success",  "code": 0,  "goodsHospitalConsumables": {  "id": 3,  "hospitalId":1,  "consumablesCode": "YHC000000001",  "consumablesName": "耗材名称",  "commonName": "通用名称",  "goodsNature": 0,  "typeId": 1,  "specsList": [  {  "id": 1,  "specs": "5mm",  "status": 1,  ......  },  {  "id": 2,  "specs": "15mm",  "status": 0,  ......  },  {  "id": 3,  "specs": "25mm",  "status": 1,  ......  }  ],  "factoryId": 1,  "factoryName": "广州白云山",  "approvalsList": [  {  "id": 1,  "approvals": "注册证号H10970001",  "status": 1,  ......  },  {  "id": 2,  "approvals": "注册证号H10970002",  "status": 0,  ......  },  {  "id": 3,  "specs": "注册证号H10970003",  "status": 1,  ......  }  ],  "status": 1,  "guid": "guid00000001",  "goodsUnit": "1",  .....  }  } |

##### 5.3.9.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.consumables |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  GoodsHospitalConsumablesEntity goodsHospitalConsumablesEntity= new GoodsHospitalConsumablesEntity();  goodsHospitalConsumablesEntity.setId(id);  //设置信息为删除状态  goodsHospitalConsumablesEntity.setStatus(StatusEnum.DELETE.getKey());  this.updateById(goodsHospitalConsumablesEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.3.9.2.6更多规格

###### 5.3.9.2.6.1分页列表

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院耗材规格（模糊匹配）、状态 |
| 列表显示 | 医院耗材规格、全球唯一编码、状态 |
| 类包 | com.ebig.hdi.modules.consumables |
| 输入 | {  "page":"1",  "limit":"10",  "hospitalId":1,  "specs":"5mm",  "status":"1"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<GoodsHospitalConsumablesSpecsEntity> page = this.selectPage(  new Query<GoodsHospitalConsumablesSpecsEntity>(params).getPage(),  new EntityWrapper<GoodsHospitalConsumablesSpecsEntity>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .like(StringUtils.isNotBlank(params.get("specs").toString()), "specs", params.get("specs").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "reagentId": 1,  "specsCode": "gg00001",  "specs": "5mm",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": 2,  "reagentId": 2,  "specsCode": "gg00002",  "specs": "15mm",  "guid": "guid00002",  "status": "1",  ......  }......  ]  }  } |

###### 5.3.9.2.6.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 新增记录可以删除，修改记录不允许删除 2. 支持批量保存新增记录和更新记录 |
| 必填字段 | 医院耗材ID、医院耗材规格、全球唯一码、状态 |
| 类包 | com.ebig.hdi.modules.consumablesId |
| 业务代码 | [  {  "id": 1,  "consumablesId": 1,  "specsCode": "gg00001",  "specs": "5mm",  "guid": "guid00001",  "status": "1",  ......  },  {  "id": null,  "consumablesId": null,  "specsCode": "",  "specs": "15mm",  "guid": "guid00002",  "status": "1",  ......  }......  ] |
| 输出 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(List<GoodsHospitalConsumablesSpecsEntity> goodsHospitalConsumablesSpecsList) {  for(GoodsHospitalConsumablesSpecsEntity goodsHospitalConsumablesSpecsEntity： goodsHospitalConsumablesSpecsList){  //判断规格是否存在(耗材下规格唯一)  if(存在){  throw new HdiException("耗材规格已存在");  }  }  for(GoodsHospitalConsumablesSpecsEntity goodsHospitalConsumablesSpecsEntity ：goodsHospitalConsumablesSpecsList){  if(StringUtil.isEmpty(goodsHospitalConsumablesSpecsEntity.getId())){  this.insert(goodsHospitalConsumablesSpecsEntity);  }else{  this.updateById(goodsHospitalConsumablesSpecsEntity);  }  }  //更新商品匹配状态为未匹配  ......  } |
| 业务代码 | {  "msg": "success",  "code": 0  } |

## 5.4证照管理

### 5.4.1证照分类信息

#### 5.4.1.1用例图



#### 5.4.1.2功能设计说明

##### 5.4.1.2.1列表查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照类型、证照分类名称 |
| 列表显示 | 分类名称、是否预警、预警天数、状态 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "type":1,  "name":"分类名称"  } |
| 业务代码 | @Override  public List<LicenseClassifyInfoEntity> listByCondition(Map<String, Object> params) {  //获取所有证照分类  List<LicenseClassifyInfoEntity> list = baseMapper.listByCondition(params);  return list;  } |
| 输出 | {  "msg": "success",  "code": 0,  "list": [  {  "id": 1,  "type": 0,  "name": "产品注册证",  "isWarning": 0,  "earlyDate": 60,  "status": "1",  ......  },  {  "id": 2,  "type": 0,  "name": "检查报告",  "isWarning": 1,  "earlyDate": 30,  "status": "1",  ......  },  {  "id": 3,  "type": 1,  "name": "营业执照",  "isWarning": 1,  "earlyDate": 60,  "status": "1",  ......  },  {  "id": 4,  "type": 0,  "name": "企业经营许可证",  "isWarning": 1,  "earlyDate": 60,  "status": "1",  ......  },  {  "id": 5,  "type": 0,  "name": "组织机构代码证",  "isWarning": 1,  "earlyDate": 30,  "status": "1",  ......  },  {  "id": 6,  "type": 0,  "name": "税务登记证",  "isWarning": 1,  "earlyDate": 60,  "status": "1",  ......  }  ]  } |

##### 5.4.1.2.1新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、证照分类名称，唯一性约束 |
| 必填字段 | 证照类型、证照分类名称、是否预警、预警天数 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "type": 0,  "name": "产品注册证",  "isWarning": 0,  "earlyDate": 60,  "status": "1",  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseClassifyInfoEntity licenseClassifyInfoEntity) {  List<LicenseClassifyInfoEntity> licenseClassifyName = baseMapper.selectByLicenseClassifyName(licenseClassifyInfoEntity.getName());  if(!StringUtil.isEmpty(licenseClassifyName)){  throw new HdiException("证照分类名称已存在");  }  //保存证照分类信息  licenseClassifyInfoEntity.setCreateTime(new Date());  this.insert(licenseClassifyInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.1.2.1编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、证照分类名称，唯一性约束 |
| 必填字段 | 证照类型、证照分类名称、是否预警、预警天数 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "id": 1,  "type": 0,  "name": "产品注册证",  "isWarning": 0,  "earlyDate": 60,  "status": "1",  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(LicenseClassifyInfoEntity licenseClassifyInfoEntity) {  List<LicenseClassifyInfoEntity> licenseClassifyName = baseMapper.selectByLicenseClassifyName(licenseClassifyInfoEntity.getName());  if(存在 ){  throw new HdiException("证照分类名称已存在");  }  //保存证照分类信息  licenseClassifyInfoEntity.setEditTime(new Date());  this.updateById(licenseClassifyInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.1.2.1删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.license |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  LicenseClassifyInfoEntity licenseClassifyInfoEntity = new LicenseClassifyInfoEntity ();  licenseClassifyInfoEntity .setId(id);  //设置信息为删除状态  licenseClassifyInfoEntity .setStatus(StatusEnum.DELETE.getKey());  this.updateById(licenseClassifyInfoEntity );  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.4.2供应商证照信息

#### 5.4.2.1用例图



#### 5.4.2.2功能设计说明

##### 5.4.2.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照分类、供应商名称（模糊匹配）、证照名称/编号（模糊匹配）、证照日期（失效时间范围）、证照状态证照状态（0：正常；1：预警；2：失效；3：换证中） |
| 列表显示 | 机构名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态 |
| 操作表 | 视图表名：view\_hdi\_license\_supplier |
| 视图表创建语句 | select a.\*,b.supplier\_name,c.type as license\_type,c.name as classify\_name,c.is\_warning,c.early\_date,datediff(a.end\_time, now()) as lose\_time,  case when c.is\_warning=1 and datediff(a.end\_time, now()) > c.early\_date then 0  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') <= a.end\_time then 0  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is null then 1  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is not null then 3  when c.is\_warning=1 and datediff(a.end\_time, now()) < c.early\_date then 2  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') > a.end\_time then 2  end as license\_status  from hdi\_license\_supplier\_info a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_license\_classify\_info c on a.classify\_id=c.id |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "page":"1",  "limit":"10",  "classifyId":3,  "supplierName":"供应商名称",  "name":"证照名称/编码",  "beginTime":"证照日期开始时间",  "endTime":"证照日期结束时间",  "licenseStatus":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<LicenseSupplierInfoEntityVo> page = this.selectPage(  new Query<LicenseSupplierInfoEntityVo>(params).getPage(),  new EntityWrapper<LicenseSupplierInfoEntityVo>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .eq(StringUtils.isNotBlank(params.get("classifyId").toString()), "classify\_id", params.get("classifyId").toString())  .like(StringUtils.isNotBlank(params.get("supplierName").toString()), "supplier\_name", params.get("supplierName").toString())  .like(StringUtils.isNotBlank(params.get("name").toString()), "name", params.get("name").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 1,  "classifyId": 1,  "name": "供应商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 2,  "supplierId": 1,  "classifyId": 2,  "name": "供应商证照2",  "number": "0000002",  "status": "1",  ......  }......  ]  }  } |

##### 5.4.2.2.2新增/换证

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 新增约束条件 | 1、机构名称，当前登陆的供应商 |
| 换证页面效果 |  |
| 换证约束条件 | 1. 从旧证信息获取机构名称和证照分类，且不能进行修改； 2. 保存新证记录时，将新证id跟新到旧证记录 |
| 必填字段 | 供应商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "classifyId": 3,  "name":"供应商证照" ,  "number": "0000000001",  "deptId": 20,  "createId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseSupplierInfoEntity licenseSupplierInfoEntity) {  //保存供应商证照信息  licenseSupplierInfoEntity.setCreateTime(new Date());  this.insert(licenseSupplierInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.2.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 编辑页面效果 |  |
| 约束条件 | 1、机构名称，不能编辑 |
| 必填字段 | 供应商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "id": 1,  "supplierId": 15,  "classifyId": 3,  "name":"供应商证照" ,  "number": "0000000001",  "editId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(LicenseSupplierInfoEntity licenseSupplierInfoEntity) {  //更新供应商证照信息  licenseSupplierInfoEntity.setEditTime(new Date());  this.updateById(licenseSupplierInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.2.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 机构名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态、关联的证照信息集合 |
| 操作表 | 视图表名：view\_hdi\_license\_supplier |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | http://localhost:2320/hdi-server-admin/license/licensesupplierinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  //获取关联的证照信息（旧证关联新证，新证关联旧证）,递归参数证照数据（向上向下递归）  List<LicenseSupplierInfoEntityVo> licenseSupplierInfoList = ......;  return Hdi.ok().put("licenseSupplierInfoList ", licenseSupplierInfoList );  } |
| 输出 | {  "msg": "success",  "code": 0,  "licenseSupplierInfoList": [  {  "id": 1,  "supplierId": 1,  "classifyId": 1,  "name": "供应商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 4,  "supplierId": 1,  "classifyId": 1,  "name": "供应商证照4",  "number": "0000001",  "status": "1",  ......  }  ]  }  } |

##### 5.4.2.2.5提交医院审批

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 获取当前供应商已绑定的医院数据 2. 选择合作的医院 |
| 获取当前供应商已绑定的医院查询语句 | select \* from hdi\_org\_supplier\_hospital\_ref where supplier\_id=7 and del\_flag=0 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "licenseId": 3,  "licenseType":1 ,  "hospitalIds":[34,35],  "deptId": 20,  "createId": 2,  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseHospitalExamineEntityVo licenseHospitalExamineEntityVo, String userName) {  for(Long hospitalId : licenseHospitalExamineEntityVo.getHospitalIds()) {  licenseHospitalExamineEntityVo.setHospitalId(hospitalId);  licenseHospitalExamineEntityVo.setCreateTime(new Date());  this.insert(licenseHospitalExamineEntityVo);  }  } |
|  | {  "msg": "success",  "code": 0  } |

##### 5.4.2.2.5查看医院审批详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 详情显示 | 医院名称、审批意见、医院审批状态、审批时间 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "licenseId": 1,  "licenseType":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<LicenseHospitalExamineEntityVo> page = new Page<LicenseHospitalExamineEntityVo>(currPage, pageSize);  //关联医院表查询  List<LicenseHospitalExamineEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("licenseId"), params.get("licenseType"));  page.setRecords(list);  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 34,  "hospitalId": "2",  "hospitalName": "南方医院",  "status": "1",  "examineOpinion": "审核通过",  "examineTime": "2019-04-05 12:00:00",  ......  },  {  "id": 2,  "supplierId": 34,  "hospitalId": "2",  "hospitalName": "南方医院",  "status": "1",  "examineOpinion": "审核不通过",  "examineTime": "2019-04-20 12:00:00",  ......  }......  ]  }  } |

##### 5.4.2.2.6删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.license |
| 功能说明 | 医院已审批通过的证照信息记录不能再删除 |
| 输入 | {  "licenseId": [1,2,3],  "licenseType":1  } |
| 业务代码 | @Override  public void delete(@RequestBody Map<String, Object> params) {  //查询证照信息是否被医院已审批通过,如果审批通过则不能删除，否则可以删除  ......  //删除证照信息记录  .....  //删除证照提交审批记录  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.4.3商品证照信息

#### 5.4.3.1用例图



#### 5.4.3.2功能设计说明

##### 5.4.3.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照分类、厂商名称（模糊匹配）、证照名称/编号（模糊匹配）、证照日期（失效时间范围）、证照状态（0：正常；1：预警；2：失效；3：换证中） |
| 列表显示 | 商品名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态 |
| 操作表 | 视图表名：view\_hdi\_license\_goods |
| 视图表创建语句 | select c.\*,d.type as license\_type,d.name as classify\_name,d.is\_warning,d.early\_date,datediff(c.end\_time, now()) as lose\_time,  case when d.is\_warning=1 and datediff(c.end\_time, now()) > d.early\_date then 0  when d.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') <= c.end\_time then 0  when d.is\_warning=1 and datediff(c.end\_time, now()) between 0 and d.early\_date and c.new\_license\_id is null then 1  when d.is\_warning=1 and datediff(c.end\_time, now()) between 0 and d.early\_date and c.new\_license\_id is not null then 3  when d.is\_warning=1 and datediff(c.end\_time, now()) < d.early\_date then 2  when d.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') > c.end\_time then 2  end as license\_status  from (select a.\*,b.supplier\_name,b.goods\_code,b.goods\_name,b.common\_name,b.factory\_id,b.factory\_name from  hdi\_license\_goods\_info a,view\_hdi\_goods\_supplier b where a.goods\_id=b.id and a.goods\_type=b.goods\_type  ) c left join hdi\_license\_classify\_info d on c.classify\_id=d.id |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "page":"1",  "limit":"10",  "classifyId":2,  "goodsName":"商品名称",  "name":"证照名称/编码",  "beginTime":"证照日期开始时间",  "endTime":"证照日期结束时间",  "licenseStatus":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<LicenseGoodsInfoEntityVo> page = this.selectPage(  new Query<LicenseGoodsInfoEntityVo>(params).getPage(),  new EntityWrapper<LicenseGoodsInfoEntityVo>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .eq(StringUtils.isNotBlank(params.get("classifyId").toString()), "classify\_id", params.get("classifyId").toString())  .like(StringUtils.isNotBlank(params.get("goodsName").toString()), "goods\_name", params.get("goodsName").toString())  .like(StringUtils.isNotBlank(params.get("name").toString()), "name", params.get("name").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "name": "商品证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 2,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "name": "商品证照2",  "number": "0000001",  "status": "1",  ......  }......  ]  }  } |

##### 5.4.3.2.2新增/换证

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询供应商商品视图表（包括供应商的药品、试剂和耗材商品） | 视图表名：view\_hdi\_goods\_supplier |
| 视图表创建语句 | select a.id,0 as goods\_type,a.supplier\_id,b.supplier\_name,a.drugs\_code as goods\_code,a.drugs\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_drugs a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id  union  select a.id,1 as goods\_type,a.supplier\_id,b.supplier\_name,a.reagent\_code as goods\_code,a.reagent\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_reagent a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id  union  select a.id,2 as goods\_type,a.supplier\_id,b.supplier\_name,a.consumables\_code as goods\_code,a.consumables\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_consumables a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id |
| 查询语句 | select \* from view\_hdi\_goods\_supplier where supplier\_id=15 |
| 约束条件 | 1. 商品名称，当前登陆供应商的商品信息 2. 生产厂商，选择商品时同时获取 |
| 换证页面效果 |  |
| 换证约束条件 | 1、从旧证信息获取商品名称、生产厂商和证照分类，且不能进行修改；  2、保存新证记录时，将新证id跟新到旧证记录 |
| 必填字段 | 商品名称、生产厂商、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "goodsId": 1,  "goodsType": 0,  "classifyId": 3,  "name":"商品证照" ,  "number": "0000000001",  "deptId": 20,  "createId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseGoodsInfoEntity licenseGoodsInfoEntity) {  //保存商品证照信息  licenseGoodsInfoEntity.setCreateTime(new Date());  this.insert(licenseGoodsInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.3.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 编辑页面效果 |  |
| 查询供应商商品视图表（包括供应商的药品、试剂和耗材商品） | 视图表名：view\_hdi\_goods\_supplier |
| 视图表创建语句 | select a.id,0 as goods\_type,a.supplier\_id,b.supplier\_name,a.drugs\_code as goods\_code,a.drugs\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_drugs a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id  union  select a.id,1 as goods\_type,a.supplier\_id,b.supplier\_name,a.reagent\_code as goods\_code,a.reagent\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_reagent a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id  union  select a.id,2 as goods\_type,a.supplier\_id,b.supplier\_name,a.consumables\_code as goods\_code,a.consumables\_name as goods\_name,a.common\_name,a.factory\_id,c.factory\_name  from hdi\_goods\_supplier\_consumables a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info c on a.factory\_id=c.id |
| 查询语句 | select \* from view\_hdi\_goods\_supplier where supplier\_id=15 |
| 约束条件 | 1、商品名称，当前登陆供应商的商品信息  2、生产厂商，选择商品时同时获取 |
| 必填字段 | 供应商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "id": 1,  "supplierId": 15,  "goodsId": 1,  "goodsType": 0,  "classifyId": 3,  "name":"商品证照" ,  "number": "0000000001",  "editId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(LicenseGoodsInfoEntity licenseGoodsInfoEntity) {  //更新商品证照信息  licenseGoodsInfoEntity.setEditTime(new Date());  this.updateById(licenseGoodsInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.3.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 商品名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态、关联的证照信息集合 |
| 操作表 | 视图表名：view\_hdi\_goods\_supplier |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | http://localhost:2320/hdi-server-admin/license/licensegoodsinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  List<LicenseGoodsInfoEntityVo> licenseGoodsInfoList = ......;  //获取关联的证照信息（旧证关联新证，新证关联旧证）,递归参数证照数据  ......  return Hdi.ok().put("licenseGoodsInfoList", licenseGoodsInfoList);  } |
| 输出 | {  "msg": "success",  "code": 0,  "licenseGoodsInfoList": [  {  "id": 1,  "supplierId": 1,  "classifyId": 1,  "name": "供应商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 4,  "supplierId": 1,  "classifyId": 1,  "name": "供应商证照4",  "number": "0000001",  "status": "1",  ......  }  ]  }  } |

##### 5.4.3.2.5提交医院审批

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1. 获取当前供应商已绑定的医院数据 2. 选择合作的医院 |
| 获取当前供应商已绑定的医院查询语句 | select \* from hdi\_org\_supplier\_hospital\_ref where supplier\_id=7 and del\_flag=0 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "licenseId": 3,  "licenseType":1 ,  "hospitalIds":[34,35],  "deptId": 20,  "createId": 2,  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseHospitalExamineEntityVo licenseHospitalExamineEntityVo, String userName) {  for(Long hospitalId : licenseHospitalExamineEntityVo.getHospitalIds()) {  licenseHospitalExamineEntityVo.setHospitalId(hospitalId);  licenseHospitalExamineEntityVo.setCreateTime(new Date());  this.insert(licenseHospitalExamineEntityVo);  }  } |
|  | {  "msg": "success",  "code": 0  } |

##### 5.4.3.2.5查看医院审批详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 详情显示 | 医院名称、审批意见、医院审批状态、审批时间 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "licenseId": 1,  "licenseType":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<LicenseHospitalExamineEntityVo> page = new Page<LicenseHospitalExamineEntityVo>(currPage, pageSize);  //关联医院表查询  List<LicenseHospitalExamineEntityVo> list = this.baseMapper.listForPage(page,params.get(Constant.SQL\_FILTER), params.get("licenseId"), params.get("licenseType"));  page.setRecords(list);  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 34,  "hospitalId": "2",  "hospitalName": "南方医院",  "status": "1",  "examineOpinion": "审核通过",  "examineTime": "2019-04-05 12:00:00",  ......  },  {  "id": 2,  "supplierId": 34,  "hospitalId": "2",  "hospitalName": "南方医院",  "status": "1",  "examineOpinion": "审核不通过",  "examineTime": "2019-04-20 12:00:00",  ......  }......  ]  }  } |

##### 5.4.3.2.6删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.license |
| 功能说明 | 医院已审批通过的证照信息记录不能再删除 |
| 输入 | {  "licenseId": [1,2,3],  "licenseType":1  } |
| 业务代码 | @Override  public void delete(@RequestBody Map<String, Object> params) {  //查询证照信息是否被医院已审批通过,如果审批通过则不能删除，否则可以删除  ......  //删除证照信息记录  .....  //删除证照提交审批记录  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.4.4厂商证照信息

#### 5.4.4.1用例图



#### 5.4.4.2功能设计说明

##### 5.4.4.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照分类、厂商名称（模糊匹配）、证照名称/编号（模糊匹配）、证照日期（失效时间范围）、证照状态（0：正常；1：预警；2：失效；3：换证中） |
| 列表显示 | 厂商名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态 |
| 操作表 | 视图表名：view\_hdi\_license\_factory |
| 视图表创建语句 | select a.\*,b.supplier\_name,d.factory\_name,c.type as license\_type,c.name as classify\_name,c.is\_warning,c.early\_date,datediff(a.end\_time, now()) as lose\_time,  case when c.is\_warning=1 and datediff(a.end\_time, now()) > c.early\_date then 0  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') <= a.end\_time then 0  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is null then 1  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is not null then 3  when c.is\_warning=1 and datediff(a.end\_time, now()) < c.early\_date then 2  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') > a.end\_time then 2  end as license\_status  from hdi\_license\_factory\_info a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_factory\_info d on a.factory\_id=d.id  left join hdi\_license\_classify\_info c on a.classify\_id=c.id |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "page":"1",  "limit":"10",  "classifyId":2,  "factoryName":"厂商名称",  "name":"证照名称/编码",  "beginTime":"证照日期开始时间",  "endTime":"证照日期结束时间",  "licenseStatus":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<LicenseFactoryInfoEntityVo> page = this.selectPage(  new Query<LicenseFactoryInfoEntityVo>(params).getPage(),  new EntityWrapper<LicenseFactoryInfoEntityVo>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .eq(StringUtils.isNotBlank(params.get("classifyId").toString()), "classify\_id", params.get("classifyId").toString())  .like(StringUtils.isNotBlank(params.get("factoryName").toString()), "factory\_name", params.get("factoryName").toString())  .like(StringUtils.isNotBlank(params.get("name").toString()), "name", params.get("name").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "factoryId": 1,  "name": "厂商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 2,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "factoryId": 2,  "name": "厂商证照2",  "number": "0000001",  "status": "1",  ......  }......  ]  }  } |

##### 5.4.4.2.2新增/换证

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、厂商名称，获取系统的厂商信息集合  2、如果厂商信息不存在，则系统生成一条待审批状态的记录 |
| 换证页面效果 |  |
| 换证约束条件 | 1、从旧证信息获取厂商名称和证照分类，且不能进行修改；  2、保存新证记录时，将新证id跟新到旧证记录 |
| 必填字段 | 厂商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "factoryId": 1,  "classifyId": 3,  "name":"厂商证照" ,  "number": "0000000001",  "deptId": 20,  "createId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseFactoryInfoEntity licenseFactoryInfoEntity) {  //判断厂商信息是否存在，如果不存在，则系统生成一条待审批状态的记录  ......  //保存厂商证照信息  licenseFactoryInfoEntity.setCreateTime(new Date());  this.insert(licenseFactoryInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.4.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 编辑页面效果 |  |
| 约束条件 | 1、厂商名称，获取系统的厂商信息集合  2、如果厂商信息不存在，则系统生成一条待审批状态的记录 |
| 必填字段 | 厂商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "id": 1,  "supplierId": 15,  "factoryId": 1,  "classifyId": 3,  "name":"厂商证照" ,  "number": "0000000001",  "deptId": 20,  "editId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(LicenseFactoryInfoEntity licenseFactoryInfoEntity) {  //判断厂商信息是否存在，如果不存在，则系统生成一条待审批状态的记录  ......  //更新厂商证照信息  licenseFactoryInfoEntity.setEditTime(new Date());  this.updateById(licenseFactoryInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.4.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 厂商名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态、关联的证照信息集合 |
| 操作表 | 视图表名：view\_hdi\_license\_factory |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | http://localhost:2320/hdi-server-admin/license/licensefactoryinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  List<LicenseFactoryInfoEntityVo> licenseFactoryInfoEntityList = ......;  //获取关联的证照信息（旧证关联新证，新证关联旧证）,递归参数证照数据  ......  return Hdi.ok().put("licenseFactoryInfoEntityList", licenseFactoryInfoEntityList);  } |
| 输出 | {  "msg": "success",  "code": 0,  "licenseGoodsInfoList": [  {  "id": 1,  "supplierId": 1,  "factoryId": 1,  "classifyId": 1,  "name": "厂商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 4,  "supplierId": 1,  "factoryId": 1,  "classifyId": 1,  "name": "供应商证照4",  "number": "0000001",  "status": "1",  ......  }  ]  }  } |

##### 5.4.4.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.license |
| 功能说明 | 厂商状态为非草稿状态时则不能删除 |
| 输入 | {  "licenseId": [1,2,3],  "licenseType":1  } |
| 业务代码 | @Override  public void delete(@RequestBody Map<String, Object> params) {  //代理商状态为非待审批状态时则不能删除  ......  //删除证照信息记录  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.4.5代理商证照信息

#### 5.4.5.1用例图



#### 5.4.5.2功能设计说明

##### 5.4.5.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照分类、代理商名称（模糊匹配）、证照名称/编号（模糊匹配）、证照日期（失效时间范围）、证照状态（0：正常；1：预警；2：失效；3：换证中） |
| 列表显示 | 代理商名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态 |
| 操作表 | 视图表名：view\_hdi\_license\_agent |
| 视图表创建语句 | select a.\*,b.supplier\_name,d.agent\_name,c.type as license\_type,c.name as classify\_name,c.is\_warning,c.early\_date,datediff(a.end\_time, now()) as lose\_time,  case when c.is\_warning=1 and datediff(a.end\_time, now()) > c.early\_date then 0  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') <= a.end\_time then 0  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is null then 1  when c.is\_warning=1 and datediff(a.end\_time, now()) between 0 and c.early\_date and a.new\_license\_id is not null then 3  when c.is\_warning=1 and datediff(a.end\_time, now()) < c.early\_date then 2  when c.is\_warning=0 and date\_format(now(),'%Y-%m-%d %H:%i:%s') > a.end\_time then 2  end as license\_status  from hdi\_license\_agent\_info a left join hdi\_org\_supplier\_info b on a.supplier\_id=b.id  left join hdi\_org\_agent\_info d on a.agent\_id=d.id  left join hdi\_license\_classify\_info c on a.classify\_id=c.id |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "page":"1",  "limit":"10",  "classifyId":2,  "agentName":"代理商名称",  "name":"证照名称/编码",  "beginTime":"证照日期开始时间",  "endTime":"证照日期结束时间",  "licenseStatus":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<LicenseAgentInfoEntityVo> page = this.selectPage(  new Query<LicenseAgentInfoEntityVo>(params).getPage(),  new EntityWrapper<LicenseAgentInfoEntityVo>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .eq(StringUtils.isNotBlank(params.get("classifyId").toString()), "classify\_id", params.get("classifyId").toString())  .like(StringUtils.isNotBlank(params.get("agentName").toString()), "agent\_name", params.get("agentName").toString())  .like(StringUtils.isNotBlank(params.get("name").toString()), "name", params.get("name").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "agentId": 1,  "name": "代理商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 2,  "supplierId": 15,  "goodsType": 0,  "classifyId": 1,  "agentId": 2,  "name": "代理商证照2",  "number": "0000001",  "status": "1",  ......  }......  ]  }  } |

##### 5.4.5.2.2新增/换证

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 约束条件 | 1、代理商名称，获取当前供应商的代理商信息集合  2、如果代理商信息不存在，则系统生成一条记录 |
| 换证页面效果 |  |
| 换证约束条件 | 1、从旧证信息获取代理商名称、代理商信用和证照分类，且不能进行修改；  2、保存新证记录时，将新证id跟新到旧证记录 |
| 必填字段 | 代理商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "supplierId": 15,  "agentId": 1,  "classifyId": 3,  "name":"厂商证照" ,  "number": "0000000001",  "deptId": 20,  "createId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(LicenseAgentInfoEntity licenseAgentInfoEntity) {  //判断代理商信息是否存在，如果代理商信息不存在，则系统生成一条记录  ......  //保存代理商证照信息  licenseAgentInfoEntity.setCreateTime(new Date());  this.insert(licenseAgentInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.5.2.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 编辑页面效果 |  |
| 约束条件 | 1、代理商名称，获取当前供应商的代理商信息集合  2、如果代理商信息不存在，则系统生成一条记录 |
| 必填字段 | 代理商名称、证照分类、证照名称、证照编号、证照效期 |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "id": 1,  "supplierId": 15,  "agentId": 1,  "classifyId": 3,  "name":"代理商证照" ,  "number": "0000000001",  "deptId": 20,  "editId": 2,  ......  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(LicenseAgentInfoEntity licenseAgentInfoEntity) {  //判断代理商信息是否存在，如果不存在，则系统生成一条记录  ......  //更新代理商证照信息  licenseAgentInfoEntity.setEditTime(new Date());  this.updateById(licenseAgentInfoEntity);  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.4.5.2.4查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 详情显示 | 代理商名称、证照名称、证照编码、证照分类、证照图片、失效日期、证照状态、关联的证照信息集合 |
| 操作表 | 视图表名：view\_hdi\_license\_agent |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | http://localhost:2320/hdi-server-admin/license/licenseagentinfo/info/1 |
| 业务代码 | @RequestMapping("/info/{id}")  public Hdi info(@PathVariable("id") Long id){  List<licenseAgentInfoEntityVo> licenseAgentInfoEntityList = ......;  //获取关联的证照信息（旧证关联新证，新证关联旧证）,递归参数证照数据  ......  return Hdi.ok().put("licenseAgentInfoList", licenseAgentInfoEntityList);  } |
| 输出 | {  "msg": "success",  "code": 0,  "licenseAgentInfoList": [  {  "id": 1,  "supplierId": 1,  "agentId": 1,  "classifyId": 1,  "name": "代理商证照1",  "number": "0000001",  "status": "1",  ......  },  {  "id": 4,  "supplierId": 1,  "agentId": 2,  "classifyId": 1,  "name": "代理商证照4",  "number": "0000001",  "status": "1",  ......  }  ]  }  } |

##### 5.4.5.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.license |
| 功能说明 | 医院已审批通过的证照信息记录不能再删除 |
| 输入 | {  "licenseId": [1,2,3],  "licenseType":1  } |
| 业务代码 | @Override  public void delete(@RequestBody Map<String, Object> params) {  //厂商状态为非待审批状态时则不能删除  ......  //删除证照信息记录  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.4.6证照预警信息

#### 5.4.6.1用例图



#### 5.4.6.2功能设计说明

##### 5.4.6.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 证照类型、证照分类、证照名称/编号（模糊匹配）、证照日期（失效时间范围）、证照状态（1：预警；2：失效；3：换证中） |
| 列表显示 | 证照类型、证照分类、证照名称、证照编码、证照图片、失效日期、证照状态 |
| 操作表 | 视图表名：view\_hdi\_license\_warning |
| 视图表创建语句 | select id,supplier\_id,supplier\_name,license\_type,classify\_id,classify\_name,name,number,begin\_time,end\_time,pic\_url,status,dept\_id  create\_id,create\_time,edit\_id,edit\_time,del\_flag,new\_license\_id,is\_warning,early\_date,lose\_time,license\_status  from view\_hdi\_license\_supplier where is\_warning=1 and license\_status in (1,2,3)  union  select id,supplier\_id,supplier\_name,license\_type,classify\_id,classify\_name,name,number,begin\_time,end\_time,pic\_url,status,dept\_id  create\_id,create\_time,edit\_id,edit\_time,del\_flag,new\_license\_id,is\_warning,early\_date,lose\_time,license\_status  from view\_hdi\_license\_goods where is\_warning=1 and license\_status in (1,2,3)  union  select id,supplier\_id,supplier\_name,license\_type,classify\_id,classify\_name,name,number,begin\_time,end\_time,pic\_url,status,dept\_id  create\_id,create\_time,edit\_id,edit\_time,del\_flag,new\_license\_id,is\_warning,early\_date,lose\_time,license\_status  from view\_hdi\_license\_factory where is\_warning=1 and license\_status in (1,2,3)  union  select id,supplier\_id,supplier\_name,license\_type,classify\_id,classify\_name,name,number,begin\_time,end\_time,pic\_url,status,dept\_id  create\_id,create\_time,edit\_id,edit\_time,del\_flag,new\_license\_id,is\_warning,early\_date,lose\_time,license\_status  from view\_hdi\_license\_agent where is\_warning=1 and license\_status in (1,2,3) |
| 类包 | com.ebig.hdi.modules.license |
| 输入 | {  "page":"1",  "limit":"10",  "licenseType":0,  "classifyId":2,  "name":"证照名称/编码",  "beginTime":"证照日期开始时间",  "endTime":"证照日期结束时间",  "licenseStatus":1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<HdiLicenseWarningVo> page = this.selectPage(  new Query<HdiLicenseWarningVo>(params).getPage(),  new EntityWrapper<HdiLicenseWarningVo>()  .addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null, (String)params.get(Constant.SQL\_FILTER))  .eq(StringUtils.isNotBlank(params.get("licenseType").toString()), "license\_type", params.get("licenseType").toString())  .eq(StringUtils.isNotBlank(params.get("classifyId").toString()), "classify\_id", params.get("classifyId").toString())  .like(StringUtils.isNotBlank(params.get("name").toString()), "name", params.get("name").toString())  .eq(StringUtils.isNotBlank(params.get("status").toString()), "status", params.get("status").toString())  );  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 15,  "supplierName": "供应商名称",  "licenseType": 0,  "classifyId": 1,  "classifyName": "分类名称",  "name": "供应商证照",  "number": "0000001",  "status": "1",  ......  },  {  "id": 2,  "supplierId": 15,  "supplierName": "供应商名称",  "licenseType": 0,  "classifyId": 1,  "classifyName": "分类名称",  "agentId": 2,  "name": "代理商证照",  "number": "0000001",  "status": "1",  ......  }......  ]  }  } |

##### 5.4.6.2.2换证

|  |  |
| --- | --- |
| 事项 | 内容 |
| 约束条件 | 前端根据证照类型进行判断，分别跳转到供应商、商品、厂商和代理商的换证页面 |
| 详细设计 | 分别查看：供应商、商品、厂商和代理商的换证相关的详细设计 |

## 5.5采购管理

### 5.5.1用例图

### 5.5.2功能设计说明

## 5.6供货管理

### 5.6.1用例图

### 5.6.2功能设计说明

## 5.7退货管理

### 5.7.1退货申请单管理

#### 5.7.1.1用例图



#### 5.7.1.2功能设计说明

##### 5.7.1.2.1主单分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院名称、库房名称、医院退货申请单编号、申请时间、状态 |
| 主单列表显示 | 1、医院名称、库房名称、医院退货申请单编号、申请时间、状态、细单条数  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "page":1,  "limit":10,  "hospitalName":"医院名称",  "storeHouseName":"库房名称",  "refundsApplyNo":"医院退货申请单编号",  "applyTimeBeginStr":"申请开始时间",  "applyTimeEndStr":"申请结束时间",  "status":null  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false, tableAlias="t1")  public PageUtils queryPage(Map<String, Object> params) {  RefundsApplyMasterVO ramVO = new RefundsApplyMasterVO();  ramVO.setHospitalName((String)params.get("hospitalName"));  ramVO.setStoreHouseName((String)params.get("storeHouseName"));  ramVO.setRefundsApplyNo((String)params.get("refundsApplyNo"));  ramVO.setApplyTimeBeginStr((String)params.get("applyTimeBeginStr"));  ramVO.setApplyTimeEndStr((String)params.get("applyTimeEndStr"));  ramVO.setStatus((Integer)params.get("status"));    String sidx = (String)params.get("sidx");  String order = (String)params.get("order");  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());    Page<RefundsApplyMasterVO> page = new Page<RefundsApplyMasterVO>(currPage, pageSize,order);  if(sidx!=null){  page.setAsc(sidx.equals("desc")?false:true);  }else{  //标志，设置默认按更新时间和创建时间排序  ramVO.setIsDefaultOrder(1);  }  //自定义分页查询列表带过滤条件  List<RefundsApplyMasterVO> list = this.baseMapper.listForPage(page,ramVO);  //查询并设置退回  setDetail(list);    page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | select a.\*,b.hospital\_name,c.storehousename from hdi\_refunds\_apply\_master a left join hdi\_org\_hospital\_info b on a.hospital\_id=b.id  left join hdi\_core\_storehouse c on a.store\_house\_id=c.storehouseid |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "supplierId": 15,  "sourcesSupplierId": "YGYS00001" ,  "hospitalId": 25,  "sourcesHospitalId": "YYY000008" ,  "storeHouseId": "1",  "sourcesStoreHouseId": "YKF000001",  "refundsApplyNo": "TH00001",  ......  },  {  "id": 2,  "supplierId": 15,  "sourcesSupplierId": "YGYS00002" ,  "hospitalId": 25,  "sourcesHospitalId": "YYY000009" ,  "storeHouseId": "1",  "sourcesStoreHouseId": "YKF000001",  "refundsApplyNo": "TH00002",  ......  }......  ]  }  } |

##### 5.7.1.2.2细单分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 主单列表显示 | 1、验收单编号、商品名称、生产厂家、商品规格、生产批号、商品单位、退货数量、退货原因  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "applyMasterId":2  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<RefundsApplyDetailVO> page = this.selectPage(  new Query<RefundsApplyDetailVO>(params).getPage(),  new EntityWrapper<RefundsApplyDetailVO>()  .eq(StringUtils.isNotBlank(params.get("applyMasterId").toString()), "apply\_master\_id", params.get("applyMasterId").toString())  );  return new PageUtils(page);  } |
| 查询语句 | select a.\*,b.supplier\_goods\_code,b.supplier\_goods\_name,b.supplier\_common\_name,b.supplier\_goods\_specs\_code,b.supplier\_goods\_specs\_name,  b.supplier\_factory\_code,b.supplier\_factory\_name,c.lotno as lot\_no,b.supplier\_goods\_unit  from hdi\_refunds\_apply\_detail a left join view\_hdi\_goods\_specs\_match b on a.goods\_type=b.goods\_type  and a.goods\_id=b.hospital\_goods\_id and a.specs\_id=b.hospital\_goods\_specs\_id  left join hdi\_core\_lot c on a.lot\_id=c.lotid |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 2,  "applyMasterId": 2,  "acceptMasterId": 2 ,  "acceptNo": "YS000001",  "acceptDetailId":1 ,  "goodsType": 3,  "goodsId": 1,  "specsId":1,  ......  },  {  "id": 3,  "applyMasterId": 3,  "acceptMasterId": 3 ,  "acceptNo": "YS000003",  "acceptDetailId":3,  "goodsType": 3,  "goodsId": 2,  "specsId":2,  ......  }......  ]  }  } |

##### 5.7.1.2.3新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 必填字段 | 医院名称、库房名称、医院退货申请单编号、申请时间、验收单编号、商品名称、生产厂商、商品规格、生产批号、商品单位、退货数量、退货原因 |
| 约束条件 | 输入主单内容后，当点击新增按钮后，主单已输入的内容不能再修改；当删除完明细单后，主单已输入的内容恢复进行修改。 |
| 查询验收单明细 | 验收单明细视图表：view\_hdi\_core\_accept\_detail |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "supplierId":15,  "hospitalId":25,  "sourcesHospitalId":"YGYS00001",  "storeHouseId":1,  "sourcesStoreHouseId":"YYY000008",  "refundsApplyNo":"THSQD00001",  "applyTime":"2019-06-06",  "refundsApplyDetailList": [  {  "acceptMasterId": 2 ,  "acceptNo": "YS000001",  "acceptDetailId":1,  "goodsType": 3,  "goodsId": 1,  "sourcesGoodsId": "1",  "specsId":1,  "sourcesSpecsId": "YYYP000002",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":2,  "realityRefundsNumber":2,  "supplyPrice":9.00,  "refundsRemark": "退货单测试1",  ......  },  {  "acceptMasterId": 3 ,  "acceptNo": "YS000003",  "acceptDetailId":3,  "goodsType": 3,  "goodsId": 2,  "sourcesGoodsId": "3",  "specsId":3,  "sourcesSpecsId": "YYYP000003",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":20,  "realityRefundsNumber":10,  "supplyPrice":15.00,  "refundsRemark": "退货单测试2",  ......  }......  ]  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(RefundsApplyMasterVO ramVO) {  if(ramVO != null){  //校验必填参数并且设置申请时间  checkValueAndSetTime(ramVO);    //设置初始状态为未确定  ramVO.setStatus(RefundsTypeEnum.UNCONFIRMED.getKey());  ramVO.setCreateTime(new Date());  //设置为系统录入  ramVO.setDataSource(HospitalDataSource.SYSTEM.getKey());  baseMapper.insert(ramVO);    //批量插入退货申请详细单  List<RefundsApplyDetailEntity> radeList = ramVO.getDetailList();  //设置详情单的退货申请单id和新增必填信息  for(RefundsApplyDetailEntity detail : radeList){  detail.setApplyMasterId(ramVO.getId());  detail.setCreateId(ramVO.getCreateId());  detail.setCreateTime(ramVO.getCreateTime());  }  //批量插入数据  refundsApplyDetailService.insertBatch(radeList);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.1.2.4修改

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 必填字段 | 医院名称、库房名称、医院退货申请单编号、申请时间、验收单编号、商品名称、生产厂商、商品规格、生产批号、商品单位、退货数量、退货原因 |
| 约束条件 | 1. 进入编辑页面后，主单已输入的内容不能进行编辑；当删除全部明细单数据后，主单已输入的内容恢复可编码；当点击新增按钮后，主单已输入的内容不能再修改； 2. 数据来源不是系统录入的，则不能编辑； 3. 数据来源是系统录入的，只能编辑状态为未确认的数据。 |
| 查询验收单明细 | 验收单明细视图表：view\_hdi\_core\_accept\_detail |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "id":1,  "supplierId":15,  "hospitalId":25,  "sourcesHospitalId":"YGYS00001",  "storeHouseId":1,  "sourcesStoreHouseId":"YYY000008",  "refundsApplyNo":"THSQD00001",  "applyTime":"2019-06-06",  "refundsApplyDetailList": [  {  "id": 2 ,  "acceptMasterId": 2 ,  "acceptNo": "YS000001",  "acceptDetailId":1,  "goodsType": 3,  "goodsId": 1,  "sourcesGoodsId": "1",  "specsId":1,  "sourcesSpecsId": "YYYP000002",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":2,  "realityRefundsNumber":2,  "supplyPrice":9.00,  "refundsRemark": "退货单测试1",  ......  },  {  "id": 3 ,  "acceptMasterId": 3 ,  "acceptNo": "YS000003",  "acceptDetailId":3,  "goodsType": 3,  "goodsId": 2,  "sourcesGoodsId": "3",  "specsId":3,  "sourcesSpecsId": "YYYP000003",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":20,  "realityRefundsNumber":10,  "supplyPrice":15.00,  "refundsRemark": "退货单测试2",  ......  }......  ]  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(RefundsApplyMasterVO ramVO) {  if(ramVO != null){  Long masterId = ramVO.getId();  if(masterId == null){  throw new HdiException("修改传入的id不能为空");  }  //修改退货申请主单信息  ramVO.setEditTime(new Date());  baseMapper.updateById(ramVO);    //修改退货申请详情信息  List<RefundsApplyDetailEntity> radeList = refundsApplyDetailService.selectList(new EntityWrapper<RefundsApplyDetailEntity>()  .eq("apply\_master\_id", masterId)  .eq("del\_flag", -1));//原来的详情信息  List<RefundsApplyDetailEntity> updateDetailList = ramVO.getDetailList();//修改的详情信息    //获取数据库里的退货申请详情的idlist  List<Long> idList = getIdList(radeList);    //循环遍历 没有id的话直接插入，有id的话把id从idList里移除，idList里剩下的就是被删除了的id  for(RefundsApplyDetailEntity updateDetail : updateDetailList){  if(updateDetail.getId() == null){  //执行新增操作  updateDetail.setApplyMasterId(masterId);  updateDetail.setCreateId(ramVO.getEditId());  updateDetail.setCreateTime(ramVO.getEditTime());  refundsApplyDetailService.insert(updateDetail);  }else{  //从idList里移除id并且执行修改操作  idList.remove(updateDetail.getId());    //执行详情修改操作  updateDetail.setEditId(ramVO.getEditId());  updateDetail.setEditTime(ramVO.getEditTime());  refundsApplyDetailService.updateById(updateDetail);  }  }    //删除idList里对应的详情记录  refundsApplyDetailService.deleteBatchIds(idList);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.1.2.5确认

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.refunds |
| 功能说明 | 可将未确认状态的退货申请单批量修改为已确认，其他状态则不能修改为已确认。 |
| 输入 | {  "id": [1,2,3]  } |
| 业务代码 | @Override  public void updateStatus(@RequestBody Long[] ids) {  //可将未确认状态的退货申请单批量修改为已确认  .......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.1.2.6删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.refunds |
| 功能说明 | 1、数据来源不是系统录入的，则不能删除；只能删除状态为未确认的数据。  2、做软删除，更新状态操作 |
| 输入 | {  "id": [1,2,3]  } |
| 业务代码 | @Override  public void delete(@RequestBody Long[] ids) {  //做软删除，更新状态操作  .......  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.1.2.7生成退货单

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 必填字段 | 退货数量、退货单价 |
| 约束条件 | 1、自动带出退货申请单数据，只能查看，不能编辑。 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "id":1,  "supplierId":15,  "hospitalId":25,  "sourcesHospitalId":"YGYS00001",  "storeHouseId":1,  "sourcesStoreHouseId":"YYY000008",  "refundsApplyNo":"THSQD00001",  "applyTime":"2019-05-06",  "regressionNumber":"THDH00001",  "refundsTime":"2019-05-16",  "refundsDetailList": [  {  "id": 2 ,  "acceptMasterId": 2 ,  "acceptNo": "YS000001",  "acceptDetailId":1,  "goodsType": 3,  "goodsId": 1,  "sourcesGoodsId": "1",  "specsId":1,  "sourcesSpecsId": "YYYP000002",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":2,  "realityRefundsNumber":2,  "refundsPrice":9.00,  "refundsRemark": "退货单测试1",  ......  },  {  "id": 3 ,  "acceptMasterId": 3 ,  "acceptNo": "YS000003",  "acceptDetailId":3,  "goodsType": 3,  "goodsId": 2,  "sourcesGoodsId": "3",  "specsId":3,  "sourcesSpecsId": "YYYP000003",  "lotId":1,  "goodsUnitCode":1,  "applyRefundsNumber":20,  "realityRefundsNumber":10,  "refundsPrice":15.00,  "refundsRemark": "退货单测试2",  ......  }......  ]  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void createRefundsDetail(RefundsApplyMasterVO ramVO) {  //更新退货申请明细单的实际退货数量  ......  //根据退货申请主单生成退货主单  ....  //根据退货申请明细单生成退货明细单  ....  } |
| 输出 | {  "msg": "success",  "code": 0  } |

### 5.7.2退货单管理

#### 5.7.2.1用例图



#### 5.7.2.2功能设计说明

##### 5.7.2.2.1主单分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院名称、库房名称、退货单编号、退货时间、状态 |
| 主单列表显示 | 1、退货单编号、销退单号、医院名称、库房名称、医院退货申请单编号、退货时间、状态、细单条数  2、排序：先按修改时间，再按创建时间倒序 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "page":1,  "limit":10,  "hospitalName":"医院名称",  "storeHouseName":"库房名称",  "refundsApplyNo":"医院退货申请单编号",  "applyTimeBeginStr":"申请开始时间",  "applyTimeEndStr":"申请结束时间",  "status":0  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false, tableAlias="t1")  public PageUtils queryPage(Map<String, Object> params) {  RefundsMasterVO rmVO = new RefundsMasterVO();  rmVO.setHospitalName((String)params.get("hospitalName"));  rmVO.setStoreHouseName((String)params.get("storeHouseName"));  rmVO.setRefundsApplyNo((String)params.get("refundsApplyNo"));  rmVO.setRefundsTimeBeginStr((String)params.get("refundsTimeBeginStr"));  rmVO.setRefundsTimeEndStr((String)params.get("refundsTimeEndStr"));  rmVO.setStatus((Integer)params.get("status"));  rmVO.setFileterDept((String)params.get(Constant.SQL\_FILTER));    String sidx = (String)params.get("sidx");  String order = (String)params.get("order");  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());    Page<RefundsMasterVO> page = new Page<RefundsMasterVO>(currPage, pageSize,order);  if(sidx!=null){  page.setAsc(sidx.equals("desc")?false:true);  }else{  //标志，设置默认按更新时间和创建时间排序  rmVO.setIsDefaultOrder(1);  }  //自定义分页查询列表带过滤条件  List<RefundsMasterVO> list = this.baseMapper.listForPage(page,rmVO);  //查询并设置退回  //setDetail(list);    page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | select  t1.\*,  (select count(id) from hdi\_refunds\_detail where master\_id = t1.id and del\_flag = 0) as detailNumber,  t2.hospital\_name as hospitalName,  t3.storehousename as storeHouseName  from hdi\_refunds\_master t1  left join hdi\_org\_hospital\_info t2 on t1.hospital\_id = t2.id  left join hdi\_core\_storehouse t3 on t1.store\_house\_id = t3.storehouseid  where t1.del\_flag = 0 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 1,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "refundsNo": "THD000001",  "supplierId": 15,  "sourcesSupplierId": "YGYS00001",  "hospitalId": 25,  "sourcesHospitalId": "YYY000008",  "storeHouseId": 1,  "sourcesStoreHouseId": "YKF000001",  "refundsApplyNo": "testrefunds",  "regressionNumber": "testxiaotui",  "refundsTime": "2019-06-06 15:49:21",  "status": 0,  "deptId": 1,  "createId": 1,  "createTime": "2019-06-06 16:47:20",  "editId": null,  "editTime": "2019-06-06 17:52:45",  "delFlag": 0,  "dataSource": 0,  "hospitalName": "广州益禾医院",  "storeHouseName": "1号库",  ……  }  ]  }  } |

##### 5.7.2.2.2细单分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 查询条件 | 医院名称、库房名称、退货单编号、退货时间、状态 |
| 细单列表显示 | 验收单编号、商品名称、生产厂商、商品规格、生产批号、商品单位、退货数量、退货原因 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "page": "1",  "limit": "10",  "id": 1  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false, tableAlias="t1")  public PageUtils queryPage(Map<String, Object> params) {  RefundsMasterVO rmVO = new RefundsMasterVO();  rmVO.setHospitalName((String)params.get("hospitalName"));  rmVO.setStoreHouseName((String)params.get("storeHouseName"));  rmVO.setRefundsApplyNo((String)params.get("refundsApplyNo"));  rmVO.setRefundsTimeBeginStr((String)params.get("refundsTimeBeginStr"));  rmVO.setRefundsTimeEndStr((String)params.get("refundsTimeEndStr"));  rmVO.setStatus((Integer)params.get("status"));  rmVO.setFileterDept((String)params.get(Constant.SQL\_FILTER));    String sidx = (String)params.get("sidx");  String order = (String)params.get("order");  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());    Page<RefundsMasterVO> page = new Page<RefundsMasterVO>(currPage, pageSize,order);  if(sidx!=null){  page.setAsc(sidx.equals("desc")?false:true);  }else{  //标志，设置默认按更新时间和创建时间排序  rmVO.setIsDefaultOrder(1);  }  //自定义分页查询列表带过滤条件  List<RefundsMasterVO> list = this.baseMapper.listForPage(page,rmVO);  //查询并设置退回  //setDetail(list);    page.setRecords(list);  return new PageUtils(page);  } |
| 查询语句 | select  t1.id,  t1.accept\_no as acceptNo,  t2.supplier\_goods\_name as goodsName,  t2.supplier\_factory\_name as factoryName,  t2.supplier\_goods\_specs\_name as specsName,  t3.lotno as lotName,  t1.goods\_unit\_code as goodsUnitName,  t1.reality\_refunds\_number as realityRefundsNumber,  t1.refunds\_remark as refundsRemark  from  hdi\_refunds\_detail t1  left join  view\_hdi\_goods\_specs\_match t2  on t1.goods\_id = t2.platform\_goods\_id  and t1.specs\_id = t2.platform\_goods\_specs\_id  and t1.goods\_type = t2.goods\_type  left join  hdi\_core\_lot t3  on t1.lot\_id = t3.lotid  and t3.lotstatus = 1  where  t1.del\_flag = 0  and master\_id = #{id} |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 1,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 1,  "acceptNo": "YS000001",  "realityRefundsNumber": 2,  "refundsRemark": "测试备注",  "goodsName": "感冒灵颗粒",  "factoryName": "白云山一厂公司",  "specsName": "8包/盒",  "lotName": "1001",  "goodsUnitName": "1"  },  {  "id": 2,  ……  },  ]  }  } |

##### 5.7.2.2.3新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 必填字段 | 医院名称、库房名称、退货时间、验收单编号、商品名称、生产厂商、商品规格、生产批号、商品规格、商品单位、生产批号、退货数量、退货单价、退货原因 |
| 约束条件 | 未保存的新增商品信息记录可删除。  点击“保存”按钮保存退货单和商品明细信息，并将退货单信息和商品明细记录显示在退货单管理列表中。 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "hospitalId": 0,  "sourcesHospitalId": "1",  "storeHouseId": 0,  "sourcesStoreHouseId": "1",  "supplierId": 0,  "refundsApplyNo": "123456",  "regressionNumber": "1234567",  "refundsTimeStr": "2019-01-01 00:00:00",  "refundsDetailList": [  {  "sourcesMasterId": 1,  "sourcesDetailId": 1,  "acceptNo": "string",  "goodsType": 0,  "goodsId": 1,  "sourcesGoodsId": "666",  "specsId": 1,  "sourcesSpecsId": "1",  "lotId": 1,  "goodsUnitCode": "WZ666",  "applyRefundsNumber": 10,  "realityRefundsNumber": 5,  "refundsRemark": "退货原因"  }  ]  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void save(RefundsMasterVO refundsMasterVO) {  if(refundsMasterVO != null){  //校验参数  checkValue(refundsMasterVO);  //设置初始状态为未提交  refundsMasterVO.setStatus(RefundsTypeEnum.UNSUBMIT.getKey());  refundsMasterVO.setCreateTime(new Date());  refundsMasterVO.setDataSource(HospitalDataSource.SYSTEM.getKey());  //设置原供应商id  refundsMasterVO.setSourcesSupplierId(orgSupplierInfoService.selectSourceSupplierId(refundsMasterVO.getSupplierId(), refundsMasterVO.getHospitalId()));  //自动生成退货单编号  String refundsNo = sysSequenceService.selectSeqValueBySeqCode(SequenceEnum.REFUNDS\_MASTER\_NO.getKey());  refundsMasterVO.setRefundsNo(refundsNo);  //保存退货单  baseMapper.insert(refundsMasterVO);    //保存退货详情单  List<RefundsDetailEntity> rdeList = refundsMasterVO.getRefundsDetailList();  //循环设置退货单id和新增必填信息  if(rdeList != null && rdeList.size() > 0 ){  int status = 3;  Long applyMasterId = null;  for(RefundsDetailEntity detail : rdeList){  if(applyMasterId == null){  //找出退货申请主单id  applyMasterId = detail.getSourcesMasterId();  }  if(StringUtils.isBlank(refundsMasterVO.getRefundsApplyNo())){  //来源验收单  detail.setSourcesType(1);  }else{  //来源退货申请单  detail.setSourcesType(0);  }  detail.setMasterId(refundsMasterVO.getId());    detail.setCreateId(refundsMasterVO.getCreateId());  detail.setCreateTime(refundsMasterVO.getCreateTime());  if(StringUtils.isNotBlank(refundsMasterVO.getRefundsApplyNo())){  //修改退货申请细单的实际退货数量  Long applyDetailId = detail.getSourcesDetailId();  RefundsApplyDetailEntity applyDetail = new RefundsApplyDetailEntity();  applyDetail.setId(applyDetailId);  applyDetail.setRealityRefundsNumber(detail.getRealityRefundsNumber());  refundsApplyDetailService.updateById(applyDetail);  if(detail.getApplyRefundsNumber() != detail.getRealityRefundsNumber()){  status = 4;  }  }    }  if(StringUtils.isNotBlank(refundsMasterVO.getRefundsApplyNo())){  //修改退货申请主单的状态  RefundsApplyMasterEntity rame = new RefundsApplyMasterEntity();  rame.setId(applyMasterId);  rame.setStatus(status);  refundsApplyMasterService.updateById(rame);  }  //批量插入退货单详情数据  refundsDetailService.insertBatch(rdeList);  }  }    } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.2.2.4编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 页面效果 |  |
| 必填字段 | 医院名称、库房名称、医院退货申请单编号、申请时间、验收单编号、商品名称、生产厂商、商品规格、生产批号、商品单位、退货数量、退货原因 |
| 约束条件 | 1. 如果是通过接口方式上传的退货单信息不允许在HDI编辑。 2. 已提交给医院的退货单信息也不能再编辑。 |
| 类包 | com.ebig.hdi.modules.refunds |
| 输入 | {  "hospitalId": 0,  "sourcesHospitalId": "1",  "storeHouseId": 0,  "sourcesStoreHouseId": "1",  "supplierId": 0,  "refundsApplyNo": "123456",  "regressionNumber": "1234567",  "refundsTimeStr": "2019-01-01 00:00:00",  "refundsDetailList": [  {  "sourcesMasterId": 1,  "sourcesDetailId": 1,  "acceptNo": "string",  "goodsType": 0,  "goodsId": 1,  "sourcesGoodsId": "666",  "specsId": 1,  "sourcesSpecsId": "1",  "lotId": 1,  "goodsUnitCode": "WZ666",  "applyRefundsNumber": 10,  "realityRefundsNumber": 5,  "refundsRemark": "退货原因"  }  ]  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void update(RefundsMasterVO rmVO) {  if(rmVO != null){  Long masterId = rmVO.getId();  if(masterId == null){  throw new HdiException("修改传入的id不能为空");  }  //设置退货时间  if(StringUtils.isNotBlank(rmVO.getRefundsTimeStr())){  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  Date refundsTime = null;  try{  refundsTime = sdf.parse(rmVO.getRefundsTimeStr());  rmVO.setRefundsTime(refundsTime);  }catch (Exception e) {  throw new HdiException("退货时间格式不正确(1960-01-01 00:00:00)");  }  }  //修改退货主单信息  rmVO.setEditTime(new Date());  baseMapper.updateById(rmVO);    //修改退货申请详情信息  List<RefundsDetailEntity> radeList = refundsDetailService.selectList(new EntityWrapper<RefundsDetailEntity>()  .eq("master\_id", masterId)  .eq("del\_flag", 0));//原来的详情信息  List<RefundsDetailEntity> updateDetailList = rmVO.getRefundsDetailList();//修改的详情信息    //获取数据库里的退货申请详情的idlist  List<Long> idList = getIdList(radeList);    //循环遍历 没有id的话直接插入，有id的话把id从idList里移除，idList里剩下的就是被删除了的id  if(updateDetailList != null && updateDetailList.size() > 0 ){  int status = 3;  Long applyMasterId = null;  for(RefundsDetailEntity updateDetail : updateDetailList){  if(updateDetail.getId() == null){  //执行新增操作  updateDetail.setMasterId(masterId);  updateDetail.setCreateId(rmVO.getEditId());  updateDetail.setCreateTime(rmVO.getEditTime());  refundsDetailService.insert(updateDetail);  }else{  //从idList里移除id并且执行修改操作  idList.remove(updateDetail.getId());    //执行详情修改操作  updateDetail.setEditId(rmVO.getEditId());  updateDetail.setEditTime(rmVO.getEditTime());  refundsDetailService.updateById(updateDetail);  }  if(StringUtils.isNotBlank(rmVO.getRefundsApplyNo())){  //修改退货申请细单的实际退货数量  Long applyDetailId = updateDetail.getSourcesDetailId();  RefundsApplyDetailEntity applyDetail = new RefundsApplyDetailEntity();  applyDetail.setId(applyDetailId);  applyDetail.setRealityRefundsNumber(updateDetail.getRealityRefundsNumber());  refundsApplyDetailService.updateById(applyDetail);  if(applyDetail.getApplyRefundsNumber() != applyDetail.getRealityRefundsNumber()){  status = 4;  }  }  }  if(StringUtils.isNotBlank(rmVO.getRefundsApplyNo())){  //修改退货申请主单的状态  RefundsApplyMasterEntity rame = new RefundsApplyMasterEntity();  rame.setId(applyMasterId);  rame.setStatus(status);  refundsApplyMasterService.updateById(rame);  }  //删除idList里对应的详情记录  if(idList !=null && idList.size() > 0 ){  refundsDetailService.deleteBatchIds(idList);  }  }  }    } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.2.2.5删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.refunds |
| 功能说明 | 1. 如果是通过接口方式上传的退货单信息记录不能删除。 2. 已提交给医院的退货单记录也不能删除。 |
| 输入 | [1,2,3] |
| 业务代码 | @RequestMapping("/delete")  //@RequiresPermissions("refunds:refundsmaster:delete")  public Hdi delete(@RequestBody Long[] ids){  refundsMasterService.deleteBatchIds(Arrays.asList(ids));  return Hdi.ok();  } |
| 输出 | {  "msg": "success",  "code": 0  } |

##### 5.7.2.2.6提交医院

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.refunds |
| 功能说明 | 提交时如果已录入发票信息，要同时提交发票信息。 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void submit(Long[] ids,Long userId) {  if(ids != null){  RefundsMasterEntity refundsMasterEntity = new RefundsMasterEntity();  refundsMasterEntity.setEditId(userId);  refundsMasterEntity.setEditTime(new Date());  refundsMasterEntity.setStatus(RefundsTypeEnum.SUBMITED.getKey());  for(Long id : ids){  refundsMasterEntity.setId(id);  baseMapper.updateById(refundsMasterEntity);  }  }    } |
| 输出 | {  "msg": "success",  "code": 0  } |

## 5.8公共模块

### 5.8.1字典查询设计

#### 5.8.1.1根据类型查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 查询条件 | 字典类型 |
| 列表显示 | 字典类型、父字典码、字典码、字典值 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "type":"province,city,area"  } |
| 业务代码 | @Override  public List<SysDictEntity> selectDictByType(String type) {  return sysDictDao.selectDictByType(StringUtil.getInParam(type));  } |
| 输出 | {  "msg": "success",  "code": 0,  "dictList": [  {  "id": null,  "name": null,  "type": "province",  "parentCode": "",  "code": "11",  "value": "北京市",  "orderNum": null,  "remark": null,  "delFlag": null  },  {  "id": null,  "name": null,  "type": "province",  "parentCode": "",  "code": "12",  "value": "天津市",  "orderNum": null,  "remark": null,  "delFlag": null  }…  ]  } |

#### 5.8.1.2根据类型和父字典码查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 查询条件 | 字典类型、父字典码 |
| 列表显示 | 字典类型、父字典码、字典码、字典值 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "type":"city ",  " parentCode ":"44"  } |
| 业务代码 | @Override  public List<SysDictEntity> selectDictByTypeAndParentCode(String type, String parentCode) {  return sysDictDao.selectDictByTypeAndParentCode(type, parentCode);  } |
| 输出 | {  "msg": "success",  "code": 0,  "dictList": [  {  "id": null,  "name": null,  "type": "city",  "parentCode": "44",  "code": "4401",  "value": "广州市",  "orderNum": null,  "remark": null,  "delFlag": null  },  {  "id": null,  "name": null,  "type": "city",  "parentCode": "44",  "code": "4402",  "value": "韶关市",  "orderNum": null,  "remark": null,  "delFlag": null  }…  ]  } |

### 5.8.2枚举查询设计

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 查询条件 | 枚举类型 |
| 列表显示 | 枚举类型、key值、value值 |
| 输入 | http://localhost:2320/hdi-server-admin/sys/dict/getEnumType/StatusEnum,IsGroupEnum |
| 业务代码 | @RequestMapping("/getEnumType/{type}")  public Hdi getEnumType(@PathVariable("type") String type) {  try {  String [] signs = type.split(",");  if(signs.length<1){  return Hdi.error("请传入正确的参数");  }  Map<String, Object> map = new HashMap<String, Object>();  for(String str : signs){  List<LinkedHashMap<String, Object>> list = new ArrayList<LinkedHashMap<String, Object>>();  Class clazz = Class.forName("com.ebig.hdi.common.enums."+str);  Method toKey = clazz.getMethod("getKey");  Method toName = clazz.getMethod("getValue");  //得到enum的所有实例  Object[] objs = clazz.getEnumConstants();  for (Object obj : objs) {  LinkedHashMap<String, Object> subMap = new LinkedHashMap<String, Object>();  subMap.put("key", toKey.invoke(obj));  subMap.put("value", toName.invoke(obj));  list.add(subMap);  }  map.put(str, list);  }  return Hdi.ok().put("list",map);  } catch (Exception e) {  throw new HdiException("字典查询出错" + e.getMessage());  }  } |
| 输出 | {  "msg": "success",  "code": 0,  "list": {  "StatusEnum": [  {  "key": 0,  "value": "停用"  },  {  "key": 1,  "value": "启用"  }  ],  "IsGroupEnum": [  {  "key": 0,  "value": "否"  },  {  "key": 1,  "value": "是"  }  ]  }  } |
|  |  |

### 5.8.3导入功能设计

### 5.8.4导出功能设计

## 5.9系统消息

### 5.9.1系统消息管理

#### 5.9.1.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 查询条件 | 消息类型 |
| 显示字段 | 消息类型、标题、简介、内容、状态、创建时间、修改时间、发布时间 |
| 排序 | 创建时间倒序 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "page":"1",  "limit":"10"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  Page<SysMessageEntity> page = this.selectPage(new Query<SysMessageEntity>(params).getPage(),  new EntityWrapper<SysMessageEntity>().addFilterIfNeed(params.get(Constant.SQL\_FILTER) != null,  (String) params.get(Constant.SQL\_FILTER)).orderBy("create\_time", false));  return new PageUtils(page);  } |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 2,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id": 2,  "type": 0,  "title": "预警提醒消息",  "description": "预警提醒简介",  "content": "预警提醒测试内容",  "status": 0,  "createId": 1,  "createTime": "2019-04-26 15:47:10",  "editId": null,  "editTime": null,  "publishTime": null  },  {  "id": 1,  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  "status": 0,  "createId": 1,  "createTime": "2019-04-26 15:46:00",  "editId": 1,  "editTime": "2019-04-26 15:59:33",  "publishTime": null  }  ]  }  } |

#### 5.9.1.2新增

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  "status": 0,  ......  } |
| 业务代码 | public Hdi save(@RequestBody SysMessageEntity sysMessage) {  ValidatorUtils.validateEntity(sysMessage);  sysMessage.setCreateId(getUserId());  sysMessageService.save(sysMessage);  return Hdi.ok();  } |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.9.1.3编辑

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "id":1,  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  "status": 0,  ......  } |
| 业务代码 | public Hdi update(@RequestBody SysMessageEntity sysMessage) {  ValidatorUtils.validateEntity(sysMessage);  sysMessage.setEditId(getUserId());  sysMessageService.update(sysMessage);  return Hdi.ok();  } |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.9.1.4发布

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "userIds" : [1,2,3,4,5],  "messageId" : "1"  } |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public void publish(Map<String, Object> params) {  List<Long> userIds = JSON.parseArray(params.get("userIds").toString(), Long.class);  if (StringUtil.isEmpty(userIds)) {  throw new HdiException("请选择发布用户");  }  Long messageId = Long.valueOf(params.get("messageId").toString());  SysMessageEntity sysMessageEntity = new SysMessageEntity();  sysMessageEntity.setId(messageId);  sysMessageEntity.setStatus(MessageStatusEnum.PUBLISH.getKey());  this.updateById(sysMessageEntity);  List<SysUserEntity> sysUserEntities = sysUserService.selectBatchIds(userIds);  for (SysUserEntity sysUserEntity : sysUserEntities) {  SysUserMessageEntity sysUserMessageEntity = new SysUserMessageEntity();  sysUserMessageEntity.setUserId(sysUserEntity.getUserId());  sysUserMessageEntity.setDeptId(String.valueOf(sysUserEntity.getDeptId()));  sysUserMessageEntity.setMessageId(messageId);  sysUserMessageEntity.setStatus(UserMessageStatusEnum.UNREAD.getKey());  sysUserMessageService.insert(sysUserMessageEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |

#### 5.9.1.5查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | http://localhost:2320/hdi-server-admin/sys/sysmessage/1 |
| 业务代码 | public Hdi info(@PathVariable("id") Long id){  SysMessageEntity sysMessage = sysMessageService.selectById(id);  return Hdi.ok().put("sysMessage", sysMessage);  } |
| 输出 | {  "msg": "success",  "code": 0,  "sysMessageEntity ": {  "id":1,  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  "status": 0,  ......  }  } |

### 5.9.2用户消息管理

#### 5.9.2.1分页查询

|  |  |
| --- | --- |
| 事项 | 内容 |
| 查询条件 | 消息类型 |
| 显示字段 | 消息类型、标题、简介、内容、创建时间、修改时间、发布时间、状态（阅读状态）、阅读时间 |
| 排序 | 创建时间倒序 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | {  "page":"1",  "limit":"10"  } |
| 业务代码 | @Override  @DataFilter(subDept = true, user = false)  public PageUtils queryPage(Map<String, Object> params) {  int currPage = Integer.parseInt(params.get("page").toString());  int pageSize = Integer.parseInt(params.get("limit").toString());  Page<SysUserMessageEntityVo> page = new Page<SysUserMessageEntityVo>(currPage, pageSize);  List<SysUserMessageEntityVo> list = this.baseMapper.listForPage(page, params.get("userId"));  page.setRecords(list);  return new PageUtils(page);  } |
| 查询SQL | select u.\*,m.type,m.title,m.description,m.content from sys\_user\_message u left join sys\_message m on u.message\_id=m.id where u.user\_id=#{userId} and u.status!=2 |
| 输出 | {  "msg": "success",  "code": 0,  "page": {  "totalCount": 6,  "pageSize": 10,  "totalPage": 1,  "currPage": 1,  "list": [  {  "id":1,  "messageId": 1,  "deptId": 7,  "status": 0,  "readTime": "2019-04-28 09:11:11",  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  ......  },  {  "id":2,  "messageId": 2,  "deptId": 7,  "status": 0,  "readTime": "2019-04-28 09:30:11",  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  ......  ......  }......  ]  }  } |

#### 5.9.1.5查询详情

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 输入 | http://localhost:2320/hdi-server-admin/sys/sysusermessage/info/1 |
| 业务代码 | @Override  @Transactional(rollbackFor = Exception.class)  public SysUserMessageEntityVo queryInfoById(Long id) {  SysUserMessageEntity sysUserMessageEntity = new SysUserMessageEntity();  sysUserMessageEntity.setId(id);  sysUserMessageEntity.setStatus(UserMessageStatusEnum.READ.getKey());  sysUserMessageEntity.setReadTime(new Date());  this.updateById(sysUserMessageEntity);  return this.baseMapper.queryInfoById(id);  } |
| 查询SQL | select u.\*,m.type,m.title,m.description,m.content from sys\_user\_message u left join sys\_message m on u.message\_id=m.id where u.id=#{id} |
| 输出 | {  "msg": "success",  "code": 0,  "sysUserMessageEntityVo": {  "id":1,  "messageId": 1,  "deptId": 7,  "status": 0,  "readTime": "2019-04-28 09:11:11",  "type": 0,  "title": "系统消息",  "description": "测试消息简介",  "content": "系统消息测试内容",  ......  }  } |

#### 5.9.1.6删除

|  |  |
| --- | --- |
| 事项 | 内容 |
| 类包 | com.ebig.hdi.modules.sys |
| 功能说明 | 做软删除，更新状态操作 |
| 输入 | [1,2,3] |
| 业务代码 | @Override  public void delete(Long[] ids) {  for(Long id : ids) {  SysUserMessageEntity sysUserMessageEntity = new SysUserMessageEntity();  sysUserMessageEntity.setId(id);  //设置信息为删除状态  sysUserMessageEntity.setStatus(UserMessageStatusEnum.DELETE.getKey());  this.updateById(sysUserMessageEntity);  }  } |
| 输出 | {  "msg": "success",  "code": 0  } |