

北京市医疗保障信息平台二期

数据服务应用子系统（初步建设阶段）

详细设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 项目编号 | XPT-DCC | 文件编号 | SXR\_TS\_DDS\_V6.0 |
| 记录号 | XPT-DCC*\_*TS\_DDS | 版本号 | V1.0.0 |
| 正文页数 | 547 | 附件个数 | 0 |
| 批准/日期**：** | | | |

首都信息发展股份有限公司

首信医联

**文件修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **变化状态\*** | **简要说明**  **（变更内容和变更范围）** | **变更日期** | **变更人** |
| 1.0.0 | C | 创建 | 2023-11-06 | 张俊杰、王自华、李阳、任保罗、杨仪珊、许畅 |
| 1.0.0 | M | 增加控制类业务逻辑和部分重要实现逻辑图，调整子系统组织形式，并添加模块和功能描述与流程设计，添加设计页面图 | 2024-05-09 | 张俊杰、王自华、任保罗、杨仪珊、许畅、郭艳来、梁健伟、王行斌、蒋飞 |
|  |  |  |  |  |

\*变化状态：C—创建，A—增加，M—修改，D—删除

**文件审批信息**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **审批人** | **角色** | **审批意见** | **签字** | **审批日期** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[6 运营管理平台 1](#_Toc12076)

[6.1 功能模块逻辑描述 1](#_Toc12287)

[6.2 审批管理 1](#_Toc20112)

[6.2.1 我的申请 2](#_Toc22960)

[6.2.2 流程审批 35](#_Toc26121)

[6.2.3 流程配置 73](#_Toc20806)

[6.3 系统配置 83](#_Toc21091)

[6.3.1 用户管理 84](#_Toc27520)

[6.3.2 角色管理 107](#_Toc23553)

[6.3.3 机构/部门管理 130](#_Toc25444)

[6.3.4 菜单管理 156](#_Toc23862)

[6.3.5 资源管理 177](#_Toc12221)

[6.3.6 场景管理 211](#_Toc26718)

[6.4 消息管理 231](#_Toc7501)

[6.4.1 消息管理 232](#_Toc21436)

[7 隐私模型服务平台 247](#_Toc7079)

[7.1 功能模块逻辑描述 247](#_Toc372)

[7.2 TEE可信计算 248](#_Toc22995)

[7.2.1 隐私查询 248](#_Toc11282)

[7.2.2 任务调度 262](#_Toc31024)

[7.3 安全多方计算 268](#_Toc1145)

[7.3.1 项目管理 269](#_Toc4133)

[7.4 配置管理 299](#_Toc5178)

[7.4.1 脱敏管理 300](#_Toc17316)

[7.4.2 证书管理 305](#_Toc12183)

[7.4.3 资源管理 314](#_Toc13717)

[7.4.4 成员管理 321](#_Toc23754)

[7.5 可信执行容器 328](#_Toc14223)

[7.5.1 数据预处理 329](#_Toc576)

[7.5.2 应用管理 348](#_Toc27806)

[7.5.3 安全沙箱容器 385](#_Toc31717)

[7.5.4 数据安全服务 439](#_Toc6620)

[7.5.5 安全审计 471](#_Toc30658)

[7.5.6 运营管理 511](#_Toc9509)

# 运营管理平台

## 功能模块逻辑描述

运营管理平台提供整体系统的配置管理，包括审批管理、系统配置、消息管理等。

主要包含的功能如下：



## 审批管理

审批管理提供系统中全部的审批流程管理，包括应用上架/申请审批、数据集发布/申请审批、资源申请审批、积分申请审批等方面。主要流程如下：



### 我的申请

#### 类设计

##### 控制类

###### SysApproveController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysApproveController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成审批流程管理与流程审批的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\approveController\SysApproveController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| selectApproveList | session[:taskId] | string | ■公有 □私有 □保护 | Result | 审批详情-审批流程 |
| selectApprovePage | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 审批列表 |
| selectMyApprovePage | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 我的申请 |
| complete | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 流程审批 |
| detail | session[:id] | long | ■公有 □私有 □保护 | Result | 审批详情-详情信息 |
| withdraw | session[:taskId] | string | ■公有 □私有 □保护 | Result | 撤销审批 |
| start | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 发起安全审核 |
| getSafeInfo | session[:id] | long | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-应用信息 |
| startStaticCheck | session[:taskId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-开始扫描 |
| applicationDetail | session[:taskId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-获取扫描信息 |
| applicationProcess | session[:JSONObject] | JSONObject | ■公有 □私有 □保护 | Result | 应用安全扫描-审批通过 |
| startDynamicInstance | session[:appId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-动态安全监测-启动实例 |
| dynamicInstanceInfo | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-实例状态 |
| dynamicInstancePage | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-动态安全监测-查询信息 |
| manualStartDynamicInstance | session[:appId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-人工审查-启动实例 |
| stopDynamicInstance | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-停止实例 |

算法描述

审批详情-审批流程

接口描述：根据审批流程id查询该审批所属审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 流程实例id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 返回流程 |
| data>>name | [string] | 是 | 节点名称 |
| data>>type | [string] | 是 | 节点审批类型 |
| data>>approveName | [string] | 是 | 审批人 |
| data>>approveId | [string] | 是 | 审批人id |
| data>>comment | [string] | 是 | 审批意见 |
| data>>status | [string] | 是 | 审批状态 |
| data>>approveTime | [number] | 是 | 审批时间 |
| code | [number] | 是 | 状态码 |

审批列表

接口描述：展示用户待审批或已审批的审批流程记录。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 否 | 审批名称 |
| status | [string] | 否 | 流程状态（REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| page | [number] | 是 | 页数 |
| pageSize | [string] | 是 | 页码 |
| startTime | [number] | 否 | 申请开始时间 |
| endTime | [number] | 否 | 申请结束时间 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 |  |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>list>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>list>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总条数 |
| data>>totalNum | [number] | 是 | 总页数 |
| code | [number] | 是 | 状态码 |

我的申请

接口描述：展示用户所申请的审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list/page/my | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 否 | 审批名称 |
| status | [string] | 否 | 流程状态（REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| page | [number] | 是 | 页数 |
| pageSize | [string] | 是 | 页码 |
| startTime | [number] | 否 | 申请开始时间 |
| endTime | [number] | 否 | 申请结束时间 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 |  |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>list>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>list>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总条数 |
| data>>totalNum | [number] | 是 | 总页数 |
| code | [number] | 是 | 状态码 |

审批

接口描述：根据各个登录用户获取该用户所属机构下的角色列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/complete | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 流程id |
| comment | [string] | 是 | 审批意见 |
| status | [string] | 是 | 审批状态（NOT-PASS 未通过/PASS 已通过) |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

审批详情-详情信息

接口描述：根据不同类型查询审批详情信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/detail | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>id | [number] | 是 | 流程id |
| data>>userId | [number] | 是 | 申请用户id |
| data>>userDept | [string] | 是 | 申请用户机构名 |
| data>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>userName | [string] | 是 | 申请用户名 |
| data>>approveRole | [string] | 是 | 审批用户角色 |
| data>>approveDept | [string] | 是 | 审批用户机构 |
| data>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>approveId | [null] | 是 | 审批用户id |
| data>>comment | [null] | 是 | 审批意见 |
| data>>taskId | [string] | 是 | 流程实例id |
| data>>taskName | [string] | 是 | 流程名称 |
| data>>name | [string] | 是 | 流程目标名称 |
| data>>relevanceContent | [null] | 是 | 流程关联详情内容 |
| data>>relevanceId | [null] | 是 | 流程关联详情内容id |
| data>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>projectId | [null] | 是 | 项目id |
| data>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>content | [string] | 是 | 根据不同类型审核流程的对应展示正文 |
| data>>projectName | [string] | 是 | 项目名 |
| code | [number] | 是 | 状态码 |

我的申请-撤销申请

接口描述：根据流程id对申请的审批流程进行撤销回退。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/withdraw | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 流程实例id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| code | [number] | 是 | 状态码 |

发起安全审核

接口描述：生产级应用进行安全审核，调用该接口发起安全审核，创建可信执行容器系统内部应用审核记录。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/start | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceId | [number] | 是 | 关联应用id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 返回信息 |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-应用信息

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时查询应用信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/safe/info | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>icon | [string] | 是 | 图标 |
| data>>description | [string] | 是 | 描述 |
| data>>ports | [string] | 是 | 端口 |
| data>>type | [string] | 是 | 应用类型（web web服务类） |
| data>>staticFileDir | [string] | 是 | 静态文件路径 |
| data>>id | [string] | 是 | 应用id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>usageSteps | [string] | 是 | 使用步骤 |
| data>>swaggerFile | [string] | 是 | swagger文件 |
| data>>outputPaths | [string] | 是 | 输出路径 |
| data>>name | [string] | 是 | 应用名称 |
| data>>outputSamples | [string] | 是 | 输出样例路径 |
| data>>taskId | [string] | 是 | 任务id |
| data>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-开始扫描

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时开始扫描。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/start/check | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 返回信息 |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-获取扫描信息

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时获取扫描信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/application/detail | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [string] | 是 | 静态扫描结果 |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-审批通过

接口描述：可信执行容器系统进行应用安全扫描审批通过。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/application/process | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 审核流程id |
| auditType | [string] | 是 | 通过类型（static静态监测,dynamic动态监测,artificial人工监测,final最后审批） |
| status | [number] | 是 | 默认传2（2审批通过） |
| result | [string] | 是 | 静态扫描结果 |
| opinion | [string] | 是 | 审批意见 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-动态安全监测-启动实例

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，启动监测应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/start/dynamicInstance | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| appId | [string] | 是 | 应用id（应用安全扫描-应用信息获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 实例id |
| code | [number] | 是 | 状态码 |

应用安全扫描-实例状态

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，查询启动监测应用实例的状态信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/dynamicInstance/info | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 实例id（启动实例接口获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 实例id |
| data>>instanceStatus | [string] | 是 | 实例状态 未知("Unknown"),已暂停 ("2"),运行中 ("Synced"),已删除 ("1"),等待中 ("Pending"),运行中1 ("containerIsRunning"),启动中 ("containerNotRunning"),创建失败 ("Failed"); |
| code | [number] | 是 | 状态码 |

应用安全扫描-动态安全监测-查询信息

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，查询应用安全信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/dynamicInstance/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 实例id（启动实例接口获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-人工审查-启动实例

接口描述：可信执行容器系统进行应用安全扫描人工审查监测时，启动应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/manual/start/dynamicInstance | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| appId | [string] | 是 | 应用id（应用安全扫描-应用信息获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-停止实例

接口描述：可信执行容器系统进行应用安全扫描时，关闭应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/detail |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysApprove

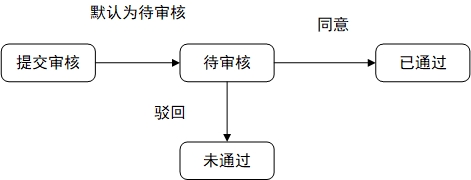
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysApprove | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统审批流程的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\approve\po\SysApprove | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 流程id |
| userId | long | | ■公有 □私有 □保护 | | | NNI | | 申请用户id |
| userDept | string | | ■公有 □私有 □保护 | | |  | | 申请用户机构名 |
| userDeptId | long | | ■公有 □私有 □保护 | | |  | | 申请用户机构id |
| userName | string | | ■公有 □私有 □保护 | | |  | | 申请用户名 |
| approveRole | string | | ■公有 □私有 □保护 | | |  | | 审批用户角色 |
| approveDept | string | | ■公有 □私有 □保护 | | |  | | 审批用户机构 |
| approveDeptId | long | | ■公有 □私有 □保护 | | |  | | 审批用户机构id |
| approveId | long | | ■公有 □私有 □保护 | | |  | | 审批用户id |
| comment | string | | ■公有 □私有 □保护 | | |  | | 审批意见 |
| taskId | string | | ■公有 □私有 □保护 | | |  | | 流程实例id |
| taskName | string | | ■公有 □私有 □保护 | | |  | | 流程名称 |
| name | string | | ■公有 □私有 □保护 | | |  | | 流程目标名称 |
| relevanceContent | string | | ■公有 □私有 □保护 | | |  | | 流程关联详情内容 |
| relevanceId | long | | ■公有 □私有 □保护 | | |  | | 流程关联详情内容id |
| relevanceSize | string | | ■公有 □私有 □保护 | | |  | | 流程关联额度 |
| projectId | long | | ■公有 □私有 □保护 | | |  | | 项目id |
| type | string | | ■公有 □私有 □保护 | | |  | | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| status | string | | ■公有 □私有 □保护 | | |  | | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| createTime | long | | ■公有 □私有 □保护 | | | NNI | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | | NNI | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_approve | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_approve、sc\_project、sc\_application、sc\_data\_set、sc\_model、sc\_sys\_flow、sc\_project\_application、sc\_project\_data、sc\_project\_model、sc\_project\_service、sc\_sys\_flow\_detail、sc\_sys\_user、sc\_project\_task、sc\_project\_process、sc\_service、sc\_data\_set\_log、sc\_business\_scenario、sc\_application\_place、sc\_service\_place、sc\_resource、sc\_sys\_dict\_data、sc\_resource\_project、sc\_operate\_indent、sc\_operate\_approve、sc\_model\_place、sc\_data\_set\_place**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  \*  FROM  sc\_sys\_approve  WHERE  task\_id = #{taskId}  ORDER BY  a.create\_time DESC  【SQL02】  SELECT  d.id,  d.offer\_dept\_id as offerDeptId,  d.offer\_user\_id as offerUserId,  d.user\_id as userId,  d.offer\_user\_name AS offerUserName,  d.NAME,  d.alias,  d.create\_time AS createTime,  d.update\_time as updateTime,  p.project\_id as projectId,  p.task\_id as taskId,  d.source\_type as sourceType,  d.safe\_data\_set\_id as safeDataSetId,  d.safe\_status as safeStatus,  d.safe\_test\_set\_id as safeTestSetId,  d.status as status,  d.points,  d.billing\_mode AS billingMode,  d.visible  FROM sc\_project\_data p  left JOIN sc\_data\_set d on p.data\_id=d.id  <where>  <if test="id != null and id != '' ">  and p.project\_id = #{id}  </if>  <if test="taskId != null and taskId != '' ">  and p.task\_id = #{taskId}  </if>  <if test="userId != null and userId != '' ">  and d.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and d.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL03】  SELECT  a.application\_id AS applicationId,  a.user\_id AS userId,  a.offer\_user\_id AS offerUserId,  a.offer\_user\_name AS offerUserName,  a.offer\_dept\_id AS offerDeptId,  a.offer\_dept\_name AS offerDeptName,  a.NAME AS applicationName,  a.description AS description,  a.type AS type,  a.icon AS icon,  a.url AS url,  a.cpu AS cpu,  a.internal\_memory AS internalMemory,  a.disk\_memory AS diskMemory,  a.points AS points,  a.safe\_application\_id as safeApplicationId,  a.safe\_application\_test\_id as safeApplicationTestId,  a.safety\_level AS safetyLevel,  a.STATUS AS STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  a.authority as authority,  a.source\_type as sourceType,  a.ports,  a.points,  a.billing\_mode AS billingMode,  a.authority as authority,  p.project\_id as projectId,  p.task\_id as taskId  FROM  sc\_project\_application p  left JOIN sc\_application a ON p.application\_id = a.application\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  <if test="userId != null and userId != '' ">  and a.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and a.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL04】  SELECT  a.id AS id,  a.point,  a.provider\_id AS providerId,  a.provider\_dept\_id AS providerDeptId,  a.billing\_mode AS billingMode,  a.NAME  FROM  sc\_service a  INNER JOIN sc\_project\_service p ON a.id = p.service\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  </where>  【SQL05】  SELECT  a.id AS id,  a.user\_id AS userId,  a.offer\_user\_id AS offerUserId,  a.offer\_user\_name AS offerUserName,  a.offer\_dept\_id AS offerDeptId,  a.offer\_dept\_name AS offerDeptName,  a.NAME,  a.description AS description,  a.type AS type,  a.icon AS icon,  a.url AS url,  a.points AS points,  a.STATUS AS STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  p.project\_id as projectId,  p.task\_id as taskId  FROM  sc\_model a  INNER JOIN sc\_project\_model p ON a.id = p.model\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  <if test="userId != null and userId != '' ">  and a.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and a.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL06】  SELECT  \*  FROM  sc\_sys\_dept  WHERE  type = "OPERATION-SIDE"  【SQL07】  SELECT  \*  FROM  sc\_sys\_flow  WHERE  type = #{type}  and dept\_id in (#{deptId})  【SQL08】  SELECT  \*  FROM  sc\_sys\_flow\_detail  WHERE  flow\_id = #{flowId}  【SQL09】  select user\_id, login\_name, user\_name from sc\_sys\_user where user\_id in  <foreach collection="ids" item="userId" open="(" separator="," close=")">  #{userId}  </foreach>  【SQL10】  SELECT id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,name,business\_id,alias,description,points,tag,type,privacy\_inquiry,visible,source\_type,update\_strategy,status,secret\_key,create\_time,update\_time  FROM  sc\_data\_set  WHERE  id = #{dataSetId}  【SQL11】  SELECT  a.id,  a.user\_id AS userId,  a.user\_dept AS userDept,  a.user\_name AS userName,  a.approve\_role AS approveRole,  a.approve\_dept AS approveDept,  a.approve\_dept\_id AS approveDeptId,  a.approve\_id AS approveId,  a.COMMENT,  a.task\_id AS taskId,  a.task\_name AS taskName,  a.NAME,  a.relevance\_content AS relevanceContent,  a.relevance\_id AS relevanceId,  a.relevance\_size AS relevanceSize,  a.project\_id AS projectId,  a.type,  a.STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  p.NAME AS projectName  FROM  sc\_sys\_approve a  LEFT JOIN sc\_project p ON p.id = a.project\_id  < WHERE >  < IF test = "userId != null" >  AND a.user\_id = #{userId}  </ IF >  < IF test = "approveDeptId != null" >  AND a.approve\_dept\_id = #{approveDeptId}  </ IF >  < IF test = "type != null and type != ''" >  AND a.type = #{type}  </ IF >  < IF test = "userName != null and userName != ''" >  AND a.user\_name = #{userName}  </ IF >  < IF test = "name != null and name != ''" >  AND a.NAME = #{name}  </ IF >  < IF test = "status != null and status != ''" >  AND a.STATUS = #{status}  </ IF >  </ WHERE >  ORDER BY  a.create\_time DESC  【SQL12】  SELECT id,user\_id,user\_dept,user\_dept\_id,user\_name,approve\_role,approve\_dept,approve\_id,approve\_dept\_id,comment,task\_id,task\_name,name,relevance\_content,relevance\_id,relevance\_size,project\_id,type,status,create\_time,update\_time,approve\_description  FROM sc\_sys\_approve  WHERE id=#{id}  【SQL13】  SELECT  a.application\_id as applicationId,  a.user\_id as userId,  a.offer\_user\_id as offerUserId,  a.offer\_user\_name as offerUserName,  a.offer\_dept\_id as offerDeptId,  a.offer\_dept\_name as offerDeptName,  a.NAME as applicationName ,  a.description as description,  a.type as type,  a.icon as icon,  a.url as url,  a.dept\_type as deptType,  a.cpu as cpu,  a.file\_url as fileUrl,  a.application\_type as applicationType,  a.internal\_memory as internalMemory,  a.disk\_memory as diskMemory,  a.points as points,  a.billing\_mode AS billingMode,  a.safety\_level as safetyLevel,  a.STATUS as status,  a.safe\_application\_id as safeApplicationId,  a.safe\_application\_test\_id as safeApplicationTestId,  a.create\_time as createTime,  a.update\_time as updateTime,  a.authority as authority,  a.ports,  a.source\_type as sourceType,  GROUP\_CONCAT(b.NAME) as businessName,  b.business\_type as businessType  FROM  sc\_application a  INNER JOIN sc\_business\_scenario b ON FIND\_IN\_SET(b.id, a.type)  <where>  a.del\_flag = 1  <if test="applicationId != null and applicationId != ''">  and a.application\_id = #{applicationId}  </if>  </where>  group by a.application\_id  【SQL14】  SELECT id,user\_id,user\_name,dept\_id,name,description,type,tag,url,status,del\_flag,create\_time,update\_time,end\_time,destruction\_time,safe\_project\_id,example\_name,safe\_status  FROM sc\_project  WHERE id=#{id}  【SQL15】  SELECT  \*  FROM  sc\_service  WHERE  id = #{id}  【SQL16】  SELECT  \*  FROM  sc\_project\_task  WHERE  id = #{id}  【SQL17】  SELECT  \*  FROM  hb\_survey\_people  WHERE  task\_id = #{taskId}  【SQL18】  SELECT  \*  FROM  sc\_business\_scenario  WHERE  id = #{id}  【SQL19】  SELECT  \*  FROM  sc\_operate\_approve  WHERE  id = #{id}    【SQL20】  SELECT id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,NAME,description,type,  icon,url,points,STATUS,del\_flag,create\_time,update\_time  FROM  sc\_model  WHERE  id = #{id} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_data\_set、sc\_service、sc\_application、sc\_model、sc\_project、sc\_project\_task、sc\_operate\_approve、sc\_project\_process、sc\_sys\_approve**>**  **操作：<修改>** |
| SQL语句：  【SQL21】  update sc\_data\_set  <set>  <if test="vo.name != null and vo.name != ''">  name = #{vo.name},  </if>  <if test="vo.alias != null and vo.alias != ''">  alias = #{vo.alias},  </if>  <if test="vo.type != null and vo.type != ''">  type = #{vo.type},  </if>  <if test="vo.businessId != null and vo.businessId != ''">  business\_id = #{vo.businessId},  </if>  <if test="vo.description != null and vo.description != ''">  description = #{vo.description},  </if>  <if test="vo.tag != null and vo.tag != ''">  tag = #{vo.tag},  </if>  <if test="vo.status != null">  status = #{vo.status},  </if>  <if test="vo.visible != null and vo.visible != ''">  visible = #{vo.visible},  </if>  <if test="vo.businessType != null and vo.businessType != ''">  business\_type = #{vo.businessType},  </if>  <if test="vo.billingMode != null">  billing\_mode = #{vo.billingMode},  </if>  <if test="vo.points != null">  points = #{vo.points},  </if>  <if test="vo.updateTime != null and vo.updateTime != ''">  update\_time = #{vo.updateTime},  </if>  <if test="vo.safeDataSetId != null and vo.safeDataSetId != ''">  safe\_data\_set\_id = #{vo.safeDataSetId},  </if>  <if test="vo.safeStatus != null">  safe\_status = #{vo.safeStatus},  </if>  <if test="vo.safeTestSetId != null and vo.safeTestSetId != ''">  safe\_test\_set\_id = #{vo.safeTestSetId},  </if>  </set>  where id = #{vo.id}  【SQL22】  UPDATE sc\_application  SET user\_id = #{userId},  offer\_user\_id =#{offerUserId},  offer\_user\_name =#{offerUserName},  offer\_dept\_id =#{offerDeptId},  offer\_dept\_name =#{offerDeptName},  NAME =#{name},  description =#{description},  type =#{type},  icon=#{icon},  url=#{url},  cpu=#{cpu},  internal\_memory=#{internalMemory},  disk\_memory=#{diskMemory},  points =#{points},  safety\_level=#{safetyLevel},  STATUS =#{status},  del\_flag =#{delFlag},  create\_time =#{createTime},  update\_time =#{updateTime}  WHERE  application\_id =#{id}  【SQL23】  UPDATE sc\_model  SET user\_id = #{userId},  offer\_user\_id =#{offerUserId},  offer\_user\_name =#{offerUserName},  offer\_dept\_id =#{offerDeptId},  offer\_dept\_name =#{offerDeptName},  NAME =#{name},  description =#{description},  type =#{type},  icon=#{icon},  url=#{url},  points =#{points},  STATUS =#{status},  del\_flag =#{delFlag},  create\_time =#{createTime},  update\_time =#{updateTime}  WHERE  id =#{id}  【SQL24】  UPDATE sc\_service  SET  STATUS =#{status}  WHERE  id =#{id}  【SQL25】  UPDATE sc\_project  SET STATUS =#{status}  WHERE id=#{id}  【SQL26】  UPDATE sc\_project\_task  SET STATUS =#{status},  approve\_status=#{approveStatus}  WHERE id=#{id}  【SQL27】  UPDATE sc\_operate\_approve  SET STATUS =#{status},  update\_time =#{updateTime}  WHERE id=#{id}  【SQL28】  UPDATE sc\_project\_process  SET approve\_status=#{approveStatus}  WHERE id=#{id}  【SQL29】  UPDATE sc\_sys\_approve  SET user\_id=#{user\_id},  user\_dept=#{user\_dept},  user\_dept\_id=#{user\_dept\_id},  user\_name=#{user\_name},  approve\_dept=#{approve\_dept},  approve\_id=#{approve\_id},  approve\_dept\_id=#{approve\_dept\_id},  comment=#{comment},  task\_id=#{task\_id},  task\_name=#{task\_name},  name=#{name},  relevance\_content=#{relevance\_content},  relevance\_id=#{relevance\_id},  relevance\_size=#{relevance\_size},  project\_id=#{project\_id},  type=#{type},  status=#{status},  create\_time=#{create\_time},  update\_time=#{update\_time}  WHERE  id=#{id} |

#### 总体设计方案

##### 设计方案

我的申请模块展示用户所申请流程，可查询用户申请详情与撤销申请。

##### 处理流程

1. 我的申请列表展示：用户可以根据申请名称、审批状态、申请类型、申请时间等条件来检索数据，调用【6.2.1.1.1.1.2.3.我的申请】接口，接口封装查询条件执行【SQL11】查询当前用户已申请可查看的所有审批列表。分别展示目标名称、申请方、申请人、审批状态、创建时间。点击查询开始检索数据，点击重置清空所有条件展示全部数据。  
   
2. 查看我的申请数据详情信息：点击详情按钮，打开弹框展示详细信息，审批详情信息包含审核基本信息与审核流程节点信息。调用【6.2.1.1.1.1.2.5.审批详情-详情信息】接口获取审核基本信息，系统内部执行【SQL12】根据id查询审核流程详情信息。根据审核流程类型进行具体详情内容查询，数据集类型使用【SQL10】查询数据详情，应用类型使用【SQL13】查询应用详情，项目类型使用【SQL14】查询项目详情，服务类型使用【SQL15】查询服务详情，任务类型使用【SQL16】查询任务详情，如果为核保或快赔任务继续执行【SQL17】查询核保人员，业务场景类型执行【SQL18】查询业务场景信息，积分申请类型执行【SQL19】查询申请详情信息，模型类型执行【SQL20】查询模型信息，最后封装为页面使用格式。



调用【6.2.1.1.1.1.2.1.审批详情-审批流程】接口获取审批流程节点信息，系统内部执行【SQL01】根据流程id查询该流程所有审批记录，再根据审批类型进行分类判断获取对应审批节点。

【项目创建审批类型】需分别执行【SQL02】、【SQL03】、【SQL04】、【SQL05】根据项目id获取该项目所关联数据集、应用、服务、模型所属机构id，再执行【SQL06】查询运营端机构id，最后通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。

【任务发起审批类型】与【个人资源审批类型】获取审批发起人所属机构id，通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。

【协作数据申请类型】先执行【SQL10】查询协作数据所属机构id，通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。



1. 撤销：点击撤销按钮，弹出提示语“你确定撤销吗？”，点击取消不执行撤销操作，点击确认调用【6.2.1.1.1.1.2.6.我的申请-撤销申请】接口，撤销当前的申请，接口先执行【SQL29】修改审批流程记录状态，撤销后状态改为“已撤销”，再根据不同类型修改对应类型关联表，数据集发布审批类型执行【SQL21】修改数据集信息表状态为未发布状态，应用上架审批类型执行【SQL22】修改应用信息表状态为未发布状态，模型上架审批类型执行【SQL23】修改模型信息表状态为未发布状态，服务上架审批类型执行【SQL24】修改服务信息表状态为未发布状态，项目创建审批类型执行【SQL25】修改项目信息表状态为未发布状态，任务发起审理类型执行【SQL26】修改任务信息表状态为未发布状态，积分申请审批类型执行【SQL27】修改积分申请信息表状态为已撤销状态，任务结果数据查看类型根据任务类型分类，普通项目的任务数据查看执行【SQL26】修改任务信息表的申请查看状态为已撤销状态，MPC项目的任务数据查看执行【SQL28】修改项目流程表对应流程的申请查询状态为已撤销状态，应用安全审核类型执行【SQL22】修改应用信息表状态为未扫描状态。



### 流程审批

#### 类设计

##### 控制类

###### SysApproveController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysApproveController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成流程审批管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\approveController\SysApproveController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| selectApproveList | session[:taskId] | string | ■公有 □私有 □保护 | Result | 审批详情-审批流程 |
| selectApprovePage | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 审批列表 |
| selectMyApprovePage | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 我的申请 |
| complete | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 流程审批 |
| detail | session[:id] | long | ■公有 □私有 □保护 | Result | 审批详情-详情信息 |
| withdraw | session[:taskId] | string | ■公有 □私有 □保护 | Result | 撤销审批 |
| start | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 发起安全审核 |
| getSafeInfo | session[:id] | long | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-应用信息 |
| startStaticCheck | session[:taskId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-开始扫描 |
| applicationDetail | session[:taskId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-静态安全监测-获取扫描信息 |
| applicationProcess | session[:JSONObject] | JSONObject | ■公有 □私有 □保护 | Result | 应用安全扫描-审批通过 |
| startDynamicInstance | session[:appId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-动态安全监测-启动实例 |
| dynamicInstanceInfo | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-实例状态 |
| dynamicInstancePage | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-动态安全监测-查询信息 |
| manualStartDynamicInstance | session[:appId] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-人工审查-启动实例 |
| stopDynamicInstance | session[:id] | string | ■公有 □私有 □保护 | Result | 应用安全扫描-停止实例 |

算法描述

审批详情-审批流程

接口描述：根据审批流程id查询该审批所属审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 流程实例id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 返回流程 |
| data>>name | [string] | 是 | 节点名称 |
| data>>type | [string] | 是 | 节点审批类型 |
| data>>approveName | [string] | 是 | 审批人 |
| data>>approveId | [string] | 是 | 审批人id |
| data>>comment | [string] | 是 | 审批意见 |
| data>>status | [string] | 是 | 审批状态 |
| data>>approveTime | [number] | 是 | 审批时间 |
| code | [number] | 是 | 状态码 |

审批列表

接口描述：展示用户待审批或已审批的审批流程记录。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 否 | 审批名称 |
| status | [string] | 否 | 流程状态（REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| page | [number] | 是 | 页数 |
| pageSize | [string] | 是 | 页码 |
| startTime | [number] | 否 | 申请开始时间 |
| endTime | [number] | 否 | 申请结束时间 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 |  |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>list>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>list>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总条数 |
| data>>totalNum | [number] | 是 | 总页数 |
| code | [number] | 是 | 状态码 |

我的申请

接口描述：展示用户所申请的审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list/page/my | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 否 | 审批名称 |
| status | [string] | 否 | 流程状态（REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| page | [number] | 是 | 页数 |
| pageSize | [string] | 是 | 页码 |
| startTime | [number] | 否 | 申请开始时间 |
| endTime | [number] | 否 | 申请结束时间 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 |  |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>list>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>list>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总条数 |
| data>>totalNum | [number] | 是 | 总页数 |
| code | [number] | 是 | 状态码 |

审批

接口描述：根据各个登录用户获取该用户所属机构下的角色列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/complete | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 流程id |
| comment | [string] | 是 | 审批意见 |
| status | [string] | 是 | 审批状态（NOT-PASS 未通过/PASS 已通过) |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

审批详情-详情信息

接口描述：根据不同类型查询审批详情信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/detail | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>id | [number] | 是 | 流程id |
| data>>userId | [number] | 是 | 申请用户id |
| data>>userDept | [string] | 是 | 申请用户机构名 |
| data>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>userName | [string] | 是 | 申请用户名 |
| data>>approveRole | [string] | 是 | 审批用户角色 |
| data>>approveDept | [string] | 是 | 审批用户机构 |
| data>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>approveId | [null] | 是 | 审批用户id |
| data>>comment | [null] | 是 | 审批意见 |
| data>>taskId | [string] | 是 | 流程实例id |
| data>>taskName | [string] | 是 | 流程名称 |
| data>>name | [string] | 是 | 流程目标名称 |
| data>>relevanceContent | [null] | 是 | 流程关联详情内容 |
| data>>relevanceId | [null] | 是 | 流程关联详情内容id |
| data>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>projectId | [null] | 是 | 项目id |
| data>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>content | [string] | 是 | 根据不同类型审核流程的对应展示正文 |
| data>>projectName | [string] | 是 | 项目名 |
| code | [number] | 是 | 状态码 |

我的申请-撤销申请

接口描述：根据流程id对申请的审批流程进行撤销回退。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/withdraw | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 流程实例id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| code | [number] | 是 | 状态码 |

发起安全审核

接口描述：生产级应用进行安全审核，调用该接口发起安全审核，创建可信执行容器系统内部应用审核记录。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/start | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceId | [number] | 是 | 关联应用id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 返回信息 |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-应用信息

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时查询应用信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/safe/info | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>icon | [string] | 是 | 图标 |
| data>>description | [string] | 是 | 描述 |
| data>>ports | [string] | 是 | 端口 |
| data>>type | [string] | 是 | 应用类型（web web服务类） |
| data>>staticFileDir | [string] | 是 | 静态文件路径 |
| data>>id | [string] | 是 | 应用id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>usageSteps | [string] | 是 | 使用步骤 |
| data>>swaggerFile | [string] | 是 | swagger文件 |
| data>>outputPaths | [string] | 是 | 输出路径 |
| data>>name | [string] | 是 | 应用名称 |
| data>>outputSamples | [string] | 是 | 输出样例路径 |
| data>>taskId | [string] | 是 | 任务id |
| data>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-开始扫描

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时开始扫描。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/start/check | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 返回信息 |
| code | [number] | 是 | 状态码 |

应用安全扫描-静态安全监测-获取扫描信息

接口描述：可信执行容器系统进行应用安全扫描中静态安全监测时获取扫描信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/application/detail | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [string] | 是 | 静态扫描结果 |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-审批通过

接口描述：可信执行容器系统进行应用安全扫描审批通过。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/approve/application/process | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 审核流程id |
| auditType | [string] | 是 | 通过类型（static静态监测,dynamic动态监测,artificial人工监测,final最后审批） |
| status | [number] | 是 | 默认传2（2审批通过） |
| result | [string] | 是 | 静态扫描结果 |
| opinion | [string] | 是 | 审批意见 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-动态安全监测-启动实例

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，启动监测应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/start/dynamicInstance | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| appId | [string] | 是 | 应用id（应用安全扫描-应用信息获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [string] | 是 | 实例id |
| code | [number] | 是 | 状态码 |

应用安全扫描-实例状态

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，查询启动监测应用实例的状态信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/dynamicInstance/info | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 实例id（启动实例接口获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 实例id |
| data>>instanceStatus | [string] | 是 | 实例状态 未知("Unknown"),已暂停 ("2"),运行中 ("Synced"),已删除 ("1"),等待中 ("Pending"),运行中1 ("containerIsRunning"),启动中 ("containerNotRunning"),创建失败 ("Failed"); |
| code | [number] | 是 | 状态码 |

应用安全扫描-动态安全监测-查询信息

接口描述：可信执行容器系统进行应用安全扫描动态安全监测时，查询应用安全信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/dynamicInstance/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 实例id（启动实例接口获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-人工审查-启动实例

接口描述：可信执行容器系统进行应用安全扫描人工审查监测时，启动应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/manual/start/dynamicInstance | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| appId | [string] | 是 | 应用id（应用安全扫描-应用信息获取） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

应用安全扫描-停止实例

接口描述：可信执行容器系统进行应用安全扫描时，关闭应用实例。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/detail |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>applicantDeptName | [string] | 是 | 申请人部门名称 |
| data>>applicantUserName | [string] | 是 | 申请人姓名 |
| data>>appName | [string] | 是 | 应用名称 |
| data>>claimUserId | [string] | 是 | 被申请人id |
| data>>claimUserName | [string] | 是 | 被申请人姓名 |
| data>>staticResult | [object] | 是 | 静态扫描结果 |
| data>>staticStatus | [string] | 是 | 静态扫描状态（2通过，1审核中） |
| data>>applicantDeptId | [string] | 是 | 申请人部门id |
| data>>appType | [string] | 是 | 应用类型 |
| data>>appId | [string] | 是 | 应用id |
| data>>id | [string] | 是 | 审核流程id |
| data>>applicantUserId | [string] | 是 | 申请人id |
| data>>taskId | [string] | 是 | 任务id |
| data>>createDate | [number] | 是 | 创建时间 |
| data>>status | [number] | 是 | 任务状态 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysApprove

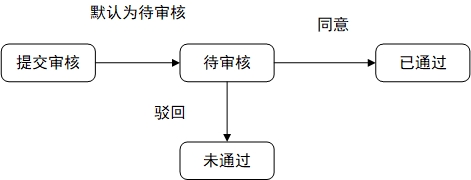
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysApprove | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统流程审批的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\approve\po\SysApprove | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 流程id |
| userId | long | | ■公有 □私有 □保护 | | | NNI | | 申请用户id |
| userDept | string | | ■公有 □私有 □保护 | | |  | | 申请用户机构名 |
| userDeptId | long | | ■公有 □私有 □保护 | | |  | | 申请用户机构id |
| userName | string | | ■公有 □私有 □保护 | | |  | | 申请用户名 |
| approveRole | string | | ■公有 □私有 □保护 | | |  | | 审批用户角色 |
| approveDept | string | | ■公有 □私有 □保护 | | |  | | 审批用户机构 |
| approveDeptId | long | | ■公有 □私有 □保护 | | |  | | 审批用户机构id |
| approveId | long | | ■公有 □私有 □保护 | | |  | | 审批用户id |
| comment | string | | ■公有 □私有 □保护 | | |  | | 审批意见 |
| taskId | string | | ■公有 □私有 □保护 | | |  | | 流程实例id |
| taskName | string | | ■公有 □私有 □保护 | | |  | | 流程名称 |
| name | string | | ■公有 □私有 □保护 | | |  | | 流程目标名称 |
| relevanceContent | string | | ■公有 □私有 □保护 | | |  | | 流程关联详情内容 |
| relevanceId | long | | ■公有 □私有 □保护 | | |  | | 流程关联详情内容id |
| relevanceSize | string | | ■公有 □私有 □保护 | | |  | | 流程关联额度 |
| projectId | long | | ■公有 □私有 □保护 | | |  | | 项目id |
| type | string | | ■公有 □私有 □保护 | | |  | | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| status | string | | ■公有 □私有 □保护 | | |  | | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| createTime | long | | ■公有 □私有 □保护 | | | NNI | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | | NNI | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_approve | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_approve、sc\_project、sc\_application、sc\_data\_set、sc\_model、sc\_sys\_flow、sc\_project\_application、sc\_project\_data、sc\_project\_model、sc\_project\_service、sc\_sys\_flow\_detail、sc\_sys\_user、sc\_project\_task、sc\_project\_process、sc\_service、sc\_data\_set\_log、sc\_business\_scenario、sc\_application\_place、sc\_service\_place、sc\_resource、sc\_sys\_dict\_data、sc\_resource\_project、sc\_operate\_indent、sc\_operate\_approve、sc\_model\_place、sc\_data\_set\_place**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  \*  FROM  sc\_sys\_approve  WHERE  task\_id = #{taskId}  ORDER BY  a.create\_time DESC  【SQL02】  SELECT  d.id,  d.offer\_dept\_id as offerDeptId,  d.offer\_user\_id as offerUserId,  d.user\_id as userId,  d.offer\_user\_name AS offerUserName,  d.NAME,  d.alias,  d.create\_time AS createTime,  d.update\_time as updateTime,  p.project\_id as projectId,  p.task\_id as taskId,  d.source\_type as sourceType,  d.safe\_data\_set\_id as safeDataSetId,  d.safe\_status as safeStatus,  d.safe\_test\_set\_id as safeTestSetId,  d.status as status,  d.points,  d.billing\_mode AS billingMode,  d.visible  FROM sc\_project\_data p  left JOIN sc\_data\_set d on p.data\_id=d.id  <where>  <if test="id != null and id != '' ">  and p.project\_id = #{id}  </if>  <if test="taskId != null and taskId != '' ">  and p.task\_id = #{taskId}  </if>  <if test="userId != null and userId != '' ">  and d.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and d.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL03】  SELECT  a.application\_id AS applicationId,  a.user\_id AS userId,  a.offer\_user\_id AS offerUserId,  a.offer\_user\_name AS offerUserName,  a.offer\_dept\_id AS offerDeptId,  a.offer\_dept\_name AS offerDeptName,  a.NAME AS applicationName,  a.description AS description,  a.type AS type,  a.icon AS icon,  a.url AS url,  a.cpu AS cpu,  a.internal\_memory AS internalMemory,  a.disk\_memory AS diskMemory,  a.points AS points,  a.safe\_application\_id as safeApplicationId,  a.safe\_application\_test\_id as safeApplicationTestId,  a.safety\_level AS safetyLevel,  a.STATUS AS STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  a.authority as authority,  a.source\_type as sourceType,  a.ports,  a.points,  a.billing\_mode AS billingMode,  a.authority as authority,  p.project\_id as projectId,  p.task\_id as taskId  FROM  sc\_project\_application p  left JOIN sc\_application a ON p.application\_id = a.application\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  <if test="userId != null and userId != '' ">  and a.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and a.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL04】  SELECT  a.id AS id,  a.point,  a.provider\_id AS providerId,  a.provider\_dept\_id AS providerDeptId,  a.billing\_mode AS billingMode,  a.NAME  FROM  sc\_service a  INNER JOIN sc\_project\_service p ON a.id = p.service\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  </where>  【SQL05】  SELECT  a.id AS id,  a.user\_id AS userId,  a.offer\_user\_id AS offerUserId,  a.offer\_user\_name AS offerUserName,  a.offer\_dept\_id AS offerDeptId,  a.offer\_dept\_name AS offerDeptName,  a.NAME,  a.description AS description,  a.type AS type,  a.icon AS icon,  a.url AS url,  a.points AS points,  a.STATUS AS STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  p.project\_id as projectId,  p.task\_id as taskId  FROM  sc\_model a  INNER JOIN sc\_project\_model p ON a.id = p.model\_id  <where>  <if test="id != null and id != '' ">  p.project\_id = #{id}  </if>  <if test="userId != null and userId != '' ">  and a.user\_id != #{userId}  </if>  <if test="deptId != null and deptId != '' ">  and a.offer\_dept\_id != #{deptId}  </if>  </where>  【SQL06】  SELECT  \*  FROM  sc\_sys\_dept  WHERE  type = "OPERATION-SIDE"  【SQL07】  SELECT  \*  FROM  sc\_sys\_flow  WHERE  type = #{type}  and dept\_id in (#{deptId})  【SQL08】  SELECT  \*  FROM  sc\_sys\_flow\_detail  WHERE  flow\_id = #{flowId}  【SQL09】  select user\_id, login\_name, user\_name from sc\_sys\_user where user\_id in  <foreach collection="ids" item="userId" open="(" separator="," close=")">  #{userId}  </foreach>  【SQL10】  SELECT id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,name,business\_id,alias,description,points,tag,type,privacy\_inquiry,visible,source\_type,update\_strategy,status,secret\_key,create\_time,update\_time  FROM  sc\_data\_set  WHERE  id = #{dataSetId}  【SQL11】  SELECT  a.id,  a.user\_id AS userId,  a.user\_dept AS userDept,  a.user\_name AS userName,  a.approve\_role AS approveRole,  a.approve\_dept AS approveDept,  a.approve\_dept\_id AS approveDeptId,  a.approve\_id AS approveId,  a.COMMENT,  a.task\_id AS taskId,  a.task\_name AS taskName,  a.NAME,  a.relevance\_content AS relevanceContent,  a.relevance\_id AS relevanceId,  a.relevance\_size AS relevanceSize,  a.project\_id AS projectId,  a.type,  a.STATUS,  a.create\_time AS createTime,  a.update\_time AS updateTime,  p.NAME AS projectName  FROM  sc\_sys\_approve a  LEFT JOIN sc\_project p ON p.id = a.project\_id  < WHERE >  < IF test = "userId != null" >  AND a.user\_id = #{userId}  </ IF >  < IF test = "approveDeptId != null" >  AND a.approve\_dept\_id = #{approveDeptId}  </ IF >  < IF test = "type != null and type != ''" >  AND a.type = #{type}  </ IF >  < IF test = "userName != null and userName != ''" >  AND a.user\_name = #{userName}  </ IF >  < IF test = "name != null and name != ''" >  AND a.NAME = #{name}  </ IF >  < IF test = "status != null and status != ''" >  AND a.STATUS = #{status}  </ IF >  </ WHERE >  ORDER BY  a.create\_time DESC  【SQL12】  SELECT id,user\_id,user\_dept,user\_dept\_id,user\_name,approve\_role,approve\_dept,approve\_id,approve\_dept\_id,comment,task\_id,task\_name,name,relevance\_content,relevance\_id,relevance\_size,project\_id,type,status,create\_time,update\_time,approve\_description  FROM sc\_sys\_approve  WHERE id=#{id}  【SQL13】  SELECT  a.application\_id as applicationId,  a.user\_id as userId,  a.offer\_user\_id as offerUserId,  a.offer\_user\_name as offerUserName,  a.offer\_dept\_id as offerDeptId,  a.offer\_dept\_name as offerDeptName,  a.NAME as applicationName ,  a.description as description,  a.type as type,  a.icon as icon,  a.url as url,  a.dept\_type as deptType,  a.cpu as cpu,  a.file\_url as fileUrl,  a.application\_type as applicationType,  a.internal\_memory as internalMemory,  a.disk\_memory as diskMemory,  a.points as points,  a.billing\_mode AS billingMode,  a.safety\_level as safetyLevel,  a.STATUS as status,  a.safe\_application\_id as safeApplicationId,  a.safe\_application\_test\_id as safeApplicationTestId,  a.create\_time as createTime,  a.update\_time as updateTime,  a.authority as authority,  a.ports,  a.source\_type as sourceType,  GROUP\_CONCAT(b.NAME) as businessName,  b.business\_type as businessType  FROM  sc\_application a  INNER JOIN sc\_business\_scenario b ON FIND\_IN\_SET(b.id, a.type)  <where>  a.del\_flag = 1  <if test="applicationId != null and applicationId != ''">  and a.application\_id = #{applicationId}  </if>  </where>  group by a.application\_id  【SQL14】  SELECT id,user\_id,user\_name,dept\_id,name,description,type,tag,url,status,del\_flag,create\_time,update\_time,end\_time,destruction\_time,safe\_project\_id,example\_name,safe\_status  FROM sc\_project  WHERE id=#{id}  【SQL15】  SELECT  \*  FROM  sc\_service  WHERE  id = #{id}  【SQL16】  SELECT  \*  FROM  sc\_project\_task  WHERE  id = #{id}  【SQL17】  SELECT  \*  FROM  hb\_survey\_people  WHERE  task\_id = #{taskId}  【SQL18】  SELECT  \*  FROM  sc\_business\_scenario  WHERE  id = #{id}  【SQL19】  SELECT  \*  FROM  sc\_operate\_approve  WHERE  id = #{id}    【SQL20】  SELECT id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,NAME,description,type,  icon,url,points,STATUS,del\_flag,create\_time,update\_time  FROM  sc\_model  WHERE  id = #{id} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_approve、sc\_data\_set\_log、sc\_data\_set\_place 、sc\_application\_place、sc\_model\_place、 sc\_service\_place、sc\_operate\_indent、sc\_operate\_check、sc\_project\_log **>**  **操作：<插入>** |
| SQL语句：  【SQL21】  insert into sc\_sys\_approve (user\_id, user\_dept\_id, user\_dept, user\_name, approve\_role, approve\_dept\_id, approve\_id, approve\_description, comment, task\_id,task\_name, relevance\_content, relevance\_id, relevance\_size, project\_id,flow\_id, flow\_detail\_id, type, mode, status, create\_time, update\_time) values (#{userId}, #{userDeptId}, #{userDept}, #{userName}, #{approveRole},#{approveDeptId}, #{approveId}, #{approveDescription}, #{comment}, #{taskId}, #{taskName}, #{relevanceContent}, #{relevanceId}, #{relevanceSize}, #{projectId}, #{flowId}, #{flowDetailId}, #{type}, #{mode}, #{status}, #{createTime}, #{updateTime});  【SQL22】  INSERT INTO sc\_data\_set\_place (data\_set\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,NAME,alias,  description,points,tag,type,privacy\_inquiry,visible,source\_type,STATUS,create\_time,update\_time  )  VALUES  (#{dataSetId},#{offerUserId},#{offerUserName},#{offerDeptId},#{offerDeptName},  #{name},#{alias},#{description},#{points},#{tag},#{type},#{privacyInquiry},  #{visible},#{sourceType},#{status},#{createTime},#{updateTime})  【SQL23】  INSERT INTO sc\_application\_place (  application\_id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,NAME,description,type,icon,url,cpu,internal\_memory,disk\_memory,points,safety\_level,STATUS,del\_flag,create\_time,update\_time )  VALUES(  #{applicationId},#{userId},#{offerUserId},#{offerUserName},#{offerDeptId},#{offerDeptName},#{name},#{description},#{type},#{icon},#{url},#{cpu},#{internalMemory},  #{diskMemory},#{points},#{safetyLevel},#{status},#{delFlag},#{createTime},#{updateTime})  【SQL24】  INSERT INTO sc\_model\_place (  model\_id,user\_id,offer\_user\_id,offer\_user\_name,offer\_dept\_id,offer\_dept\_name,NAME,description,type,icon,url,points,STATUS,del\_flag,create\_time,update\_time )  VALUES(  #{modelId},#{userId},#{offerUserId},#{offerUserName},#{offerDeptId},#{offerDeptName},#{name},#{description},#{type},#{icon},#{url},#{points},#{status},#{delFlag},#{createTime},#{updateTime})  【SQL25】  INSERT INTO sc\_service\_place(  id,user\_id,dept\_id,provider\_id,provider\_dept\_id,NAME,type,url,points,billing\_mode,tag,headers,params,body,body\_type,response\_parameters,description,STATUS,create\_time,update\_time )  VALUES(  #{id},#{userId},#{deptId},#{providerId},#{providerDeptId},#{name},#{type},#{url},#{points},#{billingMode},#{tag},#{headers},#{params},#{body},#{bodyType},#{responseParameters},#{description},#{status},#{createTime},#{updateTime})  【SQL26】  INSERT INTO sc\_operate\_indent ( id, indent\_id, procure\_dept\_id, procure\_user\_id, sale\_dept\_id, sale\_user\_id, name, type,project\_id,task\_id,unit\_price,amount, status,fraction,content, create\_time, update\_time ) VALUES (#{id},#{indentId},#{procureDeptId},#{procureUserId},#{saleDeptId},#{saleUserId},#{name},#{type},#{projectId},#{taskId},#{unitPrice},#{amount},#{status},#{fraction},#{content},#{createTime},#{updateTime})  【SQL27】  INSERT INTO sc\_operate\_check ( id, indent\_id, procure\_dept\_id, procure\_user\_id, custom\_dept\_id, custom\_user\_id, type,check\_id,charge\_type,amount, status, create\_time, update\_time ) VALUES (#{id},#{indentId},#{procureDeptId},#{procureUserId},#{customDeptId},#{customUserId},#{type},#{check\_id},#{chargeType},#{amount},#{status},#{createTime},#{updateTime})  【SQL28】  INSERT INTO sc\_project\_log(  id,user\_id,dept\_id,project\_id,user\_name,NAME,cpu,internal\_memory,disk\_memory,start\_time,end\_time,des\_time,des\_times)  VALUES(  #{id},#{userId},#{deptId},#{projectId},#{userName},#{name},#{cpu},#{internalMemory},#{diskMemory},#{startTime},#{endTime},#{desTime},#{desTimes}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 流程id |  | | userId | 申请用户id |  | | userDept | 申请用户机构名 |  | | userDeptId | 申请用户机构id |  | | userName | 申请用户名 |  | | approveRole | 审批用户角色 |  | | approveDept | 审批用户机构 |  | | approveDeptId | 审批用户机构id |  | | approveId | 审批用户id |  | | comment | 审批意见 |  | | taskId | 流程实例id |  | | taskName | 流程名称 |  | | name | 流程目标名称 |  | | relevanceContent | 流程关联详情内容 |  | | relevanceId | 流程关联详情内容id |  | | relevanceSize | 流程关联额度 |  | | projectId | 项目id |  | | type | 流程类型 | 应用上架 APP-LISTING，应用申请 APP-APPLY，  数据集发DATA-PUBLISH，数据集申请DATA-APPLY，模型上MODEL-LISTING，模型申请MODEL-APPLY，数据使用申请DATA-USE-APPLY，  机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY | | status | 流程状态 | REVIEW 待审核NOT-PASS 未通过  PASS 已通过 | | createTime | 创建时间 |  | | updateTime | 更新时间 |  | | content | 根据不同类型审核流程的对应展示正文 |  | | projectName | 项目名 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_data\_set、sc\_service、sc\_application、sc\_model、sc\_project、sc\_project\_task、sc\_operate\_approve、sc\_project\_process、sc\_sys\_approve**>**  **操作：<修改>** |
| SQL语句：  【SQL29】  update sc\_data\_set  <set>  <if test="vo.name != null and vo.name != ''">  name = #{vo.name},  </if>  <if test="vo.alias != null and vo.alias != ''">  alias = #{vo.alias},  </if>  <if test="vo.type != null and vo.type != ''">  type = #{vo.type},  </if>  <if test="vo.businessId != null and vo.businessId != ''">  business\_id = #{vo.businessId},  </if>  <if test="vo.description != null and vo.description != ''">  description = #{vo.description},  </if>  <if test="vo.tag != null and vo.tag != ''">  tag = #{vo.tag},  </if>  <if test="vo.status != null">  status = #{vo.status},  </if>  <if test="vo.visible != null and vo.visible != ''">  visible = #{vo.visible},  </if>  <if test="vo.businessType != null and vo.businessType != ''">  business\_type = #{vo.businessType},  </if>  <if test="vo.billingMode != null">  billing\_mode = #{vo.billingMode},  </if>  <if test="vo.points != null">  points = #{vo.points},  </if>  <if test="vo.updateTime != null and vo.updateTime != ''">  update\_time = #{vo.updateTime},  </if>  <if test="vo.safeDataSetId != null and vo.safeDataSetId != ''">  safe\_data\_set\_id = #{vo.safeDataSetId},  </if>  <if test="vo.safeStatus != null">  safe\_status = #{vo.safeStatus},  </if>  <if test="vo.safeTestSetId != null and vo.safeTestSetId != ''">  safe\_test\_set\_id = #{vo.safeTestSetId},  </if>  </set>  where id = #{vo.id}  【SQL30】  update sc\_data\_set\_log  <trim prefix="SET" suffixOverrides=",">  <if test="vo.dataStatus != null and vo.dataStatus != ''">  data\_status = #{vo.dataStatus},  </if>  <if test="vo.accreditUserId != null">  accredit\_user\_id = #{vo.accreditUserId},  </if>  <if test="vo.accreditUserName != null and vo.accreditUserName != ''">  accredit\_user\_name = #{vo.accreditUserName},  </if>  </trim>  <where>  data\_status = 'REVIEW'  <if test="vo.offerUserId != null">  and offer\_user\_id = #{vo.offerUserId}  </if>  <if test="vo.dataSetId != null">  and data\_set\_id = #{vo.dataSetId}  </if>  <if test="vo.type != null and vo.type != ''">  and type = #{vo.type}  </if>  </where>  【SQL31】  UPDATE sc\_model  SET user\_id = #{userId},  offer\_user\_id =#{offerUserId},  offer\_user\_name =#{offerUserName},  offer\_dept\_id =#{offerDeptId},  offer\_dept\_name =#{offerDeptName},  NAME =#{name},  description =#{description},  type =#{type},  icon=#{icon},  url=#{url},  points =#{points},  STATUS =#{status},  del\_flag =#{delFlag},  create\_time =#{createTime},  update\_time =#{updateTime}  WHERE  id =#{id}  【SQL32】  UPDATE sc\_service  SET  STATUS =#{status}  WHERE  id =#{id}  【SQL33】  UPDATE sc\_project  SET STATUS =#{status}  WHERE id=#{id}  【SQL34】  UPDATE sc\_project\_task  SET STATUS =#{status},  approve\_status=#{approveStatus}  WHERE id=#{id}  【SQL35】  UPDATE sc\_operate\_approve  SET STATUS =#{status},  update\_time =#{updateTime}  WHERE id=#{id}  【SQL36】  UPDATE sc\_project\_process  SET approve\_status=#{approveStatus}  WHERE id=#{id}  【SQL37】  UPDATE sc\_sys\_approve  SET user\_id=#{user\_id},  user\_dept=#{user\_dept},  user\_dept\_id=#{user\_dept\_id},  user\_name=#{user\_name},  approve\_dept=#{approve\_dept},  approve\_id=#{approve\_id},  approve\_dept\_id=#{approve\_dept\_id},  comment=#{comment},  task\_id=#{task\_id},  task\_name=#{task\_name},  name=#{name},  relevance\_content=#{relevance\_content},  relevance\_id=#{relevance\_id},  relevance\_size=#{relevance\_size},  project\_id=#{project\_id},  type=#{type},  status=#{status},  create\_time=#{create\_time},  update\_time=#{update\_time}  WHERE  id=#{id}  【SQL38】  UPDATE sc\_project\_data  SET project\_id = #{projectId},  data\_id =#{dataId},  dept\_id =#{deptId},  safe\_data\_id =#{safeDataId},  safe\_data\_name =#{safeDataName},  start\_time=#{startTime},  end\_time=#{endTime},  approve\_status=#{approveStatus}  WHERE  task\_id =#{taskId}  【SQL39】  UPDATE sc\_application  SET user\_id = #{userId},  offer\_user\_id =#{offerUserId},  offer\_user\_name =#{offerUserName},  offer\_dept\_id =#{offerDeptId},  offer\_dept\_name =#{offerDeptName},  NAME =#{name},  description =#{description},  type =#{type},  icon=#{icon},  url=#{url},  cpu=#{cpu},  internal\_memory=#{internalMemory},  disk\_memory=#{diskMemory},  points =#{points},  safety\_level=#{safetyLevel},  STATUS =#{status},  del\_flag =#{delFlag},  create\_time =#{createTime},  update\_time =#{updateTime}  WHERE  application\_id =#{id} |

#### 总体设计方案

##### 设计方案

审批流程展示用户待审批的审批流程，用户可以进行审批与查看详情操作。

##### 处理流程

1. 流程审批列表展示：用户可以根据审批名称、审批状态、审批类型、申请时间等条件来检索数据，调用【6.2.2.1.1.1.2.1.审批详情-审批流程】接口，接口封装查询条件执行【SQL11】查询当前用户待审批可查看的所有审批列表。分别展示目标名称、申请方、申请人、审批状态、创建时间。点击查询开始检索数据，点击重置清空所有条件展示全部数据。点击查询开始检索数据，点击重置清空所有条件展示全部数据。  
   
2. 查看审批流程数据详情信息：点击详情按钮，打开弹框展示详细信息，详情信息包含审批基础信息与流程节点。调用【6.2.2.1.1.1.2.5.审批详情-详情信息】接口获取审核基本信息，系统内部执行【SQL12】根据id查询审核流程详情信息。根据审核流程类型进行具体详情内容查询，数据集类型使用【SQL10】查询数据详情，应用类型使用【SQL13】查询应用详情，项目类型使用【SQL14】查询项目详情，服务类型使用【SQL15】查询服务详情，任务类型使用【SQL16】查询任务详情，如果为核保或快赔任务继续执行【SQL17】查询核保人员，业务场景类型执行【SQL18】查询业务场景信息，积分申请类型执行【SQL19】查询申请详情信息，模型类型执行【SQL20】查询模型信息，最后封装为页面使用格式。  
   

调用【6.2.2.1.1.1.2.1.审批详情-审批流程】接口获取审批流程节点信息，系统内部执行【SQL01】根据流程id查询该流程所有审批记录，再根据审批类型进行分类判断获取对应审批节点。

【项目创建审批类型】需分别执行【SQL02】、【SQL03】、【SQL04】、【SQL05】根据项目id获取该项目所关联数据集、应用、服务、模型所属机构id，再执行【SQL06】查询运营端机构id，最后通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。

【任务发起审批类型】与【个人资源审批类型】获取审批发起人所属机构id，通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。

【协作数据申请类型】先执行【SQL10】查询协作数据所属机构id，通过【SQL07】与【SQL08】查询所需审批的所有流程节点列表，再根据审批节点用户id集合执行【SQL09】查询用户信息，将流程节点与审批记录相互结合，封装后返回前端展示审批流程节点信息。



1. 审批：点击审批按钮调用【6.2.2.1.1.1.2.5.审批详情-详情信息】接口，查看申请的基本信息，调用【6.2.2.1.1.1.2.1.审批详情-审批流程】接口，查看当前审批的进度。输入审批意见后点击驳回或者同意按钮执行【SQL21】来完成审批操作，当前节点审批完成后查看是否存在下一个审批节点，如果所有审批节点修改完成，根据审批结果进行最后的审批业务逻辑修改。  
   

根据流程类型进行不同类型业务流程逻辑

1. 数据集发布审批，首先执行【SQL29】修改数据集状态。若审批结果为通过，修改该数据集日志信息表日志状态【SQL30】，将该数据集发布至数据集市场，在数据集市场信息表插入信息【SQL22】，最后请求可信执行容器系统接口创建数据集。
2. 应用安全审批，执行【SQL39】修改应用状态。若审批结果为通过，则改为未发布状态，若审批结果为未通过，则改为扫描失败状态。
3. 应用上架审批，首先执行【SQL39】修改应用状态。查询当前应用信息，若审批结果为通过，将该应用上架至应用市场【SQL23】，在应用市场信息表插入信息。
4. 服务上架审批，首先执行【SQL32】修改服务状态。若审批结果为通过，查询当前服务信息，在服务市场信息表插入信息【SQL25】。
5. 模型上架审批，首先执行【SQL31】修改模型状态。若审批结果为通过，将该模型发布至模型市场，在模型市场信息表插入信息【SQL24】。
6. 项目创建审批，项目审批根据类型进行逻辑处理。若审批结果为通过，可信容器类型项目会在容器系统内部创建对应的项目，TEE类型项目会执行【SQL28】保存对应的项目启动日志，MPC类型项目需将添加项目到隐私平台，最后修改项目信息。
7. 机构资源申请审批，若审批结果为通过，查询数据字典表配置系统总资源信息，并查询当前已分配的机构资源总数，相减获取可分配资源总数，若不满足分配资源数不可审批通过，若满足则给该机构新增资源，修改流程节点审批状态。最后执行【SQL26】与【SQL27】创建资源的订单与账单信息。
8. 个人资源申请审批，若审批结果为通过，查询机构的总资源信息，并查询当前已分配的机构资源总数，相减获取可分配资源总数，若不满足分配资源数不可审批通过，若满足则给该机构新增资源，修改流程节点审批状态。
9. 任务发起审理，若审批结果为通过，若是TEE项目的任务会启动tee任务，最后执行【SQL34】修改关联任务状态。
10. 积分申请审批，若审批结果为通过，将积分增加给当前用户机构。并执行【SQL35】修改积分申请信息的状态。
11. 任务结果数据查看审批，根据任务类型分类，普通项目的任务数据查看执行【SQL34】修改任务信息表的申请查看状态，MPC项目的任务数据查看执行【SQL36】修改项目流程表对应流程的申请查询状态。
12. 协作数据申请审批，首先执行【SQL38】修改项目关联的数据集申请状态。若审批通过，则将数据集推送隐私服务，最后执行【SQL33】修改项目信息。  
    

### 流程配置

#### 类设计

##### 控制类

###### SysFlowController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysFlowController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成审批流程节点管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\approveController\SysFlowController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| selectPage | session[:SysFlowVo] | SysFlowVo | ■公有 □私有 □保护 | Result | 分页查询流程配置 |
| addFlow | session[:SysFlowVo] | SysFlowVo | ■公有 □私有 □保护 | Result | 新增流程配置 |
| editFlow | session[:SysFlowVo] | SysFlowVo | ■公有 □私有 □保护 | Result | 修改流程配置 |
| remove | session[:id] | long | ■公有 □私有 □保护 | Result | 删除流程配置 |
| getTypeList |  |  | ■公有 □私有 □保护 | Result | 查询流程类型 |

算法描述

分页查询流程配置

接口描述：根据不同查询条件查询审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /sys/flow/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 否 | 流程名称 |
| type | [string] | 否 | 流程类型 |
| page | [number] | 是 | 页数 |
| pageSize | [number] | 是 | 页码 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>currentPage | [number] | 是 | 页数 |
| data>>size | [number] | 是 | 页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总条数 |
| data>>list | [array] | 是 | 返回集合 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 创建用户id |
| data>>list>>deptId | [number] | 是 | 创建机构id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>type | [string] | 是 | 流程类型 |
| data>>list>>sort | [number] | 是 | 流程排序 |
| data>>list>>content | [string] | 是 | 流程节点详情 |
| data>>list>>description | [null] | 是 | 流程说明 |
| data>>list>>sysFlowDetailVoList | [array] | 是 | 流程节点集合 |
| data>>list>>sysFlowDetailVoList>>tagId | [number] | 是 | 节点id |
| data>>list>>sysFlowDetailVoList>>roleId | [array] | 是 | 角色id集合 |
| data>>list>>sysFlowDetailVoList>>userId | [array] | 是 | 用户id集合 |
| data>>list>>sysFlowDetailVoList>>name | [string] | 是 | 节点名称 |
| data>>list>>sysFlowDetailVoList>>type | [string] | 是 | 节点类型 |
| data>>list>>createTime | [number] | 是 | 流程创建时间 |
| data>>list>>updateTime | [number] | 是 | 流程修改时间 |
| code | [number] | 是 | 状态码 |

新增流程配置

接口描述：新增流程节点配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /sys/flow/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 是 | 流程名称 |
| type | [string] | 是 | 流程类型 数据集发布审批("DATASET-APPROVE"),应用上架审批("APP-APPROVE"),服务上架审批("SERVICE-APPROVE"),机构资源审批("DEPT-RESOURCES-APPROVE"),个人资源审批("USER-RESOURCES-APPROVE"),任务发起审批("TASK-APPROVE"),项目创建审批("PROJECT-APPROVE"); |
| sysFlowDetailVoList | [array] | 是 | 节点集合 |
| sysFlowDetailVoList>>tagId | [number] | 是 | 节点id |
| sysFlowDetailVoList>>name | [string] | 是 | 节点名称 |
| sysFlowDetailVoList>>roleId | [array] | 是 | 角色id集合 |
| sysFlowDetailVoList>>userId | [array] | 是 | 用户id集合 |
| sysFlowDetailVoList>>type | [string] | 是 | 节点类型(会签 COUNTERSIGN,或签 OR-VISA,抄送 COPY-TO) |
| description | [string] | 是 | 说明 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 新增条数 |
| code | [number] | 是 | 状态码 |

修改流程配置

接口描述：修改流程节点配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /sys/flow/edit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskName | [string] | 是 | 流程名称 |
| type | [string] | 是 | 流程类型 数据集发布审批("DATASET-APPROVE"),应用上架审批("APP-APPROVE"),服务上架审批("SERVICE-APPROVE"),机构资源审批("DEPT-RESOURCES-APPROVE"),个人资源审批("USER-RESOURCES-APPROVE"),任务发起审批("TASK-APPROVE"),项目创建审批("PROJECT-APPROVE"); |
| sysFlowDetailVoList | [array] | 是 | 节点集合 |
| sysFlowDetailVoList>>tagId | [number] | 是 | 节点id |
| sysFlowDetailVoList>>name | [string] | 是 | 节点名称 |
| sysFlowDetailVoList>>roleId | [array] | 是 | 角色id集合 |
| sysFlowDetailVoList>>userId | [array] | 是 | 用户id集合 |
| sysFlowDetailVoList>>type | [string] | 是 | 节点类型(会签 COUNTERSIGN,或签 OR-VISA,抄送 COPY-TO) |
| description | [string] | 是 | 说明 |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

删除流程配置

接口描述：删除流程节点配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /sys/flow/remove/{id} | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 删除条数 |
| code | [number] | 是 | 状态码 |

查询流程类型

接口描述：根据用户所属机构类型查询对应流程类型。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /sys/flow/type/list | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回流程 |
| data>>name | [string] | 是 | 流程类型名称 |
| data>>type | [string] | 是 | 流程类型 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysFlow

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysFlow | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统审批流程节点的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\approve\po\SysFlow | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 流程id |
| userId | long | | ■公有 □私有 □保护 | | | NNI | | 用户id |
| deptId | long | | ■公有 □私有 □保护 | | |  | | 用户机构id |
| taskName | string | | ■公有 □私有 □保护 | | |  | | 流程名称 |
| type | string | | ■公有 □私有 □保护 | | |  | | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| sort | integer | | ■公有 □私有 □保护 | | |  | | 流程顺序 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| content | string | | ■公有 □私有 □保护 | | |  | | 节点内容 |
| createTime | long | | ■公有 □私有 □保护 | | | NNI | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | | NNI | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_flow | | |

算法描述

无

状态转换描述

无

#### SQL文一览

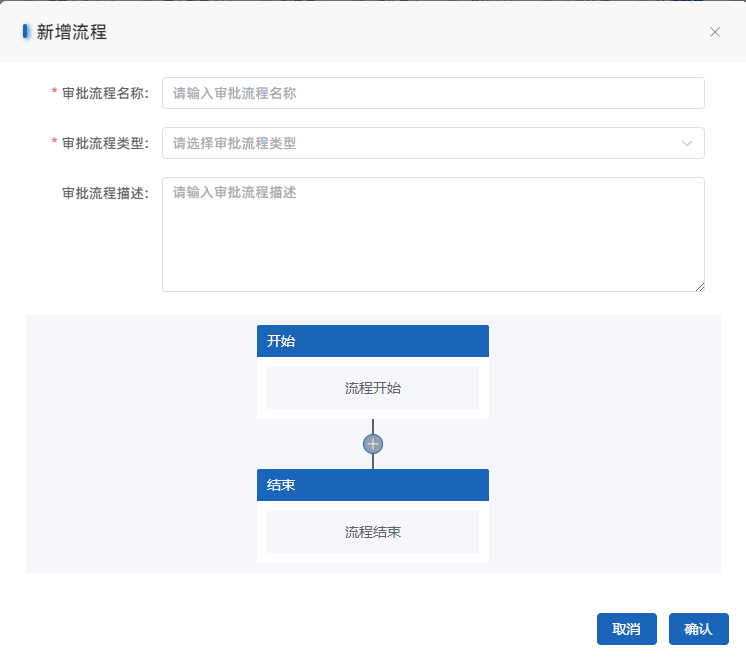
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_flow**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  select  id, user\_id, dept\_id, task\_name, type, sort, content, description, create\_time, update\_time  from  sc\_sys\_flow  <where>  <if test="deptId != null">  and dept\_id = #{deptId}  </if>  <if test="type != null and type != ''">  and type = #{type}  </if>  <if test="taskName != null and taskName != ''">  and task\_name like concat('%', #{taskName}, '%')  </if>  </where> |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_flow、sc\_sys\_flow\_detail **>**  **操作：<插入>** |
| SQL语句：  【SQL02】  insert into sc\_sys\_flow (id,user\_id, dept\_id, task\_name, type, sort, description, content, create\_time, update\_time) values (#{id},#{userId}, #{deptId},#{taskName}, #{type}, #{sort}, #{description}, #{content}, #{createTime}, #{updateTime})  【SQL03】  INSERT INTO sc\_sys\_flow\_detail(id,user\_id, dept\_id, approve\_role, approve\_user\_id, approve\_dept\_id, flow\_id, name,type, create\_time, update\_time) values (#{id},#{userId}, #{deptId},#{approveRole}, #{approveUserId}, #{approveDeptId}, #{flowId}, #{name}, #{type}, #{createTime}, #{updateTime}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 创建用户id |  | | deptId | 创建机构id |  | | taskName | 流程名称 |  | | type | 流程类型 |  | | sort | 流程排序 |  | | content | 流程节点详情 |  | | description | 流程说明 |  | | createTime | 创建时间 |  | | updateTime | 更新时间 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_flow、sc\_sys\_flow\_detail**>**  **操作：<修改>** |
| SQL语句：  【SQL04】  UPDATE sc\_sys\_flow  SET user\_id = #{userId},  dept\_id =#{deptId},  task\_name =#{taskName},  type =#{type},  sort =#{sort},  description=#{description},  content =#{content}  WHERE  id =#{id} |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | userId | 创建用户id | 创建用户id |  | | deptId | 创建机构id | 创建机构id |  | | taskName | 流程名称 | 流程名称 |  | | type | 流程类型 | 流程类型 |  | | sort | 流程排序 | 流程排序 |  | | content | 流程节点详情 | 流程节点详情 |  | | description | 流程说明 | 流程说明 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_flow、sc\_sys\_flow\_detail**>**  **操作：<删除>** |
| SQL语句：  【SQL05】  delete from sc\_sys\_flow  WHERE  id =#{id}  【SQL06】  delete from sc\_sys\_flow\_detail  WHERE  flow\_id =#{flowId} |

#### 总体设计方案

##### 设计方案

流程配置是对当前用户所属机构的审理流程及流程节点进行配置。

##### 处理流程

1. 流程配置列表展示：用户可以根据流程名称、审批流程类型等条件来检索数据，调用【6.2.3.1.1.1.2.1.分页查询流程配置】接口，接口执行【SQL01】根据条件查询审批流程列表。点击查询开始检索数据，点击重置清空所有条件展示全部数据。  
   
2. 新增流程：点击新增按钮打开弹框填写弹框信息，配置流程节点数据后，点击确认按钮，调用【6.2.3.1.1.1.2.2.新增流程配置】接口，执行【SQL02】插入审批流程信息，再执行【SQL03】插入流程节点信息，完成新增流程操作。  
   
3. 编辑流程：点击编辑按钮打开编辑弹框，在列表带入数据，展示流程数据，用户修改数据后点击确认按钮，调用【6.2.3.1.1.1.2.3.修改流程配置】接口，执行【SQL04】修改流程配置信息，再执行【SQL06】删除原流程节点信息，再执行【SQL03】插入新的流程节点信息，关闭弹框编辑流程数据完成。

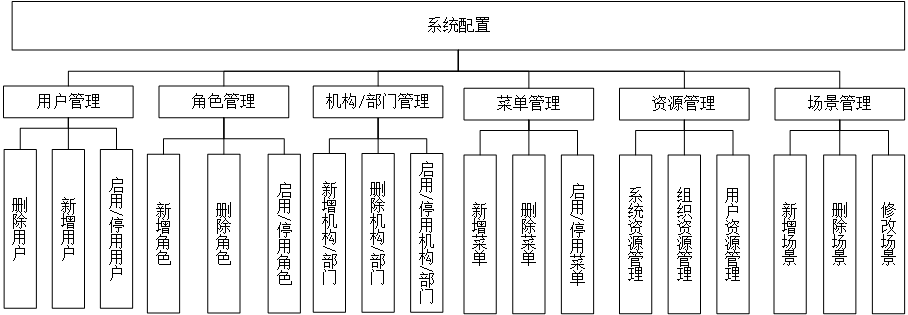


1. 删除流程：点击删除按钮，弹出提示文字，点击取消不执行删除操作，点击确认调用【6.2.3.1.1.1.2.4.删除流程配置】接口，执行【SQL05】与【SQL06】，删除流程配置信息与关联节点信息，删除当前数据，刷新列表。



## 系统配置

系统配置组件的功能分解如下图：



### 用户管理

#### 类设计

##### 控制类

###### SysUserController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysUserController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成用户管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\sysController\SysUserController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| list | session[:SysUserVo] | SysUserVo | ■公有 □私有 □保护 | Result | 用户管理-用户列表查询 |
| addSave | session[:SysUserVo] | SysUserVo | ■公有 □私有 □保护 | Result | 用户管理-新增用户 |
| editSave | session[:SysUserVo] | SysUserVo | ■公有 □私有 □保护 | Result | 用户管理-修改用户 |
| resetPwdSave | session[:userId] | long | ■公有 □私有 □保护 | Result | 用户管理-重置密码 |
| remove | session[:userId] | long | ■公有 □私有 □保护 | Result | 用户管理-删除用户 |
| deptTreeData | 无 | 无 | ■公有 □私有 □保护 | Result | 用户管理-部门列表树 |
| changeStatus | session[:SysUserVo] | SysUserVo | ■公有 □私有 □保护 | Result | 用户管理-用户状态修改 |

算法描述

用户管理-用户列表查询

接口描述：根据各个登录用户获取该用户查询权限下所有用户，支持根据用户部门、用户状态、手机号码筛选。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/list | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| page | [number] | 是 | 页数 |
| pageSize | [number] | 是 | 页码 |
| userName | [string] | 是 | 用户名称 |
| status | [boolean] | 是 | 用户状态（true正常，false异常） |
| phonenumber | [string] | 是 | 手机号码 |
| deptId | [number] | 是 | 机构部门id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒信息 |
| data | [object] | 是 | 数据详情 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总记录数量 |
| data>>list | [array] | 是 | 数据集合 |
| data>>list>>userId | [number] | 是 | 用户id |
| data>>list>>deptId | [number] | 是 | 机构部门id |
| data>>list>>userName | [string] | 是 | 用户名称 |
| data>>list>>loginName | [string] | 是 | 用户账号 |
| data>>list>>userType | [string] | 是 | 用户类型 |
| data>>list>>email | [string] | 是 | 邮箱 |
| data>>list>>phonenumber | [string] | 是 | 手机号 |
| data>>list>>sex | [string] | 是 | 性别（0男 1女 2未知） |
| data>>list>>avatar | [string] | 是 | 头像地址 |
| data>>list>>password | [string] | 是 | 密码 |
| data>>list>>salt | [string] | 是 | 密码盐 |
| data>>list>>status | [boolean] | 是 | 帐号状态（true正常，false异常） |
| data>>list>>delFlag | [boolean] | 是 | 删除标志（true存在，false删除） |
| data>>list>>loginIp | [string] | 是 | 最后登录IP |
| data>>list>>loginDate | [number] | 是 | 最后登录时间 |
| data>>list>>createUserId | [number] | 是 | 创建用户id |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新用户id |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>list>>remark | [string] | 是 | 备注 |
| data>>list>>deptName | [string] | 是 | 机构部门名称 |
| data>>list>>roleName | [string] | 是 | 角色名称 |
| code | [number] | 是 | 状态码 |

用户管理-新增用户

接口描述：根据输入用户信息，在对应机构或部门下新增对应用户。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构部门id |
| userName | [string] | 是 | 用户名称 |
| loginName | [string] | 是 | 用户账号 |
| email | [string] | 是 | 邮箱 |
| phonenumber | [string] | 是 | 联系方式 |
| remark | [string] | 是 | 备注 |
| status | [boolean] | 是 | 帐号状态（true正常，false异常） |
| roleId | [number] | 是 | 角色id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回提示信息 |
| data | [number] | 是 | 新增条数 |
| code | [number] | 是 | 状态码 |

用户管理-修改用户

接口描述：根据输入用户信息，修改对应用户。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/edit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构部门id |
| userName | [string] | 是 | 用户名称 |
| loginName | [string] | 是 | 用户账号 |
| email | [string] | 是 | 邮箱 |
| phonenumber | [string] | 是 | 联系方式 |
| remark | [string] | 是 | 备注 |
| status | [boolean] | 是 | 帐号状态（true正常，false异常） |
| roleId | [number] | 是 | 角色id |
| userId | [number] | 是 | 用户id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回提示信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

用户管理-重置密码

接口描述：将用户的密码重置为系统默认密码。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/resetPwd | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| userId | [number] | 是 | 用户id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回提示信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

用户管理-删除用户

接口描述：根据选择用户进行删除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/remove | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| userId | [number] | 是 | 用户id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回提示信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

用户管理-部门列表树

接口描述：查询用户所属机构下的部门列表树形结构。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/user/deptTreeData | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
|  |  |  |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 数据 |
| data>>id | [number] | 是 | 机构部门id |
| data>>label | [string] | 是 | 机构部门名称 |
| data>>children | [array] | 是 | 子级部门 |
| data>>children>>id | [number] | 是 | 机构部门id |
| data>>children>>label | [string] | 是 | 机构部门名称 |
| code | [number] | 是 | 状态码 |

用户管理-用户状态修改

接口描述：根据用户信息修改对应用户状态。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/user/changeStatus | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| userId | [number] | 是 | 用户id |
| status | [string] | 是 | 用户状态 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回提示信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysUser

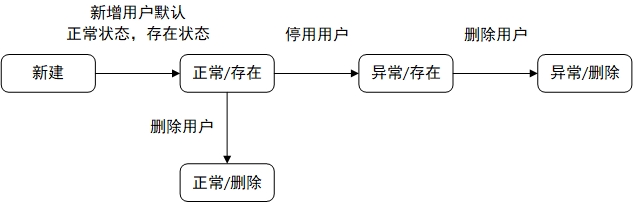
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysUser | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统用户的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\sys\po\SysUser | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| userId | long | | ■公有 □私有 □保护 | | | NNU | | 用户id |
| deptId | long | | ■公有 □私有 □保护 | | | NNI | | 机构部门id |
| userName | string | | ■公有 □私有 □保护 | | | NN | | 用户名称 |
| loginName | string | | ■公有 □私有 □保护 | | | NN | | 用户账号 |
| userType | string | | ■公有 □私有 □保护 | | |  | | 用户类型 |
| email | string | | ■公有 □私有 □保护 | | | NNIC | | 邮箱 |
| phonenumber | string | | ■公有 □私有 □保护 | | | NNIP | | 手机号 |
| sex | string | | ■公有 □私有 □保护 | | |  | | 性别（0男 1女 2未知） |
| avatar | string | | ■公有 □私有 □保护 | | |  | | 头像地址 |
| password | string | | □公有 □私有 ■保护 | | | NNS | | 密码 |
| salt | string | | ■公有 □私有 □保护 | | |  | | 密码盐 |
| status | boolean | | ■公有 □私有 □保护 | | | NN | | 帐号状态（true正常，false异常） |
| delFlag | boolean | | ■公有 □私有 □保护 | | | NN | | 删除标志（true存在，false删除） |
| loginIp | string | | ■公有 □私有 □保护 | | |  | | 最后登录IP |
| loginDate | long | | ■公有 □私有 □保护 | | |  | | 最后登录时间 |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| remark | string | | ■公有 □私有 □保护 | | |  | | 备注 |
| deptName | string | | ■公有 □私有 □保护 | | |  | | 机构部门名称 |
| roleName | string | | ■公有 □私有 □保护 | | |  | | 角色名称 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_user | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_dept、sc\_sys\_user\_role、sc\_sys\_role、sc\_resource\_user、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_sys\_user\_role**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from  sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where  u.del\_flag = 1  <if test="userId != null and userId != 0">  AND u.user\_id = #{userId}  </if>  <if test="userName != null and userName != ''">  AND u.user\_name like concat('%', #{userName}, '%')  </if>  <if test="status != null">  AND u.status = #{status}  </if>  <if test="phonenumber != null and phonenumber != ''">  AND u.phonenumber like concat('%', #{phonenumber}, '%')  </if>  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="deptId != null and deptId != 0">  AND (u.dept\_id = #{deptId} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) ))  </if>  <if test="deptIds != null">  <foreach collection="deptIds" item="id" separator="AND">  (u.dept\_id = #{id} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET(#{id},ancestors) ))  </foreach>  </if>  order by create\_time desc  【SQL02】  select user\_id, login\_name from sc\_sys\_user where login\_name=#{loginName} and del\_flag = 1 limit 1  【SQL03】  select user\_id, phonenumber from sc\_sys\_user where phonenumber=#{phonenumber} and del\_flag = 1 limit 1  【SQL04】  select user\_id, email from sc\_sys\_user where email=#{email} and del\_flag = 1 limit 1  【SQL05】  select  user\_id, login\_name  from sc\_sys\_user  <where>  del\_flag = 1  <if test="loginName != null and loginName != '' ">  and login\_name = #{loginName}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL06】  select  user\_id, phonenumber  from sc\_sys\_user  <where>  del\_flag = 1  <if test="phonenumber != null and phonenumber != '' ">  and phonenumber = #{phonenumber}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL07】  select  user\_id, email  from sc\_sys\_user  <where>  del\_flag = 1  <if test="email != null and email != ''">  and email = #{email}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL08】  select  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where u.user\_id = #{userId}  【SQL09】  select  d.dept\_id, d.parent\_id, d.dept\_name,d.dept\_short, d.order\_num, d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.update\_time, d.type  from sc\_sys\_dept d  where  d.del\_flag = 1  <if test="deptId != null and deptId != 0">  AND d.dept\_id = #{deptId}  </if>  <if test="parentId != null">  AND d.parent\_id = #{parentId}  </if>  <if test="deptName != null and deptName != ''">  AND d.dept\_name like concat('%', #{deptName}, '%')  </if>  <if test="status != null">  AND d.status = #{status}  </if>  <if test="untype != null and untype != ''">  AND (d.type = "BUSINESS-SIDE" or d.type = "OPERATION-SIDE")  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if>  order by d.parent\_id, d.order\_num  【SQL10】  select \* from sc\_sys\_dept where status = 1 and del\_flag = 1 and find\_in\_set(#{deptId}, ancestors)  【SQL11】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  where  u.user\_id = #{userId}  and r.status = 1  and r.del\_flag = 1  【SQL12】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, ur.dept\_id, d.dept\_name, r.dept\_node\_all, d.status deptStatus  from sc\_sys\_role r  left join sc\_sys\_role\_dept ur on ur.role\_id = r.role\_id  left join sc\_sys\_dept d on d.dept\_id = ur.dept\_id  where r.del\_flag = 1  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="roleName != null and roleName != ''">  AND r.role\_name like concat('%', #{roleName}, '%')  </if>  <if test="status != null">  AND r.status = #{status}  </if>  <if test="roleKey != null and roleKey != ''">  AND r.role\_key like concat('%', #{roleKey}, '%')  </if>  <if test="type != null and type != ''">  AND r.type = #{type}  </if>  <if test="deptId != null">  AND ur.dept\_id = #{deptId}  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if >  order by r.role\_sort  【SQL13】  SELECT role\_id,menu\_id FROM sc\_sys\_role\_menu WHERE role\_id IN (#{roleIds})  【SQL14】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL15】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL16】  select \* from sc\_sys\_dept  where del\_flag = 1  <choose>  <when test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, ancestors)  </foreach>  or dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </when>  <otherwise>  and find\_in\_set(#{deptId}, ancestors)  </otherwise>  </choose>  <if test="status != null">  AND status = #{status}  </if>  <if test="deptName != null and deptName != ''">  AND dept\_name like concat('%', #{deptName}, '%')  </if>  【SQL17】  select \* from sc\_sys\_dept  where dept\_id in  <foreach collection="deptIds" item="deptId" open="(" separator="," close=")">  #{deptId}  </foreach>  order by parent\_id, order\_num  【SQL18】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where d.dept\_id = #{deptId}  【SQL19】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where dept\_name=#{deptName}  and del\_flag = 1  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL20】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  <where>  <if test="menuName != null and menuName != ''">  AND menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND status = #{status}  </if>  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  <if test="pathUrl != null and pathUrl != ''">  AND path\_url = #{pathUrl}  </if>  <if test="pageUrl != null and pageUrl != ''">  AND page\_url = #{pageUrl}  </if>  </where>  order by parent\_id, order\_num  【SQL21】  select distinct m.menu\_id, m.parent\_id, m.menu\_name, m.path\_url, m.page\_url, m.status, ifnull(m.perms,'') as perms, m.menu\_type, m.icon, m.order\_num, m.create\_time,  m.ancestors,m.`show`  from sc\_sys\_menu m  left join sc\_sys\_role\_menu rm on m.menu\_id = rm.menu\_id  left join sc\_sys\_user\_role ur on rm.role\_id = ur.role\_id  left join sc\_sys\_role ro on ur.role\_id = ro.role\_id  where ur.user\_id = #{userId}  <if test="menuName != null and menuName != ''">  AND m.menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND m.status = #{status}  </if>  <if test="parentId != null">  AND m.parent\_id = #{parentId}  </if>  order by m.parent\_id, m.order\_num  【SQL22】  select \* from sc\_sys\_menu  where menu\_id in  <foreach collection="menuIds" item="menuId" index="index" open="(" separator="," close=")">  #{menuId}  </foreach>  order by parent\_id, order\_num  【SQL23】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_name=#{menuName}  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL24】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_id = #{menuId}  【SQL25】  select count(1) from sc\_sys\_menu where parent\_id = #{menuId}  【SQL26】  select count(1) from sc\_sys\_role\_menu where menu\_id = #{menuId} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_user\_role、sc\_resource\_user、sc\_sys\_oper\_log、sc\_sys\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_dept、sc\_resource、sc\_sys\_menu **>**  **操作：<插入>** |
| SQL语句：  【SQL27】  INSERT INTO sc\_sys\_user ( dept\_id, user\_name, login\_name, user\_type, email, phonenumber, password, status, del\_flag, create\_time, update\_time, remark ) VALUES (#{deptId},#{userName},#{loginName},#{userType},#{email},#{phonenumber} , #{password},#{status},#{delFlag},#{createTime},#{updateTime},#{remark } )  【SQL28】  INSERT INTO sc\_sys\_user\_role ( user\_id, role\_id ) VALUES (#{userId},#{roleId} )  【SQL29】  INSERT INTO sc\_resource\_user ( id, dept\_id, user\_id, cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_time ) VALUES (#{id},#{deptId},#{userId},#{cpu},#{internalMemory},  #{useInternalMemory},#{diskMemory},#{useDiskMemory},#{points},#{createUserId},#{createTime},#{updateTime})  【SQL30】  INSERT INTO sc\_sys\_oper\_log ( title, business\_type, method, oper\_name, oper\_ip, status, oper\_time, cost\_time ) VALUES (#{title},#{businessType},#{method},#{operName},#{operIp},  #{status},#{operTime},#{costTime})  【SQL31】  INSERT INTO sc\_sys\_role ( role\_name, role\_sort, status, del\_flag, create\_time, update\_time, remark, dept\_node\_all ) VALUES ( #{roleName},#{roleSort},#{status},#{delFlag},#{createTime},  #{updateTime},#{remark},#{deptNodeAll })  【SQL32】  INSERT INTO sc\_sys\_role\_dept ( role\_id, dept\_id ) VALUES ( #{roleId},#{deptId} )  【SQL33】  insert into sc\_sys\_role\_menu(role\_id, menu\_id) values (#{roleId},#{menuId})  【SQL34】  INSERT INTO sc\_sys\_dept ( dept\_name, dept\_short, order\_num, ancestors, leader, phone, id\_card, address, registered\_address, photo\_address\_positive, photo\_address\_negative, status, del\_flag, create\_user\_id, create\_time, update\_time, type )(  #{deptName},#{deptShort},#{orderNum},#{ancestors},#{leader},#{phone},#{id\_card},#{address},#{registered\_address},#{photo\_address\_positive},#{ photo\_address\_negative},#{status},#{del\_flag},#{delFlag},#{create\_user\_id},#{create\_time},#{update\_time},#{type})  【SQL35】  INSERT INTO sc\_resource ( id, dept\_id, cpu, use\_cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_user\_id, update\_time ) VALUES ( #{id},#{dept\_id},#{cpu},#{use\_cpu},#{internal\_memory},#{use\_internal\_memory},#{disk\_memory},#{use\_disk\_memory},#{points},#{create\_user\_id},#{create\_time},#{update\_user\_id},#{update\_time })  【SQL36】  INSERT INTO sc\_sys\_menu (menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, perms, icon, create\_time,ancestors,`show`)  VALUES( #{menu\_id},#{menu\_name},#{parent\_id},#{order\_num},#{path\_url},#{ page\_url},#{menu\_type},#{status},#{perms},#{icon},#{create\_time},#{ancestors},#{show}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 用户id |  | | deptId | 机构部门id |  | | userName | 用户名称 |  | | loginName | 用户账号 |  | | offerDeptName | 提供机构id |  | | password | 用户密码 |  | | status | 状态 |  | | delFlag | 删除状态 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_role、sc\_sys\_dept、sc\_sys\_menu**>**  **操作：<修改>** |
| SQL语句：  【SQL37】  UPDATE sc\_sys\_user  SET dept\_id = #{deptId},  user\_name = #{userName},  login\_name = #{loginName},  user\_type = #{userType},  email = #{email},  phonenumber = #{phonenumber},  password = #{password},  status = #{status},  del\_flag = #{delFlag},  create\_time = #{createTime},  update\_time = #{updateTime},  remark = #{remark}  WHERE  user\_id = #{userId}  【SQL38】  update sc\_sys\_user set del\_flag = 0 where user\_id = #{userId}  【SQL39】  UPDATE sc\_sys\_role  SET role\_name=#{roleName},  role\_sort=#{roleSort},  status=#{status},  update\_time=#{updateTime},  remark=#{remark},  dept\_node\_all=#{deptNodeAll}  WHERE  role\_id=#{roleId}  【SQL40】  update sc\_sys\_role set del\_flag = 0 where role\_id = #{roleId}  【SQL41】  UPDATE sc\_sys\_dept SET parent\_id=#{parent\_id}, dept\_name=#{dept\_name}, dept\_short=#{dept\_short}, order\_num=#{order\_num}, ancestors=#{ancestors}, leader=#{leader}, phone=#{phone}, id\_card=#{id\_card}, photo\_address\_positive=#{photo\_address\_positive}, photo\_address\_negative=#{photo\_address\_negative}, business\_license=#{business\_license}, status=#{status}, del\_flag=#{del\_flag}, create\_user\_id=#{create\_user\_id}, create\_time=#{create\_time}, update\_user\_id=#{update\_user\_id}, update\_time=#{update\_time}, type=#{type} WHERE dept\_id=#{dept\_id}  【SQL42】  update sc\_sys\_dept set status = #{status} where find\_in\_set(#{dept\_id}, ancestors)  【SQL43】  update sc\_sys\_user set status = #{status} where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL44】  update sc\_sys\_dept set del\_flag = 0 where dept\_id = #{deptId} or find\_in\_set(#{deptId}, ancestors)  【SQL45】  update sc\_sys\_user set del\_flag = 0 where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL46】  update sc\_sys\_menu  Set menu\_name =#{menu\_name},  parent\_id=#{parent\_id},  order\_num=#{order\_num},  path\_url=#{path\_url},  page\_url=#{page\_url},  menu\_type=#{menu\_type},  status=#{status},  perms=#{perms},  icon=#{icon},  create\_time=#{create\_time},  ancestors=#{ancestors},  `show`=#{show}  where menu\_id = #{menu\_id}  【SQL47】  update sc\_sys\_menu set status = #{status} where find\_in\_set(#{menuId}, ancestors) |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | status | 0 | 1 | 启动状态 | | del\_flag | 1 | 0 | 删除状态 | |  |  |  |  | |
| **程序流程中涉及的数据库表：<**sc\_sys\_user\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_resource**>**  **操作：<删除>** |
| SQL语句：  【SQL48】  delete from sc\_sys\_user\_role where user\_id = #{userId}  【SQL49】  delete from sc\_sys\_user\_role where role\_id = #{roleId}  【SQL50】  delete from sc\_sys\_role\_dept where role\_id= #{roleId}  【SQL51】  delete from sc\_sys\_role\_menu where role\_id=#{roleId}  【SQL52】  delete from sc\_sys\_role\_dept where dept\_id= #{deptId}  【SQL53】  delete from sc\_resource where dept\_id= #{deptId}  【SQL54】  delete from sc\_sys\_menu where menu\_id= #{menuId} |

#### 总体设计方案

##### 设计方案

用户管理用于部门及部门下所有用户，提供新增用户、修改用户、删除用户、停用/启用用户、重置密码功能。

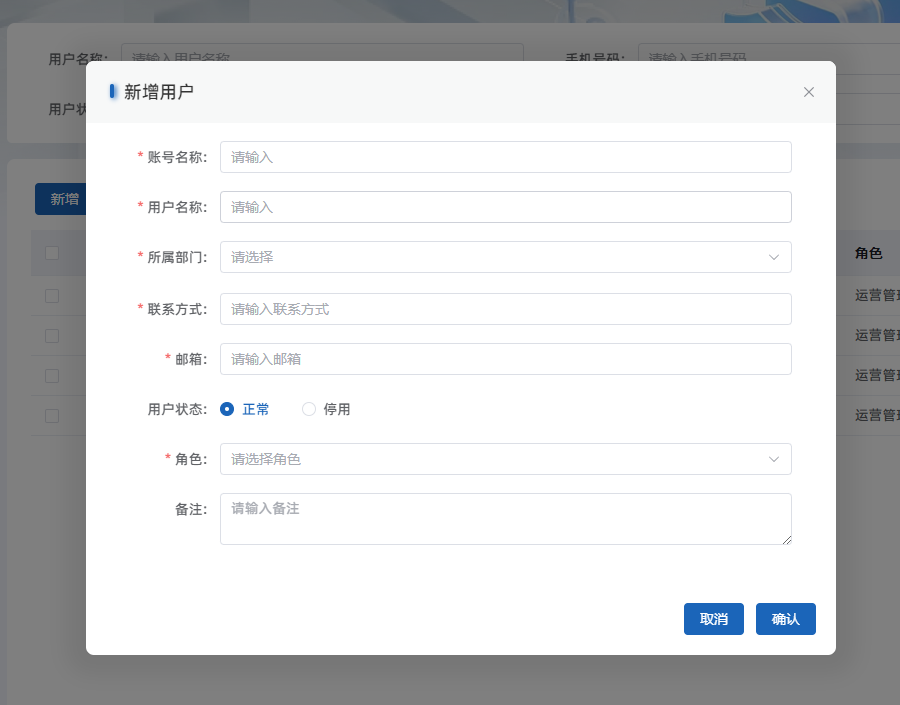
##### 处理流程

1. 查询用户列表



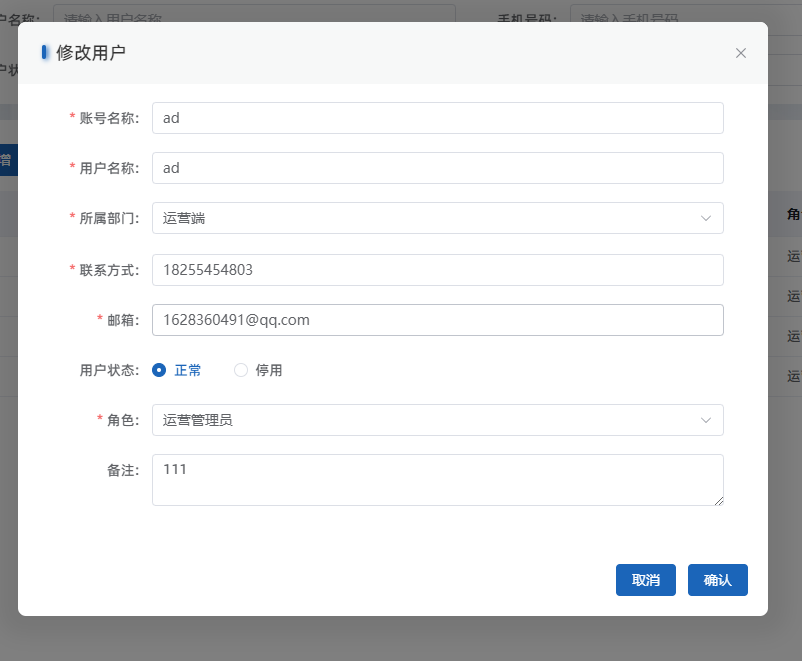
页面左侧调用【6.3.1.1.1.1.2.6.用户管理-部门列表树】接口根据token获取用户id，执行【SQL09】查询当前用户权限可查看的所有父级机构，再根据父级id集合执行【SQL10】查询所有子级部门集合，再将父级与子级组合封装成树形结构展示。页面右侧用户列表通过请求【6.3.1.1.1.1.2.1.用户管理-用户列表查询】接口分页获取用户列表，支持用户名称、手机号码、用户状态、角色、部门多个条件查询。接口内部通过token获取用户与机构部门信息，将查询条件与用户信息作为查询条件执行【SQL01】分页获取用户列表返回页面展示。分别展示账号名、用户名称、所属部门、联系方式、状态、角色、创建时间。

1. 新增用户



页面点击【新增】弹出新增用户页面，新增页面展示用户配置信息。配置完成后将配置信息作为参数调用【6.3.1.1.1.1.2.2.用户管理-新增用户】接口。接口先执行判重操作，分别根据用户账号执行【SQL02】查询用户账号判断是否重复，根据用户手机号执行【SQL03】查询用户手机号判断是否重复，根据用户邮箱执行【SQL04】查询用户邮箱判断是否重复。若用户未重复，将用户信息封装执行【SQL33】进行插入用户信息操作，用户新建成功后，将用户与角色关联信息执行【SQL34】插入用户角色关联表，最后默认给新用户分配默认用户资源执行【SQL35】。

1. 修改用户



页面点击【修改】弹出修改用户页面，修改页面展示用户配置信息。配置完成后将配置信息作为参数调用【6.3.1.1.1.1.2.3.用户管理-修改用户】接口。接口先判断是否可修改再执行判重操作，分别根据用户id执行【SQL11】查询是否可操作，根据用户账号与用户id执行【SQL05】查询用户账号判断是否重复，根据用户手机号与用户id执行【SQL06】查询用户手机号判断是否重复，根据用户邮箱与用户id执行【SQL07】查询用户邮箱判断是否重复。用户未重复，先将用户已存在的角色关联信息删除【SQL48】，再将新增的用户与角色关联信息执行【SQL34】插入用户角色关联表，最后把修改的用户配置信息执行【SQL37】进行修改。

1. 停用/启用用户



页面选择用户，点击【停用/启用】调用【6.3.1.1.1.1.2.7.用户管理-用户状态修改】接口先判断是否可修改操作，根据用户id执行【SQL11】查询是否可操作。用户可修改情况下，根据用户id执行【SQL08】查询用户信息，再根据页面传递的用户状态执行【SQL37】将用户状态修改。

1. 重置用户密码



页面选择用户，点击【重置密码】经过确认交互后调用【6.3.1.1.1.1.2.4.用户管理-重置密码】接口先判断是否可修改操作，根据用户id执行【SQL11】查询是否可操作。用户可修改情况下，将默认密码进行加密，再根据用户id执行【SQL37】将用户密码修改。

1. 删除用户



页面选择用户，点击【删除】经过确认交互后调用【6.3.1.1.1.1.2.5.用户管理-删除用户】接口先判断是否可修改操作，根据用户id执行【SQL11】查询是否可操作。用户可修改情况下，根据用户id执行【SQL38】将用户逻辑删除，同时执行【SQL48】删除用户角色关联信息。

### 角色管理

#### 类设计

##### 控制类

###### SysRoleController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysRoleController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成角色管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\sysController\SysRoleController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| list | session[:SysRoleVo] | SysRoleVo | ■公有 □私有 □保护 | Result | 角色管理-角色列表分页查询 |
| listByDept | session[:SysRoleVo] | SysRoleVo | ■公有 □私有 □保护 | Result | 角色管理-角色列表查询 |
| getInfo | session[:roleId] | long | ■公有 □私有 □保护 | Result | 角色管理-角色详情查询 |
| add | session[:SysRoleVo] | SysRoleVo | ■公有 □私有 □保护 | Result | 角色管理-新增角色 |
| edit | session[:SysRoleVo] | SysRoleVo | ■公有 □私有 □保护 | Result | 角色管理-修改角色 |
| updateStatus | session[:SysRoleVo] | SysRoleVo | ■公有 □私有 □保护 | Result | 角色管理-修改角色状态 |
| remove | session[:roleId] | long | ■公有 □私有 □保护 | Result | 角色管理-删除角色 |

算法描述

角色管理-角色列表分页查询

接口描述：根据各个登录用户获取该用户所属机构下的角色列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/role/list | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| roleName | [string] | 是 | 角色名称 |
| status | [boolean] | 是 | 角色状态（true正常 false停用） |
| type | [string] | 是 | 角色类型（数据端 DATA-SIDE,业务端 BUSINESS-SIDE,运营端 OPERATION-SIDE） |
| page | [number] | 是 | 页数 |
| pageSize | [string] | 是 | 页码 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 数据 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总记录数量 |
| data>>list | [array] | 是 | 数据行 |
| data>>list>>roleId | [number] | 是 | 角色ID |
| data>>list>>roleName | [string] | 是 | 角色名称 |
| data>>list>>roleKey | [string] | 是 | 角色权限字符串 |
| data>>list>>roleSort | [string] | 是 | 显示顺序 |
| data>>list>>type | [string] | 是 | 角色类型（数据端 DATA-SIDE,业务端 BUSINESS-SIDE,运营端 OPERATION-SIDE） |
| data>>list>>status | [boolean] | 是 | 角色状态（true正常 false停用） |
| data>>list>>delFlag | [boolean] | 是 | 删除标志（true代表存在 false代表删除） |
| data>>list>>createUserId | [number] | 是 | 创建用户id |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新用户id |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>list>>remark | [string] | 是 | 备注 |
| code | [number] | 是 | 状态码 |

角色管理-角色列表查询

接口描述：根据各个登录用户获取该用户所属机构下的角色列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/role/list/dept |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 数据 |
| data | [array] | 是 | 数据行 |
| data>>roleId | [number] | 是 | 角色ID |
| data>>roleName | [string] | 是 | 角色名称 |
| data>>roleKey | [string] | 是 | 角色权限字符串 |
| data>>roleSort | [string] | 是 | 显示顺序 |
| data>>type | [string] | 是 | 角色类型（数据端 DATA-SIDE,业务端 BUSINESS-SIDE,运营端 OPERATION-SIDE） |
| data>>status | [boolean] | 是 | 角色状态（true正常 false停用） |
| data>>delFlag | [boolean] | 是 | 删除标志（true代表存在 false代表删除） |
| data>>createUserId | [number] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [number] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>remark | [string] | 是 | 备注 |
| code | [number] | 是 | 状态码 |

角色管理-角色详情查询

接口描述：点击查询角色详细信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/role/info/{roleId} |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 数据行 |
| data>>roleId | [number] | 是 | 角色ID |
| data>>roleName | [string] | 是 | 角色名称 |
| data>>roleKey | [string] | 是 | 角色权限字符串 |
| data>>roleSort | [string] | 是 | 显示顺序 |
| data>>type | [string] | 是 | 角色类型（数据端 DATA-SIDE,业务端 BUSINESS-SIDE,运营端 OPERATION-SIDE） |
| data>>status | [boolean] | 是 | 角色状态（true正常 false停用） |
| data>>delFlag | [boolean] | 是 | 删除标志（true代表存在 false代表删除） |
| data>>createUserId | [number] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [number] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>remark | [string] | 是 | 备注 |
| code | [number] | 是 | 状态码 |

角色管理-新增角色

接口描述：在当前机构下新增对应角色，保存对应角色信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/role/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| roleName | [string] | 是 | 角色名称 |
| roleSort | [number] | 是 | 显示顺序 |
| type | [string] | 是 | 角色类型（数据端 DATA-SIDE，业务端 BUSINESS-SIDE，运营端 OPERATION-SIDE） |
| status | [boolean] | 是 | 角色状态（true正常 false停用） |
| menuIds | [array] | 是 | 菜单id集合 |
| deptId | [number] | 是 | 机构id |
| remark | [string] | 是 | 备注 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 新增条数 |
| code | [number] | 是 | 状态码 |

角色管理-修改角色

接口描述：根据角色信息修改当前角色。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/role/edit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| roleName | [string] | 是 | 角色名称 |
| roleSort | [number] | 是 | 显示顺序 |
| type | [string] | 是 | 角色类型（数据端 DATA-SIDE，业务端 BUSINESS-SIDE，运营端 OPERATION-SIDE） |
| status | [boolean] | 是 | 角色状态（true正常 false停用） |
| menuIds | [array] | 是 | 菜单id集合 |
| deptId | [number] | 是 | 机构id |
| remark | [string] | 是 | 备注 |
| roleId | [number] | 是 | 角色id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

角色管理-修改角色状态

接口描述：进行停用启动角色状态。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/role/updateStatus | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| status | [boolean] | 是 | 角色状态（true正常 false停用） |
| roleId | [number] | 是 | 角色id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

角色管理-删除角色

接口描述：根据选中角色id删除对应角色。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/role/remove | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| roleId | [number] | 是 | 角色id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysRole

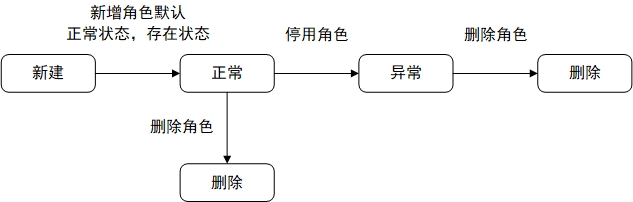
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysRole | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统角色的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\sys\po\SysRole | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| roleId | long | | ■公有 □私有 □保护 | | | NNU | | 角色ID |
| roleName | string | | ■公有 □私有 □保护 | | | NNS | | 角色名称 |
| roleKey | string | | ■公有 □私有 □保护 | | |  | | 角色权限字符串 |
| roleSort | string | | ■公有 □私有 □保护 | | | NN | | 显示顺序 |
| type | string | | ■公有 □私有 □保护 | | |  | | 角色类型（数据端 DATA-SIDE，业务端 BUSINESS-SIDE，运营端 OPERATION-SIDE） |
| status | boolean | | ■公有 □私有 □保护 | | | NN | | 角色状态（true正常 false停用） |
| delFlag | boolean | | ■公有 □私有 □保护 | | | NN | | 删除标志（true代表存在 false代表删除） |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| remark | string | | ■公有 □私有 □保护 | | |  | | 备注 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_role | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_dept、sc\_sys\_user\_role、sc\_sys\_role、sc\_resource\_user、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_sys\_user\_role**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from  sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where  u.del\_flag = 1  <if test="userId != null and userId != 0">  AND u.user\_id = #{userId}  </if>  <if test="userName != null and userName != ''">  AND u.user\_name like concat('%', #{userName}, '%')  </if>  <if test="status != null">  AND u.status = #{status}  </if>  <if test="phonenumber != null and phonenumber != ''">  AND u.phonenumber like concat('%', #{phonenumber}, '%')  </if>  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="deptId != null and deptId != 0">  AND (u.dept\_id = #{deptId} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) ))  </if>  <if test="deptIds != null">  <foreach collection="deptIds" item="id" separator="AND">  (u.dept\_id = #{id} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET(#{id},ancestors) ))  </foreach>  </if>  order by create\_time desc  【SQL02】  select user\_id, login\_name from sc\_sys\_user where login\_name=#{loginName} and del\_flag = 1 limit 1  【SQL03】  select user\_id, phonenumber from sc\_sys\_user where phonenumber=#{phonenumber} and del\_flag = 1 limit 1  【SQL04】  select user\_id, email from sc\_sys\_user where email=#{email} and del\_flag = 1 limit 1  【SQL05】  select  user\_id, login\_name  from sc\_sys\_user  <where>  del\_flag = 1  <if test="loginName != null and loginName != '' ">  and login\_name = #{loginName}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL06】  select  user\_id, phonenumber  from sc\_sys\_user  <where>  del\_flag = 1  <if test="phonenumber != null and phonenumber != '' ">  and phonenumber = #{phonenumber}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL07】  select  user\_id, email  from sc\_sys\_user  <where>  del\_flag = 1  <if test="email != null and email != ''">  and email = #{email}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL08】  select  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where u.user\_id = #{userId}  【SQL09】  select  d.dept\_id, d.parent\_id, d.dept\_name,d.dept\_short, d.order\_num, d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.update\_time, d.type  from sc\_sys\_dept d  where  d.del\_flag = 1  <if test="deptId != null and deptId != 0">  AND d.dept\_id = #{deptId}  </if>  <if test="parentId != null">  AND d.parent\_id = #{parentId}  </if>  <if test="deptName != null and deptName != ''">  AND d.dept\_name like concat('%', #{deptName}, '%')  </if>  <if test="status != null">  AND d.status = #{status}  </if>  <if test="untype != null and untype != ''">  AND (d.type = "BUSINESS-SIDE" or d.type = "OPERATION-SIDE")  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if>  order by d.parent\_id, d.order\_num  【SQL10】  select \* from sc\_sys\_dept where status = 1 and del\_flag = 1 and find\_in\_set(#{deptId}, ancestors)  【SQL11】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  where  u.user\_id = #{userId}  and r.status = 1  and r.del\_flag = 1  【SQL12】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, ur.dept\_id, d.dept\_name, r.dept\_node\_all, d.status deptStatus  from sc\_sys\_role r  left join sc\_sys\_role\_dept ur on ur.role\_id = r.role\_id  left join sc\_sys\_dept d on d.dept\_id = ur.dept\_id  where r.del\_flag = 1  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="roleName != null and roleName != ''">  AND r.role\_name like concat('%', #{roleName}, '%')  </if>  <if test="status != null">  AND r.status = #{status}  </if>  <if test="roleKey != null and roleKey != ''">  AND r.role\_key like concat('%', #{roleKey}, '%')  </if>  <if test="type != null and type != ''">  AND r.type = #{type}  </if>  <if test="deptId != null">  AND ur.dept\_id = #{deptId}  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if >  order by r.role\_sort  【SQL13】  SELECT role\_id,menu\_id FROM sc\_sys\_role\_menu WHERE role\_id IN (#{roleIds})  【SQL14】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL15】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL16】  select \* from sc\_sys\_dept  where del\_flag = 1  <choose>  <when test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, ancestors)  </foreach>  or dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </when>  <otherwise>  and find\_in\_set(#{deptId}, ancestors)  </otherwise>  </choose>  <if test="status != null">  AND status = #{status}  </if>  <if test="deptName != null and deptName != ''">  AND dept\_name like concat('%', #{deptName}, '%')  </if>  【SQL17】  select \* from sc\_sys\_dept  where dept\_id in  <foreach collection="deptIds" item="deptId" open="(" separator="," close=")">  #{deptId}  </foreach>  order by parent\_id, order\_num  【SQL18】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where d.dept\_id = #{deptId}  【SQL19】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where dept\_name=#{deptName}  and del\_flag = 1  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL20】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  <where>  <if test="menuName != null and menuName != ''">  AND menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND status = #{status}  </if>  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  <if test="pathUrl != null and pathUrl != ''">  AND path\_url = #{pathUrl}  </if>  <if test="pageUrl != null and pageUrl != ''">  AND page\_url = #{pageUrl}  </if>  </where>  order by parent\_id, order\_num  【SQL21】  select distinct m.menu\_id, m.parent\_id, m.menu\_name, m.path\_url, m.page\_url, m.status, ifnull(m.perms,'') as perms, m.menu\_type, m.icon, m.order\_num, m.create\_time,  m.ancestors,m.`show`  from sc\_sys\_menu m  left join sc\_sys\_role\_menu rm on m.menu\_id = rm.menu\_id  left join sc\_sys\_user\_role ur on rm.role\_id = ur.role\_id  left join sc\_sys\_role ro on ur.role\_id = ro.role\_id  where ur.user\_id = #{userId}  <if test="menuName != null and menuName != ''">  AND m.menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND m.status = #{status}  </if>  <if test="parentId != null">  AND m.parent\_id = #{parentId}  </if>  order by m.parent\_id, m.order\_num  【SQL22】  select \* from sc\_sys\_menu  where menu\_id in  <foreach collection="menuIds" item="menuId" index="index" open="(" separator="," close=")">  #{menuId}  </foreach>  order by parent\_id, order\_num  【SQL23】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_name=#{menuName}  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL24】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_id = #{menuId}  【SQL25】  select count(1) from sc\_sys\_menu where parent\_id = #{menuId}  【SQL26】  select count(1) from sc\_sys\_role\_menu where menu\_id = #{menuId} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_user\_role、sc\_resource\_user、sc\_sys\_oper\_log、sc\_sys\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_dept、sc\_resource、sc\_sys\_menu **>**  **操作：<插入>** |
| SQL语句：  【SQL27】  INSERT INTO sc\_sys\_user ( dept\_id, user\_name, login\_name, user\_type, email, phonenumber, password, status, del\_flag, create\_time, update\_time, remark ) VALUES (#{deptId},#{userName},#{loginName},#{userType},#{email},#{phonenumber} , #{password},#{status},#{delFlag},#{createTime},#{updateTime},#{remark } )  【SQL28】  INSERT INTO sc\_sys\_user\_role ( user\_id, role\_id ) VALUES (#{userId},#{roleId} )  【SQL29】  INSERT INTO sc\_resource\_user ( id, dept\_id, user\_id, cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_time ) VALUES (#{id},#{deptId},#{userId},#{cpu},#{internalMemory},  #{useInternalMemory},#{diskMemory},#{useDiskMemory},#{points},#{createUserId},#{createTime},#{updateTime})  【SQL30】  INSERT INTO sc\_sys\_oper\_log ( title, business\_type, method, oper\_name, oper\_ip, status, oper\_time, cost\_time ) VALUES (#{title},#{businessType},#{method},#{operName},#{operIp},  #{status},#{operTime},#{costTime})  【SQL31】  INSERT INTO sc\_sys\_role ( role\_name, role\_sort, status, del\_flag, create\_time, update\_time, remark, dept\_node\_all ) VALUES ( #{roleName},#{roleSort},#{status},#{delFlag},#{createTime},  #{updateTime},#{remark},#{deptNodeAll })  【SQL32】  INSERT INTO sc\_sys\_role\_dept ( role\_id, dept\_id ) VALUES ( #{roleId},#{deptId} )  【SQL33】  insert into sc\_sys\_role\_menu(role\_id, menu\_id) values (#{roleId},#{menuId})  【SQL34】  INSERT INTO sc\_sys\_dept ( dept\_name, dept\_short, order\_num, ancestors, leader, phone, id\_card, address, registered\_address, photo\_address\_positive, photo\_address\_negative, status, del\_flag, create\_user\_id, create\_time, update\_time, type )(  #{deptName},#{deptShort},#{orderNum},#{ancestors},#{leader},#{phone},#{id\_card},#{address},#{registered\_address},#{photo\_address\_positive},#{ photo\_address\_negative},#{status},#{del\_flag},#{delFlag},#{create\_user\_id},#{create\_time},#{update\_time},#{type})  【SQL35】  INSERT INTO sc\_resource ( id, dept\_id, cpu, use\_cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_user\_id, update\_time ) VALUES ( #{id},#{dept\_id},#{cpu},#{use\_cpu},#{internal\_memory},#{use\_internal\_memory},#{disk\_memory},#{use\_disk\_memory},#{points},#{create\_user\_id},#{create\_time},#{update\_user\_id},#{update\_time })  【SQL36】  INSERT INTO sc\_sys\_menu (menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, perms, icon, create\_time,ancestors,`show`)  VALUES( #{menu\_id},#{menu\_name},#{parent\_id},#{order\_num},#{path\_url},#{ page\_url},#{menu\_type},#{status},#{perms},#{icon},#{create\_time},#{ancestors},#{show}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 用户id |  | | deptId | 机构部门id |  | | userName | 用户名称 |  | | loginName | 用户账号 |  | | offerDeptName | 提供机构id |  | | password | 用户密码 |  | | status | 状态 |  | | delFlag | 删除状态 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_role、sc\_sys\_dept、sc\_sys\_menu**>**  **操作：<修改>** |
| SQL语句：  【SQL37】  UPDATE sc\_sys\_user  SET dept\_id = #{deptId},  user\_name = #{userName},  login\_name = #{loginName},  user\_type = #{userType},  email = #{email},  phonenumber = #{phonenumber},  password = #{password},  status = #{status},  del\_flag = #{delFlag},  create\_time = #{createTime},  update\_time = #{updateTime},  remark = #{remark}  WHERE  user\_id = #{userId}  【SQL38】  update sc\_sys\_user set del\_flag = 0 where user\_id = #{userId}  【SQL39】  UPDATE sc\_sys\_role  SET role\_name=#{roleName},  role\_sort=#{roleSort},  status=#{status},  update\_time=#{updateTime},  remark=#{remark},  dept\_node\_all=#{deptNodeAll}  WHERE  role\_id=#{roleId}  【SQL40】  update sc\_sys\_role set del\_flag = 0 where role\_id = #{roleId}  【SQL41】  UPDATE sc\_sys\_dept SET parent\_id=#{parent\_id}, dept\_name=#{dept\_name}, dept\_short=#{dept\_short}, order\_num=#{order\_num}, ancestors=#{ancestors}, leader=#{leader}, phone=#{phone}, id\_card=#{id\_card}, photo\_address\_positive=#{photo\_address\_positive}, photo\_address\_negative=#{photo\_address\_negative}, business\_license=#{business\_license}, status=#{status}, del\_flag=#{del\_flag}, create\_user\_id=#{create\_user\_id}, create\_time=#{create\_time}, update\_user\_id=#{update\_user\_id}, update\_time=#{update\_time}, type=#{type} WHERE dept\_id=#{dept\_id}  【SQL42】  update sc\_sys\_dept set status = #{status} where find\_in\_set(#{dept\_id}, ancestors)  【SQL43】  update sc\_sys\_user set status = #{status} where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL44】  update sc\_sys\_dept set del\_flag = 0 where dept\_id = #{deptId} or find\_in\_set(#{deptId}, ancestors)  【SQL45】  update sc\_sys\_user set del\_flag = 0 where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL46】  update sc\_sys\_menu  Set menu\_name =#{menu\_name},  parent\_id=#{parent\_id},  order\_num=#{order\_num},  path\_url=#{path\_url},  page\_url=#{page\_url},  menu\_type=#{menu\_type},  status=#{status},  perms=#{perms},  icon=#{icon},  create\_time=#{create\_time},  ancestors=#{ancestors},  `show`=#{show}  where menu\_id = #{menu\_id}  【SQL47】  update sc\_sys\_menu set status = #{status} where find\_in\_set(#{menuId}, ancestors) |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | status | 0 | 1 | 启动状态 | | del\_flag | 1 | 0 | 删除状态 | |  |  |  |  | |
| **程序流程中涉及的数据库表：<**sc\_sys\_user\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_resource**>**  **操作：<删除>** |
| SQL语句：  【SQL48】  delete from sc\_sys\_user\_role where user\_id = #{userId}  【SQL49】  delete from sc\_sys\_user\_role where role\_id = #{roleId}  【SQL50】  delete from sc\_sys\_role\_dept where role\_id= #{roleId}  【SQL51】  delete from sc\_sys\_role\_menu where role\_id=#{roleId}  【SQL52】  delete from sc\_sys\_role\_dept where dept\_id= #{deptId}  【SQL53】  delete from sc\_resource where dept\_id= #{deptId}  【SQL54】  delete from sc\_sys\_menu where menu\_id= #{menuId} |

#### 总体设计方案

##### 设计方案

角色管理用于管理机构权限下展示的角色，提供新增角色、修改角色、删除角色、停用/启用角色功能。

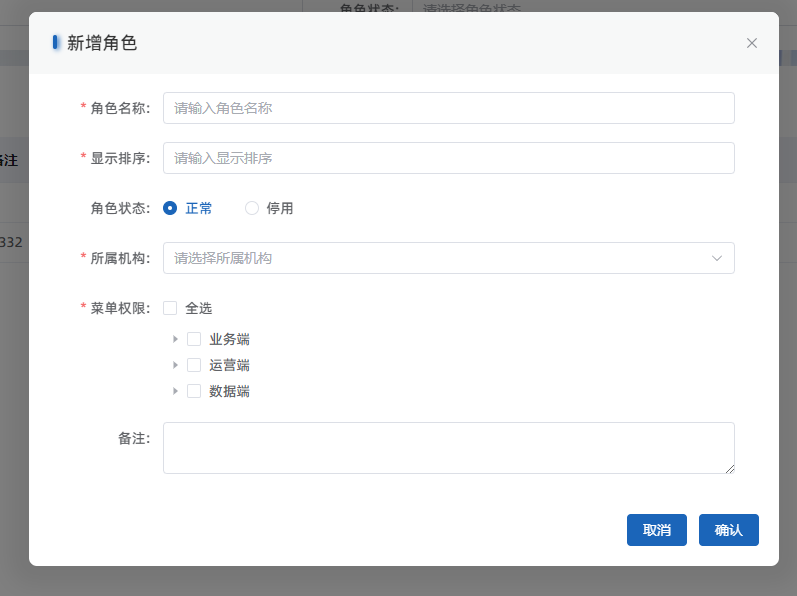
##### 处理流程

1. 查询角色列表



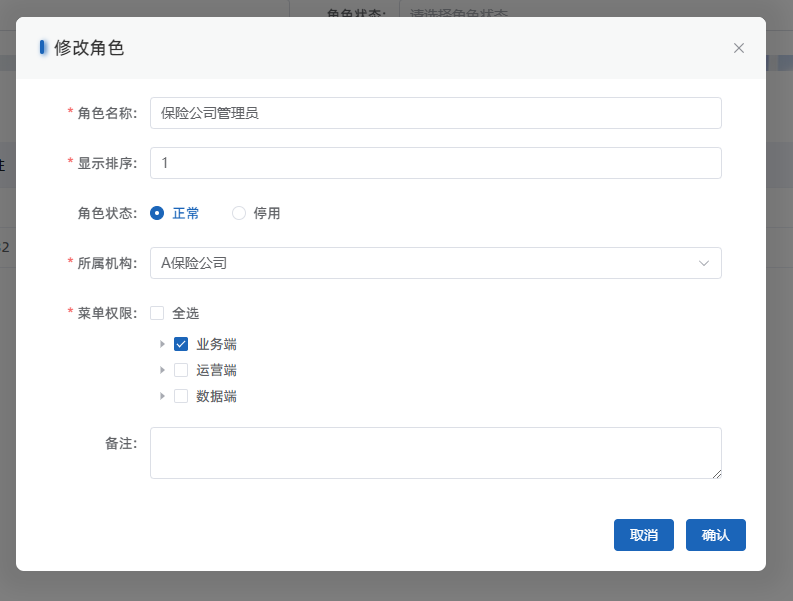
页面调用【6.3.2.1.1.1.2.1.角色管理-角色列表分页查询】接口根据token获取用户信息。判断用户角色，若用户为非超级管理员与运营端机构角色根据当前用户机构id执行【SQL12】查询所有角色集合；若用户为运营端机构角色，先执行【SQL09】查询运营端机构可查看的所有父级机构id，再根据机构id集合执行【SQL12】查询所有角色集合。查询角色集合不为空的情况，封装所有查询角色id集合，执行【SQL13】根据角色id集合查询所有角色菜单关联信息，再封装到角色列表中返回页面。分别展示角色名称、备注、排序、所属机构、状态、创建时间。

1. 新增角色



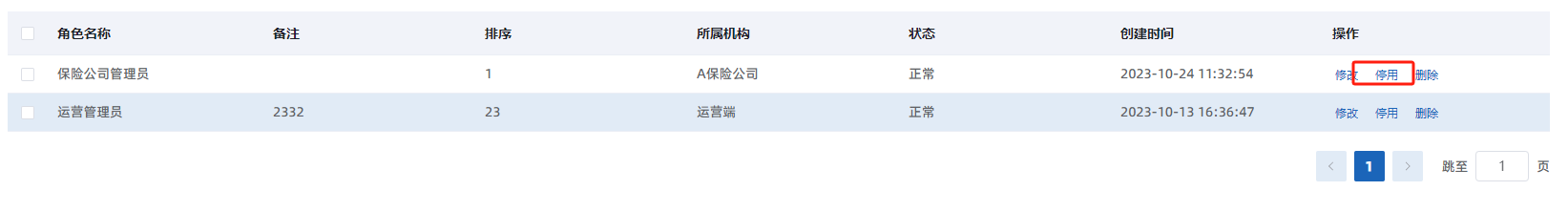
页面点击【新增】弹出新增角色页面，新增页面展示角色配置信息。配置完成后将配置信息作为参数调用【6.3.2.1.1.1.2.4.角色管理-新增角色】接口。接口先执行判重操作，分别根据角色名称执行【SQL14】查询角色名称判断是否重复，根据角色权限执行【SQL15】查询角色权限判断是否重复。若角色未重复，将角色信息封装执行【SQL31】进行插入角色信息操作，角色新建成功后，将角色与机构关联信息执行【SQL32】插入角色机构关联表，最后将角色与菜单关联信息执行【SQL39】插入角色菜单关联表。

1. 修改角色



页面点击【修改】弹出修改角色页面，修改页面展示角色配置信息。配置完成后将配置信息作为参数调用【6.3.2.1.1.1.2.5.角色管理-修改角色】接口。接口先判断角色是否为超级管理员，超级管理员不可修改，再执行判重操作，根据角色名称执行【SQL14】查询角色名称判断是否重复，根据角色权限执行【SQL15】查询角色权限判断是否重复。角色未重复，先将角色已存在的菜单关联和机构关联信息删除【SQL50】【SQL51】，再将新增的角色与机构关联信息执行【SQL32】插入角色机构关联表，新增的角色与菜单关联信息执行【SQL33】插入角色菜单关联表，最后把修改的角色配置信息执行【SQL39】进行修改。

1. 停用/启用角色



页面选择角色，点击【停用/启用】调用【6.3.2.1.1.1.2.6.角色管理-修改角色状态】接口先判断角色是否为超级管理员，超级管理员不可修改。角色可修改情况下，根据角色id执行【SQL39】修改角色状态，再根据角色id修改角色所对应用户状态修改。

1. 删除角色



页面选择角色，点击【删除】经过确认交互后调用【6.3.2.1.1.1.2.7.角色管理-删除角色】接口先判断先判断角色是否为超级管理员，超级管理员不可删除。角色可删除情况下，根据角色id执行【SQL40】将角色逻辑删除，同时执行【SQL49】删除用户角色关联信息，执行【SQL50】将角色已存在的菜单关联信息删除，执行【SQL44】将角色已存在的机构关联信息删除。

### 机构/部门管理

#### 类设计

##### 控制类

###### SysDeptController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysDeptController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成部门管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\sysController\SysDeptController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| list | session[:SysDeptVo] | SysDeptVo | ■公有 □私有 □保护 | Result | 机构管理-机构列表查询 |
| page | session[:SysDeptVo] | SysDeptVo | ■公有 □私有 □保护 | Result | 机构管理-机构列表分页查询 |
| add | session[:SysDept] | SysDept | ■公有 □私有 □保护 | Result | 机构管理-新增机构 |
| edit | session[:SysDept] | SysDept | ■公有 □私有 □保护 | Result | 机构管理-修改机构 |
| getInfo | session[:deptId] | long | ■公有 □私有 □保护 | Result | 机构管理-机构详情查询 |
| changeStatus | session[:SysDept] | SysDept | ■公有 □私有 □保护 | Result | 机构管理-修改机构状态 |
| remove | session[:deptId] | long | ■公有 □私有 □保护 | Result | 机构管理-删除机构 |
| deptTreeData | 无 | 无 | ■公有 □私有 □保护 | Result | 机构管理-所属机构 |

算法描述

机构管理-机构列表查询

接口描述：根据各个登录用户获取该用户所属机构及部门列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/dept/list | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptName | [string] | 是 | 机构部门名称 |
| status | [string] | 是 | 机构状态（true正常，false停用） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 返回数据 |
| data>>deptId | [number] | 是 | 机构id |
| data>>parentId | [number] | 是 | 父级机构id |
| data>>deptName | [string] | 是 | 机构名称 |
| data>>deptShort | [null] | 是 | 机构简称 |
| data>>orderNum | [number] | 是 | 显示顺序 |
| data>>ancestors | [string] | 是 | 祖级列表 |
| data>>leader | [string] | 是 | 法人名称 |
| data>>phone | [string] | 是 | 法人联系电话 |
| data>>idCard | [string] | 是 | 法人身份证 |
| data>>address | [string] | 是 | 归属地址 |
| data>>registeredAddress | [string] | 是 | 注册地址 |
| data>>creditCode | [string] | 是 | 社会统一信用码 |
| data>>photoAddressPositive | [string] | 是 | 法人身份证照片地址（正） |
| data>>photoAddressNegative | [string] | 是 | 法人身份证照片地址（反） |
| data>>businessLicense | [string] | 是 | 营业执照地址 |
| data>>status | [boolean] | 是 | 机构状态（true正常，false停用） |
| data>>delFlag | [boolean] | 是 | 删除标志（true正常 false停用） |
| data>>createUserId | [null] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [null] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| code | [number] | 是 | 状态码 |

机构管理-机构列表分页查询

接口描述：根据各个登录用户获取该用户所属机构及部门列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/dept/list/page | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptName | [string] | 是 | 机构部门名称 |
| status | [string] | 是 | 机构状态（true正常，false停用） |
| page | [number] | 是 | 页数 |
| pageSize | [number] | 是 | 页码 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 返回数据 |
| data>>deptId | [number] | 是 | 机构id |
| data>>parentId | [number] | 是 | 父级机构id |
| data>>deptName | [string] | 是 | 机构名称 |
| data>>deptShort | [null] | 是 | 机构简称 |
| data>>orderNum | [number] | 是 | 显示顺序 |
| data>>ancestors | [string] | 是 | 祖级列表 |
| data>>leader | [string] | 是 | 法人名称 |
| data>>phone | [string] | 是 | 法人联系电话 |
| data>>idCard | [string] | 是 | 法人身份证 |
| data>>address | [string] | 是 | 归属地址 |
| data>>registeredAddress | [string] | 是 | 注册地址 |
| data>>creditCode | [string] | 是 | 社会统一信用码 |
| data>>photoAddressPositive | [string] | 是 | 法人身份证照片地址（正） |
| data>>photoAddressNegative | [string] | 是 | 法人身份证照片地址（反） |
| data>>businessLicense | [string] | 是 | 营业执照地址 |
| data>>status | [boolean] | 是 | 机构状态（true正常，false停用） |
| data>>delFlag | [boolean] | 是 | 删除标志（true正常 false停用） |
| data>>createUserId | [null] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [null] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| code | [number] | 是 | 状态码 |

机构管理-新增机构

接口描述：根据机构信息新增机构。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/dept/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| parentId | [number] | 是 | 父级机构id |
| deptName | [string] | 是 | 机构名称 |
| deptShort | [string] | 是 | 机构简称 |
| orderNum | [number] | 是 | 显示顺序 |
| leader | [string] | 是 | 法人名称 |
| phone | [string] | 是 | 法人联系电话 |
| idCard | [string] | 是 | 法人身份证 |
| state | [boolean] | 是 | 机构状态（true正常，false停用） |
| address | [string] | 是 | 归属地址 |
| registeredAddress | [string] | 是 | 注册地址 |
| creditCode | [string] | 是 | 社会统一信用码 |
| photoAddressPositive | [string] | 是 | 法人身份证照片地址（正） |
| photoAddressNegative | [string] | 是 | 法人身份证照片地址（反） |
| businessLicense | [string] | 是 | 营业执照地址 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 新增条数 |
| code | [number] | 是 | 状态码 |

机构管理-修改机构

接口描述：根据机构id修改对应机构信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/dept/edit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| parentId | [number] | 是 | 父级机构id |
| deptId | [number] | 是 | 机构id |
| deptName | [string] | 是 | 机构名称 |
| deptShort | [string] | 是 | 机构简称 |
| orderNum | [number] | 是 | 显示顺序 |
| leader | [string] | 是 | 法人名称 |
| phone | [string] | 是 | 法人联系电话 |
| idCard | [string] | 是 | 法人身份证 |
| state | [boolean] | 是 | 机构状态（true正常，false停用） |
| address | [string] | 是 | 归属地址 |
| registeredAddress | [string] | 是 | 注册地址 |
| creditCode | [string] | 是 | 社会统一信用码 |
| photoAddressPositive | [string] | 是 | 法人身份证照片地址（正） |
| photoAddressNegative | [string] | 是 | 法人身份证照片地址（反） |
| businessLicense | [string] | 是 | 营业执照地址 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

机构管理-机构详情查询

接口描述：点击机构查看对应机构详情信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/dept/info/{deptId} | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 返回数据 |
| data>>deptId | [number] | 是 | 机构id |
| data>>parentId | [number] | 是 | 父级机构id |
| data>>deptName | [string] | 是 | 机构名称 |
| data>>deptShort | [null] | 是 | 机构简称 |
| data>>orderNum | [number] | 是 | 显示顺序 |
| data>>ancestors | [string] | 是 | 祖级列表 |
| data>>leader | [string] | 是 | 法人名称 |
| data>>phone | [string] | 是 | 法人联系电话 |
| data>>idCard | [string] | 是 | 法人身份证 |
| data>>address | [string] | 是 | 归属地址 |
| data>>registeredAddress | [string] | 是 | 注册地址 |
| data>>creditCode | [string] | 是 | 社会统一信用码 |
| data>>photoAddressPositive | [string] | 是 | 法人身份证照片地址（正） |
| data>>photoAddressNegative | [string] | 是 | 法人身份证照片地址（反） |
| data>>businessLicense | [string] | 是 | 营业执照地址 |
| data>>status | [boolean] | 是 | 机构状态（true正常，false停用） |
| data>>delFlag | [boolean] | 是 | 删除标志（true正常 false停用） |
| data>>createUserId | [null] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [null] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| code | [number] | 是 | 状态码 |

机构管理-修改机构状态

接口描述：选中机构停用启用机构状态。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/dept/changeStatus | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构id |
| state | [boolean] | 是 | 机构状态（true正常，false停用） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

机构管理-删除机构

接口描述：根据机构id删除对应机构及关联用户。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/dept/remove | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

机构管理-所属机构

接口描述：根据各个登录用户获取所属机构列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/dept/deptTreeData | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 数据 |
| data>>id | [number] | 是 | 机构部门id |
| data>>label | [string] | 是 | 机构部门名称 |
| data>>children | [array] | 是 | 子级部门 |
| data>>children>>id | [number] | 是 | 机构部门id |
| data>>children>>label | [string] | 是 | 机构部门名称 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysDept

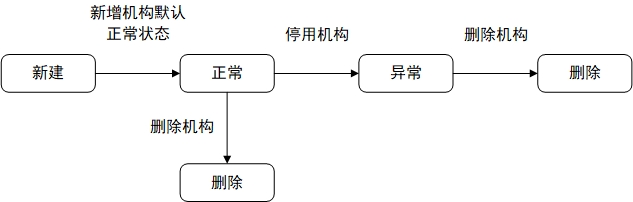
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysDept | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统机构部门的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\sys\po\SysDept | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| deptId | long | | ■公有 □私有 □保护 | | | NNU | | 机构id |
| parentId | long | | ■公有 □私有 □保护 | | | NNI | | 父级机构id |
| deptName | string | | ■公有 □私有 □保护 | | | NN | | 机构名称 |
| deptShort | string | | ■公有 □私有 □保护 | | | NN | | 机构简称 |
| orderNum | int | | ■公有 □私有 □保护 | | |  | | 显示顺序 |
| ancestors | string | | ■公有 □私有 □保护 | | | NNS | | 祖级列表 |
| leader | string | | ■公有 □私有 □保护 | | |  | | 法人名称 |
| phone | string | | ■公有 □私有 □保护 | | |  | | 法人联系电话 |
| idCard | string | | ■公有 □私有 □保护 | | |  | | 法人身份证 |
| address | string | | ■公有 □私有 □保护 | | |  | | 归属地址 |
| registeredAddress | string | | ■公有 □私有 □保护 | | |  | | 注册地址 |
| creditCode | string | | ■公有 □私有 □保护 | | |  | | 社会统一信用码 |
| photoAddressPositive | string | | ■公有 □私有 □保护 | | |  | | 法人身份证照片地址（正） |
| photoAddressNegative | string | | ■公有 □私有 □保护 | | |  | | 法人身份证照片地址（反） |
| businessLicense | string | | ■公有 □私有 □保护 | | |  | | 营业执照地址 |
| status | boolean | | ■公有 □私有 □保护 | | | NN | | 机构状态（true正常，false停用） |
| delFlag | boolean | | ■公有 □私有 □保护 | | | NN | | 删除标志（true正常 false停用） |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_dept | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_dept、sc\_sys\_user\_role、sc\_sys\_role、sc\_resource\_user、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_sys\_user\_role**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from  sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where  u.del\_flag = 1  <if test="userId != null and userId != 0">  AND u.user\_id = #{userId}  </if>  <if test="userName != null and userName != ''">  AND u.user\_name like concat('%', #{userName}, '%')  </if>  <if test="status != null">  AND u.status = #{status}  </if>  <if test="phonenumber != null and phonenumber != ''">  AND u.phonenumber like concat('%', #{phonenumber}, '%')  </if>  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="deptId != null and deptId != 0">  AND (u.dept\_id = #{deptId} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) ))  </if>  <if test="deptIds != null">  <foreach collection="deptIds" item="id" separator="AND">  (u.dept\_id = #{id} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET(#{id},ancestors) ))  </foreach>  </if>  order by create\_time desc  【SQL02】  select user\_id, login\_name from sc\_sys\_user where login\_name=#{loginName} and del\_flag = 1 limit 1  【SQL03】  select user\_id, phonenumber from sc\_sys\_user where phonenumber=#{phonenumber} and del\_flag = 1 limit 1  【SQL04】  select user\_id, email from sc\_sys\_user where email=#{email} and del\_flag = 1 limit 1  【SQL05】  select  user\_id, login\_name  from sc\_sys\_user  <where>  del\_flag = 1  <if test="loginName != null and loginName != '' ">  and login\_name = #{loginName}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL06】  select  user\_id, phonenumber  from sc\_sys\_user  <where>  del\_flag = 1  <if test="phonenumber != null and phonenumber != '' ">  and phonenumber = #{phonenumber}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL07】  select  user\_id, email  from sc\_sys\_user  <where>  del\_flag = 1  <if test="email != null and email != ''">  and email = #{email}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL08】  select  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where u.user\_id = #{userId}  【SQL09】  select  d.dept\_id, d.parent\_id, d.dept\_name,d.dept\_short, d.order\_num, d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.update\_time, d.type  from sc\_sys\_dept d  where  d.del\_flag = 1  <if test="deptId != null and deptId != 0">  AND d.dept\_id = #{deptId}  </if>  <if test="parentId != null">  AND d.parent\_id = #{parentId}  </if>  <if test="deptName != null and deptName != ''">  AND d.dept\_name like concat('%', #{deptName}, '%')  </if>  <if test="status != null">  AND d.status = #{status}  </if>  <if test="untype != null and untype != ''">  AND (d.type = "BUSINESS-SIDE" or d.type = "OPERATION-SIDE")  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if>  order by d.parent\_id, d.order\_num  【SQL10】  select \* from sc\_sys\_dept where status = 1 and del\_flag = 1 and find\_in\_set(#{deptId}, ancestors)  【SQL11】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  where  u.user\_id = #{userId}  and r.status = 1  and r.del\_flag = 1  【SQL12】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, ur.dept\_id, d.dept\_name, r.dept\_node\_all, d.status deptStatus  from sc\_sys\_role r  left join sc\_sys\_role\_dept ur on ur.role\_id = r.role\_id  left join sc\_sys\_dept d on d.dept\_id = ur.dept\_id  where r.del\_flag = 1  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="roleName != null and roleName != ''">  AND r.role\_name like concat('%', #{roleName}, '%')  </if>  <if test="status != null">  AND r.status = #{status}  </if>  <if test="roleKey != null and roleKey != ''">  AND r.role\_key like concat('%', #{roleKey}, '%')  </if>  <if test="type != null and type != ''">  AND r.type = #{type}  </if>  <if test="deptId != null">  AND ur.dept\_id = #{deptId}  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if >  order by r.role\_sort  【SQL13】  SELECT role\_id,menu\_id FROM sc\_sys\_role\_menu WHERE role\_id IN (#{roleIds})  【SQL14】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL15】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL16】  select \* from sc\_sys\_dept  where del\_flag = 1  <choose>  <when test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, ancestors)  </foreach>  or dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </when>  <otherwise>  and find\_in\_set(#{deptId}, ancestors)  </otherwise>  </choose>  <if test="status != null">  AND status = #{status}  </if>  <if test="deptName != null and deptName != ''">  AND dept\_name like concat('%', #{deptName}, '%')  </if>  【SQL17】  select \* from sc\_sys\_dept  where dept\_id in  <foreach collection="deptIds" item="deptId" open="(" separator="," close=")">  #{deptId}  </foreach>  order by parent\_id, order\_num  【SQL18】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where d.dept\_id = #{deptId}  【SQL19】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where dept\_name=#{deptName}  and del\_flag = 1  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL20】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  <where>  <if test="menuName != null and menuName != ''">  AND menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND status = #{status}  </if>  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  <if test="pathUrl != null and pathUrl != ''">  AND path\_url = #{pathUrl}  </if>  <if test="pageUrl != null and pageUrl != ''">  AND page\_url = #{pageUrl}  </if>  </where>  order by parent\_id, order\_num  【SQL21】  select distinct m.menu\_id, m.parent\_id, m.menu\_name, m.path\_url, m.page\_url, m.status, ifnull(m.perms,'') as perms, m.menu\_type, m.icon, m.order\_num, m.create\_time,  m.ancestors,m.`show`  from sc\_sys\_menu m  left join sc\_sys\_role\_menu rm on m.menu\_id = rm.menu\_id  left join sc\_sys\_user\_role ur on rm.role\_id = ur.role\_id  left join sc\_sys\_role ro on ur.role\_id = ro.role\_id  where ur.user\_id = #{userId}  <if test="menuName != null and menuName != ''">  AND m.menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND m.status = #{status}  </if>  <if test="parentId != null">  AND m.parent\_id = #{parentId}  </if>  order by m.parent\_id, m.order\_num  【SQL22】  select \* from sc\_sys\_menu  where menu\_id in  <foreach collection="menuIds" item="menuId" index="index" open="(" separator="," close=")">  #{menuId}  </foreach>  order by parent\_id, order\_num  【SQL23】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_name=#{menuName}  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL24】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_id = #{menuId}  【SQL25】  select count(1) from sc\_sys\_menu where parent\_id = #{menuId}  【SQL26】  select count(1) from sc\_sys\_role\_menu where menu\_id = #{menuId} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_user\_role、sc\_resource\_user、sc\_sys\_oper\_log、sc\_sys\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_dept、sc\_resource、sc\_sys\_menu **>**  **操作：<插入>** |
| SQL语句：  【SQL27】  INSERT INTO sc\_sys\_user ( dept\_id, user\_name, login\_name, user\_type, email, phonenumber, password, status, del\_flag, create\_time, update\_time, remark ) VALUES (#{deptId},#{userName},#{loginName},#{userType},#{email},#{phonenumber} , #{password},#{status},#{delFlag},#{createTime},#{updateTime},#{remark } )  【SQL28】  INSERT INTO sc\_sys\_user\_role ( user\_id, role\_id ) VALUES (#{userId},#{roleId} )  【SQL29】  INSERT INTO sc\_resource\_user ( id, dept\_id, user\_id, cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_time ) VALUES (#{id},#{deptId},#{userId},#{cpu},#{internalMemory},  #{useInternalMemory},#{diskMemory},#{useDiskMemory},#{points},#{createUserId},#{createTime},#{updateTime})  【SQL30】  INSERT INTO sc\_sys\_oper\_log ( title, business\_type, method, oper\_name, oper\_ip, status, oper\_time, cost\_time ) VALUES (#{title},#{businessType},#{method},#{operName},#{operIp},  #{status},#{operTime},#{costTime})  【SQL31】  INSERT INTO sc\_sys\_role ( role\_name, role\_sort, status, del\_flag, create\_time, update\_time, remark, dept\_node\_all ) VALUES ( #{roleName},#{roleSort},#{status},#{delFlag},#{createTime},  #{updateTime},#{remark},#{deptNodeAll })  【SQL32】  INSERT INTO sc\_sys\_role\_dept ( role\_id, dept\_id ) VALUES ( #{roleId},#{deptId} )  【SQL33】  insert into sc\_sys\_role\_menu(role\_id, menu\_id) values (#{roleId},#{menuId})  【SQL34】  INSERT INTO sc\_sys\_dept ( dept\_name, dept\_short, order\_num, ancestors, leader, phone, id\_card, address, registered\_address, photo\_address\_positive, photo\_address\_negative, status, del\_flag, create\_user\_id, create\_time, update\_time, type )(  #{deptName},#{deptShort},#{orderNum},#{ancestors},#{leader},#{phone},#{id\_card},#{address},#{registered\_address},#{photo\_address\_positive},#{ photo\_address\_negative},#{status},#{del\_flag},#{delFlag},#{create\_user\_id},#{create\_time},#{update\_time},#{type})  【SQL35】  INSERT INTO sc\_resource ( id, dept\_id, cpu, use\_cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_user\_id, update\_time ) VALUES ( #{id},#{dept\_id},#{cpu},#{use\_cpu},#{internal\_memory},#{use\_internal\_memory},#{disk\_memory},#{use\_disk\_memory},#{points},#{create\_user\_id},#{create\_time},#{update\_user\_id},#{update\_time })  【SQL36】  INSERT INTO sc\_sys\_menu (menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, perms, icon, create\_time,ancestors,`show`)  VALUES( #{menu\_id},#{menu\_name},#{parent\_id},#{order\_num},#{path\_url},#{ page\_url},#{menu\_type},#{status},#{perms},#{icon},#{create\_time},#{ancestors},#{show}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 用户id |  | | deptId | 机构部门id |  | | userName | 用户名称 |  | | loginName | 用户账号 |  | | offerDeptName | 提供机构id |  | | password | 用户密码 |  | | status | 状态 |  | | delFlag | 删除状态 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_role、sc\_sys\_dept、sc\_sys\_menu**>**  **操作：<修改>** |
| SQL语句：  【SQL37】  UPDATE sc\_sys\_user  SET dept\_id = #{deptId},  user\_name = #{userName},  login\_name = #{loginName},  user\_type = #{userType},  email = #{email},  phonenumber = #{phonenumber},  password = #{password},  status = #{status},  del\_flag = #{delFlag},  create\_time = #{createTime},  update\_time = #{updateTime},  remark = #{remark}  WHERE  user\_id = #{userId}  【SQL38】  update sc\_sys\_user set del\_flag = 0 where user\_id = #{userId}  【SQL39】  UPDATE sc\_sys\_role  SET role\_name=#{roleName},  role\_sort=#{roleSort},  status=#{status},  update\_time=#{updateTime},  remark=#{remark},  dept\_node\_all=#{deptNodeAll}  WHERE  role\_id=#{roleId}  【SQL40】  update sc\_sys\_role set del\_flag = 0 where role\_id = #{roleId}  【SQL41】  UPDATE sc\_sys\_dept SET parent\_id=#{parent\_id}, dept\_name=#{dept\_name}, dept\_short=#{dept\_short}, order\_num=#{order\_num}, ancestors=#{ancestors}, leader=#{leader}, phone=#{phone}, id\_card=#{id\_card}, photo\_address\_positive=#{photo\_address\_positive}, photo\_address\_negative=#{photo\_address\_negative}, business\_license=#{business\_license}, status=#{status}, del\_flag=#{del\_flag}, create\_user\_id=#{create\_user\_id}, create\_time=#{create\_time}, update\_user\_id=#{update\_user\_id}, update\_time=#{update\_time}, type=#{type} WHERE dept\_id=#{dept\_id}  【SQL42】  update sc\_sys\_dept set status = #{status} where find\_in\_set(#{dept\_id}, ancestors)  【SQL43】  update sc\_sys\_user set status = #{status} where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL44】  update sc\_sys\_dept set del\_flag = 0 where dept\_id = #{deptId} or find\_in\_set(#{deptId}, ancestors)  【SQL45】  update sc\_sys\_user set del\_flag = 0 where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL46】  update sc\_sys\_menu  Set menu\_name =#{menu\_name},  parent\_id=#{parent\_id},  order\_num=#{order\_num},  path\_url=#{path\_url},  page\_url=#{page\_url},  menu\_type=#{menu\_type},  status=#{status},  perms=#{perms},  icon=#{icon},  create\_time=#{create\_time},  ancestors=#{ancestors},  `show`=#{show}  where menu\_id = #{menu\_id}  【SQL47】  update sc\_sys\_menu set status = #{status} where find\_in\_set(#{menuId}, ancestors) |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | status | 0 | 1 | 启动状态 | | del\_flag | 1 | 0 | 删除状态 | |  |  |  |  | |
| **程序流程中涉及的数据库表：<**sc\_sys\_user\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_resource**>**  **操作：<删除>** |
| SQL语句：  【SQL48】  delete from sc\_sys\_user\_role where user\_id = #{userId}  【SQL49】  delete from sc\_sys\_user\_role where role\_id = #{roleId}  【SQL50】  delete from sc\_sys\_role\_dept where role\_id= #{roleId}  【SQL51】  delete from sc\_sys\_role\_menu where role\_id=#{roleId}  【SQL52】  delete from sc\_sys\_role\_dept where dept\_id= #{deptId}  【SQL53】  delete from sc\_resource where dept\_id= #{deptId}  【SQL54】  delete from sc\_sys\_menu where menu\_id= #{menuId} |

#### 总体设计方案

##### 设计方案

机构管理用于管理机构，提供新增机构、修改机构、删除机构、停用/启用机构、查看详情功能。

##### 处理流程

1. 查询机构列表



页面调用【6.3.3.1.1.1.2.2.机构管理-机构列表分页查询】接口根据token获取用户信息。判断用户角色，若用户为非超级管理员与运营端机构角色根据当前用户机构id执行【SQL16】查询所有角色集合；若用户为运营端机构角色，先执行【SQL09】查询运营端与业务端机构可查看的所有父级机构id，再根据机构id集合执行【SQL16】查询所有角色集合；若用户为超级关联员则执行【SQL09】查询所有机构部门信息。查询角色集合不为空的情况，封装所有查询角色id集合，执行【SQL17】根据角色id集合查询所有角色信息，再封装到角色列表中返回页面。分别展示机构名称、排序、状态、创建时间。

1. 查询机构详情



页面点击【查看】弹出详情页面，机构详情页面调用【6.3.3.1.1.1.2.5.机构管理-机构详情查询】接口根据机构id执行【SQL18】查询消息详情。

1. 新增机构



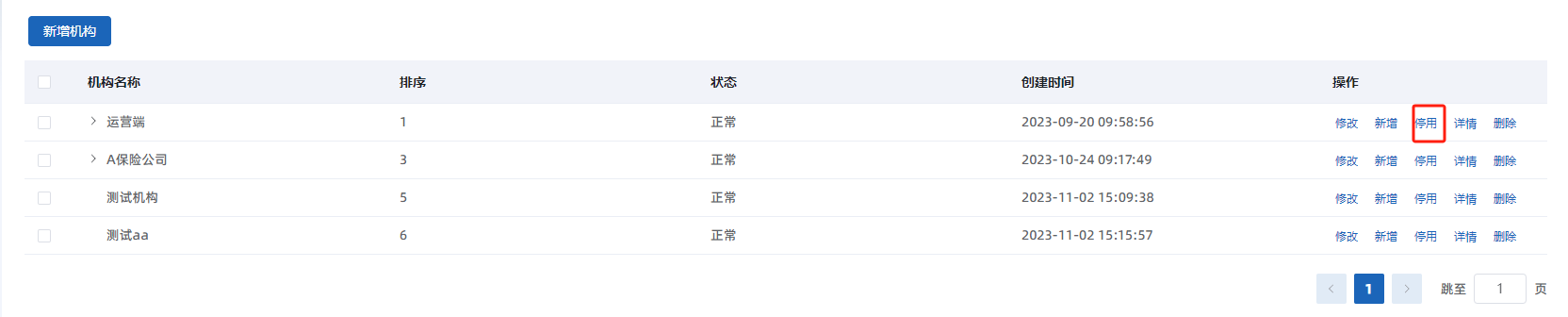
页面点击【新增】弹出新增机构页面，新增页面展示机构配置信息。配置完成后将配置信息作为参数调用【6.3.3.1.1.1.2.3.机构管理-新增机构】接口。接口先执行判重操作，根据机构名称执行【SQL19】查询机构名称判断是否重复。若机构未重复，将机构信息封装执行【SQL34】进行插入机构信息操作，机构新建成功后，默认分配给机构默认资源，执行【SQL35】插入系统资源表。

1. 修改机构



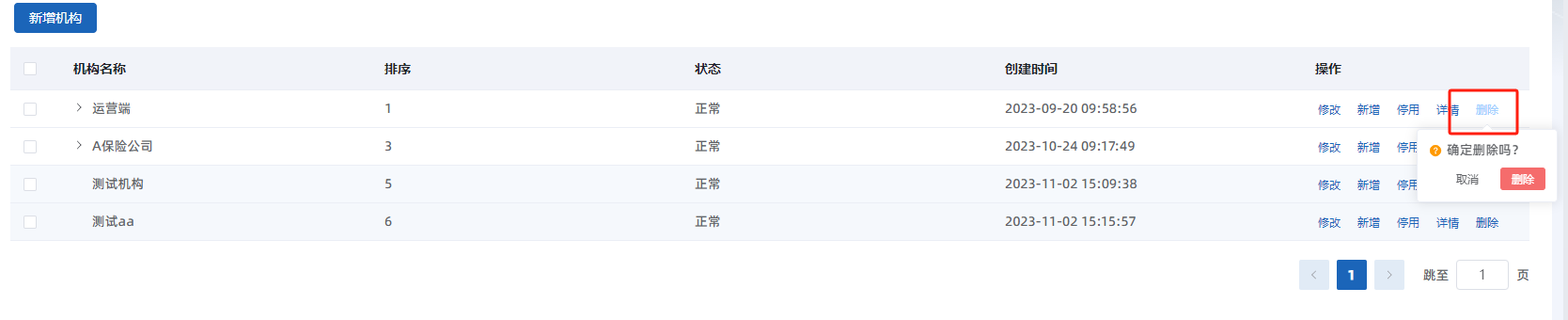
页面点击【修改】弹出修改机构页面，修改页面展示机构配置信息。配置完成后将配置信息作为参数调用【6.3.3.1.1.1.2.4.机构管理-修改机构】接口。接口先判断是否可修改，本机构不可作为上级机构再执行判重操作，根据机构名称执行【SQL19】查询机构名称判断是否重复。机构未重复，先执行【SQL18】分别查询新的父级机构和旧的机构信息，再组合比对其祖级列表修改变化，将修改的配置信息执行【SQL41】修改机构表，最后把原机构配置影响的机构部门执行【SQL42】进行修改。

1. 停用/启用机构



页面选择机构，点击【停用/启用】调用【6.3.3.1.1.1.2.6.机构管理-修改机构状态】接口根据机构id执行【SQL41】进行信息修改，根据机构id执行【SQL43】修改机构及其子部门下所有用户状态。

1. 删除机构



页面选择机构，点击【删除】经过确认交互后调用【6.3.3.1.1.1.2.7.机构管理-删除机构】接口根据机构id执行【SQL44】逻辑删除。根据机构id执行【SQL52】将机构角色关联信息删除，同时执行【SQL53】删除机构资源释放资源，执行【SQL45】逻辑删除该机构及其子部门下所有用户。

### 菜单管理

#### 类设计

##### 控制类

###### SysMenuController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysMenuController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成菜单管理的相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\sysController\SysMenuController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| add | session[:SysMenu] | SysMenu | ■公有 □私有 □保护 | Result | 菜单管理-新增菜单 |
| edit | session[:SysMenu] | SysMenu | ■公有 □私有 □保护 | Result | 菜单管理-修改菜单/修改菜单状态 |
| list | session[:SysMenuVo] | SysMenuVo | ■公有 □私有 □保护 | Result | 菜单管理-菜单列表查询 |
| treeselect | session[:SysMenuVo] | SysMenuVo | ■公有 □私有 □保护 | Result | 菜单管理-获取菜单下拉树列表 |
| remove | session[:menuId] | long | ■公有 □私有 □保护 | Result | 菜单管理-删除菜单 |

算法描述

菜单管理-新增菜单

接口描述：根据菜单信息新增对菜单。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/menu/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| menuName | [string] | 是 | 菜单名称 |
| parentId | [number] | 是 | 父级菜单id |
| orderNum | [number] | 是 | 显示顺序 |
| menuType | [string] | 是 | 菜单类型（M端口 C菜单 F按钮） |
| status | [boolean] | 是 | 菜单状态（true正常 false停用） |
| perms | [string] | 是 | 权限标识 |
| pathUrl | [string] | 是 | 路由地址 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 新增条数 |
| code | [number] | 是 | 状态码 |

菜单管理-修改菜单/修改菜单状态

接口描述：根据填写菜单信息修改菜单。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/menu/edit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| menuName | [string] | 是 | 菜单名称 |
| parentId | [number] | 是 | 父级菜单id |
| orderNum | [number] | 是 | 显示顺序 |
| menuType | [string] | 是 | 菜单类型（M端口 C菜单 F按钮） |
| status | [boolean] | 是 | 菜单状态（true正常 false停用） |
| perms | [string] | 是 | 权限标识 |
| pathUrl | [string] | 是 | 路由地址 |
| menuId | [number] | 是 | 菜单id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

菜单管理-菜单列表查询

接口描述：超级管理员角色可查询全局的菜单信息列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/menu/list |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| menuName | [string] | 是 | 菜单名称 |
| status | [string] | 是 | 菜单状态（true正常 false停用） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 信息 |
| data>>menuId | [number] | 是 | 菜单id |
| data>>menuName | [string] | 是 | 菜单名称 |
| data>>parentId | [number] | 是 | 上级菜单id |
| data>>orderNum | [string] | 是 | 显示顺序 |
| data>>pathUrl | [string] | 是 | 路由地址 |
| data>>menuType | [string] | 是 | 菜单类型（M端口 C菜单 F按钮） |
| data>>status | [boolean] | 是 | 菜单状态（true正常 false停用） |
| data>>perms | [string] | 是 | 权限标识 |
| data>>icon | [string] | 是 | 菜单图标 |
| data>>createUserId | [null] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [null] | 是 | 更新用户id |
| data>>updateTime | [null] | 是 | 更新时间 |
| data>>remark | [null] | 是 | 备注 |
| data>>children | [array] | 是 | 子级 |
| code | [number] | 是 | 状态码 |

菜单管理-获取菜单下拉树列表

接口描述：获取菜单下拉框所展示的菜单权限下拉树列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/menu/treeselect |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| menuName | [string] | 是 | 菜单名称 |
| status | [string] | 是 | 菜单状态（true正常 false停用） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [array] | 是 | 信息 |
| data>>id | [number] | 是 | 菜单id |
| data>>label | [string] | 是 | 菜单名称 |
| data>>children | [array] | 是 | 子级 |
| code | [number] | 是 | 请求返回码 |

菜单管理-删除菜单

接口描述：根据所选菜单删除菜单，若菜单被分配不可删除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /system/menu/remove/{menuId} |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 |  |  |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 请求返回信息 |
| data | [number] | 是 | 修改条数 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### SysMenu

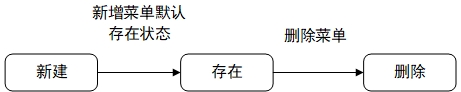
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysMenu | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统菜单的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\sys\po\SysMenu | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| menuId | long | | ■公有 □私有 □保护 | | | NNU | | 菜单id |
| menuName | string | | ■公有 □私有 □保护 | | | NNS | | 菜单名称 |
| parentId | long | | ■公有 □私有 □保护 | | | NNI | | 上级菜单id |
| orderNum | int | | ■公有 □私有 □保护 | | | NN | | 显示顺序 |
| pathUrl | string | | ■公有 □私有 □保护 | | |  | | 路由地址 |
| menuType | string | | ■公有 □私有 □保护 | | | NNS | | 菜单类型（M端口 C菜单 F按钮） |
| status | boolean | | ■公有 □私有 □保护 | | | NNI | | 菜单状态（true正常 false停用） |
| perms | string | | ■公有 □私有 □保护 | | |  | | 权限标识 |
| icon | string | | ■公有 □私有 □保护 | | |  | | 菜单图标 |
| createUserId | long | | □公有 □私有 ■保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| remark | string | | ■公有 □私有 □保护 | | |  | | 备注 |
| children | array | | ■公有 □私有 □保护 | | |  | | 子级 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_menu | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_dept、sc\_sys\_user\_role、sc\_sys\_role、sc\_resource\_user、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_sys\_user\_role**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from  sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where  u.del\_flag = 1  <if test="userId != null and userId != 0">  AND u.user\_id = #{userId}  </if>  <if test="userName != null and userName != ''">  AND u.user\_name like concat('%', #{userName}, '%')  </if>  <if test="status != null">  AND u.status = #{status}  </if>  <if test="phonenumber != null and phonenumber != ''">  AND u.phonenumber like concat('%', #{phonenumber}, '%')  </if>  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="deptId != null and deptId != 0">  AND (u.dept\_id = #{deptId} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) ))  </if>  <if test="deptIds != null">  <foreach collection="deptIds" item="id" separator="AND">  (u.dept\_id = #{id} OR u.dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET(#{id},ancestors) ))  </foreach>  </if>  order by create\_time desc  【SQL02】  select user\_id, login\_name from sc\_sys\_user where login\_name=#{loginName} and del\_flag = 1 limit 1  【SQL03】  select user\_id, phonenumber from sc\_sys\_user where phonenumber=#{phonenumber} and del\_flag = 1 limit 1  【SQL04】  select user\_id, email from sc\_sys\_user where email=#{email} and del\_flag = 1 limit 1  【SQL05】  select  user\_id, login\_name  from sc\_sys\_user  <where>  del\_flag = 1  <if test="loginName != null and loginName != '' ">  and login\_name = #{loginName}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL06】  select  user\_id, phonenumber  from sc\_sys\_user  <where>  del\_flag = 1  <if test="phonenumber != null and phonenumber != '' ">  and phonenumber = #{phonenumber}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL07】  select  user\_id, email  from sc\_sys\_user  <where>  del\_flag = 1  <if test="email != null and email != ''">  and email = #{email}  </if>  <if test="userId != null and userId != ''">  and user\_id != #{userId}  </if>  limit 1  </where>  【SQL08】  select  u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password, u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark, d.dept\_name, r.role\_name, r.role\_id, sru.cpu, sru.internal\_memory, sru.disk\_memory,r.status roleStatus  from sc\_sys\_user u  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role r on r.role\_id = ur.role\_id  left join sc\_resource\_user sru on sru.user\_id = u.user\_id  where u.user\_id = #{userId}  【SQL09】  select  d.dept\_id, d.parent\_id, d.dept\_name,d.dept\_short, d.order\_num, d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.update\_time, d.type  from sc\_sys\_dept d  where  d.del\_flag = 1  <if test="deptId != null and deptId != 0">  AND d.dept\_id = #{deptId}  </if>  <if test="parentId != null">  AND d.parent\_id = #{parentId}  </if>  <if test="deptName != null and deptName != ''">  AND d.dept\_name like concat('%', #{deptName}, '%')  </if>  <if test="status != null">  AND d.status = #{status}  </if>  <if test="untype != null and untype != ''">  AND (d.type = "BUSINESS-SIDE" or d.type = "OPERATION-SIDE")  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if>  order by d.parent\_id, d.order\_num  【SQL10】  select \* from sc\_sys\_dept where status = 1 and del\_flag = 1 and find\_in\_set(#{deptId}, ancestors)  【SQL11】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  where  u.user\_id = #{userId}  and r.status = 1  and r.del\_flag = 1  【SQL12】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, ur.dept\_id, d.dept\_name, r.dept\_node\_all, d.status deptStatus  from sc\_sys\_role r  left join sc\_sys\_role\_dept ur on ur.role\_id = r.role\_id  left join sc\_sys\_dept d on d.dept\_id = ur.dept\_id  where r.del\_flag = 1  <if test="roleId != null">  AND r.role\_id = #{roleId}  </if>  <if test="roleName != null and roleName != ''">  AND r.role\_name like concat('%', #{roleName}, '%')  </if>  <if test="status != null">  AND r.status = #{status}  </if>  <if test="roleKey != null and roleKey != ''">  AND r.role\_key like concat('%', #{roleKey}, '%')  </if>  <if test="type != null and type != ''">  AND r.type = #{type}  </if>  <if test="deptId != null">  AND ur.dept\_id = #{deptId}  </if>  <if test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, d.ancestors)  </foreach>  or d.dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </if >  order by r.role\_sort  【SQL13】  SELECT role\_id,menu\_id FROM sc\_sys\_role\_menu WHERE role\_id IN (#{roleIds})  【SQL14】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL15】  select  r.role\_id, r.role\_name, r.role\_key, r.role\_sort, r.type, r.create\_user\_id, r.update\_user\_id, r.update\_time,r.status, r.del\_flag, r.create\_time, r.remark, r.dept\_node\_all  from sc\_sys\_role r  left join sc\_sys\_user\_role ur on ur.role\_id = r.role\_id  left join sc\_sys\_user u on u.user\_id = ur.user\_id  left join sc\_sys\_dept d on u.dept\_id = d.dept\_id  where r.role\_name=#{roleName} and r.del\_flag = 1 limit 1  【SQL16】  select \* from sc\_sys\_dept  where del\_flag = 1  <choose>  <when test="deptIds != null">  and (  <foreach collection="deptIds" item="id" open="(" separator="or" close=")">  find\_in\_set(#{id}, ancestors)  </foreach>  or dept\_id in  <foreach collection="deptIds" item="id" open="(" separator="," close=")">  #{id}  </foreach>  )  </when>  <otherwise>  and find\_in\_set(#{deptId}, ancestors)  </otherwise>  </choose>  <if test="status != null">  AND status = #{status}  </if>  <if test="deptName != null and deptName != ''">  AND dept\_name like concat('%', #{deptName}, '%')  </if>  【SQL17】  select \* from sc\_sys\_dept  where dept\_id in  <foreach collection="deptIds" item="deptId" open="(" separator="," close=")">  #{deptId}  </foreach>  order by parent\_id, order\_num  【SQL18】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where d.dept\_id = #{deptId}  【SQL19】  select  d.dept\_id, d.parent\_id, d.dept\_name, d.dept\_short, d.order\_num,d.ancestors, d.leader, d.phone, d.id\_card, d.address, d.registered\_address, d.credit\_code ,d.photo\_address\_positive, d.photo\_address\_negative, d.business\_license, d.status, d.del\_flag, d.create\_user\_id, d.create\_time, d.update\_user\_id, d.type , d.update\_time,(select dept\_name from sc\_sys\_dept where dept\_id = d.parent\_id) parentName  from sc\_sys\_dept d  where dept\_name=#{deptName}  and del\_flag = 1  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL20】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  <where>  <if test="menuName != null and menuName != ''">  AND menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND status = #{status}  </if>  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  <if test="pathUrl != null and pathUrl != ''">  AND path\_url = #{pathUrl}  </if>  <if test="pageUrl != null and pageUrl != ''">  AND page\_url = #{pageUrl}  </if>  </where>  order by parent\_id, order\_num  【SQL21】  select distinct m.menu\_id, m.parent\_id, m.menu\_name, m.path\_url, m.page\_url, m.status, ifnull(m.perms,'') as perms, m.menu\_type, m.icon, m.order\_num, m.create\_time,  m.ancestors,m.`show`  from sc\_sys\_menu m  left join sc\_sys\_role\_menu rm on m.menu\_id = rm.menu\_id  left join sc\_sys\_user\_role ur on rm.role\_id = ur.role\_id  left join sc\_sys\_role ro on ur.role\_id = ro.role\_id  where ur.user\_id = #{userId}  <if test="menuName != null and menuName != ''">  AND m.menu\_name like concat('%', #{menuName}, '%')  </if>  <if test="status != null">  AND m.status = #{status}  </if>  <if test="parentId != null">  AND m.parent\_id = #{parentId}  </if>  order by m.parent\_id, m.order\_num  【SQL22】  select \* from sc\_sys\_menu  where menu\_id in  <foreach collection="menuIds" item="menuId" index="index" open="(" separator="," close=")">  #{menuId}  </foreach>  order by parent\_id, order\_num  【SQL23】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_name=#{menuName}  <if test="parentId != null">  AND parent\_id = #{parentId}  </if>  limit 1  【SQL24】  select  menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, ifnull(perms,'') as perms, icon, create\_time,ancestors,`show`  from sc\_sys\_menu  where menu\_id = #{menuId}  【SQL25】  select count(1) from sc\_sys\_menu where parent\_id = #{menuId}  【SQL26】  select count(1) from sc\_sys\_role\_menu where menu\_id = #{menuId} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_user\_role、sc\_resource\_user、sc\_sys\_oper\_log、sc\_sys\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_dept、sc\_resource、sc\_sys\_menu **>**  **操作：<插入>** |
| SQL语句：  【SQL27】  INSERT INTO sc\_sys\_user ( dept\_id, user\_name, login\_name, user\_type, email, phonenumber, password, status, del\_flag, create\_time, update\_time, remark ) VALUES (#{deptId},#{userName},#{loginName},#{userType},#{email},#{phonenumber} , #{password},#{status},#{delFlag},#{createTime},#{updateTime},#{remark } )  【SQL28】  INSERT INTO sc\_sys\_user\_role ( user\_id, role\_id ) VALUES (#{userId},#{roleId} )  【SQL29】  INSERT INTO sc\_resource\_user ( id, dept\_id, user\_id, cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_time ) VALUES (#{id},#{deptId},#{userId},#{cpu},#{internalMemory},  #{useInternalMemory},#{diskMemory},#{useDiskMemory},#{points},#{createUserId},#{createTime},#{updateTime})  【SQL30】  INSERT INTO sc\_sys\_oper\_log ( title, business\_type, method, oper\_name, oper\_ip, status, oper\_time, cost\_time ) VALUES (#{title},#{businessType},#{method},#{operName},#{operIp},  #{status},#{operTime},#{costTime})  【SQL31】  INSERT INTO sc\_sys\_role ( role\_name, role\_sort, status, del\_flag, create\_time, update\_time, remark, dept\_node\_all ) VALUES ( #{roleName},#{roleSort},#{status},#{delFlag},#{createTime},  #{updateTime},#{remark},#{deptNodeAll })  【SQL32】  INSERT INTO sc\_sys\_role\_dept ( role\_id, dept\_id ) VALUES ( #{roleId},#{deptId} )  【SQL33】  insert into sc\_sys\_role\_menu(role\_id, menu\_id) values (#{roleId},#{menuId})  【SQL34】  INSERT INTO sc\_sys\_dept ( dept\_name, dept\_short, order\_num, ancestors, leader, phone, id\_card, address, registered\_address, photo\_address\_positive, photo\_address\_negative, status, del\_flag, create\_user\_id, create\_time, update\_time, type )(  #{deptName},#{deptShort},#{orderNum},#{ancestors},#{leader},#{phone},#{id\_card},#{address},#{registered\_address},#{photo\_address\_positive},#{ photo\_address\_negative},#{status},#{del\_flag},#{delFlag},#{create\_user\_id},#{create\_time},#{update\_time},#{type})  【SQL35】  INSERT INTO sc\_resource ( id, dept\_id, cpu, use\_cpu, internal\_memory, use\_internal\_memory, disk\_memory, use\_disk\_memory, points, create\_user\_id, create\_time, update\_user\_id, update\_time ) VALUES ( #{id},#{dept\_id},#{cpu},#{use\_cpu},#{internal\_memory},#{use\_internal\_memory},#{disk\_memory},#{use\_disk\_memory},#{points},#{create\_user\_id},#{create\_time},#{update\_user\_id},#{update\_time })  【SQL36】  INSERT INTO sc\_sys\_menu (menu\_id, menu\_name, parent\_id, order\_num, path\_url, page\_url, menu\_type, status, perms, icon, create\_time,ancestors,`show`)  VALUES( #{menu\_id},#{menu\_name},#{parent\_id},#{order\_num},#{path\_url},#{ page\_url},#{menu\_type},#{status},#{perms},#{icon},#{create\_time},#{ancestors},#{show}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 用户id |  | | deptId | 机构部门id |  | | userName | 用户名称 |  | | loginName | 用户账号 |  | | offerDeptName | 提供机构id |  | | password | 用户密码 |  | | status | 状态 |  | | delFlag | 删除状态 |  | |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_user、sc\_sys\_role、sc\_sys\_dept、sc\_sys\_menu**>**  **操作：<修改>** |
| SQL语句：  【SQL37】  UPDATE sc\_sys\_user  SET dept\_id = #{deptId},  user\_name = #{userName},  login\_name = #{loginName},  user\_type = #{userType},  email = #{email},  phonenumber = #{phonenumber},  password = #{password},  status = #{status},  del\_flag = #{delFlag},  create\_time = #{createTime},  update\_time = #{updateTime},  remark = #{remark}  WHERE  user\_id = #{userId}  【SQL38】  update sc\_sys\_user set del\_flag = 0 where user\_id = #{userId}  【SQL39】  UPDATE sc\_sys\_role  SET role\_name=#{roleName},  role\_sort=#{roleSort},  status=#{status},  update\_time=#{updateTime},  remark=#{remark},  dept\_node\_all=#{deptNodeAll}  WHERE  role\_id=#{roleId}  【SQL40】  update sc\_sys\_role set del\_flag = 0 where role\_id = #{roleId}  【SQL41】  UPDATE sc\_sys\_dept SET parent\_id=#{parent\_id}, dept\_name=#{dept\_name}, dept\_short=#{dept\_short}, order\_num=#{order\_num}, ancestors=#{ancestors}, leader=#{leader}, phone=#{phone}, id\_card=#{id\_card}, photo\_address\_positive=#{photo\_address\_positive}, photo\_address\_negative=#{photo\_address\_negative}, business\_license=#{business\_license}, status=#{status}, del\_flag=#{del\_flag}, create\_user\_id=#{create\_user\_id}, create\_time=#{create\_time}, update\_user\_id=#{update\_user\_id}, update\_time=#{update\_time}, type=#{type} WHERE dept\_id=#{dept\_id}  【SQL42】  update sc\_sys\_dept set status = #{status} where find\_in\_set(#{dept\_id}, ancestors)  【SQL43】  update sc\_sys\_user set status = #{status} where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL44】  update sc\_sys\_dept set del\_flag = 0 where dept\_id = #{deptId} or find\_in\_set(#{deptId}, ancestors)  【SQL45】  update sc\_sys\_user set del\_flag = 0 where dept\_id = #{deptId} OR dept\_id IN ( SELECT t.dept\_id FROM sc\_sys\_dept t WHERE FIND\_IN\_SET (#{deptId},ancestors) )  【SQL46】  update sc\_sys\_menu  Set menu\_name =#{menu\_name},  parent\_id=#{parent\_id},  order\_num=#{order\_num},  path\_url=#{path\_url},  page\_url=#{page\_url},  menu\_type=#{menu\_type},  status=#{status},  perms=#{perms},  icon=#{icon},  create\_time=#{create\_time},  ancestors=#{ancestors},  `show`=#{show}  where menu\_id = #{menu\_id}  【SQL47】  update sc\_sys\_menu set status = #{status} where find\_in\_set(#{menuId}, ancestors) |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | status | 0 | 1 | 启动状态 | | del\_flag | 1 | 0 | 删除状态 | |  |  |  |  | |
| **程序流程中涉及的数据库表：<**sc\_sys\_user\_role、sc\_sys\_role\_dept、sc\_sys\_role\_menu、sc\_sys\_role\_dept、sc\_resource**>**  **操作：<删除>** |
| SQL语句：  【SQL48】  delete from sc\_sys\_user\_role where user\_id = #{userId}  【SQL49】  delete from sc\_sys\_user\_role where role\_id = #{roleId}  【SQL50】  delete from sc\_sys\_role\_dept where role\_id= #{roleId}  【SQL51】  delete from sc\_sys\_role\_menu where role\_id=#{roleId}  【SQL52】  delete from sc\_sys\_role\_dept where dept\_id= #{deptId}  【SQL53】  delete from sc\_resource where dept\_id= #{deptId}  【SQL54】  delete from sc\_sys\_menu where menu\_id= #{menuId} |

#### 总体设计方案

##### 设计方案

菜单管理用于管理整个系统展示菜单，提供新增菜单、修改菜单、删除菜单、停用/启用菜单功能。

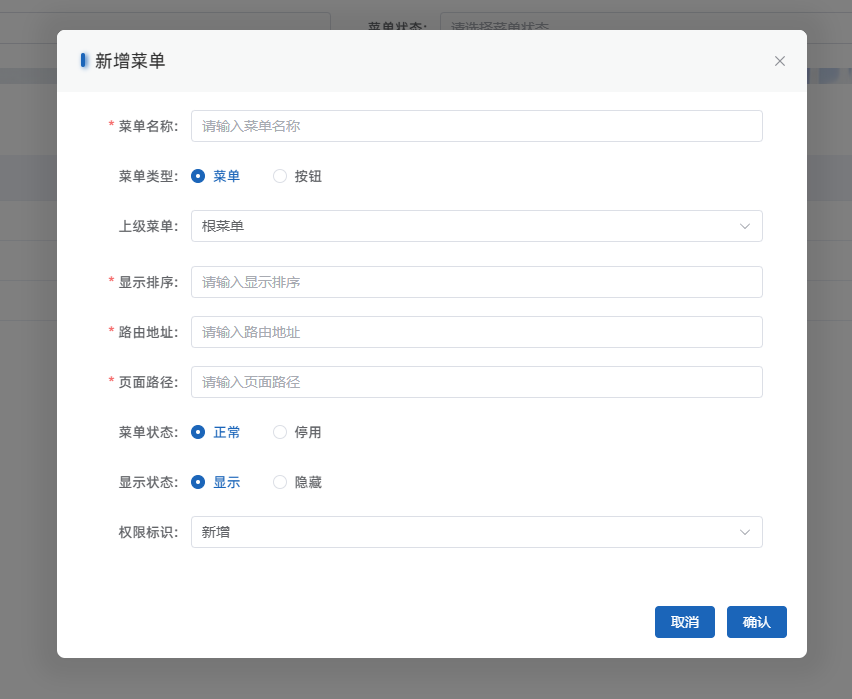
##### 处理流程

1. 查询菜单列表



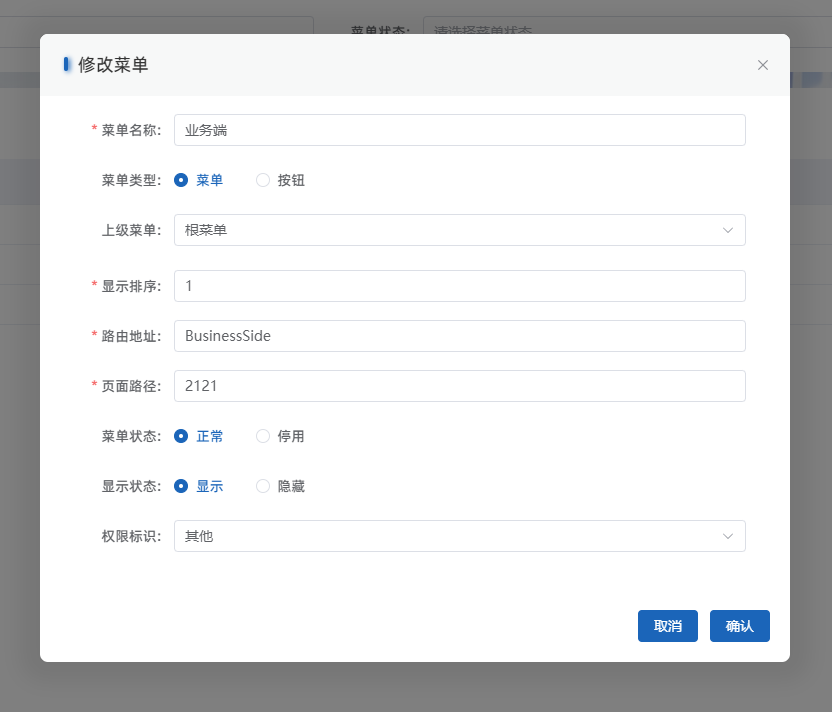
页面调用【6.3.4.1.1.1.2.3.菜单管理-菜单列表查询】接口根据token获取用户信息，根据当前用户机构判断获取用户菜单。若是超级管理员或运营端则执行【SQL20】查询当前用户权限的所有菜单，若是其他端用户则执行【SQL21】查询当前用户权限的所有菜单。查询菜单集合不为空的情况，封装所有查询菜单id集合，执行【SQL22】根据菜单id集合查询所有菜单信息，再封装到菜单列表中返回页面。分别展示菜单名称、排序、路由地址、状态、创建时间。

1. 新增菜单



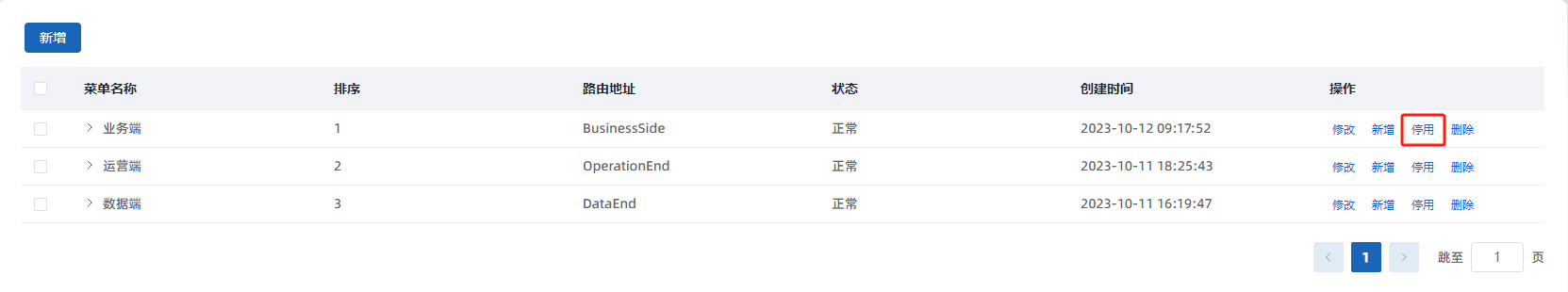
页面点击【新增】弹出新增菜单页面，新增页面展示菜单配置信息。配置完成后将配置信息作为参数调用【6.3.4.1.1.1.2.1.菜单管理-新增菜单】接口。接口先执行判重操作，分别根据菜单名称执行【SQL23】查询菜单名称判断是否重复，根据菜单路由和菜单页面路径执行【SQL20】查询菜单路由和菜单页面路径判断是否重复。若菜单未重复，将菜单信息封装执行【SQL36】进行插入菜单信息操作。

1. 修改菜单



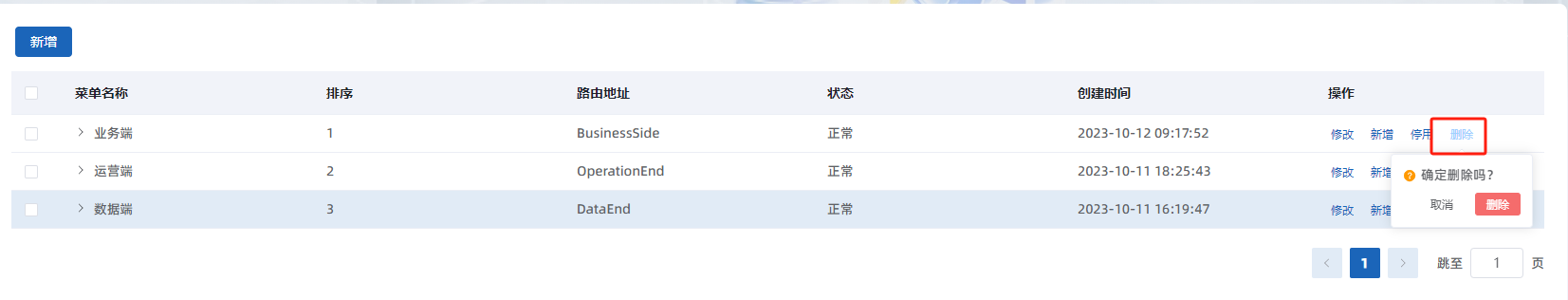
页面点击【修改】弹出修改菜单页面，修改页面展示用户配置信息。配置完成后将配置信息作为参数调用【6.3.4.1.1.1.2.2.菜单管理-修改菜单/修改菜单状态】接口。接口先执行判重操作，分别根据菜单名称执行【SQL23】查询菜单名称判断是否重复，根据菜单路由和菜单页面路径执行【SQL20】查询菜单路由和菜单页面路径判断是否重复。菜单未重复，执行【SQL24】分别查询新的父级菜单和旧的菜单信息，再组合比对其祖级列表修改变化，将修改的配置信息执行【SQL46】修改菜单，最后把原菜单配置影响的菜单执行【SQL47】进行修改。

1. 停用/启用菜单



页面选择用户，点击【停用/启用】调用【6.3.4.1.1.1.2.2.菜单管理-修改菜单/修改菜单状态】执行相同逻辑。

1. 删除菜单



页面选择菜单，点击【删除】经过确认交互后调用【6.3.4.1.1.1.2.5.菜单管理-删除菜单】接口先判断是否可修改操作，根据菜单id执行【SQL25】与【SQL-6】查询是否可操作。菜单可修改情况下，根据菜单id执行【SQL48】将菜单删除。

### 资源管理

#### 类设计

##### 控制类

###### ResourceController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ResourceController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、文件传输、数据更新相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/mhsdip/controller/resourceController/ResourceController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| get | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 获取某个机构资源 |
| list | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 机构资源列表 |
| update | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 更新机构资源 |
| Insert | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 插入机构资源 |
| all | 无 | 无 | ■公有 □私有 □保护 | Result | 系统资源的使用量 |
| remaining | session[:deptId] | long | ■公有 □私有 □保护 | Result | 系统资源的剩余使用量 |

算法描述

资源管理-机构资源使用详情

接口描述：运营端资源管理-某一个机构拥有的资源查看。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/get | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| depeId | [number] | 否 | 机构id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>id | [number] | 是 | 数据列表 |
| data>>deptId | [number] | 是 | 机构id |
| data>>cpu | [number] | 是 | cpu |
| data>>useCpu | [number] | 是 | cpu使用量 |
| data>>internalMemory | [number] | 是 | 内存 |
| data>>useInternalMemory | [number] | 是 | 已用内存 |
| data>>diskMemory | [number] | 是 | 磁盘存储 |
| data>>useDiskMemory | [number] | 是 | 已用磁盘存储 |
| data>>points | [number] | 是 | 积分 |
| data>>createUserId | [number] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [number] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>deptName | [string] | 是 | 机构名称 |
| data>>jsonObjectList | [array] | 是 | 项目资源使用量 |
| data>>jsonObjectList>>percentage | [number] | 是 | 百分比 |
| data>>jsonObjectList>>num | [number] | 是 | 数值 |
| data>>jsonObjectList>>type | [string] | 是 | 类型 |
| code | [number] | 是 | 状态码 |

资源管理-机构资源使用列表

接口描述：运营端资源管理-机构资源列表查看，支持机构名称查询。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/list | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| pageSize | [number] | 是 | 页数 |
| page | [number] | 是 | 页码 |
| deptName | [string] | 否 | 机构名称 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回状态 |
| data | [object] | 是 | 返回数据 |
| data>>page | [number] | 是 | 页码 |
| data>>pageSize | [number] | 是 | 页数 |
| data>>total | [number] | 是 | 总数据量 |
| data>>list | [array] | 是 | 数据集 |
| data>>list>>id | [number] | 是 | 资源id |
| data>>list>>deptId | [number] | 是 | 机构id |
| data>>list>>cpu | [number] | 是 | cpu核数 |
| data>>list>>useCpu | [number] | 是 | 已用cpu核数 |
| data>>list>>internalMemory | [number] | 是 | 内存（mb） |
| data>>list>>useInternalMemory | [number] | 是 | 已用内存（mb） |
| data>>list>>diskMemory | [number] | 是 | 磁盘存储（mb） |
| data>>list>>useDiskMemory | [number] | 是 | 已用磁盘存储（mb） |
| data>>list>>points | [number] | 是 | 积分 |
| data>>list>>createUserId | [number] | 是 | 创建用户id |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新用户id |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>list>>deptName | [string] | 是 | 机构名称 |
| code | [number] | 是 | 状态码 |

资源管理-机构资源更新

接口描述：运营端资源管理-某一个机构拥有的资源进行资源更新。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/update | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构id |
| resourceQuotaVoList | [object] | 否 | 项目资源配额 |
| resourceQuotaVoList>>useDiskMemory | [number] | 否 | 已用磁盘存储（mb） |
| resourceQuotaVoList>>useInternalMemory | [number] | 否 | 已用内存（mb） |
| resourceQuotaVoList>>useCpu | [number] | 否 | 已用cpu核数 |
| points | [number] | 否 | 积分 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

机构管理-机构资源插入

接口描述：机构创建时插入一个机构资源。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/insert | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [number] | 是 | 机构id |
| diskMemory | [number] | 否 | 磁盘存储 |
| internalMemory | [number] | 否 | 内存 |
| cpu | [number] | 否 | cpu核数 |
| points | [number] | 否 | 积分 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

资源管理-系统资源的使用量

接口描述：运营端资源管理-系统整体资源的使用情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Get | /resource/all | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 |  |  |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [array] | 是 | 数据 |
| data>>percentage | [number] | 是 | 百分比 |
| data>>num | [number] | 是 | 数量 |
| data>>type | [string] | 是 | 类型 |
| code | [number] | 是 | 状态码 |

资源管理-系统资源剩余情况

接口描述：运营端资源管理-系统资源剩余情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Get | /resource/remaining/{deptId} | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| depeId | [number] | 否 | 机构id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>diskMemory | [number] | 是 | 磁盘空间 |
| data>>cpu | [number] | 是 | Cpu |
| data>>internalMemory | [number] | 是 | 内存空间 |
| code | [number] | 是 | 状态码 |

资源管理-系统资源申请

接口描述：机构资源申请。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /resource/insert/approve | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceResourceContent | [object] | 是 | 资源内容 |
| relevanceResourceContent>>cpu | [number] | 是 |  |
| relevanceResourceContent>>internalMemory | [number] | 是 |  |
| relevanceResourceContent>>diskMemory | [number] | 是 |  |
| approveDescription | [string] | 是 | 申请描述 |
| points | [number] | 是 | 积分 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

###### ResourceProjectController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ResourceProjectController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、文件传输、数据更新相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/mhsdip/controller/resourceController/ResourceProjectController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| get | session[:ResourceProjectVo] | ResourceProjectVo | ■公有 □私有 □保护 | Result | 获取某个机构资源 |
| list | session[:ResourceProjectVo] | ResourceProjectVo | ■公有 □私有 □保护 | Result | 机构资源列表 |
| update | session[:ResourceProjectVo] | ResourceProjectVo | ■公有 □私有 □保护 | Result | 更新机构资源 |
| Insert | session[:ResourceProjectVo] | ResourceProjectVo | ■公有 □私有 □保护 | Result | 插入机构资源 |
| all | 无 | 无 | ■公有 □私有 □保护 | Result | 系统资源的使用量 |
| remaining | session[:ResourceProjectVo] | ResourceProjectVo | ■公有 □私有 □保护 | Result | 系统资源的剩余使用量 |
| insertApprove | session[:SysApproveVo] | SysApproveVo | ■公有 □私有 □保护 | Result | 资源申请 |

算法描述

资源管理-机构项目资源使用详情

接口描述：业务端资源管理-某一个机构拥有的项目资源查看。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Get | /resource/project/get/{projectId} | PathVariable |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| projectId | [number] | 否 | 项目id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>id | [number] | 是 | Id |
| data>>deptId | [number] | 是 | 机构id |
| data>>cpu | [number] | 是 | cpu |
| data>>useCpu | [number] | 是 | cpu使用量 |
| data>>internalMemory | [number] | 是 | 内存 |
| data>>useInternalMemory | [number] | 是 | 已用内存 |
| data>>diskMemory | [number] | 是 | 磁盘存储 |
| data>>useDiskMemory | [number] | 是 | 已用磁盘存储 |
| data>>points | [number] | 是 | 积分 |
| data>>createUserId | [number] | 是 | 创建用户id |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [number] | 是 | 更新用户id |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>deptName | [string] | 是 | 机构名称 |
| data>>jsonObjectList | [array] | 是 | 项目资源使用量 |
| data>>jsonObjectList>>percentage | [number] | 是 | 百分比 |
| data>>jsonObjectList>>num | [number] | 是 | 数值 |
| data>>jsonObjectList>>type | [string] | 是 | 类型 |
| code | [number] | 是 | 状态码 |

资源管理-机构项目资源使用列表

接口描述：运营端资源管理-机构项目资源列表查看，支持机构名称查询，项目状态，项目开始时间和排序查询。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/project/list | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| pageSize | [number] | 是 | 页数 |
| page | [number] | 是 | 页码 |
| projectName | [string] | 否 | 项目名称 |
| status | [string] | 否 | 项目状态 |
| startTime | [number] | 否 | 开始时间 |
| endTime | [number] | 否 | 结束时间 |
| sort | [string] | 否 | 排序类型，从大到小LARGE，从小到大SMALL，创建时间CREATE |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>currentPage | [number] | 是 | 页数 |
| data>>size | [number] | 是 | 页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总数 |
| data>>list | [array] | 是 | 数据列表 |
| data>>list>>id | [number] | 是 | Id |
| data>>list>>deptId | [number] | 是 | 机构id |
| data>>list>>userId | [number] | 是 | 用户id |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>cpu | [number] | 是 | cpu核数 |
| data>>list>>useCpu | [number] | 是 | 已用cpu核数 |
| data>>list>>internalMemory | [number] | 是 | 内存 |
| data>>list>>useInternalMemory | [number] | 是 | 已用内存 |
| data>>list>>diskMemory | [number] | 是 | 磁盘存储 |
| data>>list>>useDiskMemory | [number] | 是 | 已用磁盘存储 |
| data>>list>>createUserId | [number] | 是 | 创建用户id |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新用户id |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>list>>projectName | [string] | 是 | 项目名称 |
| data>>list>>status | [string] | 是 | 项目状态（NOT-RUN/未运行，RUN/运行中，STOP/停用，RUIN/销毁） |
| code | [number] | 是 |  |

资源管理-机构项目资源更新

接口描述：资源管理-某一个项目拥有的资源进行资源更新。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/project/update | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| projectId | [number] | 是 | 项目id |
| resourceQuotaVoList | [object] | 否 | 项目资源配额 |
| resourceQuotaVoList>>useDiskMemory | [number] | 否 | 已用磁盘存储（mb） |
| resourceQuotaVoList>>useInternalMemory | [number] | 否 | 已用内存（mb） |
| resourceQuotaVoList>>useCpu | [number] | 否 | 已用cpu核数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

资源管理-项目资源插入

接口描述：项目创建时插入一个项目资源。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/project/insert | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| projectId | [number] | 是 | 项目id |
| diskMemory | [number] | 否 | 磁盘存储 |
| internalMemory | [number] | 否 | 内存 |
| cpu | [number] | 否 | cpu核数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

资源管理-机构资源的使用量

接口描述：运营端资源管理-机构整体资源的使用情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| get | /resource/project/all | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
|  |  |  |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [array] | 是 | 数据 |
| data>>percentage | [number] | 是 | 百分比 |
| data>>num | [number] | 是 | 数量 |
| data>>type | [string] | 是 | 类型 |
| code | [number] | 是 | 状态码 |

资源管理-机构资源剩余情况

接口描述：运营端资源管理-机构资源剩余情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Get | /resource/project/remaining/{deptId} | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| depeId | [number] | 否 | 机构id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>diskMemory | [number] | 是 | 磁盘空间 |
| data>>cpu | [number] | 是 | Cpu |
| data>>internalMemory | [number] | 是 | 内存空间 |
| code | [number] | 是 | 状态码 |

资源管理-申请资源

接口描述：业务端资源管理-用户申请资源。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/project/insert/approve | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceContent | [string] | 是 | 申请的资源，需要string类型的json |
| approveDescription | [string] | 是 | 申请说明 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [boolean] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

###### ResourceUserController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ResourceUserController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、文件传输、数据更新相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/mhsdip/controller/resourceController/ResourceUserController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| update | session[:ResourceUser] | ResourceUser | ■公有 □私有 □保护 | Result | 更新机构资源 |
| Insert | session[:ResourceUserVo] | ResourceUserVo | ■公有 □私有 □保护 | Result | 插入机构资源 |
| remaining | session[:ResourceUserVo] | ResourceUserVo | ■公有 □私有 □保护 | Result | 系统资源的剩余使用量 |

算法描述

用户资源-用户资源更新

接口描述：用户管理-某一个机构对机构下的用户进行资源更新。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/user/update | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [string] | 是 | 机构id |
| userId | [string] | 是 | 用户 |
| cpu | [number] | 是 | cpu核数 |
| useCpu | [number] | 否 | 已用cpu核数 |
| internalMemory | [number] | 是 | 内存（mb） |
| useInternalMemory | [number] | 否 | 已用内存（mb） |
| diskMemory | [number] | 是 | 磁盘存储（mb） |
| useDiskMemory | [number] | 否 | 已用磁盘存储（mb） |
| points | [number] | 是 | 积分 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

用户资源-用户资源插入

接口描述：用户管理-创建用户时，默认插入一条用户资源。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/user/insert | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| deptId | [string] | 是 | 机构id |
| userId | [string] | 是 | 用户 |
| cpu | [number] | 是 | cpu核数 |
| useCpu | [number] | 否 | 已用cpu核数 |
| internalMemory | [number] | 是 | 内存（mb） |
| useInternalMemory | [number] | 否 | 已用内存（mb） |
| diskMemory | [number] | 是 | 磁盘存储（mb） |
| useDiskMemory | [number] | 否 | 已用磁盘存储（mb） |
| points | [number] | 是 | 积分 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

用户资源-用户资源剩余情况

接口描述：业务端用户资源管理-机构资源剩余情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Get | /resource/user/remaining | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| depeId | [number] | 是 | 机构id |
| userId | [number] | 是 | 用户id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [number] | 是 | 数据 |
| code | [number] | 是 | 状态码 |

用户资源-资源列表

接口描述：用户资源，根据机构id查询出机构下用户资源情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/user/list | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| page | [number] | 是 | 页码 |
| pageSize | [number] | 是 | 页数 |
| deptId | [number] | 否 | 机构id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>currentPage | [number] | 是 | 页数 |
| data>>size | [number] | 是 | 页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总数 |
| data>>list | [array] | 是 | 数据列表 |
| data>>list>>id | [number] | 是 | 资源id |
| data>>list>>deptId | [number] | 是 | 机构id |
| data>>list>>userId | [number] | 是 | 用户id |
| data>>list>>userName | [string] | 是 | 用户名称 |
| data>>list>>cpu | [number] | 是 | cpu核数 |
| data>>list>>useCpu | [number] | 是 | 已用cpu核数 |
| data>>list>>internalMemory | [number] | 是 | 内存 |
| data>>list>>useInternalMemory | [number] | 是 | 已用内存 |
| data>>list>>diskMemory | [number] | 是 | 磁盘存储 |
| data>>list>>useDiskMemory | [number] | 是 | 已用磁盘存储 |
| data>>list>>points | [number] | 是 | 积分 |
| data>>list>>createUserId | [number] | 是 | 创建用户id |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新用户id |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>list>>jsonObjectList | [null] | 是 | 资源统计情况 |
| code | [number] | 是 | 状态码 |

用户资源-资源详情

接口描述：用户资源，根据用户id查询出用户资源情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /resource/user/insert | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 否 | 用户id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 返回信息 |
| data | [object] | 是 | 数据 |
| data>>id | [null] | 是 |  |
| data>>deptId | [null] | 是 |  |
| data>>userId | [null] | 是 |  |
| data>>userName | [null] | 是 | 用户姓名 |
| data>>cpu | [null] | 是 |  |
| data>>useCpu | [null] | 是 |  |
| data>>internalMemory | [null] | 是 |  |
| data>>useInternalMemory | [null] | 是 |  |
| data>>diskMemory | [null] | 是 |  |
| data>>useDiskMemory | [null] | 是 |  |
| data>>points | [null] | 是 |  |
| data>>createUserId | [null] | 是 |  |
| data>>createTime | [null] | 是 |  |
| data>>updateUserId | [null] | 是 |  |
| data>>updateTime | [null] | 是 |  |
| data>>jsonObjectList | [array] | 是 | 资源百分比 |
| data>>jsonObjectList>>percentage | [number] | 是 | 百分比 |
| data>>jsonObjectList>>num | [number] | 是 | 数量 |
| data>>jsonObjectList>>type | [string] | 是 | 类型 |
| code | [number] | 是 | 状态码 |

##### 实体类

###### scResource

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | scResource | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存机构资源的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\resource\po\scResource | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 资源id |
| deptId | long | | ■公有 □私有 □保护 | | | NNI | | 机构id |
| cpu | int | | ■公有 □私有 □保护 | | |  | | cpu核数 |
| useCpu | int | | ■公有 □私有 □保护 | | |  | | 已用cpu核数 |
| internalMemory | int | | ■公有 □私有 □保护 | | |  | | 内存（mb） |
| useInternalMemory | int | | ■公有 □私有 □保护 | | |  | | 已用内存（mb） |
| diskMemory | int | | ■公有 □私有 □保护 | | |  | | 磁盘存储（mb） |
| useDiskMemory | int | | ■公有 □私有 □保护 | | |  | | 已用磁盘存储（mb） |
| points | int | | ■公有 □私有 □保护 | | |  | | 积分 |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_resource | | |

算法描述

无

状态转换描述

无

###### ResourceProject

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ResourceProject | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存机构资源的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\resource\po\ResourceProject | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 资源id |
| deptId | long | | ■公有 □私有 □保护 | | | NNI | | 机构id |
| userId | long | | ■公有 □私有 □保护 | | | NNI | | 用户id |
| projectId | long | | ■公有 □私有 □保护 | | | NNI | | 项目id |
| cpu | long | | ■公有 □私有 □保护 | | |  | | cpu核数 |
| useCpu | int | | ■公有 □私有 □保护 | | |  | | 已用cpu核数 |
| internalMemory | int | | ■公有 □私有 □保护 | | |  | | 内存（mb） |
| useInternalMemory | int | | ■公有 □私有 □保护 | | |  | | 已用内存（mb） |
| diskMemory | int | | ■公有 □私有 □保护 | | |  | | 磁盘存储（mb） |
| useDiskMemory | int | | ■公有 □私有 □保护 | | |  | | 已用磁盘存储（mb） |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_resource\_project | | |

算法描述

无

状态转换描述

无

###### ResourceUser

摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | | ResourceUser | | | | | |
| 父类 | | 无 | | | | | |
| 类职责简述 | | 保存用户资源的详细信息 | | | | | |
| 持久化 | | 是 | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\resource\po\ResourceUser | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | 资源id |
| deptId | long | | ■公有 □私有 □保护 | | | NNI | 机构id |
| userId | long | | ■公有 □私有 □保护 | | | NNI | 用户id |
| cpu | long | | ■公有 □私有 □保护 | | |  | cpu核数 |
| useCpu | int | | ■公有 □私有 □保护 | | |  | 已用cpu核数 |
| internalMemory | int | | ■公有 □私有 □保护 | | |  | 内存（mb） |
| useInternalMemory | int | | ■公有 □私有 □保护 | | |  | 已用内存（mb） |
| diskMemory | int | | ■公有 □私有 □保护 | | |  | 磁盘存储（mb） |
| useDiskMemory | int | | ■公有 □私有 □保护 | | |  | 已用磁盘存储（mb） |
| point | int | | ■公有 □私有 □保护 | | |  | 积分 |
| createUserId | long | | ■公有 □私有 □保护 | | |  | 创建用户id |
| createTime | long | | ■公有 □私有 □保护 | | |  | 创建时间 |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | 更新用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | 更新时间 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 |
| 无 | 无 | | | 无 | 无 | 无 | 无 |
| 对应数据库表 | | | | | | sc\_resource\_user | |

算法描述

无

状态转换描述

无

#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_resource、sc\_sys\_dict\_data、sc\_sys\_dept、sc\_resource\_project、sc\_project**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  r.id,  r.dept\_id AS deptId,  r.cpu,  r.use\_cpu AS useCpu,  r.internal\_memory AS internalMemory,  r.use\_internal\_memory AS useInternalMemory,  r.disk\_memory AS diskMemory,  r.use\_disk\_memory AS useDiskMemory,  r.points,  r.create\_user\_id AS createUserId,  r.create\_time AS createTime,  r.update\_user\_id AS updateUserId,  r.update\_time AS updateTime,  d.dept\_name AS deptName  FROM  sc\_resource r  INNER JOIN sc\_sys\_dept d ON r.dept\_id = d.dept\_id  < WHERE > 1 = 1  < IF test = "deptName != null and deptName != '' " >  AND d.dept\_name LIKE concat('%',#{deptName}, '%')  </ IF >  < IF test = "deptId != null and deptId != '' " >  AND d.dept\_id = #{deptId}  </ IF >  </ WHERE>  【SQL02】  SELECT COALESCE  ( sum( use\_cpu ), 0 ) AS useCpu,  COALESCE ( sum( use\_internal\_memory ), 0 ) AS useInternalMemory,  COALESCE ( sum( use\_disk\_memory ), 0 ) AS useDiskMemory,  COALESCE ( sum( cpu ), 0 ) AS cpu,  COALESCE ( sum( internal\_memory ), 0 ) AS internalMemory,  COALESCE ( sum( disk\_memory ), 0 ) AS diskMemory  FROM  sc\_resource  < WHERE >  < IF test = "deptId != 0 and deptId != null" >  dept\_id = #{deptId}  </ IF >  </ WHERE>  【SQL03】  SELECT  dict\_code,  dict\_sort,  dict\_label,  dict\_value,  dict\_type,  css\_class,  list\_class,  is\_default,  STATUS,  create\_user\_id,  create\_time,  update\_user\_id,  update\_time,  remark  FROM  sc\_sys\_dict\_data  WHERE  dict\_type = 'RESOURCE'  【SQL04】  SELECT  r.id,  r.dept\_id AS deptId,  r.user\_id AS userId,  r.project\_id AS projectId,  r.cpu,  r.use\_cpu AS useCpu,  r.internal\_memory AS internalMemory,  r.use\_internal\_memory AS useInternalMemory,  r.disk\_memory AS diskMemory,  r.use\_disk\_memory AS useDiskMemory,  r.create\_user\_id AS createUserId,  r.create\_time AS createTime,  r.update\_user\_id AS updateUserId,  p.create\_time AS startTime,  p.end\_time AS endTime,  p.NAME AS projectName,  p.STATUS AS STATUS,  p.destruction\_time AS destructionTime  FROM  sc\_resource\_project r  LEFT JOIN sc\_project p ON r.project\_id = p.id  <where>  1=1  <if test="projectName != null and projectName != '' ">  and p.name like concat('%', #{projectName}, '%')  </if>  <if test="status != null and status != '' ">  and p.status = #{status}  </if>  <if test="startTime != null and startTime != '' ">  and p.create\_time &gt;= #{startTime}  </if>  <if test="endTime !=null and endTime != '' ">  and p.create\_time &lt;= #{endTime}  </if>  <if test="deptId !=null and deptId != 0l ">  and p.dept\_id = #{deptId}  </if>  <if test="userId != null and userId != 0l ">  and r.user\_id = #{userId}  </if>  </where>  <if test="sort == 'LARGE' ">  order by r.cpu DESC,r.internal\_memory DESC,r.disk\_memory DESC  </if>  <if test="sort == 'SMALL' ">  order by r.cpu ASC,r.internal\_memory ASC,r.disk\_memory ASC  </if>  <if test="sort == 'CREATE' ">  order by r.create\_time asc  </if>  【SQL05】  SELECT COALESCE  ( sum( use\_cpu ), 0 ) AS useCpu,  COALESCE ( sum( use\_internal\_memory ), 0 ) AS useInternalMemory,  COALESCE ( sum( use\_disk\_memory ), 0 ) AS useDiskMemory,  COALESCE ( sum( cpu ), 0 ) AS cpu,  COALESCE ( sum( internal\_memory ), 0 ) AS internalMemory,  COALESCE ( sum( disk\_memory ), 0 ) AS diskMemory  FROM  sc\_resource  < WHERE >  < IF test = "deptId != 0 and deptId != null" >  dept\_id != #{deptId}  </ IF >  </ WHERE>  【SQL06】  SELECT  id,  dept\_id,  cpu,  use\_cpu,  internal\_memory,  use\_internal\_memory,  disk\_memory,  use\_disk\_memory,  points,  create\_user\_id,  create\_time,  update\_user\_id,  update\_time  FROM  sc\_resource  WHERE  dept\_id = #{id}  【SQL07】  SELECT  sru.id,  sru.dept\_id as deptId,  sru.user\_id as userId,  sru.cpu,  sru.use\_cpu as useCpu,  sru.internal\_memory as internalMemory,  sru.use\_internal\_memory as useInternalMemory,  sru.disk\_memory as diskMemory,  sru.use\_disk\_memory as useDiskMemory,  sru.points,  sru.create\_user\_id as createUserId,  sru.create\_time as createTime,  sru.update\_user\_id as updateUserId,  sru.update\_time as updateTime,  ssu.user\_name as userName FROM  sc\_resource\_user sru LEFT JOIN sc\_sys\_user ssu ON sru.user\_id = ssu.user\_id <where>  <if test="userName != null and userName != '' ">  and ssu.user\_name = #{userName}  </if>  <if test="deptId != null and deptId != '' ">  and ssu.dept\_id = #{deptId}  </if> </where>  【SQL09】  SELECT  sru.id,  sru.dept\_id as deptId,  sru.user\_id as userId,  sru.cpu,  sru.use\_cpu as useCpu,  sru.internal\_memory as internalMemory,  sru.use\_internal\_memory as useInternalMemory,  sru.disk\_memory as diskMemory,  sru.use\_disk\_memory as useDiskMemory,  sru.points,  sru.create\_user\_id as createUserId,  sru.create\_time as createTime,  sru.update\_user\_id as updateUserId,  sru.update\_time as updateTime,  ssu.user\_name as userName FROM  sc\_resource\_user sru LEFT JOIN sc\_sys\_user ssu ON sru.user\_id = ssu.user\_id <where>  <if test="userName != null and userName != '' ">  and ssu.user\_name = #{userName}  </if>  <if test="deptId != null and deptId != '' ">  and ssu.dept\_id = #{deptId}  </if> </where>  【SQL13】  select coalesce(sum(use\_cpu),0) as useCpu, coalesce(sum(use\_internal\_memory),0) as useInternalMemory, coalesce(sum(use\_disk\_memory),0) as useDiskMemory, coalesce(sum(cpu),0) as cpu, coalesce(sum(internal\_memory),0) as internalMemory, coalesce(sum(disk\_memory),0) as diskMemory from sc\_resource\_user <where>  1=1  <if test="deptId != 0">  and dept\_id = #{deptId}  </if>  <if test="userId != 0">  and user\_id != #{userId}  </if> </where> |
| **程序流程中涉及的数据库表：<表名：**sc\_resource\_project、sc\_resource **>**  **操作：<修改>** |
| SQL语句：  【SQL10】  UPDATE sc\_resource  SET dept\_id =#{deptId},  cpu =#{cpu },  use\_cpu =#{use\_cpu },  internal\_memory =#{internal\_memory},  use\_internal\_memory =#{use\_internal\_memory},  disk\_memory =#{disk\_memory},  use\_disk\_memory =#{use\_disk\_memory},  points =#{points},  create\_user\_id =#{create\_user\_id},  create\_time =#{create\_time},  update\_user\_id =#{update\_user\_id},  update\_time =#{update\_time}  WHERE  dept\_id = #{deptId}  【SQL11】  UPDATE sc\_resource\_project  SET dept\_id =#{deptId},  cpu =#{cpu },  use\_cpu =#{use\_cpu },  internal\_memory =#{internal\_memory},  use\_internal\_memory =#{use\_internal\_memory},  disk\_memory =#{disk\_memory},  use\_disk\_memory =#{use\_disk\_memory},  points =#{points},  create\_user\_id =#{create\_user\_id},  create\_time =#{create\_time},  update\_user\_id =#{update\_user\_id},  update\_time =#{update\_time}  WHERE  dept\_id = #{deptId}  【SQL12】  UPDATE sc\_resource\_project  SET dept\_id =#{deptId},  cpu =#{cpu },  use\_cpu =#{use\_cpu },  internal\_memory =#{internal\_memory},  use\_internal\_memory =#{use\_internal\_memory},  disk\_memory =#{disk\_memory},  use\_disk\_memory =#{use\_disk\_memory},  points =#{points},  create\_user\_id =#{create\_user\_id},  create\_time =#{create\_time},  update\_user\_id =#{update\_user\_id},  update\_time =#{update\_time}  WHERE  dept\_id = #{deptId} |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | cpu | 0 | 资源值 | Cpu资源 | | use\_cpu | 0 | 资源值 | Cpu已用资源 | | internal\_memory | 0 | 资源值 | 内存资源 | | use\_internal\_memory | 0 | 资源值 | 内存已用资源 | | disk\_memory | 0 | 资源值 | 磁盘内存资源 | | use\_disk\_memory | 0 | 资源值 | 磁盘已用资源 | |
| **程序流程中涉及的数据库表：<表名：**sc\_resource**>**  **操作：<新增>** |
| 【SQL14】  insert into sc\_sys\_approve (user\_id, user\_dept\_id, user\_dept, user\_name, approve\_role,   approve\_dept\_id, approve\_id, approve\_description, comment, task\_id,   task\_name, relevance\_content, relevance\_id, relevance\_size, project\_id,  flow\_id, flow\_detail\_id, type, mode, status, create\_time, update\_time) values (#{userId}, #{userDeptId}, #{userDept}, #{userName}, #{approveRole},   #{approveDeptId}, #{approveId}, #{approveDescription}, #{comment}, #{taskId},   #{taskName}, #{relevanceContent}, #{relevanceId}, #{relevanceSize}, #{projectId},  #{flowId}, #{flowDetailId}, #{type}, #{mode}, #{status}, #{createTime}, #{updateTime}); |

#### 总体设计方案

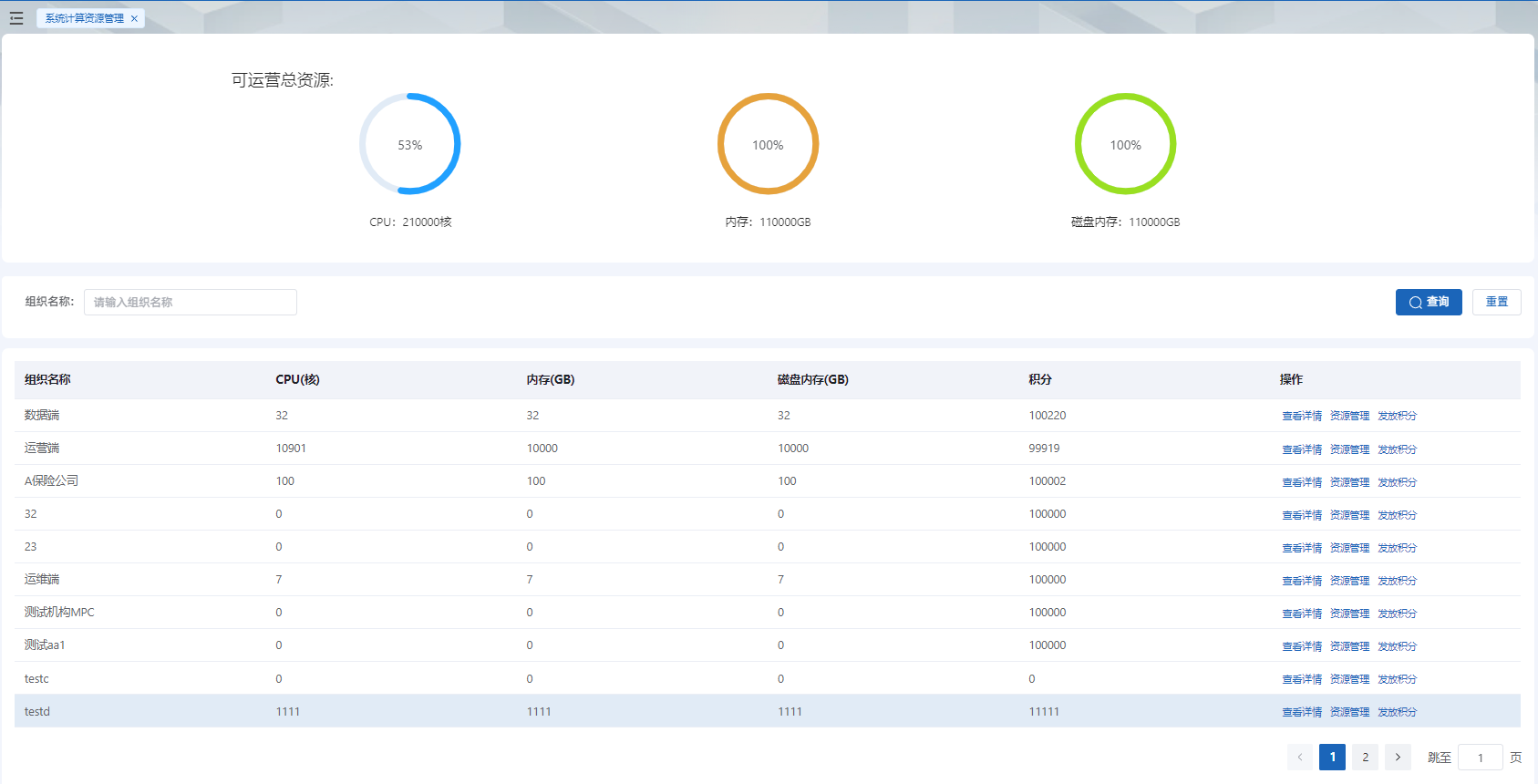
##### 设计方案

资源管理为用户提供系统CPU、内存、磁盘等资源的查看和申请等功能，支持将资源分配给指定的机构、用户。

##### 处理流程

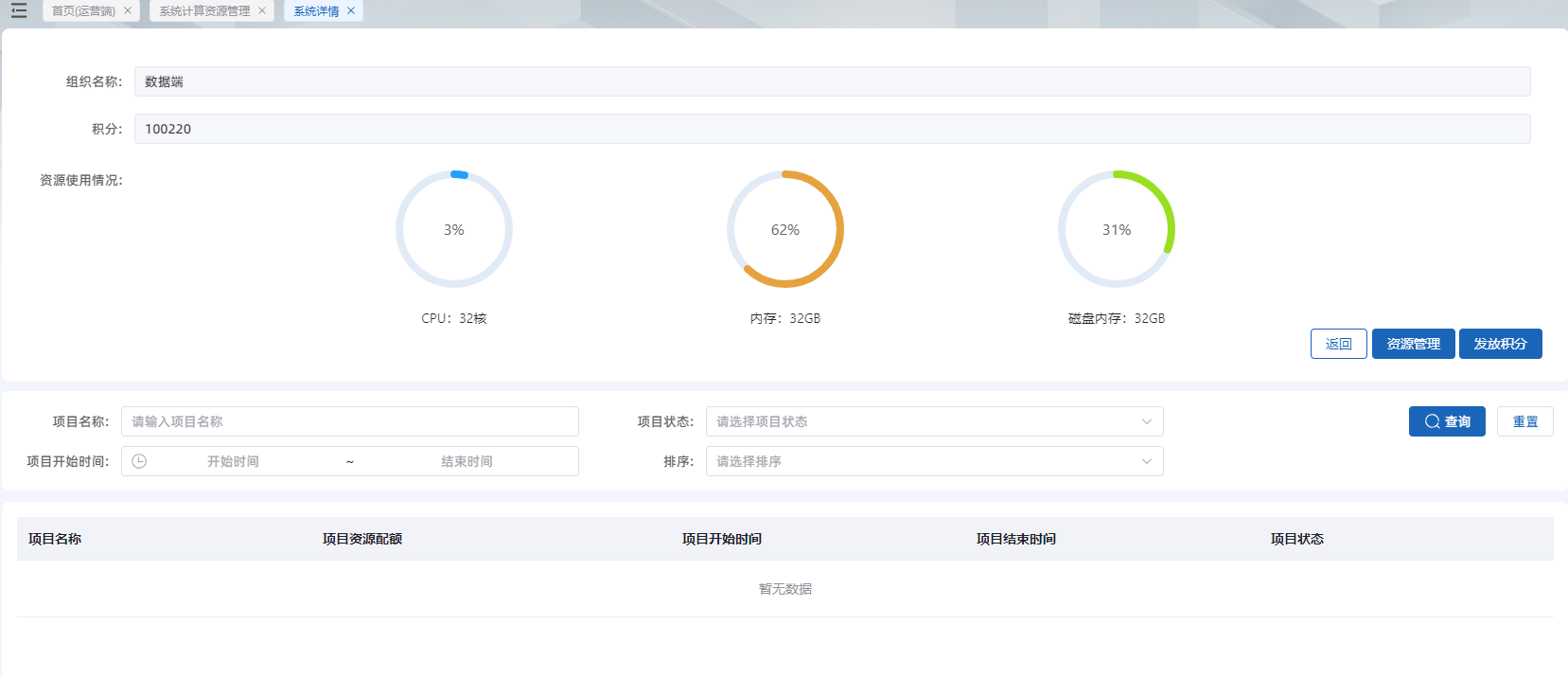
###### 系统计算资源管理

1.查询资源列表



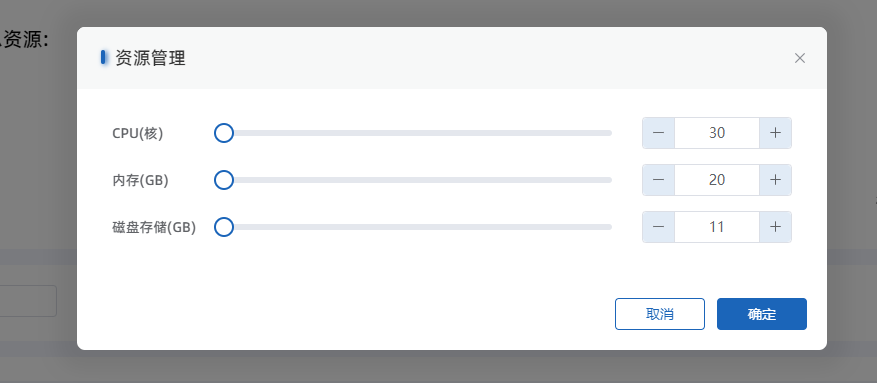
进入资源管理系统计算资源管理页面，下部分组织资源列表为页面调用【6.3.5.1.1.1.2.2.资源管理-机构资源使用列表】接口，在未输入筛选条件情况下，使用【SQL01】默认返回所有组织的资源使用情况,若前端输入筛选条件，则在【SQL01】基础上增加筛选条件查询。上部分资源详情为页面调用【6.3.5.1.1.1.2.5.资源管理-系统资源的使用量】接口。接口执行【SQL05】查询出三个类型的资源分配额度情况和系统资源总量处理图表的格式返回页面。

2.查看详情



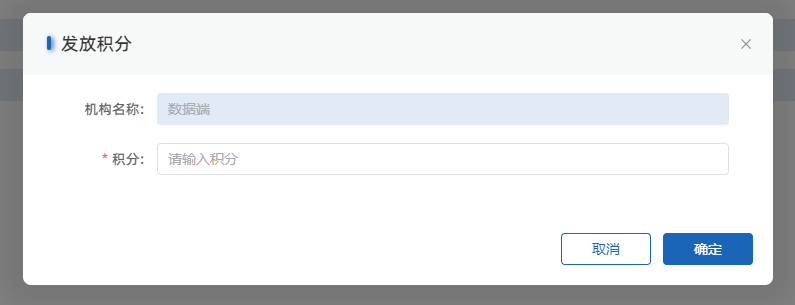
页面点击【查看详情】弹出系统资源详情页面，调用【6.3.5.1.1.1.2.1.资源管理-机构资源使用详情】接口查询资源详情。接口根据机构id执行【SQL01】查询组织基本信息和组织资源占用情况信息，再根据前端需求处理图表的格式返回页面。下面项目资源列表调用【6.3.5.1.1.2.2.2.资源管理-机构项目资源使用列表】接口，在未输入筛选条件情况下，使用【SQL04】默认返回所有机构的资源使用情况,若前端输入筛选条件，则在【SQL04】基础上增加筛选条件查询。

3.资源管理



页面点击【资源管理】弹出资源管理页面，页面调用【6.3.5.1.1.1.2.6.资源管理-系统资源剩余情况】执行【SQL05】查询出不包括当前机构的已分配资源量，再执行【SQL02】查询出系统资源总量，封装返回资源剩余值。配置完成资源后将配置信息作为参数调用【6.3.5.1.1.1.2.3.资源管理-机构资源更新】接口。接口执行【SQL06】查询机构原始资源，把修改的资源配置信息执行【SQL07】进行修改。

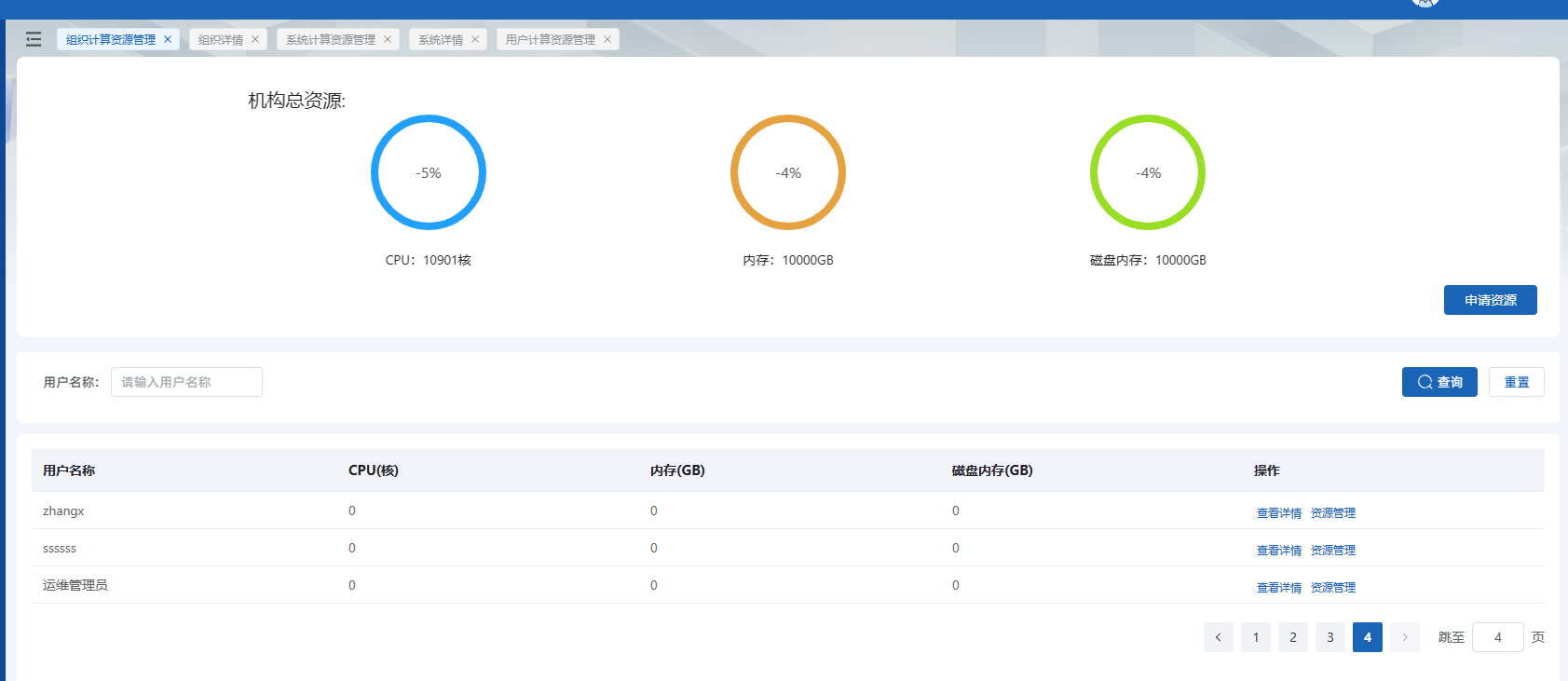
4.发放积分



发送积分配置完成积分资源后将配置信息作为参数调用【6.3.5.1.1.1.2.3.资源管理-机构资源更新】接口。接口执行【SQL02】查询机构原始资源，把修改的资源配置信息执行【SQL08】进行修改。

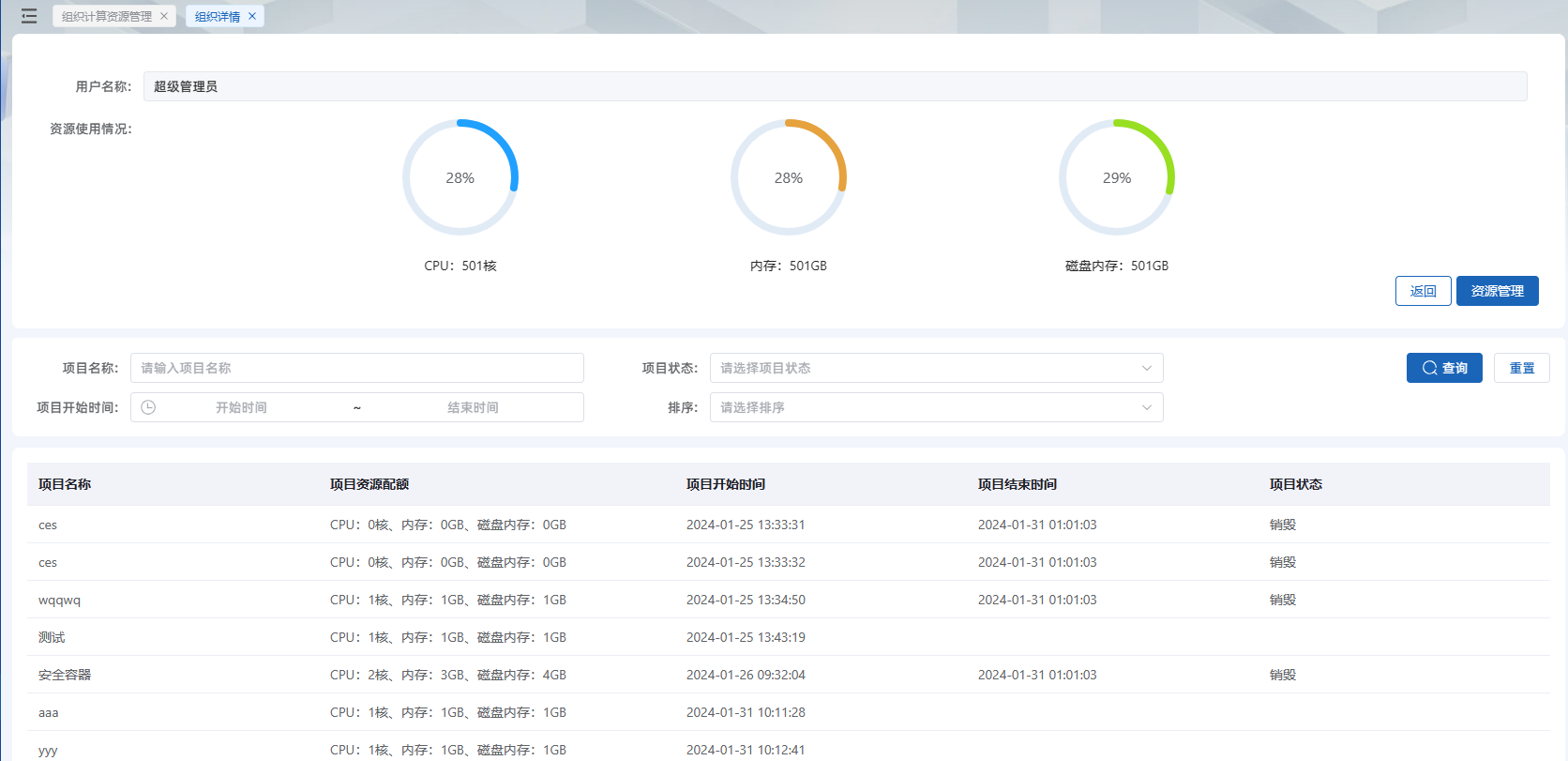
###### 组织计算资源管理

1. 查询资源列表



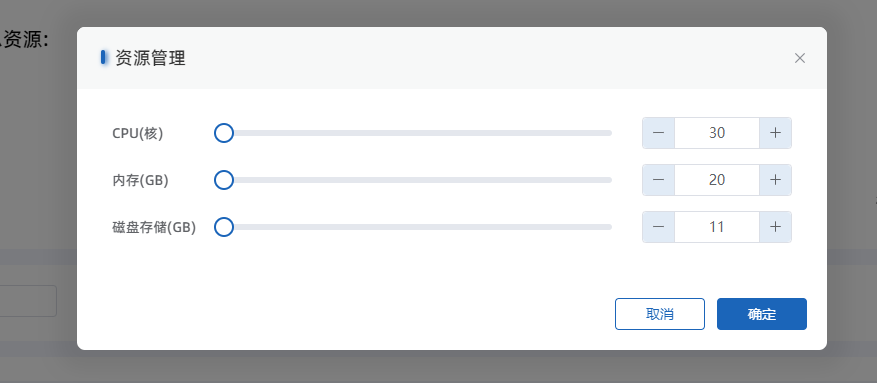
进入资源管理组织计算资源管理页面，下部分用户资源列表为页面调用【6.3.5.1.1.3.2.4.用户资源-资源列表】接口，在未输入筛选条件情况下，使用【SQL07】默认返回所有用户的资源使用情况,若前端输入筛选条件，则在【SQL07】基础上增加筛选条件查询。上部分资源详情为页面调用【6.3.5.1.1.2.2.5.资源管理-机构资源的使用量】接口。接口先执行【SQL02】查询出三个类型的资源分配额度情况和系统资源总量，再根据前端需求处理图表的格式返回页面。

2.查看详情



页面点击【查看详情】弹出组织资源详情页面，调用【6.3.5.1.1.3.2.5.用户资源-资源详情】接口查询资源详情。接口根据机构id执行【SQL04】查询组织基本信息，根据用户id执行【SQL04】查询资源占用情况信息，再根据前端需求处理图表的格式返回页面。下面项目资源列表调用【6.3.5.1.1.2.2.2.资源管理-机构项目资源使用列表】接口，在未输入筛选条件情况下，使用【SQL07】默认返回所有机构的资源使用情况,若前端输入筛选条件，则在【SQL07】基础上增加筛选条件查询。

3.资源管理



页面点击【资源管理】弹出资源管理页面，页面调用【6.3.5.1.1.3.2.3.用户资源-用户资源剩余情况】执行【SQL13】查询出不包括当前用户的已分配资源量，再执行【SQL05】查询出系统资源总量，封装返回资源剩余值。配置完成资源后将配置信息作为参数调用【6.3.5.1.1.3.2.1.用户资源-用户资源更新】接口。接口执行【SQL10】查询用户原始资源，把修改的资源配置信息执行【SQL11】进行修改。

4.申请资源



页面点击【申请资源】弹出资源管理页面，输入需要申请的cpu、内存、磁盘内存资源（显示对应所需积分）调用【6.3.5.1.1.1.2.7.资源管理-系统资源申请】接口，执行【SQL14】插入审核数据，完成该机构资源申请功能。

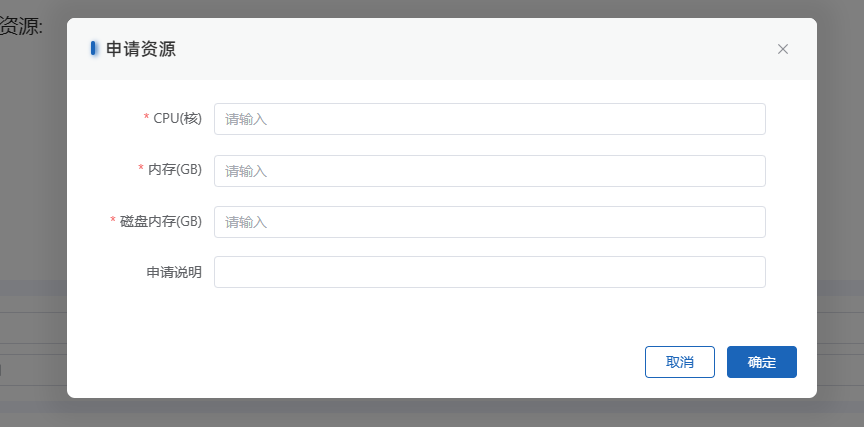
###### 用户计算资源管理

1.查询资源列表



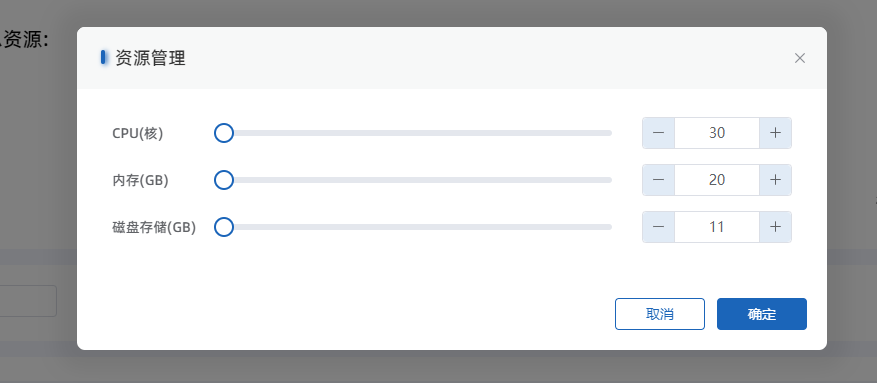
进入资源管理用户计算资源管理页面，下部分项目资源列表为页面调用【6.3.5.1.1.2.2.2.资源管理-机构项目资源使用列表】接口，在未输入筛选条件情况下，使用【SQL04】默认返回所有项目的资源使用情况,若前端输入筛选条件，则在【SQL04】基础上增加筛选条件查询。上部分资源详情为页面调用【6.3.5.1.1.1.2.5.资源管理-系统资源的使用量】接口。接口先执行【SQL05】查询出三个类型的资源分配额度情况和系统资源总量，再根据前端需求处理图表的格式返回页面。

2.申请资源



页面点击【申请资源】弹出资源管理页面，输入需要申请的cpu、内存、磁盘内存资源，以及申请说明（非必填），调用【6.3.5.1.1.2.2.7.资源管理-申请资源】接口，执行【SQL14】插入审核数据，完成该资源申请功能。

3.资源管理



页面点击【资源管理】弹出资源管理页面，页面调用【6.3.5.1.1.2.2.6.资源管理-机构资源剩余情况】执行【SQL05】查询出不包括当前项目的已分配资源量和系统资源总量，封装返回资源剩余值。配置完成资源后将配置信息作为参数调用【6.3.5.1.1.2.2.3.资源管理-机构项目资源更新】接口。接口执行【SQL04】查询项目原始资源，把修改的资源配置信息执行【SQL12】进行修改。

### 场景管理

#### 类设计

##### 控制类

###### BusinessScenarioController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | BusinessScenarioController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 业务场景管理 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\otherController\BusinessScenarioController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| insertBusinessScenario | session[:BusinessScenarioVo] | BusinessScenarioVo | ■公有 □私有 □保护 | Result | 新增业务场景数据 |
| updateBusinessScenario | session[:BusinessScenarioVo] | BusinessScenarioVo | ■公有 □私有 □保护 | Result | 修改业务场景数据 |
| delectBusinessScenario | session[:id] | Long | ■公有 □私有 □保护 | Result | 软删除业务场景数据 |
| selectBusinessScenario | 无 | 无 | ■公有 □私有 □保护 | Result | 查询业务场景数据 |
| selectPageBusinessScenario | session[:BusinessScenarioVo] | BusinessScenarioVo | ■公有 □私有 □保护 | Result | 分页查询业务场景数据 |
| getBusinessScenario | session[:BusinessScenarioVo] | BusinessScenarioVo | ■公有 □私有 □保护 | Result | 业务场景详情获取 |

算法描述

新增业务场景数据

接口描述：增加业务场景在数据集、应用、项目等上进行业务分类。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /businessScenario/insert | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 是 | 场景名称 |
| description | [string] | 是 | 场景备注 |
| isCommon | [string] | 是 | 是否公用（yes是公用，id为机构私有） |
| sceneChoose | [string] | 是 | 业务场景选择 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

修改业务场景数据

接口描述：修改业务场景在数据集、应用、项目等上进行业务分类。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /businessScenario/update | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 场景id |
| name | [string] | 是 | 场景名称 |
| description | [string] | 是 | 场景备注 |
| isCommon | [string] | 是 | 是否公用（yes是公用，id为机构私有） |
| sceneChoose | [string] | 是 | 业务场景选择 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

软删除业务场景数据

接口描述：删除业务场景在数据集、应用、项目等上进行业务分类。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /businessScenario/delect | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 场景id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

查询业务场景数据

接口描述：查询业务场景在数据集、应用、项目等上进行业务分类使用。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /businessScenario/select | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>id | [number] | 是 | 场景id |
| data>>userId | [number] | 是 | 用户id |
| data>>userName | [string] | 是 | 用户名称 |
| data>>deptId | [null] | 是 | 用户所属机构id |
| data>>deptName | [null] | 是 | 用户所属机构名称 |
| data>>name | [string] | 是 | 场景名称 |
| data>>description | [string] | 是 | 场景备注 |
| data>>delFlag | [number] | 是 | 删除标志（1代表存在 0代表删除） |
| data>>sceneChooseList | [array] | 是 | 业务场景选择 |
| data>>applicationDtoList | [array] | 是 | 绑定的应用 |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateTime | [number] | 是 | 修改时间 |

分页查询业务场景数据

接口描述：分页查询业务场景在数据集、应用、项目等上进行业务分类使用。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /businessScenario/selectPage | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 否 | 场景名称 |
| userName | [string] | 否 | 用户名称 |
| deptName | [string] | 否 | 用户所属机构名称 |
| startTime | [string] | 否 | 开始时间 |
| endTime | [string] | 否 | 结束时间 |
| page | [string] | 否 | 当前页 |
| pageSize | [string] | 否 | 页面大小 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总记录数量 |
| data>>list | [array] | 是 | 数据集合 |
| data>>list>>id | [number] | 是 | 场景id |
| data>>list>>userId | [number] | 是 | 用户id |
| data>>list>>userName | [string] | 是 | 用户名称 |
| data>>list>>deptId | [number] | 是 | 用户所属机构id |
| data>>list>>deptName | [string] | 是 | 用户所属机构名称 |
| data>>list>>name | [string] | 是 | 场景名称 |
| data>>list>>description | [string] | 是 | 场景备注 |
| data>>list>>delFlag | [number] | 是 | 删除标志（1代表存在 0代表删除） |
| data>>list>>sceneChooseList | [array] | 是 | 业务场景选择 |
| data>>list>>applicationDtoList | [array] | 是 | 绑定的应用 |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 修改时间 |

业务场景详情获取

接口描述：查询业务场景详情信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /businessScenario/get | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 场景id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>id | [number] | 是 | 场景id |
| data>>userId | [number] | 是 | 用户id |
| data>>userName | [string] | 是 | 用户名称 |
| data>>deptId | [null] | 是 | 用户所属机构id |
| data>>deptName | [null] | 是 | 用户所属机构名称 |
| data>>name | [string] | 是 | 场景名称 |
| data>>description | [string] | 是 | 场景备注 |
| data>>delFlag | [number] | 是 | 删除标志（1代表存在 0代表删除） |
| data>>sceneChooseList | [array] | 是 | 业务场景选择 |
| data>>applicationDtoList | [array] | 是 | 绑定的应用 |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateTime | [number] | 是 | 修改时间 |

##### 实体类

###### BusinessScenario

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | BusinessScenario | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存业务场景的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\po\BusinessScenario.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | |  | | 场景id |
| userId | long | | ■公有 □私有 □保护 | | |  | | 用户id |
| userName | string | | ■公有 □私有 □保护 | | |  | | 用户名 |
| deptId | long | | ■公有 □私有 □保护 | | |  | | 用户所属机构id |
| deptName | string | | ■公有 □私有 □保护 | | |  | | 用户所属机构名称 |
| businessType | string | | ■公有 □私有 □保护 | | |  | | 场景类型 |
| name | string | | ■公有 □私有 □保护 | | |  | | 场景名称 |
| description | string | | ■公有 □私有 □保护 | | |  | | 场景备注 |
| delFlag | long | | ■公有 □私有 □保护 | | |  | | 删除标志（1代表存在 0代表删除） |
| isCommon | string | | ■公有 □私有 □保护 | | |  | | 是否公用（yes是公用，id为机构私有） |
| sceneChoose | string | | ■公有 □私有 □保护 | | |  | | 业务场景选择 |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 修改时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sc\_business\_scenario | | |

算法描述

无

状态转换描述

无

#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_evaluate、 sc\_business\_scenario、sc\_data\_set\_place、sc\_application\_place、 sc\_model\_place**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询客户评价  select  id,  offer\_user\_name,  product\_name,  score,  evaluate\_content,  create\_time,  problem\_type,  problem\_summary,  problem\_attribution  from  sc\_evaluate  <where>  evaluate\_dept\_id = #{vo.evaluateDeptId} and evaluate\_type = 0  <if test="vo.offerUserName != null and vo.offerUserName != ''">  and offer\_user\_name like concat('%',#{vo.offerUserName},'%')  </if>  <if test="vo.productName != null and vo.productName != ''">  and product\_name like concat('%', #{vo.productName},'%')  </if>  <if test="vo.problemType != null">  and problem\_type = #{vo.problemType}  </if>  <if test="vo.startTime != null and vo.startTime != '' and vo.endTime != null and vo.endTime != '' ">  and create\_time between #{vo.startTime} and #{vo.endTime}  </if>  </where>  【SQL02】查询机构评价  select  id,  offer\_user\_name,  offer\_dept\_name,  product\_name,  score,  evaluate\_content,  create\_time,  problem\_type,  problem\_summary,  problem\_attribution  from  sc\_evaluate  <where>  evaluate\_dept\_id = #{vo.evaluateDeptId} and evaluate\_type = 1  <if test="vo.offerUserName != null and vo.offerUserName != ''">  and offer\_user\_name like concat('%',#{vo.offerUserName},'%')  </if>  <if test="vo.productName != null and vo.productName != ''">  and product\_name like concat('%', #{vo.productName},'%')  </if>  <if test="vo.startTime != null and vo.startTime != '' and vo.endTime != null and vo.endTime != '' ">  and create\_time between #{vo.startTime} and #{vo.endTime}  </if>  </where>  【SQL03】产品优化  SELECT  a.problemType as problemType,  b.problemTypeNum as problemTypeNum,  a.problemAttribution as problemAttribution,  a.attributionNum as attributionNum  FROM  (SELECT  x.problem\_type as problemType,  COUNT(x.id) as attributionNum,  x.type as problemAttribution  FROM(  SELECT a.problem\_type,a.id,  substring\_index( substring\_index( a.problem\_attribution, ',', b.help\_topic\_id + 1 ), ',',- 1 ) as type  FROM sc\_evaluate a  JOIN mysql.help\_topic b ON b.help\_topic\_id &lt; ( length( a.problem\_attribution ) - length( REPLACE ( a.problem\_attribution, ',', '' ) ) + 1 )  where a.evaluate\_dept\_id = #{vo.evaluateDeptId}  ORDER BY a.id  ) x  GROUP BY x.type,x.problem\_type  )a  LEFT JOIN  (SELECT  x.problem\_type as problemType,  COUNT(x.id) as problemTypeNum  FROM(  SELECT a.problem\_type,a.id,  substring\_index( substring\_index( a.problem\_attribution, ',', b.help\_topic\_id + 1 ), ',',- 1 ) as type  FROM sc\_evaluate a  JOIN mysql.help\_topic b ON b.help\_topic\_id &lt; ( length( a.problem\_attribution ) - length( REPLACE ( a.problem\_attribution, ',', '' ) ) + 1 )  where a.evaluate\_dept\_id = #{vo.evaluateDeptId}  ORDER BY a.id  ) x  GROUP BY x.problem\_type)b on a.problemType = b.problemType  【SQL04】产品优化总数量  SELECT  count(id)  FROM(  SELECT a.problem\_type,a.id,  substring\_index( substring\_index( a.problem\_attribution, ',', b.help\_topic\_id + 1 ), ',',- 1 ) as type  FROM sc\_evaluate a  JOIN mysql.help\_topic b ON b.help\_topic\_id &lt; ( length( a.problem\_attribution ) - length( REPLACE ( a.problem\_attribution, ',', '' ) ) + 1 )  where a.evaluate\_dept\_id = #{vo.evaluateDeptId}  ORDER BY a.id  ) x  【SQL05】产品优化详情  select  offer\_user\_name,  product\_name,  score,  evaluate\_content,  create\_time,  problem\_type  from  sc\_evaluate  <where>  evaluate\_dept\_id = #{vo.evaluateDeptId}  <if test="vo.problemType != null">  and problem\_type = #{vo.problemType}  </if>  <if test="vo.problemAttribution != null and vo.problemAttribution != ''">  and FIND\_IN\_SET(#{vo.problemAttribution},problem\_attribution)  </if>  </where>  【SQL06】业务场景列表不分页查询  select \* from sc\_business\_scenario  <where>  del\_flag = 1  <if test="vo.id != null ">  and id != #{vo.id}  </if>  <if test="vo.name != null and vo.name != ''">  and name = #{vo.name}  </if>  <if test="vo.businessType != null and vo.businessType != ''">  and business\_type = #{vo.businessType}  </if>  </where>  limit 1  【SQL07】业务场景列表不分页查询  select \* from sc\_business\_scenario  where del\_flag = 1 and (is\_common is null or is\_common = 'yes')  【SQL08】业务场景列表不分页查询  select \* from sc\_business\_scenario  where del\_flag = 1 and is\_common is not null and  is\_common != 'yes' and dept\_id = #{deptId}  【SQL09】获取市场的所有数据集名称  select name from sc\_data\_set\_place where del\_flag = 1  【SQL10】获取应用市场中应用名称  select name from sc\_application\_place where del\_flag = 1  【SQL11】获取模型市场中模型名称  select name from sc\_model\_place where del\_flag = 1  【SQL12】根据id查询业务场景  select \* from sc\_business\_scenario  where id = #{id}  【SQL13】查询业务场景  select \* from sc\_business\_scenario  <where>  del\_flag = 1  <if test="businessType != null and businessType != ''">  and business\_type like concat ('%',#{businessType},'%')  </if>  <if test="userName!= null and userName!= ''">  and user\_name like concat ('%',#{userName},'%')  </if>  <if test="deptName!= null and deptName!= ''">  and dept\_name like concat ('%',#{deptName},'%')  </if>  <if test="startTime!= null and endTime!= null>  and create\_time between #{startTime} and #{endTime}  </if>  </where>  【SQL14】  SELECT  a.application\_id as applicationId,  a.user\_id as userId,  a.offer\_user\_id as offerUserId,  a.offer\_user\_name as offerUserName,  a.offer\_dept\_id as offerDeptId,  a.offer\_dept\_name as offerDeptName,  a.NAME as applicationName ,  a.description as description,  a.type as type,  a.icon as icon,  a.dept\_type as deptType,  a.url as url,  a.cpu as cpu,  a.file\_url as fileUrl,  a.application\_type as applicationType,  a.internal\_memory as internalMemory,  a.disk\_memory as diskMemory,  a.points as points,  a.billing\_mode AS billingMode,  a.safety\_level as safetyLevel,  a.STATUS as status,  a.safe\_application\_id as safeApplicationId,  a.safe\_application\_test\_id as safeApplicationTestId,  a.create\_time as createTime,  a.update\_time as updateTime,  a.authority as authority,  a.source\_type as sourceType,  a.ports,  GROUP\_CONCAT(b.NAME) as businessName,  b.business\_type as businessType  FROM  sc\_application a  INNER JOIN sc\_business\_scenario b ON FIND\_IN\_SET(b.id, a.type)  <where>  a.del\_flag = 1  <if test="applicationName != null and applicationName != ''">  and a.name like concat('%' #{applicationName} '%')  </if>  <if test="offerDeptId != null and offerDeptId != ''">  and a.offer\_dept\_id = #{offerDeptId}  </if>  <if test="offerDeptName != null and offerDeptName != ''">  and a.offer\_dept\_name like concat('%' #{offerDeptName} '%')  </if>  <if test="offerUserName != null and offerUserName != ''">  and a.offer\_user\_name like concat('%' #{offerUserName} '%')  </if>  <if test="status != null and status != ''">  and a.status = #{status}  </if>  <if test="businessName != null and businessName != ''">  and b.name like concat('%' #{businessName} '%')  </if>  <if test="startPoints != null">  and a.points &gt;= #{startPoints}  </if>  <if test="endPoints != null">  and a.points &lt;= #{endPoints}  </if>  <if test="safetyLevel != null and safetyLevel != ''">  and b.safety\_level like concat('%' #{safetyLevel} '%')  </if>  <if test="userId != null and userId != 0l">  and a.user\_id = #{userId}  </if>  <if test="sourceType != null and sourceType != 0l">  and a.source\_type = #{sourceType}  </if>  <if test="type != null and type != 0l">  and a.type = #{type}  </if>  <if test="deptType != null and deptType != ''">  and a.dept\_type = #{deptType}  </if>  </where>  group by a.application\_id  order by a.create\_time desc |
|  |
| **程序流程中涉及的数据库表：<表名：sc\_evaluate、 sc\_business\_scenario>**  **操作：<插入>** |
| SQL语句：  【SQL15】新增数据  insert into sc\_evaluate(offer\_user\_id, offer\_user\_name, offer\_dept\_id, offer\_dept\_name, evaluate\_dept\_id, product\_name, relevance\_id, score, evaluate\_content, problem\_type, problem\_summary, problem\_attribution, evaluate\_type, create\_time, update\_time)  values  (#{entity.offerUserId}, #{entity.offerUserName}, #{entity.offerDeptId}, #{entity.offerDeptName}, #{entity.evaluateDeptId}, #{entity.productName}, #{entity.relevanceId}, #{entity.score}, #{entity.evaluateContent}, #{entity.problemType}, #{entity.problemSummary}, #{entity.problemAttribution}, #{entity.evaluateType}, #{entity.createTime}, #{entity.updateTime})  【SQL16】新增业务  insert into sc\_business\_scenario(user\_id, user\_name, dept\_id, dept\_name, business\_type, description, del\_flag, create\_time, update\_time, icon, is\_common, scene\_choose, name)  values  (#{entity.userId}, #{entity.userName}, #{entity.deptId}, #{entity.deptName}, #{entity.businessType}, #{entity.description}, #{entity.delFlag}, #{entity.createTime}, #{entity.updateTime}, #{entity.icon}, #{entity.isCommon}, #{entity.sceneChoose}, #{entity.name}) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | offerUserId | 提供用户id |  | | offerUserName | 提供用户名 |  | | offerDeptId | 提供机构id |  | | offerDeptName | 提供机构名称 |  | | evaluateDeptId | 被评价机构id |  | | productName | 评价产品名称 |  | | relevanceId | 关联id | 数据集、应用、模型等 | | score | 评分 |  | | evaluateContent | 评价内容 |  | | problemType | 问题类型 | 0-bug类，1-功能完善类，2-功能新增类 | | problemSummary | 问题概述 |  | | problemAttribution | 问题归因 |  | | evaluateType | 评价类型 | 0-客户评价，1-机构评价 | | createTime | 创建时间 |  | |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_evaluate、sc\_business\_scenario**>**  **操作：<修改>** |
| SQL语句：  【SQL17】编辑评价  update sc\_evaluate  <set>  <if test="vo.problemType != null">  problem\_type = #{vo.problemType},  </if>  <if test="vo.problemSummary != null">  problem\_summary = #{vo.problemSummary},  </if>  <if test="vo.problemAttribution != null">  problem\_attribution = #{vo.problemAttribution},  </if>  </set>  where id = #{vo.id}  【SQL18】修改业务场景数据  update sc\_business\_scenario <set>  <if test="vo.userId != null">  user\_id = #{vo.userId},  </if>  <if test="vo.userName != null">  user\_name = #{vo.userName},  </if>  <if test="vo.deptId != null">  dept\_id = #{vo.deptId},  </if>  <if test="vo.deptName != null">  dept\_name = #{vo.deptName},  </if>  <if test="vo.name != null">  name = #{vo.name},  </if>  <if test="vo.businessType != null">  business\_type = #{vo.businessType},  </if>  <if test="vo.description != null">  description = #{vo.description},  </if>  <if test="vo.updateTime != null">  update\_time = #{vo.updateTime},  </if>  <if test="vo.isCommon != null">  is\_common = #{vo.isCommon},  </if>  <if test="vo.sceneChoose != null ">  scene\_choose = #{vo.sceneChoose},  </if> </set> where id = #{vo.id}  【SQL19】场景数据修改删除状态  update sc\_business\_scenario set del\_flag = 0 where id = #{id}  【SQL20】发布后调用修改状态  update sc\_business\_scenario  set is\_common = 'yes'  where is\_common = #{dataSetId} |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | problemType | 0 | 页面配置评价类型 | 评价类型(0-客户评价，1-机构评价) | | problemSummary | null | 页面配置问题概述 | 问题概述 | | problemAttribution | null | 页面配置问题归因 | 问题归因 | | userId | 0 | 页面配置用户id | 用户id | | userName | null | 页面配置用户名 | 用户名 | | deptId | 0 | 页面配置用户所属机构id | 用户所属机构id | | deptName | null | 页面配置用户所属机构名称 | 用户所属机构名称 | | businessType | 0 | 页面配置场景类型 | 场景类型 | | name | null | 页面配置场景名称 | 场景名称 | | description | null | 页面配置场景备注 | 场景备注 | | delFlag | 0 | 页面配置删除标志 | 删除标志（1代表存在 0代表删除） | | isCommon | null | 页面配置是否公用 | 是否公用（yes是公用，id为机构私有） | | sceneChoose | null | 页面配置业务场景选择 | 业务场景选择 | | updateTime | null | 页面配置修改时间 | 修改时间 | |

#### 总体设计方案

##### 设计方案

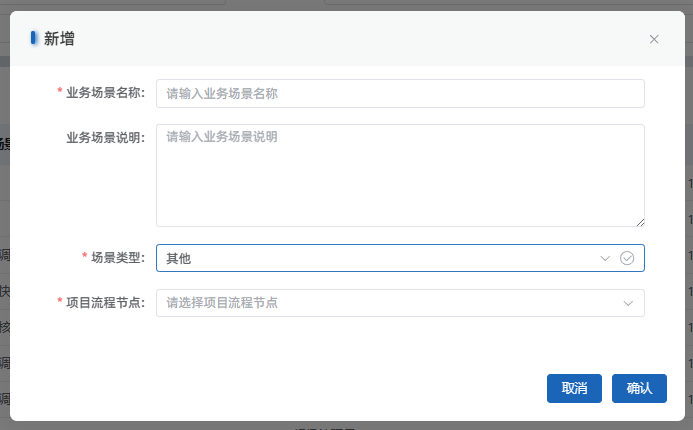
业务场景是操作的依赖，选择相对于的业务场景，方便进行区分数据集、项目、应用、模型等。

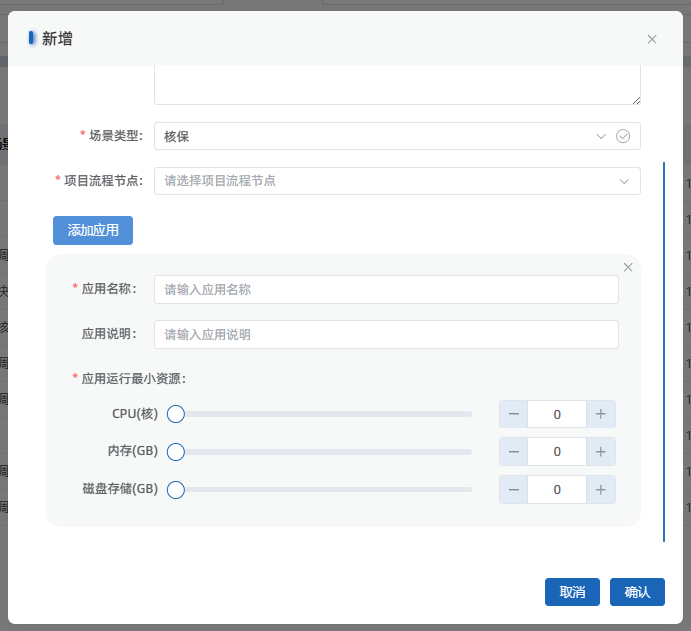
##### 处理流程

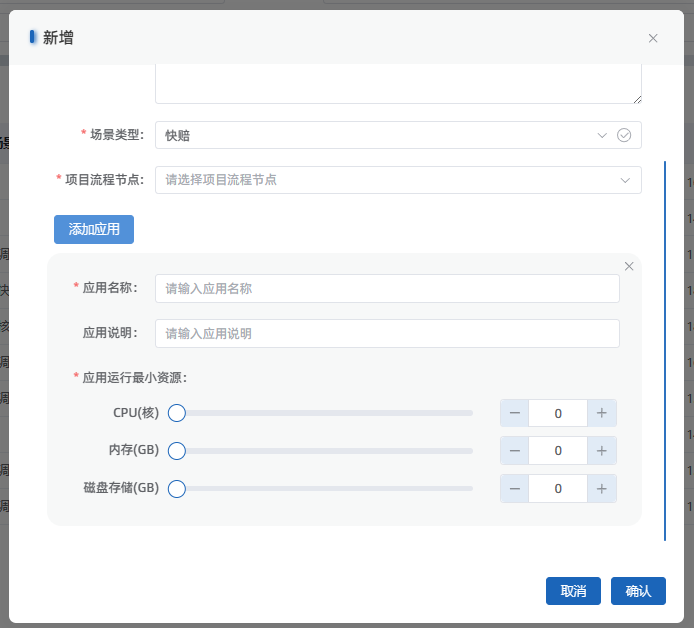
1.查询业务场景管理

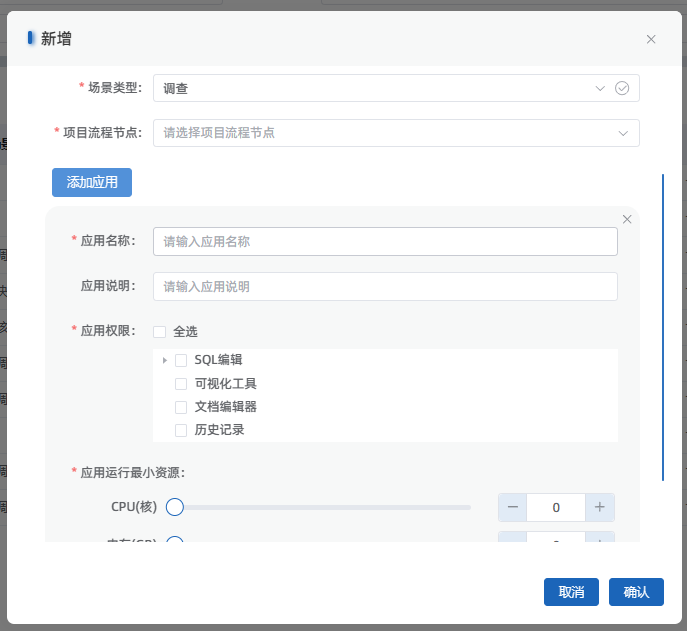


页面支持业务场景名称、业务场景说明、上传用户、上传时间作为查询条件，调用【6.3.6.1.1.1.2.4.查询业务场景数据】接口。接口将查询条件与分页信息作为查询条件执行【SQL13】分页获取业务场景列表，如果不为空则循环获取业务场景主键id执行【SQL14】查询绑定的应用返回页面展示。









新增业务场景：点击【新增】按钮，页面弹出新增业务场景配置弹框。业务场景配置业务场景名称、业务场景说明、场景类型、项目流程节点（控制项目中需要添加的内容），场景类型选择核保、调查、快赔时下方有添加应用的按钮，场景类型选择其他时不可添加应用，没有添加应用的按钮，点击添加应用按钮可以新增不同场景类型的应用（可以添加多个），且控制该应用的功能，调查场景类型应用（有应用权限的控制），核保/快赔场景类型应用（无应用权限的控制），配置完成后点击【确认】页面调用【6.3.6.1.1.1.2.1.新增业务场景数据】接口，接口执行【SQL15】进行业务场景新增操作。



修改业务场景：点击【修改】按钮，页面弹出修改业务场景配置弹框。修改对应业务场景名称、业务场景说明、场景类型、项目流程节点（控制项目中需要添加的内容）、应用配置，修改完成后点击【确认】页面调用【6.3.6.1.1.1.2.2.修改业务场景数据】接口，接口执行【SQL18】进行业务场景修改操作。

删除业务场景：在业务场景列表中可对未使用的业务场景进行删除操作，请求【2.10.1.1.2.2.3.软删除业务场景数据】接口，判断应用、数据集、模型是否被使用，使用则不能删除，不使用执行【SQL19】进行删除操作。

## 消息管理

消息管理组件的功能分解如下图：



### 消息管理

#### 类设计

##### 控制类

###### SysNoticeController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | SysNoticeController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 消息管理 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\controller\sysController\SysNoticeController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| selectNotice | session[:SysNoticeVo] | SysNoticeVo | ■公有 □私有 □保护 | Result | 查询消息 |
| updateNotice | session[:SysNoticeVo] | SysNoticeVo | ■公有 □私有 □保护 | Result | 标为已读、全部设为已读 |
| updateDelFlag | session[:id] | Long | ■公有 □私有 □保护 | Result | 删除消息 |
| insert | session[:SysNoticeVo] | SysNoticeVo | ■公有 □私有 □保护 | Result | 新增消息 |
| details | session[:Long] | id | ■公有 □私有 □保护 | Result | 查询消息详情 |
| insertDataNotice | session[:SysNoticeVo] | SysNoticeVo | ■公有 □私有 □保护 | Result | 向数据端发送消息 |
| insertDeptNotice | session[:SysNoticeVo] | SysNoticeVo | ■公有 □私有 □保护 | Result | 根据机构名称发送消息 |
| goApprove | session[:String] | taskId | ■公有 □私有 □保护 | Result | 去审批 |

算法描述

查询消息

接口描述：查询数据集、应用、项目、任务、服务、积分、资源审批、数据需求的消息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/notice/select | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| noticeType | [string] | 是 | 消息类型（审批授权 APPROVE，项目邀请 PROJECT-INVITATION，用户反馈 USER-FEEDBACK，监控告警 MONITORING-ALARM，账单结算 BILL-SETTLEMENT，数据需求 DATA-DEMAND） |
| status | [number] | 是 | 消息状态（0未读 1已读） |
| updateUserId | [number] | 是 | 通知用户id |
| page | [number] | 是 | 当前页 |
| pageSize | [number] | 是 | 页面大小 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总页数 |
| data>>totalNum | [number] | 是 | 总记录数量 |
| data>>list | [array] | 是 | 数据集合 |
| data>>list>>id | [number] | 是 | 消息ID |
| data>>list>>relevanceId | [number] | 是 | 关联id（项目、资源、数据集、模型等） |
| data>>list>>relevanceType | [number] | 是 | 关联类型（应用APP、数据集DATASet、模型MODEL、数据DATA、机构ORGANIZATION、资源RESOURCE） |
| data>>list>>noticeTitle | [string] | 是 | 消息标题 |
| data>>list>>noticeType | [string] | 是 | 消息类型（审批授权 APPROVE，项目邀请 PROJECT-INVITATION，用户反馈 USER-FEEDBACK，监控告警 MONITORING-ALARM，账单结算 BILL-SETTLEMENT，数据需求 DATA-DEMAND） |
| data>>list>>noticeContent | [string] | 是 | 消息内容 |
| data>>list>>status | [number] | 是 | 消息状态（0未读 1已读） |
| data>>list>>createUserId | [number] | 是 | 创建(申请)用户id |
| data>>list>>createUserName | [string] | 是 | 创建(申请)用户名称 |
| data>>list>>createDeptId | [number] | 是 | 创建(申请)机构id |
| data>>list>>createDeptName | [string] | 是 | 创建(申请)机构名称 |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateUserId | [number] | 是 | 更新者 |
| data>>list>>updateTime | [string] | 是 | 更新时间 |
| data>>list>>delFlag | [number] | 是 | 删除标志（1代表存在 0代表删除 |
| data>>list>>approvalStatus | [string] | 是 | 未通过(NOT-PASS),已通过(PASS),已撤销(REVOKE) |
| data>>list>>taskId | [string] | 是 | 流程实例id |

标为已读、全部设为已读

接口描述：把消息标为已读进行分类，也可以批量进行已读。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/notice/update | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 消息ID，单个已读传 |
| updateUserId | [number] | 是 | 通知用户id，全部已读传 |
| noticeType | [string] | 是 | 消息类型（审批授权 APPROVE，项目邀请 PROJECT-INVITATION，用户反馈 USER-FEEDBACK，监控告警 MONITORING-ALARM，账单结算 BILL-SETTLEMENT，数据需求 DATA-DEMAND） |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

删除信息

接口描述：把消息进行逻辑删除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/notice/updateDelFlag | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 消息id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

新增信息

接口描述：在数据集、应用、项目、任务、服务、积分、资源审批、数据需求的时候新增信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/notice/insert | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| noticeTitle | [string] | 是 | 消息标题 |
| relevanceId | [string] | 是 | 关联id（项目、资源、数据集、模型等） |
| relevanceType | [string] | 是 | 关联类型（应用APP、数据集DATASet、模型MODEL、数据DATA、机构ORGANIZATION、资源RESOURCE） |
| noticeType | [string] | 是 | 消息类型（审批授权 APPROVE，项目邀请 PROJECT-INVITATION，用户反馈 USER-FEEDBACK，监控告警 MONITORING-ALARM，账单结算 BILL-SETTLEMENT，数据需求 DATA-DEMAND） |
| updateUserId | [number] | 是 | 通知用户id |
| noticeContent | [string] | 是 | 消息内容 |
| createUserId | [number] | 是 | 创建用户id |
| createUserName | [string] | 是 | 创建(申请)用户名称 |
| createDeptId | [string] | 是 | 创建(申请)机构id |
| createDeptName | [string] | 是 | 创建(申请)机构名称 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

消息详情

接口描述：查询消息详情进行前端展示，并根据类型跳转。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/notice/details | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [number] | 是 | 消息id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>id | [number] | 是 | 消息ID |
| data>>relevanceId | [number] | 是 | 关联id（项目、资源、数据集、模型等） |
| data>>relevanceType | [number] | 是 | 关联类型（应用APP、数据集DATASet、模型MODEL、数据DATA、机构ORGANIZATION、资源RESOURCE） |
| data>>noticeTitle | [string] | 是 | 消息标题 |
| data>>noticeType | [string] | 是 | 消息类型（审批授权 APPROVE，项目邀请 PROJECT-INVITATION，用户反馈 USER-FEEDBACK，监控告警 MONITORING-ALARM，账单结算 BILL-SETTLEMENT，数据需求 DATA-DEMAND） |
| data>>noticeContent | [string] | 是 | 消息内容 |
| data>>status | [number] | 是 | 消息状态（0未读 1已读） |
| data>>createUserId | [number] | 是 | 创建(申请)用户id |
| data>>createUserName | [string] | 是 | 创建(申请)用户名称 |
| data>>createDeptId | [number] | 是 | 创建(申请)机构id |
| data>>createDeptName | [string] | 是 | 创建(申请)机构名称 |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateUserId | [number] | 是 | 更新者 |
| data>>updateTime | [string] | 是 | 更新时间 |
| data>>delFlag | [number] | 是 | 删除标志（1代表存在 0代表删除 |
| data>>points | [number] | 是 | 积分 |
| data>>approvalStatus | [string] | 是 | 未通过(NOT-PASS),已通过(PASS),已撤销(REVOKE) |
| data>>taskId | [string] | 是 | 流程实例id |

向数据端发送消息

接口描述：向数据端进行发送信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/notice/insertDataNotice | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceId | [number] | 是 | 关联id（项目、资源、数据集、模型等） |
| noticeContent | [string] | 是 | 消息内容 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

根据机构名称发送消息

接口描述：根据更新者的用户id查询对应的机构id并进行发送信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /system/notice/insertDeptNotice | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| relevanceId | [number] | 是 | 关联id（项目、资源、数据集、模型等） |
| updateUserId | [number] | 是 | 更新者 |
| noticeContent | [string] | 是 | 消息内容 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |

去审批

接口描述：根据消息详情，点击去审批跳转到审批页面中。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/notice/goApprove | 无 |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| taskId | [string] | 是 | 流程实例id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 | 提醒消息 |
| code | [number] | 是 | 状态码 |
| data | [array] | 是 | 数据详情 |
| data>>id | [number] | 是 | 主键id |
| data>>userId | [number] | 是 | 申请用户id |
| data>>userDept | [string] | 是 | 申请用户机构 |
| data>>userDeptId | [number] | 是 | 申请用户机构 |
| data>>userName | [string] | 是 | 申请用户名 |
| data>>approveRole | [string] | 是 | 审批用户角色 |
| data>>approveDept | [null] | 是 | 审批用户机构 |
| data>>approveId | [string] | 是 | 审批用户id |
| data>>approveDeptId | [null] | 是 | 申请用户机构 |
| data>>comment | [null] | 是 | 审批意见 |
| data>>taskId | [string] | 是 | 流程实例id |
| data>>taskName | [string] | 是 | 流程名称 |
| data>>relevanceContent | [null] | 是 | 流程关联详情内容 |
| data>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>relevanceSize | [null] | 是 | 流程关联额度 |
| data>>projectId | [null] | 是 | 项目id |
| data>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING,应用申请 APP-APPLY,数据集发布 DATA-PUBLISH,数据集申请 DATA-APPLY,模型上架 MODEL-LISTING,模型申请 MODEL-APPLY,数据使用申请 DATA-USE-APPLY,机构注册 ORGANIZATION-APPLY,资源申请 RESOURCE-APPLY) |
| data>>mode | [string] | 是 | 流程方式 |
| data>>flowId | [number] | 是 | 流程id |
| data>>flowDetailId | [number] | 是 | 节点id |
| data>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>createTime | [number] | 是 | 创建时间 |
| data>>updateTime | [number] | 是 | 更新时间 |
| data>>spendTime | [null] | 是 | 花费时间 |
| data>>content | [null] | 是 | 详情 |
| data>>projectName | [null] | 是 | 项目id |
| data>>approveDescription | [null] | 是 | 申请说明 |
| data>>approveName | [null] | 是 | 审批节点名称 |
| data>>elapsedTime | [null] | 是 | 耗时 |
| data>>projectDuration | [null] | 是 | 项目期限 |
| data>>projectResources | [null] | 是 | 项目资源 |

##### 实体类

###### SysNotice

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | SysNotice | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存消息的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\model\sys\po\SysNotice.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | |  | | 消息id |
| relevanceId | long | | ■公有 □私有 □保护 | | |  | | 关联id（项目、资源、数据集、模型等） |
| relevanceType | string | | ■公有 □私有 □保护 | | |  | | 关联类型（应用APP、数据集DATASet、模型MODEL、数据DATA、机构ORGANIZATION、资源RESOURCE） |
| noticeTitle | string | | ■公有 □私有 □保护 | | |  | | 消息标题 |
| noticeType | string | | ■公有 □私有 □保护 | | |  | | 消息类型（监控告警 MONITORING-ALARM,审批授权 APPROVE,资源更新 RESOURCE-RENEWAL,用户反馈 USER-FEEDBACK） |
| noticeContent | string | | ■公有 □私有 □保护 | | |  | | 消息内容 |
| status | string | | ■公有 □私有 □保护 | | |  | | 消息状态（0未读 1已读） |
| createUserId | long | | ■公有 □私有 □保护 | | |  | | 创建(申请)用户id |
| createUserName | string | | ■公有 □私有 □保护 | | |  | | 创建(申请)用户名称 |
| createDeptId | long | | ■公有 □私有 □保护 | | |  | | 创建(申请)机构id |
| createDeptName | string | | ■公有 □私有 □保护 | | |  | | 创建(申请)机构名称 |
| approvalStatus | string | | ■公有 □私有 □保护 | | |  | | 未通过(NOT-PASS),已通过(PASS),已撤销(REVOKE) |
| taskId | string | | ■公有 □私有 □保护 | | |  | | 流程实例id |
| updateUserId | long | | ■公有 □私有 □保护 | | |  | | 更新者 |
| delFlag | integer | | ■公有 □私有 □保护 | | |  | | 删除标志（1代表存在 0代表删除） |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 修改时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | sc\_sys\_notice | | |

算法描述

无

状态转换描述

无

#### SQL文一览

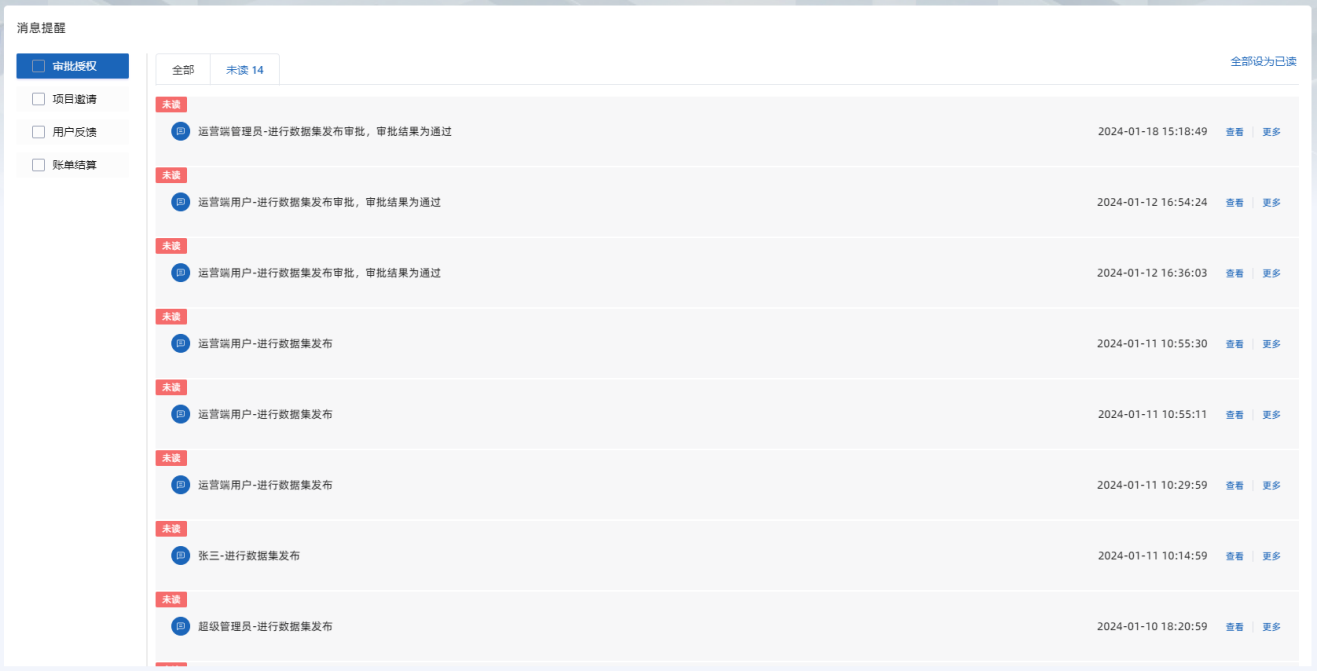
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**sc\_business\_scenario、sc\_sys\_notice、sc\_sys\_approve、sc\_sys\_use、 sc\_sys\_user\_role、 sc\_sys\_role\_dept、 sc\_sys\_dept**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询信息  select \* from sc\_business\_scenario  <where>  del\_flag = 1  <if test="noticeType!= null and noticeType!= ''">  and notice\_type =#{noticeType}  </if>  <if test="status!= null and status!= ''">  and statuslike concat #{status}  </if>  <if test="updateUserId!= null and updateUserId!= ''">  and update\_user\_id =#{updateUserId}  </if>  </where>  【SQL02】消息详情  select \*  from sc\_sys\_notice  where del\_flag = 1  and id = #{id}  【SQL03】统计时间范围内的数据集反馈数  SELECT  CASE WHEN u.a  <if test = 'operator == "greater" ' >  <![CDATA[>]]>  </if>  <if test = 'operator == "less" ' >  <![CDATA[<]]>  </if>  #{threshold} THEN 1 ELSE 0 END  from(  SELECT  count(id) as a  FROM  sc\_sys\_notice  <where>  notice\_type = 'USER-FEEDBACK'  and relevance\_type = 'DATASET'  <if test="stTime != null and stTime != 0 ">  and create\_time &gt;= #{stTime}  </if>  <if test="edTime !=null and edTime != 0 ">  and create\_time &lt;= #{edTime}  </if>  </where>  ) u  【SQL04】去审批  select \*  from sc\_sys\_approve  where task\_id = #{taskId} and find\_in\_set(#{userId}, approve\_id)  【SQL05】根据端获取所有的id  select u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password  , u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark  , d.dept\_name  from sc\_sys\_user u  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role\_dept rd on rd.role\_id = ur.role\_id  left join sc\_sys\_dept d on d.dept\_id = rd.dept\_id  where u.del\_flag = 1 and u.status = 1 and d.del\_flag = 1 and d.status = 1 AND d.parent\_id = 0  <if test="type != null and type != ''">  and d.type = #{type}  </if>  【SQL06】根据机构获取所有的用户id  select u.user\_id, u.dept\_id, u.login\_name, u.user\_name, u.user\_type, u.email, u.avatar, u.phonenumber, u.password  , u.sex, u.salt, u.status, u.del\_flag, u.login\_ip, u.login\_date, u.create\_user\_id, u.create\_time, u.remark  , d.dept\_name  from sc\_sys\_user u  left join sc\_sys\_user\_role ur on u.user\_id = ur.user\_id  left join sc\_sys\_role\_dept rd on rd.role\_id = ur.role\_id  left join sc\_sys\_dept d on d.dept\_id = rd.dept\_id  where u.del\_flag = 1 and u.status = 1 and d.del\_flag = 1 and d.status = 1 AND d.parent\_id = 0  <if test="deptId != null">  and d.dept\_id = #{deptId} |
|  |
| **程序流程中涉及的数据库表：<表名：**sc\_sys\_notice**>**  **操作：<插入>** |
| SQL语句：  【SQL07】新增消息  insert into sc\_sys\_notice (relevance\_id, relevance\_type, notice\_title, notice\_type, notice\_content, status,  create\_user\_id, create\_user\_name, create\_dept\_id, create\_dept\_name, create\_time,  update\_user\_id, update\_time, del\_flag, task\_id) values  <foreach item="vos" collection="vos" separator=",">  (#{vos.relevanceId}, #{vos.relevanceType}, #{vos.noticeTitle}, #{vos.noticeType}, #{vos.noticeContent}, #{vos.status},  #{vos.createUserId}, #{vos.createUserName},#{vos.createDeptId}, #{vos.createDeptName}, #{vos.createTime},  #{vos.updateUserId}, #{vos.updateTime}, #{vos.delFlag}, #{vos.taskId})  </foreach> |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | relevanceId | 关联id | 项目、资源、数据集、模型等 | | relevanceType | 关联类型 | 应用APP、数据集DATASet、模型MODEL、数据DATA、机构ORGANIZATION、资源RESOURCE | | noticeTitle | 消息标题 |  | | noticeType | 消息类型 | 监控告警 MONITORING-ALARM,审批授权 APPROVE,资源更新 RESOURCE-RENEWAL,用户反馈 USER-FEEDBACK | | noticeContent | 消息内容 |  | | status | 消息状态 | 0未读 1已读 | | createUserId | 创建(申请)用户id |  | | createUserName | 创建(申请)用户名称 |  | | createDeptId | 创建(申请)机构id |  | | createDeptName | 创建(申请)机构名称 |  | | approvalStatus | 未通过 | NOT-PASS),已通过(PASS),已撤销(REVOKE | | taskId | 流程实例id |  | | updateUserId | 更新者 | 0不允许，1允许 | | delFlag | 删除标志 | 1代表存在 0代表删除 | | createTime | 创建时间 |  | |
|  |
| **程序流程中涉及的数据库表：<表名：** sc\_sys\_notice**>**  **操作：<修改>** |
| SQL语句：  【SQL08】标为已读、全部设为已读  update sc\_sys\_notice  set status = 1  <where>  del\_flag = 1  <if test="vo.noticeType != null and vo.noticeType != ''">  and notice\_type = #{vo.noticeType}  </if>  <if test="vo.updateUserId != null and vo.updateUserId != ''">  and update\_user\_id = #{vo.updateUserId}  </if>  <if test="vo.id != null and vo.id != ''">  and id = #{vo.id}  </if>  </where>  【SQL09】修改删除状态  update sc\_sys\_notice  set del\_flag = 0  where id = #{id}  【SQL10】修改审批状态  update sc\_sys\_notice  set approval\_status = #{vo.approvalStatus}  <where>  del\_flag = 1  <if test="vo.taskId != null and vo.taskId != ''">  and task\_id = #{vo.taskId}  </if>  </where> |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | points | 0 | 页面配置积分 | 积分 | | privacyInquiry | 0 | 页面配置查询状态 | 隐私查询 | | visible | 0 | 页面配置可见性 | 数据可见性 | | updateTime | 创建时间 | 发布时间 | 更新时间 | |  |  |  |  | |

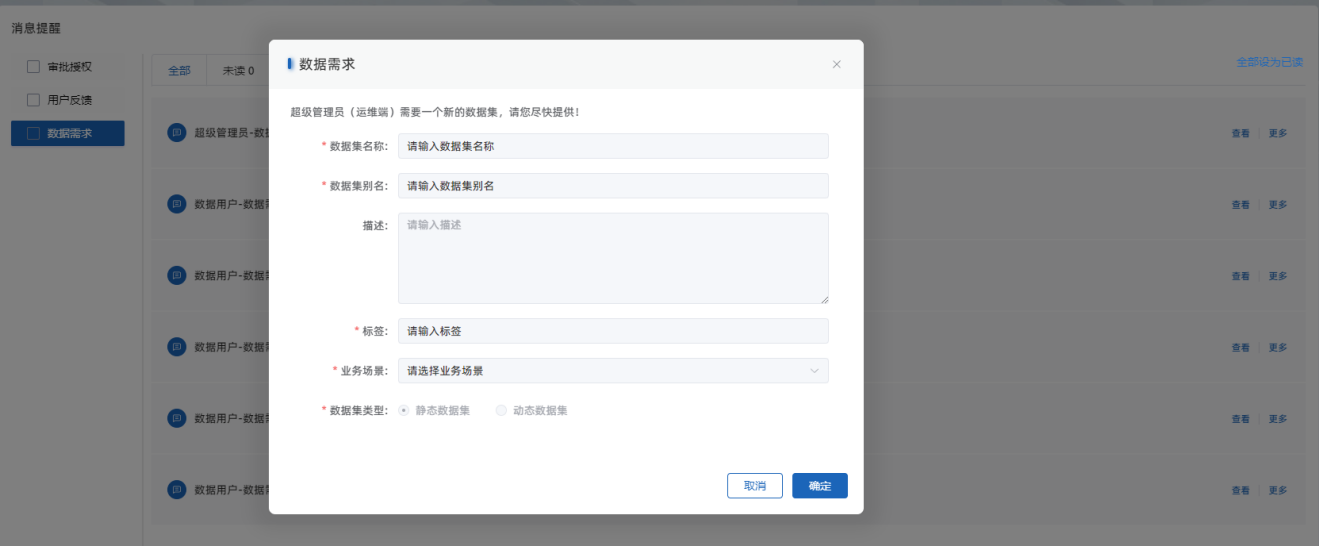
#### 总体设计方案

##### 设计方案

消息管理根据类型展示监控告警、审批授权、资源更新、用户反馈、数据需求五种类型消息。在消息类型基础上又设计了查看详情、修改已读状态、删除等基础功能。

##### 处理流程





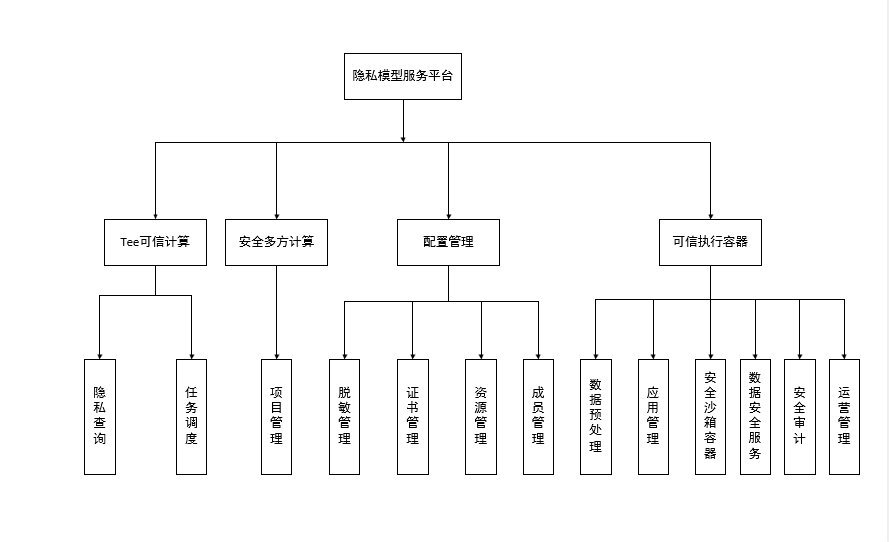
页面传递消息类型请求【6.4.1.1.1.1.2.1.查询消息】接口，接口执行【SQL01】返回消息列表，可进行查看和删除操作。点击【查看】按钮，弹出详情弹窗，页面传递数据id请求【6.4.1.1.1.1.2.5.消息详情】接口，接口执行【SQL02】返回数据详情,按照关联id进行匹配分类并返回。点击【去审批】按钮，页面传递实例id，执行【6.4.1.1.1.1.2.8.去审批】接口，接口执行【SQL04】返回前端并进行跳转审批页面。点击【删除】按钮，页面传递数据id请求【6.4.1.1.1.1.2.3.删除信息】，接口执行【SQL09】返回是否删除成功。点击【已读】按钮，页面传递的id请求【6.4.1.1.1.1.2.2.标为已读、全部设为已读】接口，接口执行【SQL08】数据修改为已读，点击【全部标为已读】按钮页面传递的ids请求【6.4.1.1.1.1.2.2.标为已读、全部设为已读】接口，接口执行【SQL08】这个用户的消息数据修全部改为已读状态。

# 隐私模型服务平台

## 功能模块逻辑描述

隐私模型服务平台提供三种隐私计算能力，包括可信执行容器、TEE可信计算环境和安全多方计算MPC，使得平台上的数据和应用能够在进行计算分析时得到隐私和安全保护。

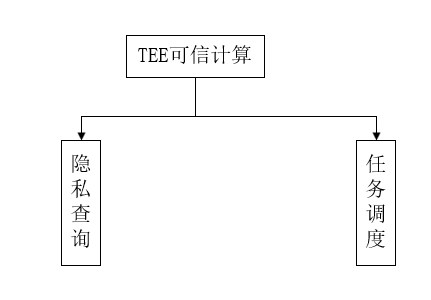
主要包含的功能如下：



## TEE可信计算

基于可信环境下安全计算模块，通过拖拽式数据集和数据预处理组件进行连接和参数设置形成数据分析模型。

主要包含的功能如下：



### 隐私查询

基于可信环境下安全计算模块，通过拖拽式数据集和数据预处理组件进行连接和参数设置形成数据分析模型。主要流程如下：



#### 类设计

##### 控制类

###### AnalysisModelController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | AnalysisModelController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 隐私查询任务管理操作 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\analysis\controller\AnalysisModelController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| addModel | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 创建任务 |
| baseupdate | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 修改任务信息 |
| getmodelobject | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 查看任务详情 |
| delete | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 任务 |

算法描述

创建任务

接口描述：页面传入隐私任务相关创建信息传递给后端，并存入数据库中，然后将创建好的信息回传给前端显示。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /add | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 是 | 任务名称 |
| description | [string] | 否 | 任务描述 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>id | [string] | 是 | 任务id |
| data>>name | [number] | 是 | 任务名称 |
| data>>runType | [number] | 是 | 执行类型 |
| data>>status | [number] | 是 | 任务状态 |
| data>>types | [number] | 是 | 任务类型 |
| data>>creatorName | [number] | 是 | 创建者名称 |
| Data>>createdTime | [string] | 是 | 创建时间 |
| data>>modifiedTime | [string] | 是 | 更新时间 |

修改任务信息

接口描述：页面传入任务更新信息传递给后端，并存入数据库中，然后将修改后的信息回传给前端显示。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /baseupdate | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 是 | 任务名称 |
| description | [string] | 否 | 任务描述 |
| Id | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>id | [string] | 是 | 任务id |
| data>>name | [number] | 是 | 任务名称 |
| data>>runType | [number] | 是 | 执行类型 |
| data>>status | [number] | 是 | 任务状态 |
| data>>types | [number] | 是 | 任务类型 |
| data>>creatorName | [number] | 是 | 创建者名称 |
| Data>>createdTime | [string] | 是 | 创建时间 |
| data>>modifiedTime | [string] | 是 | 更新时间 |

查看任务信息

接口描述：查看任务相关信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /info | String |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Id | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

**删除任务**

接口描述：删除已创建的任务。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /delete | String |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Id | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

###### AnalysisModelTaskController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | AnalysisModelTaskController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 隐私查询任务执行操作 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\analysis\controller\AnalysisModelTaskController  .java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| tasklist | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 任务执行列表 |
| dealmodeldata | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 执行任务 |
| querymodeldata | 无 | 无 | ■公有 □私有 □保护 | AjaxResult | 数据预览 |

算法描述

任务列表

接口描述：获取任务列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /tasklist | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| currentPage | [number] | 是 | 分页页数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>id | [string] | 是 | 任务id |
| data>>list>>name | [number] | 是 | 任务名称 |
| data>>list>>runType | [number] | 是 | 执行类型 |
| data>>list>>status | [number] | 是 | 任务状态 |
| data>>list>>types | [number] | 是 | 任务类型 |
| data>>list>>creatorName | [number] | 是 | 创建者名称 |
| data>>list>>createdTime | [string] | 是 | 创建时间 |
| data>>list>>modifiedTime | [string] | 是 | 更新时间 |

执行任务

接口描述：执行任务配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /dealmodeldata | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 任务id |
| file | [file] | 是 | 证书文件 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

数据预览

接口描述：获取任务执行数据列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /querymodeldata | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

##### 实体类

###### TTruModelDo

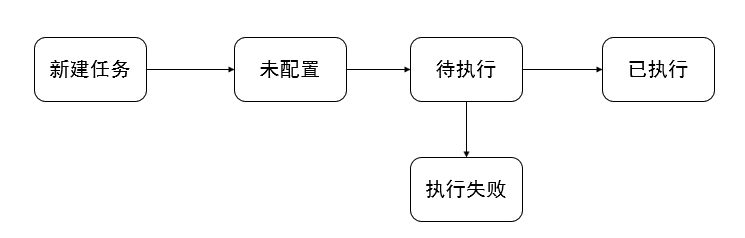
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TTruModelDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存隐私查询任务的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\bydb\TTruModelDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| modelType | Iznteger | | ■公有 □私有 □保护 | | |  | | 可视化组件类型 1为纵向，0为横向 |
| runSql | string | | ■公有 □私有 □保护 | | |  | | sql语句 |
| buildType | integer | | ■公有 □私有 □保护 | | |  | | 构建类型 1为可视化组件，0为编写sql |
| description | string | | ■公有 □私有 □保护 | | |  | | 模型详情 |
| tags | string | | ■公有 □私有 □保护 | | |  | | 标签 |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 模型名称 |
| total | integer | | ■公有 □私有 □保护 | | |  | | 模型数据输出大小值 |
| folderId | string | | ■公有 □私有 □保护 | | |  | | 路径id |
| types | integer | | ■公有 □私有 □保护 | | |  | | 1为离线，0为实时 |
| disable | integer | | ■公有 □私有 □保护 | | |  | | 是否显示 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 任务状态 0失败，1成功，2正在运行，3等待，4其他，9删除 |
| nodePartyId | string | | ■公有 □私有 □保护 | | | NNU | | 节点id |
| synFlag | integer | | ■公有 □私有 □保护 | | |  | | 上传标志 1已上传，3待上传 |
| config | string | | ■公有 □私有 □保护 | | |  | | 模型类型 olk，spark，flink |
| dcId | string | | ■公有 □私有 □保护 | | |  | | 节点id |
| cacheFlag | integer | | ■公有 □私有 □保护 | | |  | | 是否缓存 1缓存，0不缓存 |
| runType | integer | | ■公有 □私有 □保护 | | |  | | 运行类型 1为手动运行，0为调度运行 |
| runCorn | string | | ■公有 □私有 □保护 | | |  | | 运行频率 |
| lastRunTime | date | | ■公有 □私有 □保护 | | |  | | 最后运行时间 |
| viewName | string | | ■公有 □私有 □保护 | | |  | | 视图名称 |
| cacheTaskNo | string | | ■公有 □私有 □保护 | | |  | | 缓存编号 |
| cacheTableName | string | | ■公有 □私有 □保护 | | |  | | 缓存表 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建者id |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建者账号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| createDeptNo | string | | ■公有 □私有 □保护 | | |  | | 归属部门编号 |
| createDeptNa | string | | ■公有 □私有 □保护 | | |  | | 归属部门名称 |
| outputType | integer | | ■公有 □私有 □保护 | | |  | | 输出组件类型 |
| outputId | string | | ■公有 □私有 □保护 | | |  | | 输出组件id |
| paramConfig | string | | ■公有 □私有 □保护 | | |  | | sql语句字段 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_tru\_model | | |

算法描述

无

状态转换描述



###### TTruModelElementDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TTruModelElementDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存任务节点的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\bydb\TTruModelElementDo | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| config | String | | ■公有 □私有 □保护 | | |  | | 组件配置 |
| creatorAccount | String | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorId | String | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatorName | String | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| element | String | | ■公有 □私有 □保护 | | |  | | 模型编码 |
| elementType | Integer | | ■公有 □私有 □保护 | | |  | | 组件类型 1为表 0为系统组件 |
| icon | String | | ■公有 □私有 □保护 | | |  | | 图标 |
| modelId | String | | ■公有 □私有 □保护 | | | NNU | | 项目ID |
| name | String | | ■公有 □私有 □保护 | | |  | | 组件名称 |
| origName | String | | ■公有 □私有 □保护 | | |  | | 原名称 |
| runSql | String | | ■公有 □私有 □保护 | | |  | | 运行SQL语句 |
| remark | String | | ■公有 □私有 □保护 | | |  | | 备注 |
| runOrder | Integer | | ■公有 □私有 □保护 | | |  | | 运行顺序 |
| tableSql | String | | ■公有 □私有 □保护 | | |  | | flink数据源语句 |
| runStatus | Integer | | ■公有 □私有 □保护 | | |  | | 运行状态 -1代表配置未保存无法运行，0配置保存未审核，1可以运行 |
| shape | String | | ■公有 □私有 □保护 | | |  | | 图标形状 |
| tcId | String | | ■公有 □私有 □保护 | | | NNU | | 实例化表或者组件ID |
| total | Integer | | ■公有 □私有 □保护 | | |  | | 限制条数 |
| x | String | | ■公有 □私有 □保护 | | |  | | x坐标 |
| y | String | | ■公有 □私有 □保护 | | |  | | y坐标 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_tru\_model\_element | | |

算法描述

无

状态转换描述



###### TTruModelElementRelDo

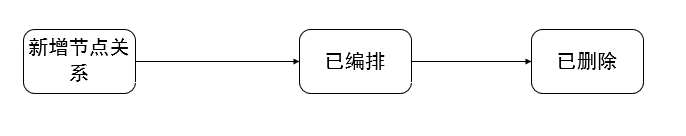
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TTruModelElementRelDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存任务节点边的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\bydb\TTruModelElementRelDo | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建了姓名 |
| endElementId | string | | ■公有 □私有 □保护 | | |  | | 目标元素ID |
| endPortId | string | | ■公有 □私有 □保护 | | | NNU | | 终点瞄点 |
| modelId | string | | ■公有 □私有 □保护 | | | NNU | | 模型ID |
| startElementId | string | | ■公有 □私有 □保护 | | | NNU | | 起始元素ID |
| startPortId | string | | ■公有 □私有 □保护 | | |  | | 起始瞄点 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_tru\_model\_element\_rel | | |

算法描述

无

状态转换描述



###### TTruModelField

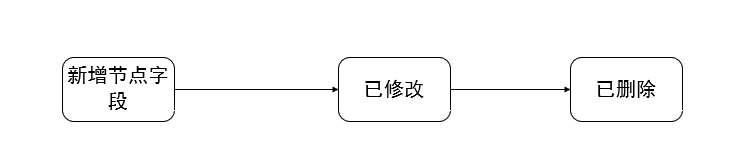
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TTruModelField | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存任务组件字段的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\bydb\TTruModelField,java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| elementId | string | | ■公有 □私有 □保护 | | | NNU | | 模型ID |
| extendsId | string | | ■公有 □私有 □保护 | | |  | | 继承字段的上级节点 |
| fromFieldId | string | | ■公有 □私有 □保护 | | |  | | 来源字段id |
| tableId | string | | ■公有 □私有 □保护 | | | NNU | | 表id |
| fieldExpr | string | | ■公有 □私有 □保护 | | |  | |  |
| fieldAlias | string | | ■公有 □私有 □保护 | | |  | | 字段别名 |
| tableAlias | string | | ■公有 □私有 □保护 | | |  | | 继承所属表别名 |
| fieldName | string | | ■公有 □私有 □保护 | | |  | | 字段名称 |
| aggregation | string | | ■公有 □私有 □保护 | | |  | |  |
| orderFunc | string | | ■公有 □私有 □保护 | | |  | | 排序方式 desc asc |
| fieldType | string | | ■公有 □私有 □保护 | | |  | | 类型默认值 |
| origFieldType | string | | ■公有 □私有 □保护 | | |  | | 类型默认值 |
| origFlag | Integer | | ■公有 □私有 □保护 | | |  | | 原字段 1是 0增加 |
| columnType | string | | ■公有 □私有 □保护 | | |  | | 对应java类型类型 |
| filterConfig | string | | ■公有 □私有 □保护 | | |  | | 过滤条件 |
| filterType | string | | ■公有 □私有 □保护 | | |  | | 和前一个字段过滤 连接方式 （and、or 等） |
| filterStatus | Integer | | ■公有 □私有 □保护 | | |  | | 是否启用筛选 1 是 0 否 |
| filterSort | Integer | | ■公有 □私有 □保护 | | |  | | 过滤字段排序 |
| filterValue | string | | ■公有 □私有 □保护 | | |  | | 过滤筛选值 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人帐号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| fieldLength | string | | ■公有 □私有 □保护 | | |  | |  |
| isSelect | Integer | | ■公有 □私有 □保护 | | |  | | 是否选用 1 是 0 否 |
| isPrimary | Boolean | | ■公有 □私有 □保护 | | |  | | 主键标识 |
|  |  | |  | | |  | |  |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_tru\_model\_field | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**t\_tru\_model  、t\_tru\_model\_element、t\_tru\_model\_element\_rel、t\_tru\_model\_field **>**  **操作：<查询>** |
| SQL语句：  【SQL01】通过用户账号查询任务列表  SELECT \* from t\_tru\_model where 1=1 and creator\_account = ? and create\_dept\_no = ? and status!=9 order by modified\_time desc limit 0,10  【SQL02】根据任务ID查询任务信息  SELECT created\_time,modified\_time,id,model\_type,run\_sql,build\_type,description,tags,name,total,folder\_id,types,disable,status,node\_party\_id,syn\_flag,config,dc\_id,cache\_flag,run\_type,run\_corn,last\_run\_time,view\_name,cache\_task\_no,cache\_table\_name,creator\_id,creator\_account,creator\_name,create\_dept\_no,create\_dept\_na,output\_type,output\_id,param\_config FROM t\_tru\_model WHERE id = ?  【SQL03】根据任务ID查询任务节点信息  SELECT \* FROM t\_tru\_model\_element WHERE model\_id = ?  【SQL04】根据任务ID查询任务节点关系信息  SELECT \* FROM t\_tru\_model\_element\_rel WHERE model\_id = ?  【SQL05】根据节点ID查询节点字段信息  SELECT \* from t\_bydb\_field where 1=1 and object\_id = ? and enable = ? order by norder |
|  |
| **程序流程中涉及的数据库表：<表名：**t\_tru\_model  **>**  **操作：<插入>** |
| SQL语句：  【SQL06】插入任务信息  INSERT INTO t\_tru\_model ( created\_time,modified\_time,id,model\_type,run\_sql,build\_type,description,tags,name,total,folder\_id,types,disable,status,node\_party\_id,syn\_flag,config,dc\_id,cache\_flag,run\_type,run\_corn,last\_run\_time,view\_name,cache\_task\_no,cache\_table\_name,creator\_id,creator\_account,creator\_name,create\_dept\_no,create\_dept\_na,output\_type,output\_id,param\_config ) VALUES( ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,? ) |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 任务id |  | | name | 任务名称 |  | | node\_party\_id | 节点id |  | | status | 任务状态 |  | | created\_time | 创建时间 |  | | creator\_id | 用户id |  | |
|  |

#### 总体设计方案

##### 设计方案

根据用户需求，点击新建任务按钮，在弹出框中填入任务名称等相关信息进行创建，创建成功刷新任务列表页面。

##### 处理流程



页面默认无查询条件请求【7.2.1.1.2.2.1.隐私查询-查询任务列表】接口执行【SQL01】分页获取系统的数据资产。接口内部发起HTTP请求调用隐私计算平台隐私查询列表接口分页获取其所有数据资产列表返回前端展示。分别展示任务名称、运行类型、更新时间等字段。

页面输入任务名称作为查询条件，点击【查询】按钮，页面重新请求【7.2.1.1.2.2.1.隐私查询-查询用户下的任务表】接口执行【SQL01】，并将数据名称作为参数调用隐私计算平台隐私查询列模糊搜索获取对应任务名称的是任务列表。

### 任务调度

任务调度为用户提供隐私查询历史任务情况信息，展示不同任务的执行日志。主要流程如下：



#### 类设计

##### 控制类

###### OcclumProcessDefinitionController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | OcclumProcessDefinitionController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 任务调度主页 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\fedral\controller\occlum  \OcclumProcessDefinitionController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| queryProcessDefinitionListPaging | session[:DataHomeVo] | DataHomeVo | ■公有 □私有 □保护 | Result | 获取任务调度列表 |
| queryProcessInstanceList | session[:DataHomeVo] | DataHomeVo | ■公有 □私有 □保护 | Result | 获取任务调度工作流实例列表 |
| queryTaskListPaging | session[:DataHomeVo] | DataHomeVo | ■公有 □私有 □保护 | Result | 获取任务调度任务实例列表 |

算法描述

任务配置

接口描述：展示任务配置相关的数据。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /projects/occlum/process/list-paging | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>name | [string] | 是 | 任务名称 |
| data>>list>>flag | [string] | 是 | 任务标识 |
| data>>list>>createTime | [string] | 是 | 创建时间 |
| data>>list>>projectId | [string] | 是 | 项目id |
| data>>list>>updateTime | [string] | 是 | 更新时间 |

工作流实例

接口描述：展示工作流实例相关的数据。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /projects/occlum/instance/list-paging | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>name | [string] | 是 | 任务名称 |
| data>>list>>state | [string] | 是 | 任务标识 |
| data>>list>>endTime | [string] | 是 | 结束时间 |
| data>>list>>projectId | [string] | 是 | 项目id |
| data>>list>>startTime | [string] | 是 | 开始时间 |

任务实例

接口描述：展示任务实例相关的数据。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /projects/occlum/task-instance/list-paging | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 | 无 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>name | [string] | 是 | 任务名称 |
| data>>list>>state | [string] | 是 | 任务标识 |
| data>>list>>endTime | [string] | 是 | 结束时间 |
| data>>list>>projectId | [string] | 是 | 项目id |
| data>>list>>startTime | [string] | 是 | 开始时间 |

##### 实体类

###### ProcessInstance

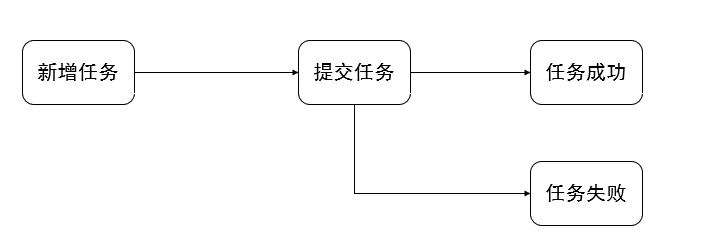
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ProcessInstance | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存任务任务的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\entity\ProcessInstance.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | int | | ■公有 □私有 □保护 | | | NNU | | id |
| code | long | | ■公有 □私有 □保护 | | |  | | 编码 |
| name | string | | ■公有 □私有 □保护 | | |  | | 任务名称 |
| version | int | | ■公有 □私有 □保护 | | |  | | version |
| projectCode | long | | ■公有 □私有 □保护 | | |  | | 项目编码 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| globalParams | string | | ■公有 □私有 □保护 | | |  | | 配置 |
| createTime | date | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateTime | date | | ■公有 □私有 □保护 | | |  | | 更新时间 |
| userId | int | | ■公有 □私有 □保护 | | |  | | 用户id |
| userName | string | | ■公有 □私有 □保护 | | |  | | 用户名称 |
| projectName | string | | ■公有 □私有 □保护 | | |  | | 模型名称 |
| timeout | int | | ■公有 □私有 □保护 | | |  | | 超时时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_ds\_process\_instance | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**t\_ds\_process\_instance  **>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询任务配置信息  SELECT start\_time FROM t\_ds\_process\_instance WHERE process\_definition\_code = ? ORDER BY id DESC |
|  |

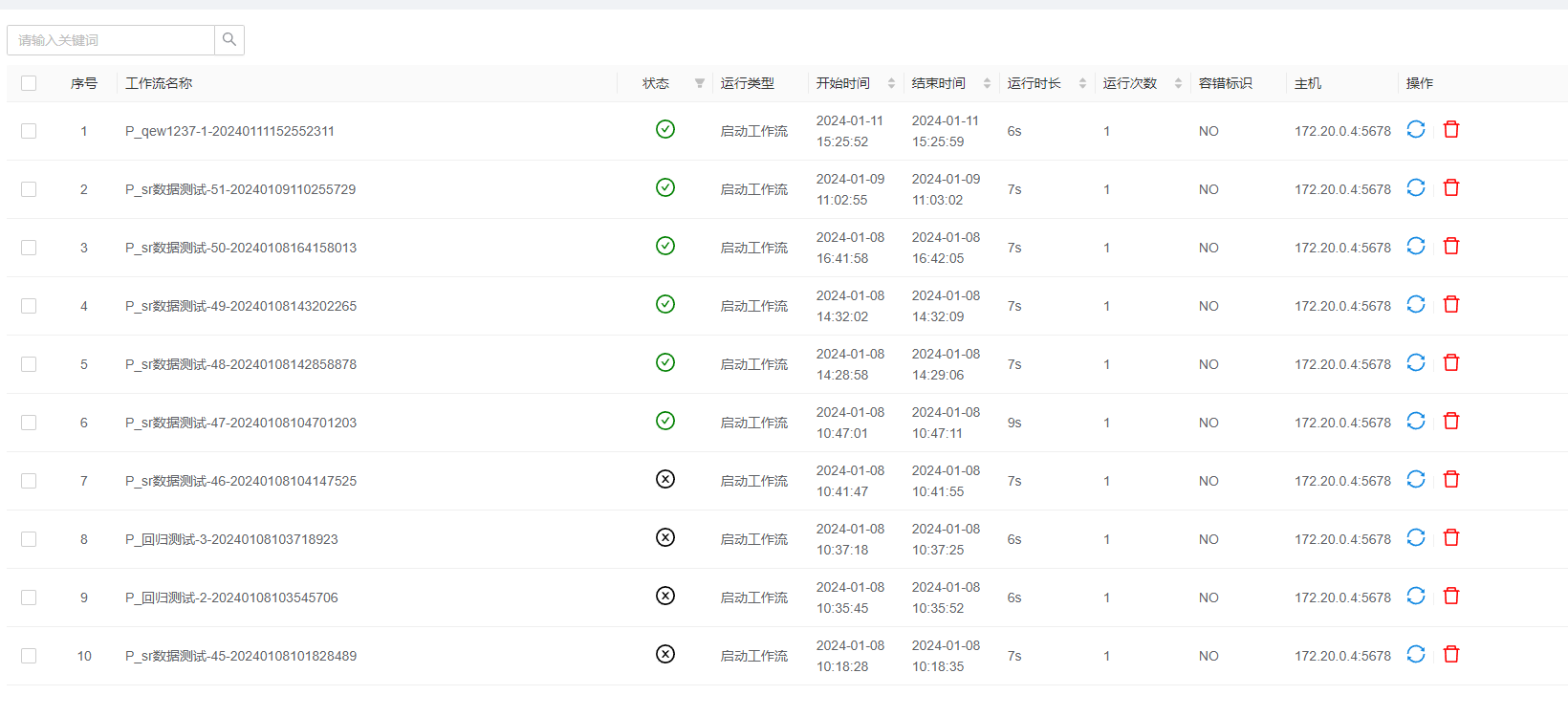
#### 总体设计方案

##### 设计方案

提供项目以及任务流程的管理功能，可以查看、新建、删除、编辑和执行项目的任务流程，并且支持任务的日志分析查看。

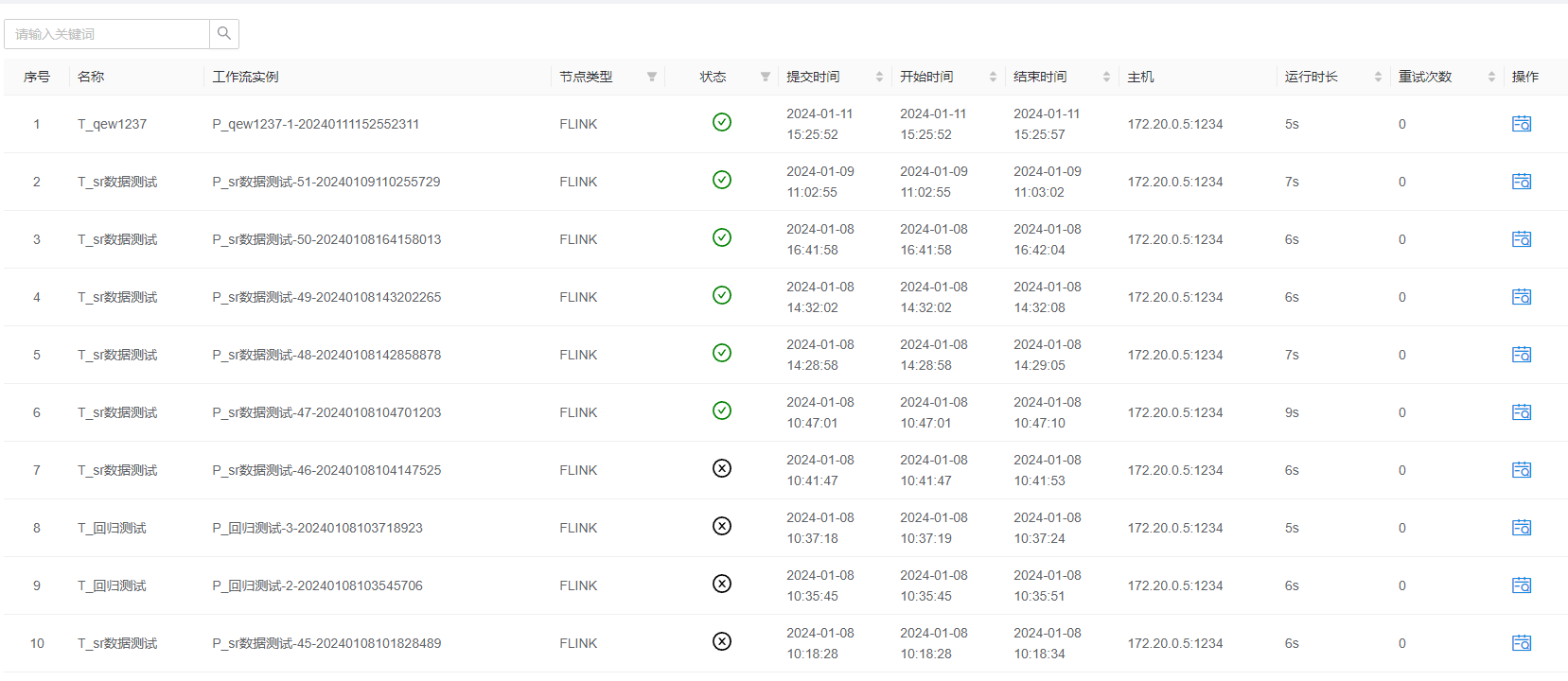
##### 处理流程

1. 根据不同类型展示任务工作流实例基本信息



执行【SQL01】，请求【2.3.1.1.1.2.1.任务调度-工作流实例接口列表】接口获取任务基本信息包含工作流名称、状态、运行类型、开始时间、结束时间、运行次数等数据。

1. 根据不同类型展示任务实例信息

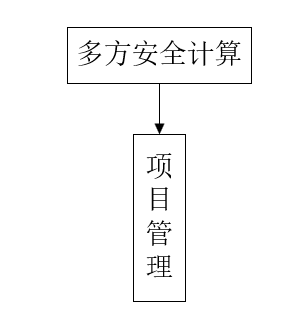


执行【SQL01】，请求【2.3.1.1.1.2.1.任务调度-任务实例接口列表】获取任务基本信息包含名称、节点类型、状态、提交时间、开始时间、运行时长等数据。

## 安全多方计算

基于多方安全计算的组件，提供项目以及任务流程的管理功能，可以查看、新建、删除、编辑和执行项目的任务流程，并且支持任务的日志分析查看。

主要包含的功能如下：



### 项目管理

项目管理为用户提供项目、流程、数据集等资产的管理。主要流程如下：



#### 类设计

##### 控制类

###### ProjectController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ProjectController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 多方安全项目中心管理相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/fedral/controller/fedral/ProjectController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | ■公有 □私有 □保护 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| info | session[:id] | string | ■公有 □私有 □保护 | Result | 项目详情 |
| page | session[:ProjectVo] | ProjectVo | ■公有 □私有 □保护 | Result | 项目列表 |
| update | session[:ProjectVo] | ProjectVo | ■公有 □私有 □保护 | Result | 项目更新 |
| addModel | session[:ProjectVo  ] | ProjectVo | ■公有 □私有 □保护 | Result | 项目添加 |
| delete | session[:id] | String | ■公有 □私有 □保护 | Result | 删除项目 |

算法描述

项目管理-项目详情

接口描述：根据项目id查询项目详情。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /project/info?id={id} | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 项目id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **示例** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>creatorId | [string] | 是 | 用户id |
| data>>host | [string] | 是 | 发起方信息 |
| data>>guests | [string] | 是 | 协助方信息 |
| data>>name | [string] | 是 | 项目名称 |
| data>>id | msg | [string] | 是 |
| data>>modifiedTime | data | [string] | 是 |
| data>>createdTime | success | [boolean] | 是 |
| data>>description | [string] | 是 | 描述 |
| data>>types | [number] | 是 | 类型 |
| data>>icon | [string] | 是 | 应用图标 |
| data>>icon | [string] | 是 | 图标 |

项目管理-项目列表

接口描述：根据各个登录用户获取该用户创建的项目资产，支持根据项目名称、等方式筛选。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /project/page | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| currentPage | [number] | 是 | 页数 |
| pageSize | [number] | 是 | 页码 |
| Name | [string] | 是 | 项目名称 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>creatorId | [string] | 是 | 用户id |
| data>>list>>host | [string] | 是 | 发起方信息 |
| data>>list>>guests | [string] | 是 | 协助方信息 |
| data>>list>>name | [string] | 是 | 项目名称 |
| data>>list>>id | msg | [string] | 是 |
| data>>list>>modifiedTime | data | [string] | 是 |
| data>>list>>createdTime | success | [boolean] | 是 |
| data>>list>>description | [string] | 是 | 描述 |
| data>>list>>types | [number] | 是 | 类型 |
| data>>list>>icon | [string] | 是 | 应用图标 |
| page | [number] | 是 | 页数 |
| total | [number] | 是 | 数据 |

项目管理-项目添加

接口描述：根据用户填写的项目信息为该用户创建的项目资产。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /project/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 是 | 项目名称 |
| description | [string] | 是 | 项目描述 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-项目更新

接口描述：根据用户填写的项目信息为选中的应用项目资产进行数据更新。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /project/update | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 是 | 项目名称 |
| description | [string] | 是 | 项目备注 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-项目删除

接口描述：根据用户选中的应用资产进行数据删除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /application/delete/?id={id} | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 应用id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |

###### ModelController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ModelController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成项目流程存储、数据检索、数据更新相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/fedral/controller/fedral/ModelController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
|  |  |  | | |  |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| info | session[:id] | long | ■公有 □私有 □保护 | Result | 项目管理-模型流程详情 |
| page | session[:FModelDo  ] | FModelDo | ■公有 □私有 □保护 | Result | 项目管理-模型列表 |
| update | session[:FModelDo] | FModelDo | ■公有 □私有 □保护 | Result | 项目管理-更新模型 |
| add | session[:FModelDo] | FModelDo | ■公有 □私有 □保护 | Result | 项目管理-创建项目流程 |
| delete | session[:id] | long | ■公有 □私有 □保护 | Result | 项目管理-删除项目流程 |
| addelement | session[:ModelElementVto  ] | ModelElementVto | ■公有 □私有 □保护 | Result | 项目管理-添加流程点 |
| addelementrel | session[:FModelElementRelDo  ] | FModelElementRelDo | ■公有 □私有 □保护 | Result | 项目管理-添加流程边 |
| getelements | session[:string  ] | string | ■公有 □私有 □保护 | Result | 项目管理-流程编排信息 |
| elementconfiginfo | session[:string  ] | string | ■公有 □私有 □保护 | Result | 项目管理-查看流程中点模板配置 |
| deleteelement | session[:string  ] | string | ■公有 □私有 □保护 | Result | 项目管理-删除流程中的节点 |

算法描述

项目管理-获取项目流程详情

接口描述：用户项目管理页面，可以点击某个项目查看流程详细信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /model/info?id={id} | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| **id** | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>name | [string] | 是 | 流程名称 |
| data>>id | [string] | 是 | 流程id |
| data>>createdTime | [string] | 是 | 创建时间 |
| data>>creatorId | [string] | 是 | 用户id |
| data>>modifiedTime | [string] | 是 | 修改时间 |
| data>>projectId | [string] | 是 | 项目id |
| data>>types | [number] | 是 | 流程类型 |
| data>>staus | [string] | 是 | 流程状态 |

项目管理-获取流程列表

接口描述：用户项目管理页面，可以查看流程列表信息，可以根据流程名称查询。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/page | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| name | [string] | 否 | 流程名称 |
| currentPage | [number] | 是 | 页码 |
| pageSize | [number] | 是 | 页数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list>>name | [string] | 是 | 流程名称 |
| data>>list>>id | [string] | 是 | 流程id |
| data>>list>>createdTime | [string] | 是 | 创建时间 |
| data>>list>>creatorId | [string] | 是 | 用户id |
| data>>list>>modifiedTime | [string] | 是 | 修改时间 |
| data>>list>>projectId | [string] | 是 | 项目id |
| data>>list>>types | [number] | 是 | 流程类型 |
| data>>list>>status | [string] | 是 | 流程状态 |
| page | [number] | 是 | 页数 |
| total | [number] | 是 | 数据 |

项目管理-流程删除

接口描述：用户项目管理下详情页面，可以点击某个流程进行删除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /model/delete/id={id} | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| **id** | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程添加

接口描述：用户项目管理页面，可以创建流程。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/add | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 | 无 |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程修改

接口描述：用户项目管理页面，用户可以对需要修改的流程进行修改。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/update | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| description | [string] | 否 | 流程描述 |
| name | [string] | 否 | 流程名称 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程点添加

接口描述：用户进入流程编排页面，新增组件节点。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/addelement | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| shape | [string] | 否 | 节点类型 |
| name | [string] | 否 | 节点名称 |
| x | [string] | 否 | 节点横坐标 |
| y | [string] | 否 | 节点纵坐标 |
| modelId | [string] | 否 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程边添加

接口描述：用户进入流程编排页面，新增组件节点之间的边。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/addelementrel | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| endElementId | [string] | 否 | 结束节点id |
| startElementId | [string] | 否 | 开始节点id |
| modelId | [string] | 否 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程编排信息

接口描述：用户进入流程编排页面，获取页面点和边信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /model/getelements | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回信息 |
| data | [string] | 是 | 接口返回数据 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>id | [string] | 是 | 接口返回数据 |
| data>>createdTime | [string] | 是 | 创建时间 |
| data>>creatorId | [string] | 是 | 用户id |
| data>>creatorName | [string] | 是 | 用户名称 |
| data>>currentPage | [number] | 是 | 页码 |
| data>>description: | [string] | 是 | 描述 |
| data>>disable | [number] | 是 | 是否显示标识 |
| data>>hasTemplate | [number] | 是 | 模板类型 |
| data>>id | [string] | 是 | 接口返回数据 |
| data>>model | [boolean] | 是 | 模型类型 |
| data>>modifiedTime | [string] | 是 | 修改时间 |
| data>>name | [string] | 是 | 流程名称 |
| data>>pageSize | [number] | 是 | 页数 |
| data>>projectId | [string] | 是 | 项目id |
| data>>status | [number] | 是 | 流程状态 |
| node |  |  |  |
| edge |  |  |  |
| node>>list>>componentId | [string] | 是 | 组件id |
| node>>list>>componentType | [string] | 是 | 组件类型 |
| node>>list>>id | [string] | 是 | 节点id |
| node>>list>>modelId | [string] | 是 | 流程id |
| node>>list>>name | [string] | 是 | 节点名称 |
| node>>list>>shape | [string] | 是 | 节点形状 |
| node>>list>>types | [string] | 是 | 节点类型 |
| node>>list>>x | [string] | 是 | 节点横坐标 |
| node>>list>>y | [string] | 是 | 节点纵坐标 |
| edge>>list>>endElementId | [string] | 是 | 起始节点 |
| edge>>list>>startElementId | [string] | 是 | 结束节点 |
| edge>>list>>id | [string] | 是 | 边id |
| edge>>list>>modelId | [string] | 是 | 流程id |

项目管理-流程点信息

接口描述：用户进入流程编排页面，获取页面点配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /model/elementdataconfiginfo | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| elementId | [string] | 是 | 流程点id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| node>>componentId | [string] | 是 | 组件id |
| node>>componentType | [string] | 是 | 组件类型 |
| node>>id | [string] | 是 | 节点id |
| node>>modelId | [string] | 是 | 流程id |
| node>>name | [string] | 是 | 节点名称 |
| node>>shape | [string] | 是 | 节点形状 |
| node>>types | [string] | 是 | 节点类型 |
| node>>x | [string] | 是 | 节点横坐标 |
| node>>y | [string] | 是 | 节点纵坐标 |
| node | [string] | 是 |  |

项目管理-流程点删除

接口描述：用户进入流程编排页面，获取页面点配置信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /model/deleteelement | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 流程点id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

###### JobController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | JobController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 项目流程执行信息 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/fedral/controller/fateapi/JobController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | ■公有 □私有 □保护 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| submit | session[:id] | string | ■公有 □私有 □保护 | Result | 提交流程任务 |
| log | session[:modelJobId  ] | string | ■公有 □私有 □保护 | Result | 下载日志 |
| query | session[:id  ] | string | ■公有 □私有 □保护 | Result | 任务执行详情 |

算法描述

项目管理-流程提交

接口描述：用户进入流程编排页面，点击执行按钮提交流程任务。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /job/submit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 否 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程任务查看

接口描述：用户进入流程编排页面，点击查看按钮，查看任务执行情况。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /job/query | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 否 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

项目管理-流程提交

接口描述：用户进入流程编排页面，点击执行按钮提交流程任务。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /job/submit | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 否 | 流程id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

###### TrackController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | TrackController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 任务执行记录信息 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/fedral/controller/fateapi/TrackController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | ■公有 □私有 □保护 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| data | session[:ComponentVos  ] | ComponentVos | ■公有 □私有 □保护 | Result | 查询任务执行组件数据 |
| download | session[:ComponentVos  ] | ComponentVos | ■公有 □私有 □保护 | Result | 下载任务执行组件数据 |

算法描述

项目管理-查询流程组件数据

接口描述：用户进入流程任务页面，点击组件查看组件数据。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /track/data | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| componentName | [string] | 否 | 组件名称 |
| jobId | [string] | 否 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| meta | [string] | 是 | 接口返回数据元数据 |
| meta>>header | [string] | 是 | 元数据信息 |
| data>>names | [string] | 是 | 组件类型 |
| data>>total | [number] | 是 | 组件数据总量 |

项目管理-下载流程组件数据

接口描述：用户进入流程任务页面，点击下载组件数据。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /track/download | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| componentName | [string] | 否 | 组件名称 |
| jobId | [string] | 否 | 任务id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

##### 实体类

###### FProjectDo

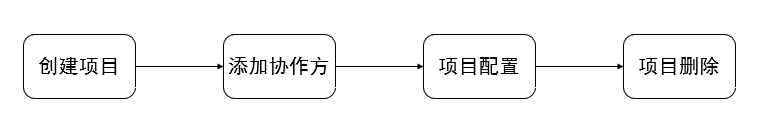
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FProjectDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存项目的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FProjectDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 项目目名称 |
| project | string | | ■公有 □私有 □保护 | | |  | |  |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| disable | integer | | ■公有 □私有 □保护 | | |  | | 是否可修改 |
| host | string | | ■公有 □私有 □保护 | | |  | | 发起方 |
| types | integer | | ■公有 □私有 □保护 | | |  | | 类型 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人帐号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| name | string | | ■公有 □私有 □保护 | | |  | | 项目目名称 |
| project | string | | ■公有 □私有 □保护 | | |  | |  |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| disable | integer | | ■公有 □私有 □保护 | | |  | | 是否可修改 |
| host | string | | ■公有 □私有 □保护 | | |  | | 发起方 |
| types | integer | | ■公有 □私有 □保护 | | |  | | 类型 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人帐号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_project | | |

算法描述

无

状态转换描述



###### FModelDo

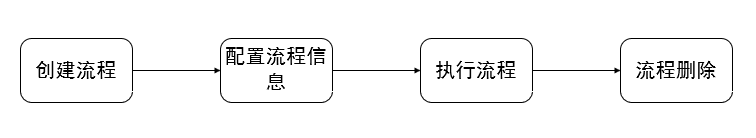
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FModelDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存流程的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FModelDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 模型名称 |
| model | string | | ■公有 □私有 □保护 | | | NN | | 模型编码 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 状态 |
| disable | integer | | ■公有 □私有 □保护 | | |  | | 是否可修改 |
| projectId | string | | ■公有 □私有 □保护 | | |  | | 项目id |
| config | string | | ■公有 □私有 □保护 | | |  | | 模型dsl 配置 |
| hasTemplate | integer | | ■公有 □私有 □保护 | | |  | | 是否为模板 |
| types | integer | | ■公有 □私有 □保护 | | |  | | 类型 |
| creatorId | string | | ■公有 □私有 □保护 | | | NN | | 创建人ID |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人帐号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| name | string | | ■公有 □私有 □保护 | | |  | | 模型名称 |
| model | string | | ■公有 □私有 □保护 | | |  | | 模型编码 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 状态 |
| disable | integer | | ■公有 □私有 □保护 | | |  | | 是否可修改 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_model | | |

算法描述

无

状态转换描述



###### FDataPartyDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FDataPartyDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存数据集的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FDataPartyDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| dataType | integer | | ■公有 □私有 □保护 | | |  | | 1 为 csv 0 为 联邦分析sql源 |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 数据集名称 |
| tags | string | | ■公有 □私有 □保护 | | |  | | 标签 |
| tableName | string | | ■公有 □私有 □保护 | | |  | | 数据集英文名 |
| description | string | | ■公有 □私有 □保护 | | |  | | 数据集详情 |
| namespace | string | | ■公有 □私有 □保护 | | |  | | 数据集空间 英文 |
| head | integer | | ■公有 □私有 □保护 | | |  | | 数据集是否有标题 |
| featrueNum | integer | | ■公有 □私有 □保护 | | |  | | 特征字段数量 |
| dataTotal | integer | | ■公有 □私有 □保护 | | |  | | 特征数据总量 |
| useCnt | integer | | ■公有 □私有 □保护 | | |  | | 数据集引用数量 |
| sid | string | | ■公有 □私有 □保护 | | |  | | 特征id |
| featureColumn | string | | ■公有 □私有 □保护 | | |  | | 特征名 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 数据集状态 1为启用 0 为下线 |
| isShow | integer | | ■公有 □私有 □保护 | | |  | | 0 为 私有 1 为 公有 2 为指定节点 |
| jobId | string | | ■公有 □私有 □保护 | | |  | | 上传数据集任务id |
| withLabel | integer | | ■公有 □私有 □保护 | | |  | | 数据集是否有标签 1为有 0 为没有 |
| labelName | string | | ■公有 □私有 □保护 | | |  | | 标签名称 |
| labelType | string | | ■公有 □私有 □保护 | | |  | | 标签类型 |
| part | integer | | ■公有 □私有 □保护 | | |  | | 分区 |
| dbSourceId | string | | ■公有 □私有 □保护 | | |  | | 数据源id |
| sqlContent | string | | ■公有 □私有 □保护 | | |  | | 语句内容 |
| nodeId | string | | ■公有 □私有 □保护 | | |  | | 节点id |
| edaHtml | string | | ■公有 □私有 □保护 | | |  | | eda html 文佳地址 |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatorAccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorName | string | | ■公有 □私有 □保护 | | |  | | 创建者名称 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_data\_party | | |

算法描述

无

状态转换描述

无

###### FModelElementDo

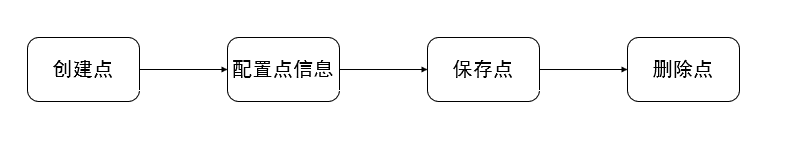
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FModelElementDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存流程点的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FModelElementDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| componentid | string | | ■公有 □私有 □保护 | | | NNU | | 组件id |
| ports | string | | ■公有 □私有 □保护 | | |  | | 前端所需参数 |
| name | string | | ■公有 □私有 □保护 | | | NN | | 组件名称 |
| modelid | string | | ■公有 □私有 □保护 | | | NN | | 模型id |
| data | string | | ■公有 □私有 □保护 | | |  | | 组件数据 |
| config | string | | ■公有 □私有 □保护 | | |  | | 组件配置 |
| x | string | | ■公有 □私有 □保护 | | | NN | | x坐标 |
| y | string | | ■公有 □私有 □保护 | | | NN | | y坐标 |
| shape | string | | ■公有 □私有 □保护 | | |  | | 图标形状 |
| icon | string | | ■公有 □私有 □保护 | | |  | | 图标 |
| creatorid | string | | ■公有 □私有 □保护 | | | NN | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | 创建者名称 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_model\_element | | |

算法描述

无

状态转换描述



###### FModelElementRelDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FModelElementRelDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存流程边的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FModelElementRelDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 主键id |
| modelid | string | | ■公有 □私有 □保护 | | | NN | | 模型id |
| startelementid | string | | ■公有 □私有 □保护 | | | NN | | 起始元素id |
| endelementid | string | | ■公有 □私有 □保护 | | | NN | | 目标元素id |
| startportid | string | | ■公有 □私有 □保护 | | |  | | 起点瞄点 |
| endportid | string | | ■公有 □私有 □保护 | | |  | | 终点瞄点 |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | 创建者名称 |
| modelid | string | | ■公有 □私有 □保护 | | |  | | 模型id |
| startelementid | string | | ■公有 □私有 □保护 | | |  | | 起始元素id |
| endelementid | string | | ■公有 □私有 □保护 | | |  | | 目标元素id |
| startportid | string | | ■公有 □私有 □保护 | | |  | | 起点瞄点 |
| endportid | string | | ■公有 □私有 □保护 | | |  | | 终点瞄点 |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | creatoraccount |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | creatorname |
| modelid | string | | ■公有 □私有 □保护 | | |  | | modelid |
| startelementid | string | | ■公有 □私有 □保护 | | |  | | startelementid |
| endelementid | string | | ■公有 □私有 □保护 | | |  | | endelementid |
| startportid | string | | ■公有 □私有 □保护 | | |  | | startportid |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_model\_element\_rel | | |

算法描述

无

状态转换描述

无

###### FProjectDataDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FProjectDataDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存项目挂载数据集的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FProjectDataDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 主键id |
| dataSetId | long | | ■公有 □私有 □保护 | | | NNI | | 数据集id |
| name | string | | ■公有 □私有 □保护 | | | NNS | | 文件名称 |
| url | string | | ■公有 □私有 □保护 | | | NNS | | 文件上传地址 |
| size | double | | ■公有 □私有 □保护 | | |  | | 文件大小 |
| fields | string | | ■公有 □私有 □保护 | | |  | | 数据集字段内容 |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 修改时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_project\_data | | |

算法描述

无

状态转换描述

无

###### FProjectGuestDo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FProjectGuestDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存项目协助方的详细信息； | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FProjectGuestDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | | 主键id |
| ProjectId | long | | ■公有 □私有 □保护 | | | NNI | | 项目id |
| creatorName | string | | ■公有 □私有 □保护 | | | NNS | | 用户名称 |
| nodeId | string | | ■公有 □私有 □保护 | | | NNS | | 协助方节点id |
| creatorId | string | | ■公有 □私有 □保护 | | |  | | 用户id |
| updateTime | long | | ■公有 □私有 □保护 | | |  | | 修改时间 |
| createTime | long | | ■公有 □私有 □保护 | | |  | | 创建时间 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_project\_guest | | |

摘要

算法描述

无

状态转换描述

无

###### FDataApproveDo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FDataApproveDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存项目数据集审批的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\federal\FDataApproveDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| projectid | string | | ■公有 □私有 □保护 | | | NNU | | 项目数据id |
| dataid | string | | ■公有 □私有 □保护 | | | NNU | | 数据集id |
| datacatalog | string | | ■公有 □私有 □保护 | | |  | | 数据归类 |
| content | string | | ■公有 □私有 □保护 | | |  | | 申请内容 |
| approval | string | | ■公有 □私有 □保护 | | |  | | 审批理由 |
| nodeid | string | | ■公有 □私有 □保护 | | |  | | 节点id |
| userid | string | | ■公有 □私有 □保护 | | |  | | 用户id |
| username | string | | ■公有 □私有 □保护 | | |  | | 用户名称 |
| approve | integer | | ■公有 □私有 □保护 | | |  | | 审批状态 |
| types | integer | | ■公有 □私有 □保护 | | |  | | 类型 |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | creatorname |
| projectid | string | | ■公有 □私有 □保护 | | |  | | projectid |
| dataid | string | | ■公有 □私有 □保护 | | |  | | dataid |
| datacatalog | string | | ■公有 □私有 □保护 | | |  | | datacatalog |
| content | string | | ■公有 □私有 □保护 | | |  | | content |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_data\_approve | | |

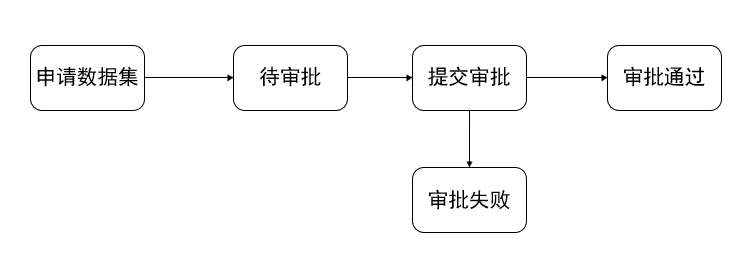
摘要

无

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：fl\_project**  **、fl\_model、fl\_data\_party、fl\_model\_elemen、fl\_model\_element\_rel**  **、fl\_project\_data、fl\_project\_guest、fl\_data\_approve**  **>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT a.\* from fl\_project a where 1=1 and a.creator\_account = ? order by a.created\_time desc limit 0,100  【SQL02】  SELECT \* FROM fl\_project\_guest WHERE project\_id = #{id}  【SQL03】  SELECT \* FROM fl\_model where project\_id = #{projectId} order by created\_time desc  【SQL04】  SELECT \* FROM fl\_data\_approve where approve!=9 and project\_id=#{projectId} and node\_id=#{id} and types=#{types}  【SQL05】  SELECT \* FROM fl\_data\_party where id in ( ? , ? , ? , ? ) order by created\_time desc  【SQL06】  SELECT created\_time,modified\_time,id,name,model,description,status,disable,project\_id,config,has\_template,types,creator\_id,creator\_account,creator\_name FROM fl\_model WHERE id = {id}  【SQL07】  SELECT \* FROM fl\_model\_element\_rel WHERE model\_id ={id}  【SQL08】  SELECT e.\*,g.types as types,g.component\_en as componentType FROM fl\_model\_element e JOIN fl\_component g on e.component\_id = g.id and e.model\_id = ?  【SQL9】  INSERT INTO fl\_model ( created\_time,modified\_time,id,name,model,status,project\_id,types,creator\_id,creator\_account,creator\_name ) VALUES( ?,?,?,?,?,?,?,?,?,?,? ) |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | userId | 用户id |  | | modelId | 流程id |  | | projectId | 项目id |  | | name | 项目名称 |  | | creatorid | 创建者id |  | | dataId | 数据id |  | | types | 审批类型 |  | | nodeId | 节点id |  | | Config | 主键配置 |  | | shape | 组件形状 |  | | tag | 数据集标签 |  | | type | 数据集类型 |  | | icon | 图标 |  | | y | 点纵坐标 |  | | x | 点横坐标 |  | | status | 状态 |  | | createTime | 创建时间 |  | | updateTime | 更新时间 |  | |

#### 总体设计方案

##### 设计方案

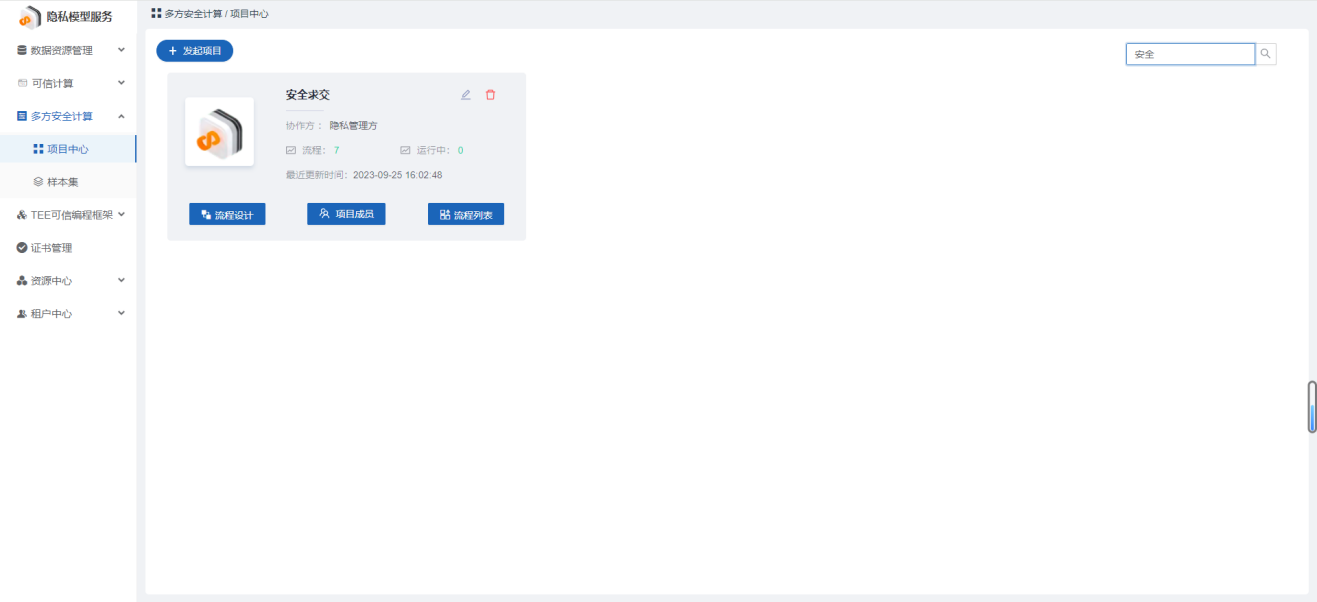
项目管理包多方含项目管理、流程管理等模块，系统根据用户机构类型分别展示。用户拥有创建项目与项目流程的权限，其中展示项目资产与项目流程资产信息。

项目列表展示该用户所有项目资源信息，并支持对项目详情信息查询。项目创建、项目配置删除等功能。项目流程列表展示用户项目下所有创建的流程列表。支持创建流程、修改流程、查看流程编排配置，流程执行结果、删除\执行流程、等多功能。

##### 处理流程

###### 项目列表

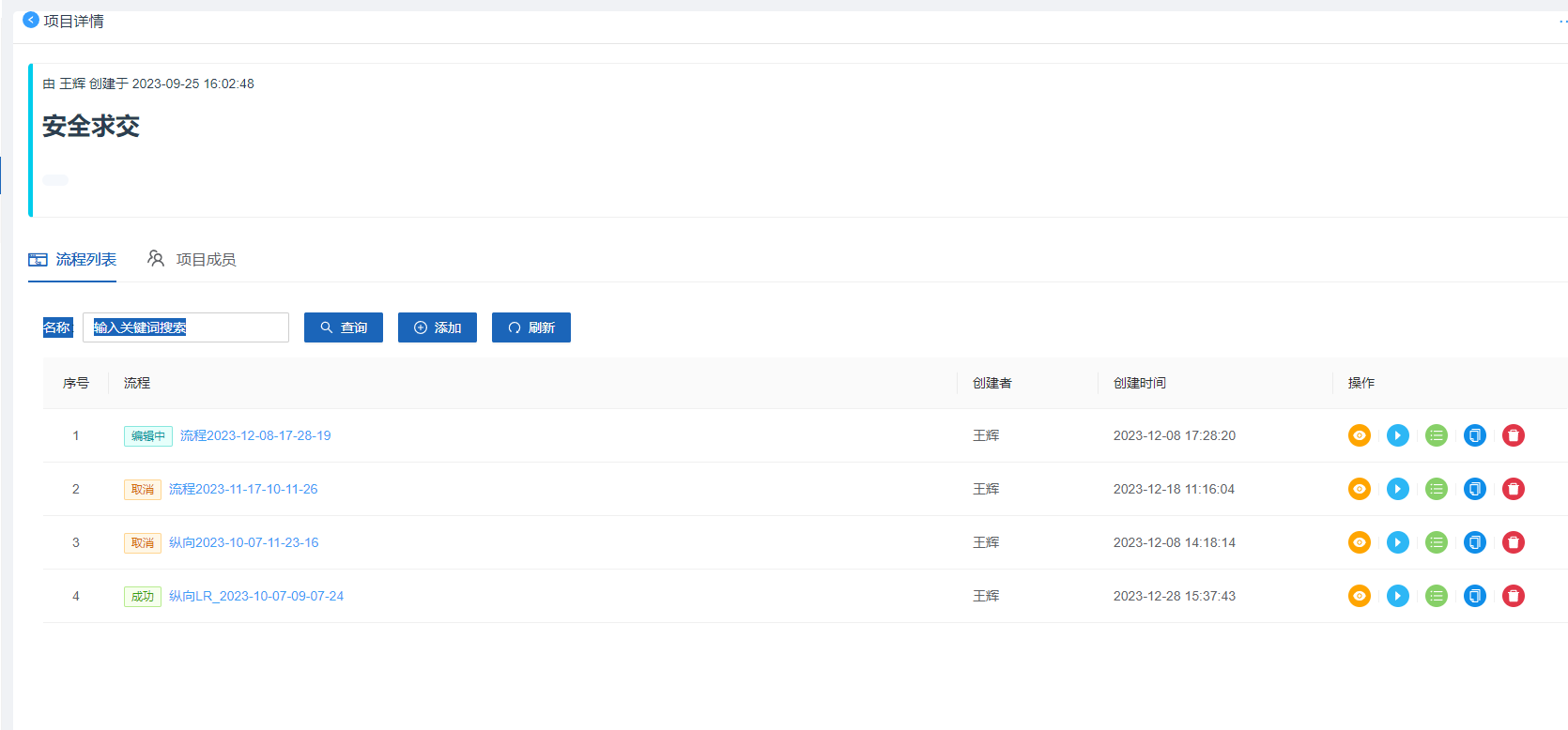
1. 查询项目列表



页面默认无查询条件请求【7.3.1.2.1.2.2.项目管理-项目列表】接口执行【SQL01】分页获取用户的项目资产。接口内部发起HTTP请求调用项目列表接口分页获取其所有项目列表返回前端展示。分别展示项目名称、更新时间、流程数量等字段。

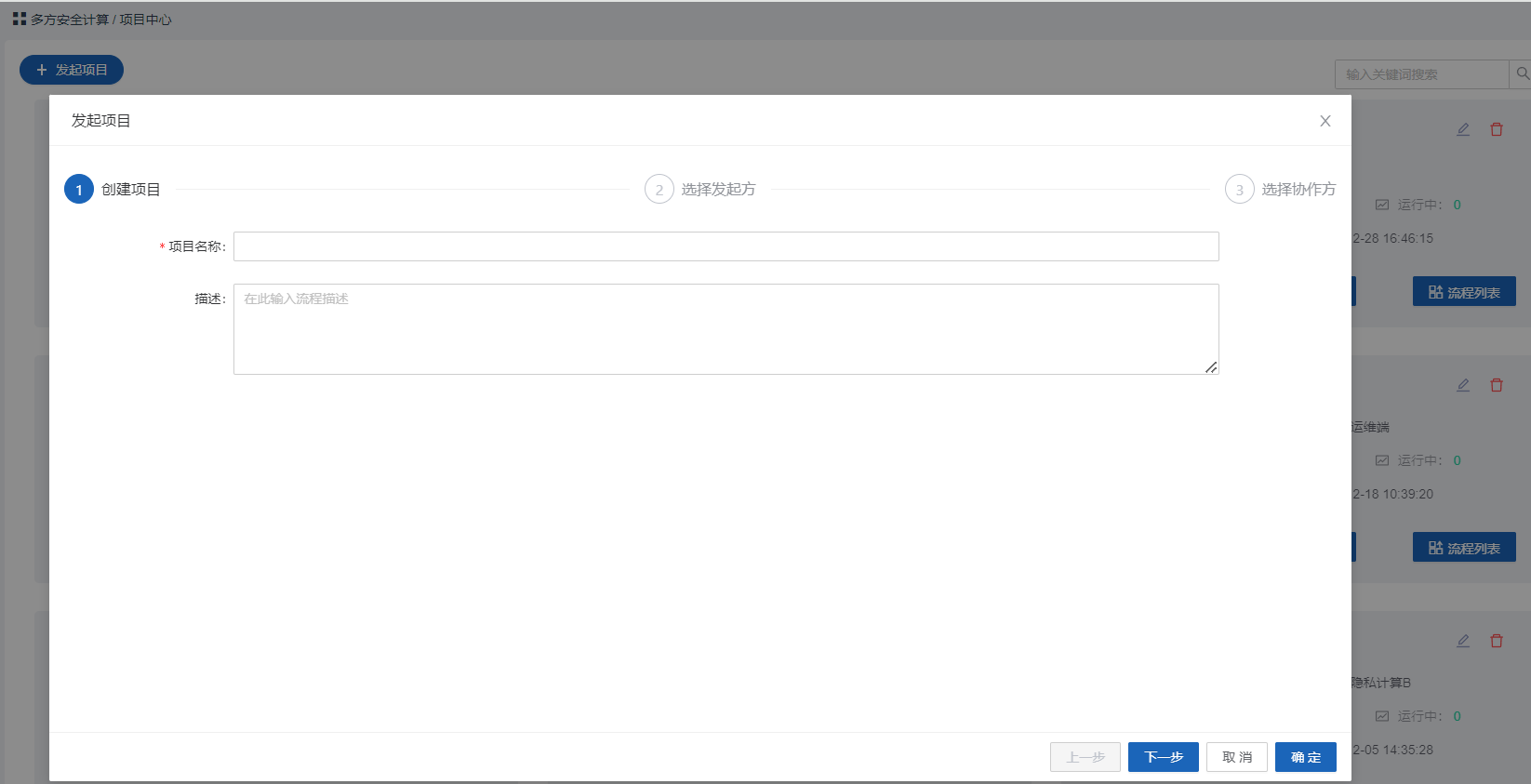
页面输入数据名称作为查询条件，点击【查询】按钮，页面重新请求【7.3.1.2.1.2.2.项目管理-项目列表】接口，执行【SQL01】并将项目名称作为参数调用平台系统接口模糊搜索获取对应项目名称的是项目资产列表。

1. 执行查看项目详情



页面点击【查看】按钮请求【7.3.1.2.1.2.1.项目管理-查看项目需求】接口执行【SQL03】获取项目详情信息。接口内部根据分页信息与项目id发起HTTP请求调用系统资源详情接口分页获取该数据id对应项目信息。页面根据接口返回展示。

1. 创建项目



页面选择发起项目按钮，填写数据集基本信息，将项目信息作为参数请求【7.4.1.2.1.2.4.项目管理-新增项目】接口执行【SQL09】。接口将数据集信息封装在【7.4.1.2.5.project】实体类中，再根据token获取用户信息，将用户信息封装在实体类中，插入项目信息表。

###### 流程列表

1. 查询项目流程列表



页面默认无查询条件请求页面重新请求【7.2.1.2.1.2.3.项目管理-查询流程列表】接口，接口内部执行分页获取项目流程列表返回页面展示。分别展示流程名称、创建者、创建时间等多个字段。

页面输入流程名称为模糊搜索，点击【查询】按钮，页面重新请求【7.2.1.2.1.2.3项目管理-查询流程列表】接口执行【SQL06】，接口将页面传递条件作为查询条件执行获取项目流程列表返回页面展示。

1. 查询流程详情

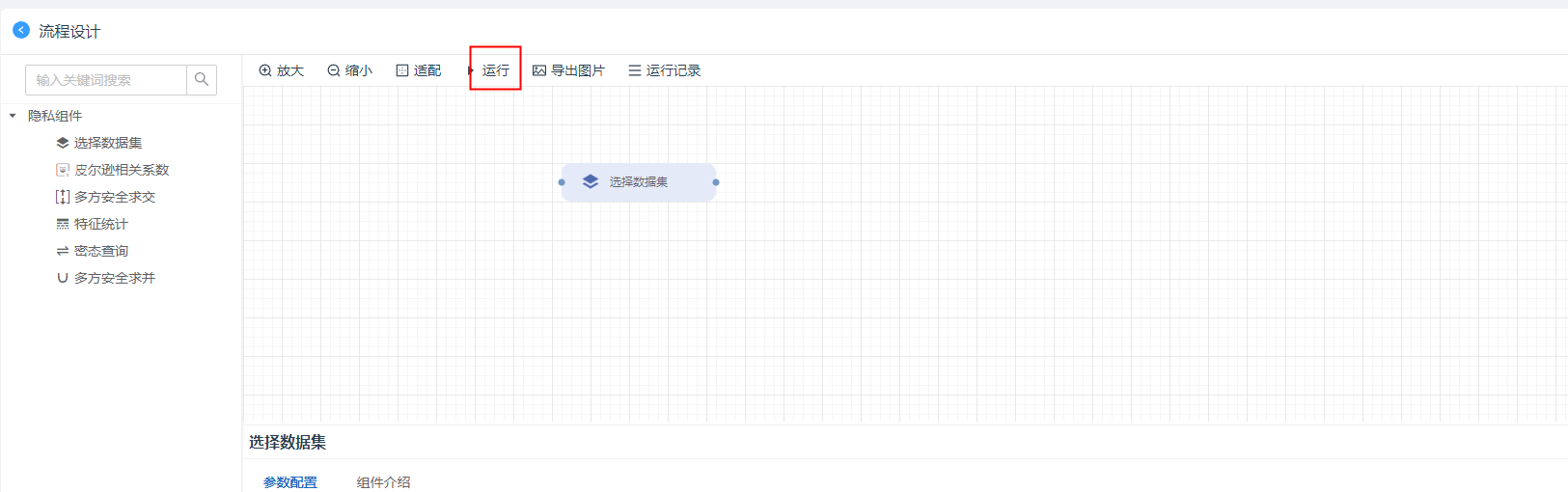
页面点击流程进入流程详情页面，流程详情页面包含组件概览、可视化界面等功能。

1. 组件详情-概览



概览页面调用【7.2.1.2.1.2.4.项目管理-流程详情】接口执行【SQL05】根据流程id获取项目流程详情信。

1. 流程详情-启动

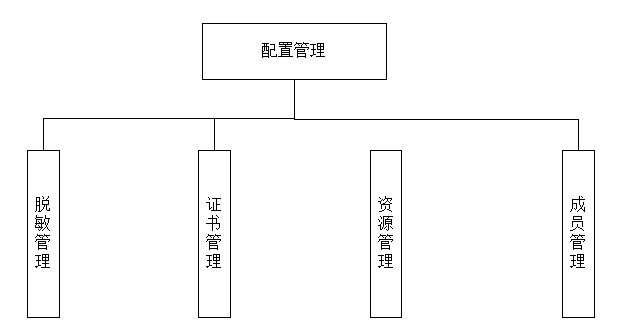


数据页面调用【7.2.1.2.3.2.3.项目管理-流程启动】接口执行【SQL07】根据流程id执行流程任务。

## 配置管理

户提供隐私模型服务平台相关配置功能，包括脱敏管理、证书管理、资源管理、成员管理等板块。

主要包含的功能如下：



### 脱敏管理

脱敏管理为用户提供项目脱敏函数的管理功能，支持用户数据脱敏业务需求。主要流程如下：



#### 类设计

##### 控制类

###### TruUdfController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | TruUdfController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、数据更新相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\bydb\controller\TruUdfController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| option | session[:string] | HbSurveyPeopleVo | ■公有 □私有 □保护 | Result | 脱敏函数列表 |
| update | session[:TTruUdfDo  ] | TTruUdfDo | ■公有 □私有 □保护 | Result | 修改脱敏函数 |

算法描述

脱敏管理-脱敏列表

接口描述：查询系统的脱敏函数列表信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /truudf/option | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 表**参数名** | **类型** | **必填** | **说明** |
| qryCond | [string] | 否 | 函数名称 |
| pageSize | [number] | 是 | 页码 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>total | [string] | 是 |  |
| data>>list | [array] | 是 |  |
| data>>list>>id | [string] | 是 |  |
| data>>list>>norder | [string] | 是 | 排序 |
| data>>list>>catalogNo | [string] | 是 | 函数分类 |
| data>>list>>chnName | [string] | 是 | 函数中文名称 |
| data>>list>>modifiedTime | [string] | 是 | 修改时间 |
| data>>list>>remark | [string] | 是 | 备注 |
| data>>list>>sqlFun | [string] | 是 | 案例 |
| data>>list>>tableFun | [string] | 是 | 使用方式 |
| data>>list>>createTime | [string] | 是 | 创建时间 |

脱敏管理-修改脱敏状态

接口描述：用户在脱敏函数列表上对脱敏信息进行修改。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /truudf/update | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 核保人主键 |
| enable | [string] | 是 | 状态 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

##### 实体类

###### TTruUdfDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TTruUdfDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存脱敏函数的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\bydb  \TTruUdfDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| projectId | long | | ■公有 □私有 □保护 | | | NNU | | id |
| belongtype | string | | ■公有 □私有 □保护 | | |  | |  |
| dbtype | string | | ■公有 □私有 □保护 | | |  | | 数据库类型 |
| catalogno | string | | ■公有 □私有 □保护 | | |  | | 分类 |
| functionname | string | | ■公有 □私有 □保护 | | |  | | 函数名称 |
| sqlfun | string | | ■公有 □私有 □保护 | | |  | | SQL函数 |
| tablefun | string | | ■公有 □私有 □保护 | | |  | | TABLE函数 |
| chnname | string | | ■公有 □私有 □保护 | | |  | | 中文名称 |
| mdfile | string | | ■公有 □私有 □保护 | | |  | | md文件 |
| remark | string | | ■公有 □私有 □保护 | | |  | | 说明 |
| functiontype | string | | ■公有 □私有 □保护 | | |  | | 函数类型 |
| functiondefine | string | | ■公有 □私有 □保护 | | |  | | 定义 |
| argumenttype | string | | ■公有 □私有 □保护 | | |  | | 参数类型 |
| argumentdesc | string | | ■公有 □私有 □保护 | | |  | | 参数是说明 |
| returntype | string | | ■公有 □私有 □保护 | | |  | | 返回类型 |
| variablearity | string | | ■公有 □私有 □保护 | | |  | | 是否变长 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| slanguage | string | | ■公有 □私有 □保护 | | |  | | 语言 |
| builtin | string | | ■公有 □私有 □保护 | | |  | | 内置 |
| sdeterministic | string | | ■公有 □私有 □保护 | | |  | | 确定性 |
| icon | string | | ■公有 □私有 □保护 | | |  | | 图标 |
| example | string | | ■公有 □私有 □保护 | | |  | |  |
| norder | integer | | ■公有 □私有 □保护 | | |  | | 排序 |
| enable | integer | | ■公有 □私有 □保护 | | |  | | 是否启用 1启用 0 不启用 |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | 创建者名称 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | t\_tru\_udf | | |

算法描述

无

状态转换描述

无

#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**t\_tru\_udf  **>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT \* from t\_tru\_udf where 1=1 and db\_type = ? and catalog\_no = ? order by norder  【SQL02】  UPDATE t\_tru\_udf SET created\_time = ?,modified\_time = ?,belong\_type = ?,db\_type = ?,catalog\_no = ?,function\_name = ?,sql\_fun = ?,table\_fun = ?,chn\_name = ?,md\_file = ?,remark = ?,function\_type = ?,function\_define = ?,argument\_type = ?,argument\_desc = ?,return\_type = ?,variable\_arity = ?,description = ?,slanguage = ?,built\_in = ?,sdeterministic = ?,icon = ?,example = ?,norder = ?,enable = ?,creator\_id = ?,creator\_account = ?,creator\_name = ? WHERE id = ? |
|  |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | norder | 0 | 1 | 排序 | |  |  |  |  | |

#### 总体设计方案

##### 设计方案

脱敏管理为用户提供数据分析下的字段脱敏功能。

##### 处理流程

1. 脱敏列表



脱敏函数列表，调用【7.4.1.2.1.2.1..脱敏管理-脱敏函数列表】执行【SQL01】根据条件查询脱敏函数信息列表

脱敏函数信息修改，页面展示脱敏函数配置项，页面将修改后的配置作为参数调用【7.4.1.2.1.2.2.脱敏管理-修改脱敏函数】接口，接口执行【SQL02】将修改后的配置进行修改。

### 证书管理

证书管理提供已生成证书的管理。主要流程如下：



#### 类设计

##### 控制类

###### TeeLicenseController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | TeeLicenseController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、数据更新相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\tee\controller\TeeLicenseController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | session[:TeeLicenseDo  ] | TeeLicenseDo | ■公有 □私有 □保护 | Result | 证书管理-证书列表 |
| add | session[:TeeLicenseDo  ] | TeeLicenseDo | ■公有 □私有 □保护 | Result | 证书管理-证书添加 |
| update | session[:TeeLicenseDo  ] | TeeLicenseDo | ■公有 □私有 □保护 | Result | 证书管理-证书修改 |
| delete | session[:string  ] | string | ■公有 □私有 □保护 | Result | 证书管理-证书删除 |
| put | session[:string  ] | string | ■公有 □私有 □保护 | Result | 证书管理-证书生成 |
| download | session[:string  ] | string | ■公有 □私有 □保护 | Result | 证书管理-证书下载 |

算法描述

证书管理-证书列表

接口描述：根据用户创建的证书获取对应证书列表，支持根据证书名称进行筛选。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /license/page | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| pageSize | [number] | 是 | 页码 |
| name | [string] | 否 | 证书名称 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>pageSize | [number] | 是 | 0 |
| data>>total | [number] | 是 | 1 |
| data>>page | [number] | 是 | 1 |
| data>>list | [array] | 是 |  |
| data>>list>>licenseDays | [number] | 是 | 证书有效天数 |
| data>>list>>creatorId | [number] | 是 | 用户id |
| data>>list>>status | [number] | 是 | 证书状态 |
| data>>list>>content | [string] | 是 | 证书生成地址 |
| data>>list>>name | [string] | 是 | 证书名称 |
| data>>list>>id | [string] | 是 |  |
| data>>list>>description | [string] | 是 | 备注 |
| data>>list>>createTime | [string] | 是 | 创建时间 |
| data>>list>>modifiedTime | [string] | 是 | 修改时间 |

证书管理-证书添加

接口描述：为用户添加对应证书。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /license/add | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| licenseDays | [number] | 是 | 证书有效天数 |
| creatorId | [string] | 是 | 用户id |
| status | [string] | 是 | 证书状态 |
| content | [string] | 否 | 证书配置 |
| name | [string] | 否 | 证书名称 |
| description | [string] | 否 | 证书描述 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

证书管理-证书修改

接口描述：修改对应证书信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| POST | /license/update | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| licenseDays | [number] | 是 | 证书有效天数 |
| creatorId | [string] | 是 | 用户id |
| status | [string] | 是 | 证书状态 |
| content | [string] | 否 | 证书配置 |
| name | [string] | 是 | 证书名称 |
| description | [string] | 否 | 证书描述 |
| Id | [string] | 是 | 证书id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

证书管理-证书生成

接口描述：为用户添加对应证书。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| PUT | /license/touch | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| licenseDays | [number] | 是 | 证书有效天数 |
| Id | [string] | 是 | 证书id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

证书管理-证书下载

接口描述：为用户添加对应证书。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /license/download | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Id | [string] | 是 | 证书id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

证书管理-证书删除

接口描述：为用户添加对应证书。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /license/delete | Json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| id | [string] | 是 | 证书id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |

##### 实体类

###### TeeLicenseDo

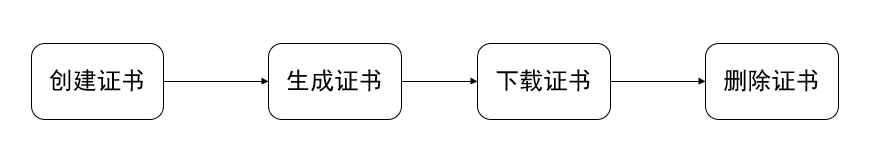
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | TeeLicenseDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存证书详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\fedral\bean\tee  \TeeLicenseDo.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 许可证名称 |
| description | string | | ■公有 □私有 □保护 | | |  | | 描述 |
| licensedays | integer | | ■公有 □私有 □保护 | | |  | | 证书使用天数 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 证书状态 |
| content | string | | ■公有 □私有 □保护 | | |  | | 许可证信息 |
| nodeid | string | | ■公有 □私有 □保护 | | | NNU | | 节点id |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人id |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | 创建者名称 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | tee\_license | | |

算法描述

无

状态转换描述



#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**tee\_license  **>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT \* from tee\_license where 1=1 and creator\_Id = ? order by created\_time desc limit 0,10  【SQL02】  INSERT INTO tee\_license ( created\_time,modified\_time,id,  name,licenseDays,status,creator\_id,c  reator\_account,creator\_name )  VALUES( ?,?,?,?,?,?,?,?,? )  【SQL03】  UPDATE tee\_license SET modified\_time = ?,name = ?,licenseDays = ? WHERE id = ?  【SQL04】  DELETE FROM tee\_license WHERE id = ? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | Id | 证书id | 证书id | | name | 证书名 | 证书名称 | | description | 备注 | 备注 | | licenseDays | 过期天数 | 过期天数 | | createTime | 业务场景id | 创建时间 | | updateTime | 数据集别名 | 修改时间 | |

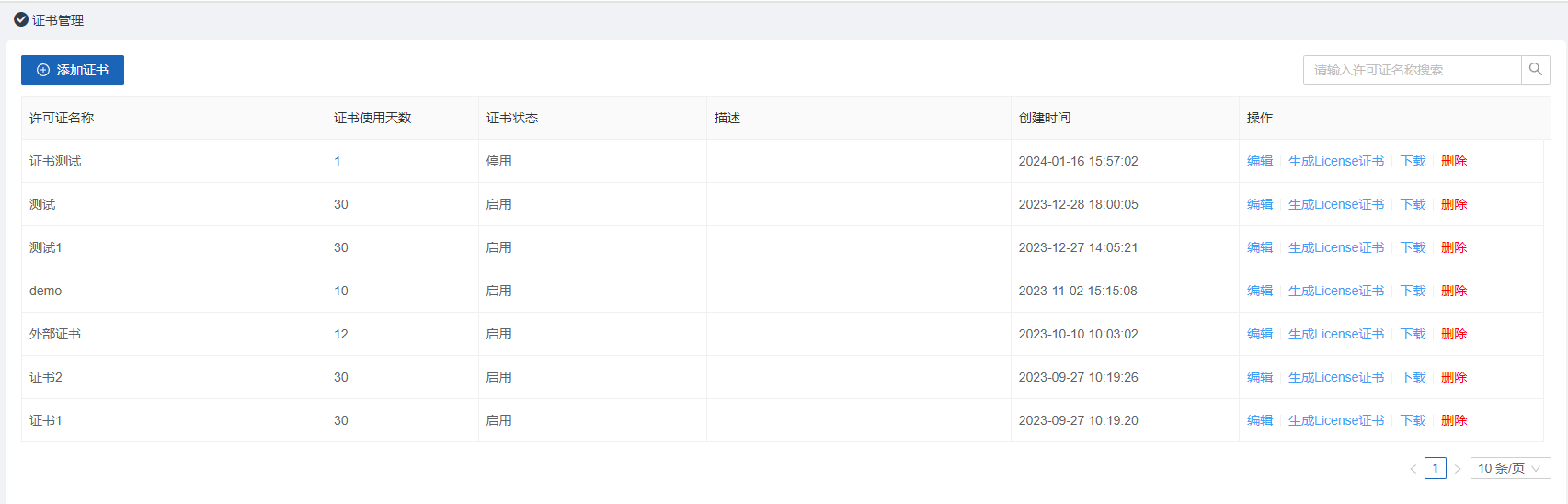
#### 整体设计方案

##### 设计方案

证书管理提供对系统生成证书进行管理，提供证书生成、证书下载等功能。

##### 处理流程

1. 查询证书列表



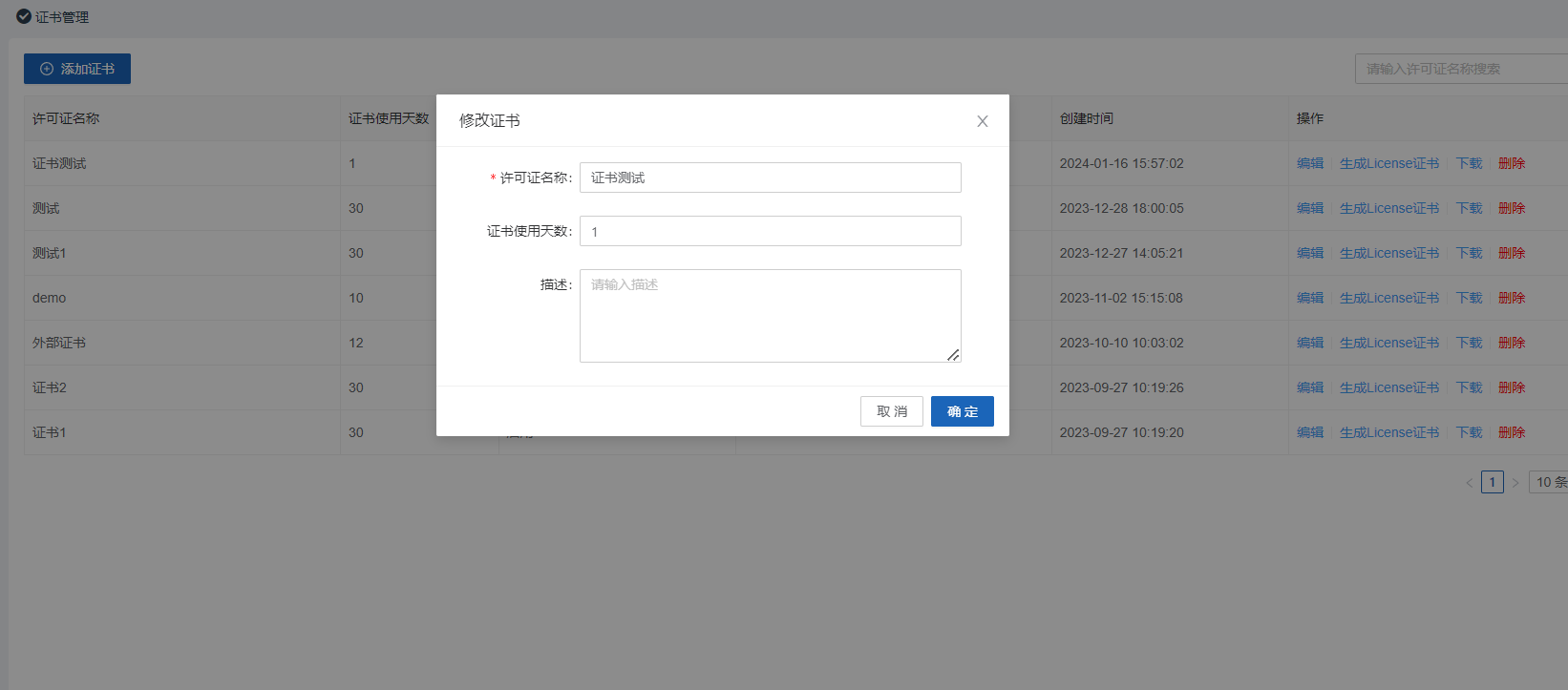
页面支持证书名称为查询条件，调用【7.4.2.2.1.2.1.证书管理-证书列表】接口。接口将查询条件与分页信息作为查询条件执行【SQL01】分页获取证书列表返回页面展示。

1. 新增证书



页面点击【新增】弹出新增业务场景页面，新增页面展示业务场景配置信息。配置完成后将配置信息作为参数调用【7.4.2.2.1.2.2.证书管理-证书新增】接口。接口将证书信息封装执行【SQL02】进行插入证书信息操作。

1. 修改证书



页面点击【修改】弹出修改证书页面，修改页面展示证书配置信息。配置完成后将配置信息作为参数调用【7.4.4.2.1.2.3.证书管理-证书修改】接口。接口把修改的证书信息执行【SQL03】进行修改。

1. 删除证书



页面选择证书，点击【删除】确认调用【7.4.2.2.1.2.4.证书管理-证书删除】接口根据证书id执行【SQL04】删除证书。

### 资源管理

资源管理为用户提供系统CPU、内存、磁盘等资源的查看等功能。主要流程如下：



#### 类设计

##### 控制类

###### ResourceController

算法描述

资源管理-节点列表

接口描述：资源管理-查看系统接入的节点信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /api/node/local | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| page | [number] | 否 | 分页数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 |  |
| data | [object] | 是 |  |
| data>>total | [number] | 是 |  |
| data>>page | [number] | 是 |  |
| data>>list | [number] | 是 |  |
| data>>list>>id | [number] | 是 |  |
| data>>list>>nodeIp | [number] | 是 | 节点ip |
| data>>list>>nodeName | [number] | 是 | 节点名称 |
| data>>list>>remark | [number] | 是 | 备注 |
| data>>list>>share | [number] | 是 | 分享数 |
| data>>list>>status | [number] | 是 | 状态 |
| data>>list>>type | [number] | 是 | 类型 |
| data>>list>>systemType | [string] | 是 | 系统版本 |
| code | [number] | 是 |  |

资源管理-资源预览

接口描述：对接入的节点资源情况进行查看，支持节点名称查询。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /apiCentral/central/system-resources-overview | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 |  |  |  |

返回参数

资源管理-节点更新

接口描述：对接入的节点配置信息进行更新。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| PUT | /api/node/local/ | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Id | [number] | 是 | 节点id |
| remark | [object] | 否 | 备注 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 |  |
| data | [number] | 是 |  |
| code | [number] | 是 |  |

资源管理-节点接入

接口描述：接入一个服务器资源。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| Post | /api/node/local/ | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Name | [string] | 是 | 节点名称 |
| Type | [number] | 是 | 类型 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 |  |
| data | [number] | 是 |  |
| code | [number] | 是 |  |

资源管理-节点删除

接口描述：资源管理-节点移除。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| DELETE | /api/node/local/ | json |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| Id | [string] | 是 | 节点id |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| message | [string] | 是 |  |
| data | [array] | 是 |  |
| code | [number] | 是 |  |

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ResourceController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、数据更新相关内容。 | | | | |
| 代码发布路径 | $com/capinfo/healthcare/fedral/controller/resource/ResourceController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| local | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 获取节点列表 |
| systemResourcesOverview | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 资源预览 |
| delete | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 删除节点 |
| insert | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 接入节点 |
| update | session[:ResourceVo] | ResourceVo | ■公有 □私有 □保护 | Result | 更新节点 |

##### 实体类

###### NodeInfo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | NodeInfo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存平台资源的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\central\bean\scResource | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | string | | ■公有 □私有 □保护 | | | NNU | | id |
| name | string | | ■公有 □私有 □保护 | | | NNI | | 资源名称 |
| remark | string | | ■公有 □私有 □保护 | | |  | | 备注 |
| useCpu | string | | ■公有 □私有 □保护 | | |  | | cpu核数 |
| node\_name | string | | ■公有 □私有 □保护 | | |  | | 节点名称 |
| node\_ip | string | | ■公有 □私有 □保护 | | |  | | 节点ip |
| type | int | | ■公有 □私有 □保护 | | |  | | 类型 |
| system\_type | int | | ■公有 □私有 □保护 | | |  | | 系统版本 |
| gpu | int | | ■公有 □私有 □保护 | | |  | | Gpu |
| has\_gpu | long | | ■公有 □私有 □保护 | | |  | | 是否有gpu |
| share | long | | ■公有 □私有 □保护 | | |  | | 是否分享 |
| cluster | string | | ■公有 □私有 □保护 | | |  | | 集群 |
| status | long | | ■公有 □私有 □保护 | | |  | | 状态 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | node\_info | | |

算法描述

无

状态转换描述

无

#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**node\_info**>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT  id,uid,name,remark,node\_name,node\_ip,`type`,system\_type,gpu,has\_gpu,share,cluster,status  FROM  node\_info  【SQL02】  INSERT INTO `vela-v3`.node\_info (id,uid,name,remark,node\_name,node\_ip,`type`,system\_type,gpu,has\_gpu,share,cluster,status) VALUES  ('15d0bcc8-b0f3-4080-839b-ad1460107788','dcb239a9-5503-4870-b2b0-6ac7cfd6beb8','eqweq','','k8s-master-01','192.168.100.98','localAi','3.10.0-1160.el7.x86\_64','','false','noShare','local',''),  ('834b2832-d585-4f6d-9593-f17e0e642bcc','d3300442-30bc-4664-af5c-2b726f57784d','adad','','node01','192.168.100.55','localAi','3.10.0-1127.13.1.el7.x86\_64','','false','noShare','local','');  【SQL03】  DELETE  FROM  node\_info  WHERE  id= #{id} |
|  |
| 关键字段赋值说明：   |  |  |  |  | | --- | --- | --- | --- | | 字段名 | 原值 | 新值 | 备注 | | name | 0 |  | 资源名称 | | remark | 0 |  | 备注 | | useCpu | 0 |  | cpu核数 | | node\_name | 0 |  | 节点名称 | | node\_ip | 0 |  | 节点ip | | type | 0 |  | 类型 | |

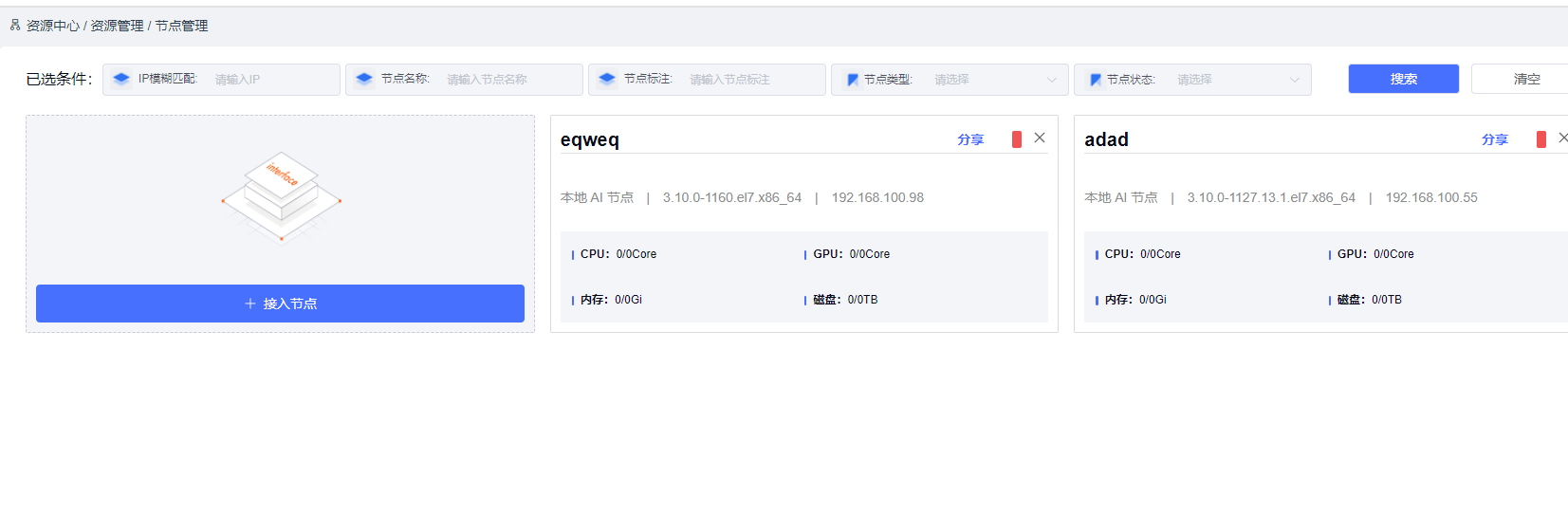
#### 总体设计方案

##### 设计方案

资源管理为用户提供系统CPU、内存、磁盘等资源的查看等功能。

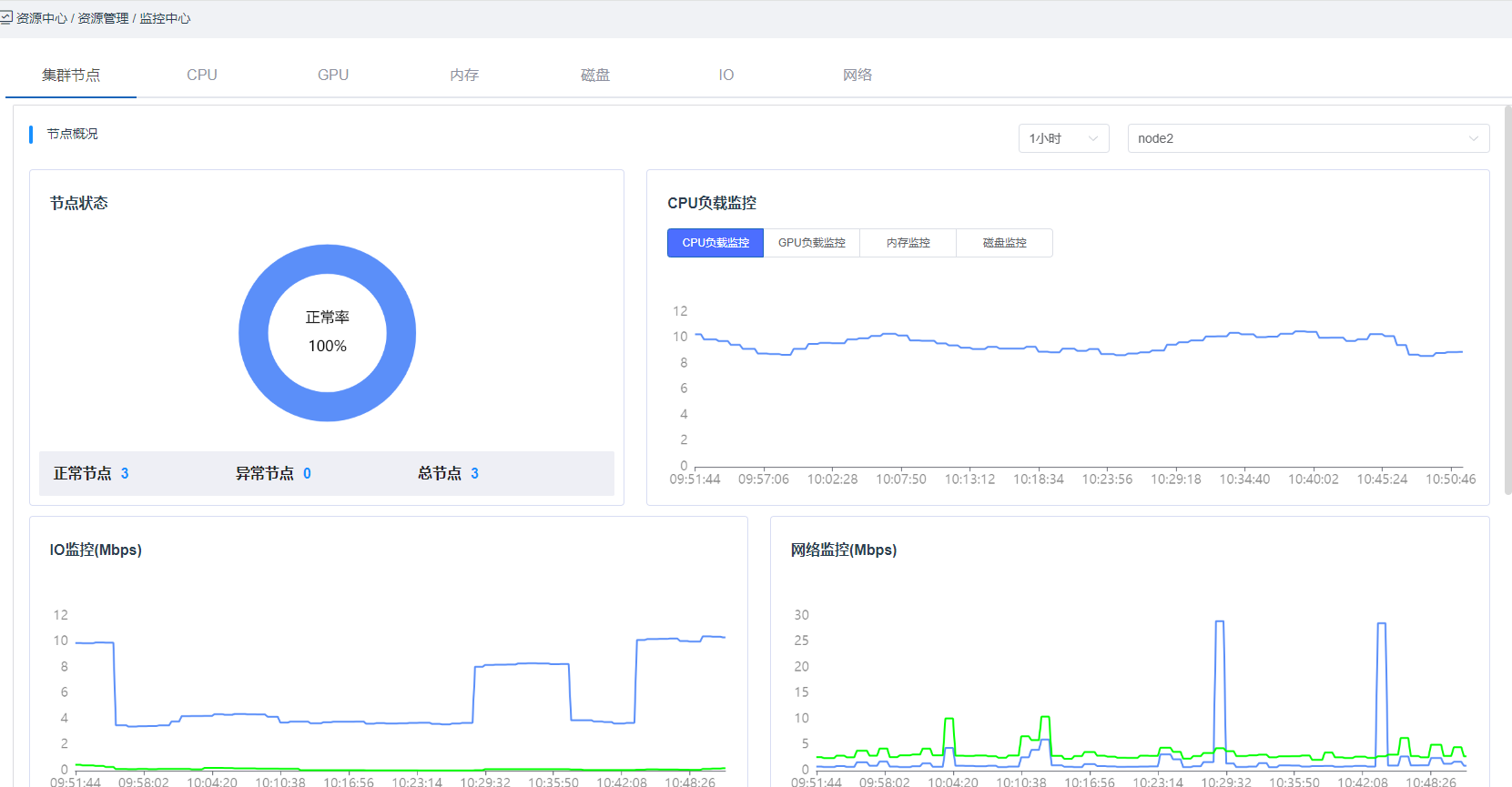
##### 处理流程

1. 查询节点列表



进入资源管理页面，资源列表为页面调用【7.4.3.2.1.2.1.资源管理-节点列表】接口，在未输入筛选条件情况下，使用【SQL01】默认返回所有节点的资源使用情况,若前端输入筛选条件，则在【SQL01】基础上增加筛选条件查询返回页面。

1. 节点资源预览



页面进入【监控中心】页面，调用【7.3.4.2.1.2.1..资源管理-资源预览】接口查询资源详情。接口查询节点使用基本信息，再根据前端需求处理图表的格式返回页面。

### 成员管理

为用户提供注册到平台的组织成员信息，并且支持对成员资源配置情况。主要流程如下：



#### 类设计

##### 控制类

###### TruPmsNodeController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | TruPmsNodeController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 接受用户通过页面点请求，完成数据存储、数据检索、数据更新相关内容。 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\analysis\controller\TruPmsNodeController  .java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| pmsnodepartypage | 无 | 无 | ■公有 □私有 □保护 | Result | 成员管理-成员列表 |
| pmstabledatasetlist | session[:string] | string | ■公有 □私有 □保护 | Result | 成员管理-成员数据集列表 |

算法描述

成员管理-成员列表

接口描述：根据审批流程id查询该审批所属审批流程列表。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | /system/approve/list |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| 无 | 无 |  |  |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>createdTime | [string] | 是 | 创建时间 |
| data>>list>>modifiedTime | [string] | 是 | 修改时间 |
| data>>page | [number] | 是 | 当前页数 |
| data>>pageSize | [number] | 是 | 当前页码 |
| data>>total | [number] | 是 | 总条数 |

成员管理-成员数据集

接口描述：展示各成员下授权的数据集信息。

请求方式

|  |  |  |
| --- | --- | --- |
| **HTTP方法** | **URI** | **消息体** |
| GET | trupmsnode/pmstabledatasetlist |  |

请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| nodePartyId | [string] | 是 | 成员id |
| pageSize | [number] | 否 | 页码 |
| currentPage | [number] | 是 | 页数 |

返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必填** | **说明** |
| msg | [string] | 是 | 接口返回 |
| data | [string] | 是 | 接口返回数据信息 |
| success | [boolean] | 是 | 是否成功标识 |
| data>>list | [array] | 是 | 返回流程 |
| data>>list>>id | [number] | 是 | 流程id |
| data>>list>>userId | [number] | 是 | 申请用户id |
| data>>list>>userDept | [string] | 是 | 申请用户机构名 |
| data>>list>>userDeptId | [string] | 是 | 申请用户机构id |
| data>>list>>userName | [string] | 是 | 申请用户名 |
| data>>list>>approveRole | [string] | 是 | 审批用户角色 |
| data>>list>>approveDept | [string] | 是 | 审批用户机构 |
| data>>list>>approveDeptId | [number] | 是 | 审批用户机构id |
| data>>list>>approveId | [number] | 是 | 审批用户id |
| data>>list>>comment | [string] | 是 | 审批意见 |
| data>>list>>taskId | [string] | 是 | 流程实例id |
| data>>list>>taskName | [string] | 是 | 流程名称 |
| data>>list>>name | [string] | 是 | 流程目标名称 |
| data>>list>>relevanceContent | [string] | 是 | 流程关联详情内容 |
| data>>list>>relevanceId | [number] | 是 | 流程关联详情内容id |
| data>>list>>relevanceSize | [string] | 是 | 流程关联额度 |
| data>>list>>projectId | [number] | 是 | 项目id |
| data>>list>>type | [string] | 是 | 流程类型(应用上架 APP-LISTING，应用申请 APP-APPLY，数据集发布 DATA-PUBLISH，数据集申请 DATA-APPLY，模型上架 MODEL-LISTING，模型申请 MODEL-APPLY，数据使用申请 DATA-USE-APPLY，机构注册 ORGANIZATION-APPLY，资源申请 RESOURCE-APPLY) |
| data>>list>>status | [string] | 是 | 流程状态（(REVIEW 待审核/NOT-PASS 未通过/PASS 已通过) |
| data>>list>>createTime | [number] | 是 | 创建时间 |
| data>>list>>updateTime | [number] | 是 | 更新时间 |
| data>>currentPage | [number] | 是 | 当前页数 |
| data>>size | [number] | 是 | 当前页码 |
| data>>totalPage | [number] | 是 | 总条数 |
| data>>totalNum | [number] | 是 | 总页数 |
| code | [number] | 是 | 请求返回码 |

##### 实体类

###### FNodePartyDo

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | FNodePartyDo | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 保存系统审批流程的详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\federal\bean\federal\FNodePartyDo | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | long | | ■公有 □私有 □保护 | | | NNU | |  |
| name | string | | ■公有 □私有 □保护 | | | NNU | | 节点名称 |
| serverconfig | string | | ■公有 □私有 □保护 | | |  | | 服务类型 |
| partyid | integer | | ■公有 □私有 □保护 | | | NNU | | 节点识别码 |
| ip | string | | ■公有 □私有 □保护 | | | NNU | | 节点ip |
| port | integer | | ■公有 □私有 □保护 | | |  | | 节点端口 |
| status | integer | | ■公有 □私有 □保护 | | |  | | 状态 |
| backend | integer | | ■公有 □私有 □保护 | | |  | | 运行模式 0为eggroll 1为spark rabbitmq 2为spark pulsar |
| workmode | integer | | ■公有 □私有 □保护 | | |  | | 存储类型 1为mysql |
| prefixsymbol | string | | ■公有 □私有 □保护 | | |  | | 前缀符号 |
| numexecutors | integer | | ■公有 □私有 □保护 | | |  | | spark执行任务的并发数 |
| drivermemory | string | | ■公有 □私有 □保护 | | |  | | spark应用可分配的内存大小 |
| executormemory | string | | ■公有 □私有 □保护 | | |  | | spark executor进程的内存大小 |
| executorcores | integer | | ■公有 □私有 □保护 | | |  | | spark executor进程的CPU核心数量 |
| creatorid | string | | ■公有 □私有 □保护 | | |  | | 创建人ID |
| creatoraccount | string | | ■公有 □私有 □保护 | | |  | | 创建人账号 |
| creatorname | string | | ■公有 □私有 □保护 | | |  | | 创建人姓名 |
| flowaddress | string | | ■公有 □私有 □保护 | | |  | | 流程服务资源地址 |
| icon | string | | ■公有 □私有 □保护 | | |  | | 图标 |
| email | string | | ■公有 □私有 □保护 | | |  | | 邮箱 |
| cpulimit | float | | ■公有 □私有 □保护 | | |  | | CPU最大使用核心数 |
| memlimit | integer | | ■公有 □私有 □保护 | | |  | | 内存最大使用大小 |
| chiplimit | float | | ■公有 □私有 □保护 | | |  | | 芯片最大使用数 |
| mobile | string | | ■公有 □私有 □保护 | | |  | | 电话 |
| isstatus | integer | | ■公有 □私有 □保护 | | |  | | 是否在线 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | fl\_node\_party | | |

算法描述

无

状态转换描述

无

#### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：fl\_node\_party**  **>**  **操作：<查询>** |
| SQL语句：  【SQL01】  SELECT a.\*,data\_party\_cnt from fl\_node\_party a left join( select count(\*) data\_party\_cnt ,node\_id nid from fl\_data\_party group by node\_id ) b on a.id =b.nid where is\_status =0 order by a.created\_time desc limit 0,10  【SQL02】  select e.\*,case when use\_cnt is not null then use\_cnt else 0 end use\_cnt,case when user\_id=? then 3 when grant\_flag = 1 then 1 when apply\_flag = 1 then 1 else 0 end user\_priv\_grant, f.node\_party\_name,apply\_count,apply\_flag,grant\_count,grant\_flag,favourite\_count,favourite\_flag,discuss\_total\_count,discuss\_user\_count from ( select a.id, 'db' scatalog, a.id obj\_id, a.user\_id,a.user\_name,a.user\_account, a.node\_party\_id, a.schema\_id, a.db\_id, dbsource\_id datasource\_id ,a.object\_name , a.obj\_full\_name ,a.obj\_chn\_name ,a.dc\_id,a.created\_time ,a.creator\_name,a.creator\_account, 1 priv\_flag ,null ds\_label,'ide' stype, null remark,a.share\_flag, b.schema\_name , b.schema\_chn\_name , c.db\_name ,c.dc\_db\_name ,c.db\_chn\_name from t\_bydb\_object a, t\_bydb\_schema b, t\_bydb\_database c where a.db\_id = c.id and a.schema\_id =b.id and a.enable = 1 ) e left join (select id ,name node\_party\_name from fl\_node\_party where status = 1 ) f on e.node\_party\_id = f.id left join ( select count(\*) apply\_count,max(case when creator\_id=? then approve else 0 end) apply\_flag, data\_id rel\_id from fl\_data\_approve where approve = 2 group by data\_id ) g on e.id = g.rel\_id left join ( select count(\*) favourite\_count,max(case when creator\_id=? and node\_party\_id=? then 1 else 0 end) favourite\_flag, rel\_id from t\_tru\_favourite\_object group by rel\_id ) h on e.id = h.rel\_id left join ( select count(\*) discuss\_total\_count,sum(case when discuss\_account=? and node\_party\_id=? then 1 else 0 end) discuss\_user\_count, rel\_id from t\_tru\_discuss\_object group by rel\_id ) j on e.id = j.rel\_id left join ( select count(\*) grant\_count,sum(case when creator\_id=? then 1 else 0 end) grant\_flag, data\_id rel\_id from fl\_data\_approve where approve = 1 group by data\_id ) k on e.id = k.rel\_id left join ( select count(id) use\_cnt, real\_obj\_id from t\_tru\_model\_object group by real\_obj\_id ) l on e.id = l.real\_obj\_id where f.id is not null and ( ( e.share\_flag = 2 and e.id in ( select data\_id from t\_bydb\_data\_node where node\_id = ? ) ) or ( e.share\_flag =1) or ( e.user\_id = ? ) ) and scatalog = ? and node\_party\_id= ? order by scatalog,created\_time desc limit 0,1 |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | creatorid | 用户id |  | | partyid | 节点识别码 |  | | creatoraccount | 创建人账号 |  | | flowaddress | 服务资源地址 |  | | chiplimit | 芯片最大使用数 |  | | name | 成员名称 |  | | cpulimit | CPU最大使用核心数 |  | | memlimit | 内存最大使用大小 |  | | description | 描述 |  | | status | 状态 |  | |

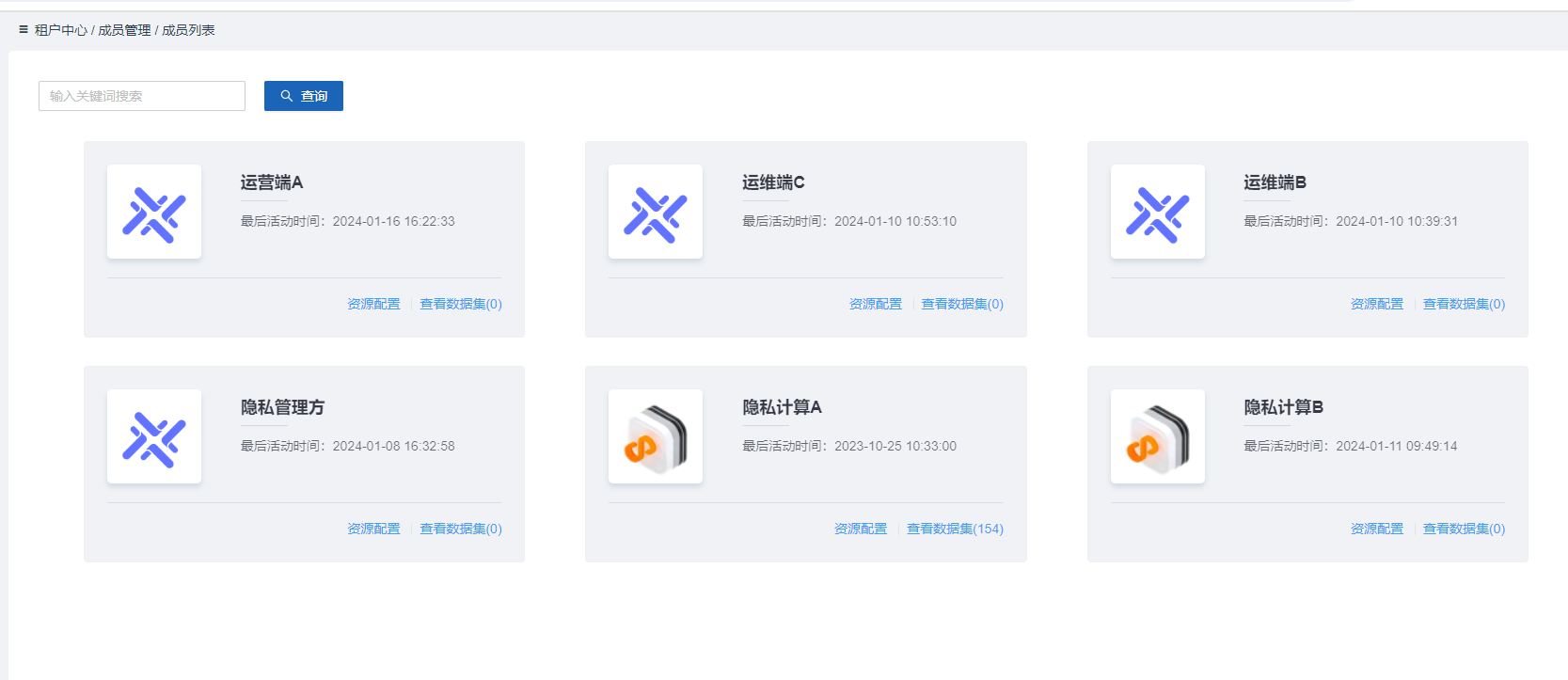
#### 总体设计方案

##### 设计方案

成员管理展示用户待审批的审批流程，用户可以进行审批与查看详情操作。

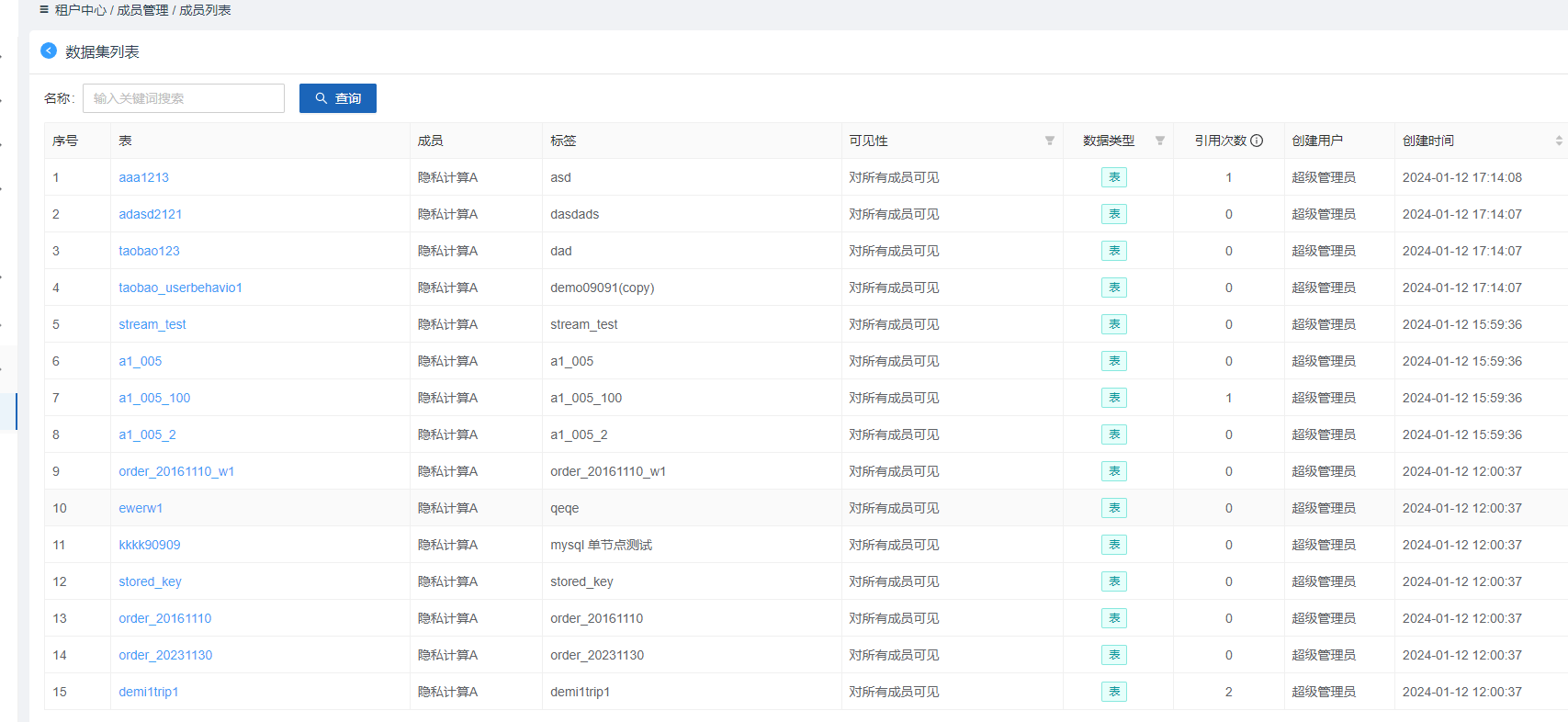
##### 处理流程

1. 查询成员列表



页面调用【7.4.4.2.1.2.1.成员管理-成员列表】接口，接口封装查询条件执行【SQL01】查询当前所有成员列表数据。分别展示成员名称最近活动时间。

1. 成员数据集

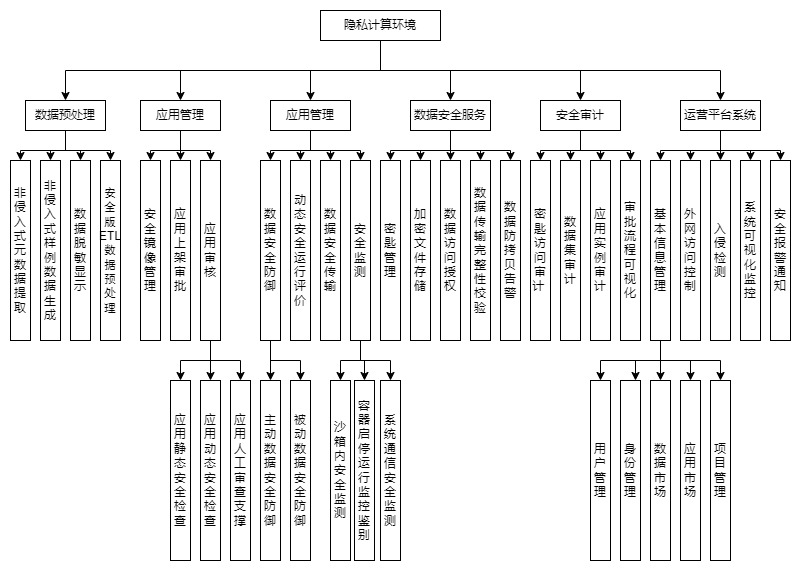


页面点击【详情】弹出成员数据集详情页面，页面调用【7.4.4.2.1.2.2成员管理-成员数据集】根据成员id执行【SQL02】查询展示数据集详情信息。

## 可信执行容器

可信执行容器提供数据预处理、应用管理、安全沙箱容器、数据安全服务、安全审计、运营管理共6个方面的支撑服务。

主要包含的功能如下：



### 数据预处理

#### 非侵入式元数据提取

非侵入式元数据提取文件的元数据。主要流程如下：



##### 类设计

###### 控制类

RsdpDatasetDetailController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpDatasetDetailController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 数据集详情处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/controller/RsdpDatasetDetailController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| setFileMetadata | dto | RsdpDatasetDetailDTO | ■公有  □私有  □保护 | Result | 保存数据集文件元数据 |

算法描述

保存数据集文件元数据

接口描述：保存元数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /rsdpdataset/rsdpdatasetdetail/setmetadata | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| datasetId | Long | 是 | 数据集id |
| filepath | String | 是 | 文件路径 |
| metadata | String | 是 | 元数据 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpDatasetDetailEntity

摘要

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | |
| 英文名称 | | RsdpDatasetDetailEntity | | | | |
| 父类 | | BaseEntity | | | | |
| 类职责简述 | | 数据集详情实体类 | | | | |
| 持久化 | | 是 | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/entity/RsdpDatasetDetailEntity.java | | | | |
| 属性 | | | | | | |
| 属性名称 | | 类型 | 可见性 | | 数据库校验\* | 说明 |
| datasetId | | Long | □公有 ■私有 □保护 | |  | 数据集ID |
| filename | | String | □公有 ■私有 □保护 | |  | 文件名称 |
| filepath | | String | □公有 ■私有 □保护 | |  | 文件路径 |
| filesize | | Long | □公有 ■私有 □保护 | |  | 文件大小 |
| metadata | | String | □公有 ■私有 □保护 | |  | 元数据 |
| lastModified | | Date | □公有 ■私有 □保护 | |  | 最后修改日期 |
| previewPath | | String | □公有 ■私有 □保护 | |  | 预览文件路径 |
| updater | | Long | □公有 ■私有 □保护 | |  | 更新者 |
| updateDate | | Date | □公有 ■私有 □保护 | |  | 更新日期 |
| 方法 | | | | | | |
| 方法名称 | 参数名称 | | 参数类型 | 可见性 | 返回值 | 说明 |
| 无 | 无 | | 无 | 无 | 无 | 无 |
| 对应数据库表 | | | | | rsdp\_dataset\_detail | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_dataset\_detail **>**  **操作：<更新>** |
| SQL语句：  【SQL01】保存数据集文件元数据  UPDATE rsdp\_dateset\_detail SET metadata=? WHERE dataset\_id=? AND filepath=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | dataset\_id | 数据集ID |  | | filepath | 文件路径 |  | | metadata | 元数据 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户上传csv文件，系统在不读取文件内容的前提下读取其元数据，并保存到数据库。

###### 处理流程



1.上传文件

用户进入数据集页面，点击上传，弹出上传文件对话框。

2.元数据生成

系统会对上传的文件进行读取，在不读取文件内容的前提下，调用【7.5.1.1.1.1.1.2.1保存数据集文件元数据】，执行【SQL01】保存数据集文件元数据。

#### 非侵入式样例数据生成

非侵入式样例数据生成提供从数据集自动化生成样例数据的功能。主要流程如下：



##### 类设计

###### 控制类

RsdpDatasetDetailController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpDatasetDetailController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 数据集详情处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/controller/RsdpDatasetDetailController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| setFilePreview | dto | RsdpDatasetDetailDTO | ■公有 □私有 □保护 | Result | 保存数据集样例文件 |

算法描述

保存数据集样例文件

接口描述：保存数据集样例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /rsdpdataset/rsdpdatasetdetail/setpreview | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| datasetId | Long | 是 | 数据集id |
| filepath | String | 是 | 文件路径 |
| preview\_path | String | 是 | 样例文件路径 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpDatasetDetailEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | RsdpDatasetDetailEntity | | | | | | | |
| 父类 | BaseEntity | | | | | | | |
| 类职责简述 | 数据集详情实体类 | | | | | | | |
| 持久化 | 是 | | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/entity/RsdpDatasetDetailEntity.java | | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | | 类型 | 可见性 | | | 数据库校验\* | | 说明 |
| datasetId | | Long | □公有 ■私有 □保护 | | |  | | 数据集ID |
| filename | | String | □公有 ■私有 □保护 | | |  | | 文件名称 |
| filepath | | String | □公有 ■私有 □保护 | | |  | | 文件路径 |
| filesize | | Long | □公有 ■私有 □保护 | | |  | | 文件大小 |
| metadata | | String | □公有 ■私有 □保护 | | |  | | 元数据 |
| lastModified | | Date | □公有 ■私有 □保护 | | |  | | 最后修改日期 |
| previewPath | | String | □公有 ■私有 □保护 | | |  | | 预览文件路径 |
| updater | | Long | □公有 ■私有 □保护 | | |  | | 更新者 |
| updateDate | | Date | □公有 ■私有 □保护 | | |  | | 更新日期 |
| 方法 | | | | | | | | |
| 方法名称 | | 参数名称 | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | | 无 | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_dataset\_detail | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_dataset\_detail **>**  **操作：<更新>** |
| SQL语句：  【SQL01】保存数据集样例文件  UPDATE rsdp\_dateset\_detail SET preview\_path=? WHERE dataset\_id=? AND filepath=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | dataset\_id | 数据集ID |  | | filepath | 文件路径 |  | | preview\_path | 预览文件路径 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户上传csv文件，系统在不读取整体文件内容的前提下生成其样例数据，并保存到数据库。

###### 处理流程



1. 上传文件

用户进入数据集页面，点击上传，弹出上传文件对话框。从资源管理器中选择文件或拖动文件到选择框中，上传文件。

1. 样例数据生成

选择自动生成预览数据，同时选择采样数量，可以选择数值或百分比。选择完成后点击确定。系统调用【7.5.1.2.1.1.1.2.1保存数据集样例文件】执行【SQL01】保存数据集样例文件。然后点击左侧文件列表的文件，就可以在右侧预览栏看到预览数据。

#### 数据脱敏显示

##### 类设计

###### 控制类

RsdpDatasetDetailController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpDatasetDetailController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 数据集详情处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/controller/RsdpDatasetDetailController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| set Desensitization | dto | RsdpDatasetDetailDTO | ■公有 □私有 □保护 | Result | 设置数据文件脱敏字段 |

算法描述

设置数据文件脱敏字段

接口描述：设置脱敏字段。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /rsdpdataset/rsdpdatasetdetail/setdesensitization | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| datasetId | Long | 是 | 数据集id |
| filepath | String | 是 | 文件路径 |
| desensitization | String | 是 | 脱敏字段 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpDatasetDetailEntity

摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | RsdpDatasetDetailEntity | | | | | | |
| 父类 | BaseEntity | | | | | | |
| 类职责简述 | 数据集详情实体类 | | | | | | |
| 持久化 | 是 | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpdataset/entity/RsdpDatasetDetailEntity.java | | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | 数据库校验\* | | 说明 |
| datasetId | Long | □公有 ■私有 □保护 | | |  | | 数据集ID |
| filename | String | ☑公有 ■私有 □保护 | | |  | | 文件名称 |
| filepath | String | □公有 ■私有 □保护 | | |  | | 文件路径 |
| filesize | Long | □公有 ■私有 □保护 | | |  | | 文件大小 |
| metadata | String | □公有 ■私有 □保护 | | |  | | 元数据 |
| lastModified | Date | □公有 ■私有 □保护 | | |  | | 最后修改日期 |
| previewPath | String | □公有 ■私有 □保护 | | |  | | 预览文件路径 |
| updater | Long | □公有 ■私有 □保护 | | |  | | 更新者 |
| updateDate | Date | □公有 ■私有 □保护 | | |  | | 更新日期 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | | 返回值 | 说明 | |
| 无 | 无 | 无 | 无 | | 无 | 无 | |
| 对应数据库表 | | | | rsdp\_dataset\_detail | | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_dataset\_detail **>**  **操作：<更新>** |
| SQL语句：  【SQL01】设置数据文件脱敏字段  UPDATE rsdp\_dateset\_detail SET desensitization=? WHERE dataset\_id=? AND filepath=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | dataset\_id | 数据集ID |  | | filepath | 文件路径 |  | | desensitization | 脱敏字段 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户上传csv文件，读取到元数据后，可以设置数据脱敏，用于隐藏敏感信息。

###### 处理流程



1.上传文件

用户进入数据集页面，点击上传，弹出上传文件对话框。从资源管理器中选择文件或拖动文件到选择框中，上传文件。

2.设置脱敏字段

在左侧选择文件，右侧点击脱敏设置，会弹出数据脱敏对话框，选择启用脱敏，然后选择脱敏规则，系统一共预置了8种规则，分别为：姓名、身份证、手机号、邮箱、日期、年龄、地址以及密码。选择相应的规则，然后点击确定。系统调用【7.5.1.3.1.1.1.2.1-设置数据文件脱敏字段】执行【SQL01】设置数据文件脱敏字段。切换到预览一栏，可以看到设置了数据脱敏的列，其数据显示的是脱敏后的结果。

#### 安全版ETL数据预处理

##### 类设计

###### 控制类

RsdpProjectInstanceController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpProjectInstanceController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 项目实例处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpproject/controller/RsdpProjectInstanceController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| save | dto | RsdpProjectInstanceDTO | ■公有 □私有 □保护 | Result | 创建项目实例 |

算法描述

创建项目实例

接口描述：创建项目实例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /rsdpprojectinstance/rsdpprojectinstance | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| projectId | Long | 是 | 项目ID |
| appName | String | 是 | 应用名称 |
| appId | Long | 是 | 应用Id |
| instanceName | String | 是 | 实例名称 |
| instanceNamespace | String | 是 | 实例命名空间 |
| instanceType | String | 是 | 实例类型 |
| instanceStatus | String | 是 | 实例状态 |
| instanceStorage | int | 是 | 实例网盘存储配额 |
| instanceCache | String | 是 | 实例缓存文件 |
| instanceChartName | String | 是 | 实例Helm Chart应用名称 |
| instanceChartVersion | String | 是 | 实例Helm Chart应用版本 |
| instanceImageName | String | 是 | 实例镜像名称 |
| instanceImageTag | String | 是 | 实例镜像Tag |
| instanceCpu | String | 是 | 实例cpu |
| instanceMem | String | 是 | 实例内存(G) |
| instanceValues | String | 是 | 实例其他参数 |
| instanceDatasetIds | List<String> | 是 | 实例挂载的数据集Id列表 |
| instanceUseLoadBalancer | Integer | 是 | 实例使用loadbalancer |
| instanceLastStartDate | Date | 是 | 实例上次启动时间 |
| instanceLastStopRunningSeconds | Long | 是 | 实例上次启动前总运行秒数 |
| instanceReplicaCount | Integer | 是 | 实例副本数量 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpProjectInstanceEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | RsdpProjectInstanceEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 项目实例实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpproject/entity/RsdpProjectInstanceEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| appName | String | | □公有 ■私有 □保护 | | |  | | 应用名称 |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用Id |
| instanceName | String | | □公有 ■私有 □保护 | | |  | | 实例名称 |
| instanceNamespace | String | | □公有 ■私有 □保护 | | |  | | 实例命名空间 |
| instanceType | String | | □公有 ■私有 □保护 | | |  | | 实例类型 |
| instanceStatus | String | | □公有 ■私有 □保护 | | |  | | 实例状态 |
| instanceStorage | int | | □公有 ■私有 □保护 | | |  | | 实例网盘存储配额 |
| instanceCache | String | | □公有 ■私有 □保护 | | |  | | 实例缓存文件 |
| instanceChartName | String | | □公有 ■私有 □保护 | | |  | | 实例Helm Chart应用名称 |
| instanceChartVersion | String | | □公有 ■私有 □保护 | | |  | | 实例Helm Chart应用版本 |
| instanceImageName | String | | □公有 ■私有 □保护 | | |  | | 实例镜像名称 |
| instanceImageTag | String | | □公有 ■私有 □保护 | | |  | | 实例镜像Tag |
| instanceCpu | String | | □公有 ■私有 □保护 | | |  | | 实例cpu |
| instanceMem | String | | □公有 ■私有 □保护 | | |  | | 实例内存(G) |
| instanceValues | String | | □公有 ■私有 □保护 | | |  | | 实例其他参数 |
| instanceDatasetIds | List<String> | | □公有 ■私有 □保护 | | |  | | 实例挂载的数据集Id列表 |
| instanceUseLoadBalancer | Integer | | □公有 ■私有 □保护 | | |  | | 实例使用loadbalancer |
| instanceLastStartDate | Date | | □公有 ■私有 □保护 | | |  | | 实例上次启动时间 |
| instanceLastStopRunningSeconds | Long | | □公有 ■私有 □保护 | | |  | | 实例上次启动前总运行秒数 |
| instanceReplicaCount | Integer | | □公有 ■私有 □保护 | | |  | | 实例副本数量 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_project\_instance | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_project\_instance **>**  **操作：<新增>** |
| SQL语句：  【SQL01】创建项目实例  INSERT INTO rsdp\_project\_instance ( id, project\_id, app\_name, app\_id, instance\_name, instance\_namespace, instance\_type, instance\_storage, instance\_status, instance\_chart\_name, instance\_chart\_version, instance\_image\_name, instance\_image\_tag, instance\_cpu, instance\_mem, instance\_values, instance\_use\_load\_balancer, updater, update\_date, creator, create\_date ) VALUES |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | projectId | 项目ID |  | | appName | 应用名称 |  | | appId | 应用Id |  | | instanceName | 实例名称 |  | | instanceNamespace | 实例命名空间 |  | | instanceType | 实例类型 |  | | instanceStatus | 实例状态 |  | | instanceStorage | 实例网盘存储配额 |  | | instanceCache | 实例缓存文件 |  | | instanceChartName | 实例Helm Chart应用名称 |  | | instanceChartVersion | 实例Helm Chart应用版本 |  | | instanceImageName | 实例镜像名称 |  | | instanceImageTag | 实例镜像Tag |  | | instanceCpu | 实例cpu |  | | instanceMem | 实例内存(G) |  | | instanceValues | 实例其他参数 |  | | instanceDatasetIds | 实例挂载的数据集Id列表 |  | | instanceUseLoadBalancer | 实例使用loadbalancer |  | | instanceLastStartDate | 实例上次启动时间 |  | | instanceLastStopRunningSeconds | 实例上次启动前总运行秒数 |  | | instanceReplicaCount | 实例副本数量 |  | | creator | 创建者 |  | | creatorName | 创建者名称 |  | | createDate | 创建日期 |  | | updater | 更新者 |  | | updateDate | 更新时间 |  | | extraInfo | 额外信息 |  | | useUnreadableDataset | 应用安全级别 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户创建kettle项目实例，通过容器暴露的端口进入spoon页面，在此页面设计ETL数据预处理。

###### 处理流程



1.创建项目实例

用户进入项目详情页，点击应用栏，添加ETL Tool应用到项目，然后点击启动实例按钮，弹出新建服务实例对话框，输入各种信息后，点击确定。系统调用【7.5.1.4.1.1.1.2.1-创建项目实例】调用【SQL01】创建项目实例。

2.ETL数据预处理

页面会自动跳转到工作实例标签，可以看到刚刚创建的实例，点击刷新按钮刷新实例状态，当实例完全启动后，点击在浏览器中打开，打开容器开放的端口，进入spoon程序。在spoon中，可以设计数据来源及数据处理方式，数据处理完成后可以导出。

### 应用管理

#### 安全镜像管理

##### 类设计

###### 控制类

ApplicationController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用安全镜像相关接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\application\controller\ApplicationController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| list | 无 | int  int | ■公有 □私有 □保护 | List<ApplicationImage> | 获取镜像列表 |
| download | params | Map<String, Object> | ■公有 □私有 □保护 | TgzFile | 下载应用 |

算法描述

获取应用列表接口

接口描述：查询应用列表，支持分页查询。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /list | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| list | [array] | 是 | 应用镜像列表 |

应用下载接口

接口描述：下载用户所需应用。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /download | application/x-gzip |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

###### 实体类

ApplicationEntity

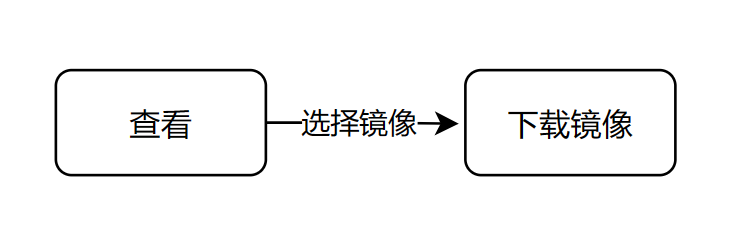
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ApplicationEntity | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 应用详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\rsdpapplication\entity\ApplicationEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| deptId | Long | | □公有 ■私有 □保护 | | | NNIC | | 所属机构ID |
| deptName | String | | □公有 ■私有 □保护 | | |  | | 所属机构名称 |
| name | String | | □公有 ■私有 □保护 | | |  | | 应用名称 |
| chartName | String | | □公有 ■私有 □保护 | | |  | | Helm Chart 应用名称 |
| useUnreadableDataset | Integer | | □公有 ■私有 □保护 | | |  | | 可以使用不可见数据集 |
| status | Integer | | □公有 ■私有 □保护 | | | NNIC | | 应用状态 |
| imageName | string | | □公有 ■私有 □保护 | | |  | | 镜像名称 |
| imageTag | string | | □公有 ■私有 □保护 | | |  | | 镜像Tag |
| imageId | string | | □公有 ■私有 □保护 | | |  | | 镜像ID |
| imageStatus | string | | □公有 ■私有 □保护 | | |  | | 镜像状态：0commit成功 1commit出错 2已同步到仓库 3同步仓库失败 |
| ports | string | | □公有 ■私有 □保护 | | |  | | 应用服务端口 |
| icon | string | | □公有 ■私有 □保护 | | |  | | 应用图标 |
| credit | string | | □公有 ■私有 □保护 | | |  | | 所需积分 |
| platform | string | | □公有 ■私有 □保护 | | |  | | 所属平台 |
| description | string | | □公有 ■私有 □保护 | | |  | | 描述 |
| author | string | | □公有 ■私有 □保护 | | |  | | 应用提供者 |
| type | string | | □公有 ■私有 □保护 | | |  | | 应用类型 |
| version | string | | □公有 ■私有 □保护 | | |  | | 应用版本 |
| updater | Long | | □公有 ■私有 □保护 | | | NNIC | | 更新者 |
| updateDate | Date | | □公有 ■私有 □保护 | | |  | | 更新时间 |
| canDelete | string | | □公有 ■私有 □保护 | | |  | | 能否删除 |
| canDownload | string | | □公有 ■私有 □保护 | | |  | | 是否允许下载 |
| creatorName | string | | □公有 ■私有 □保护 | | |  | | 创建者名称 |
| staticFileDir | string | | □公有 ■私有 □保护 | | |  | | 静态文件目录 |
| swaggerFile | string | | □公有 ■私有 □保护 | | |  | | API描述文档 |
| outputPaths | string | | □公有 ■私有 □保护 | | |  | | 输出文件路径 |
| outputSamples | string | | □公有 ■私有 □保护 | | |  | | 输出文件样例 |
| dataImportMethod | string | | □公有 ■私有 □保护 | | |  | | 数据导入方式 |
| datasetsToUse | string | | □公有 ■私有 □保护 | | |  | | 计划使用的数据集 |
| datasetFiles | string | | □公有 ■私有 □保护 | | |  | | 数据集文件 |
| usageSteps | string | | □公有 ■私有 □保护 | | |  | | 使用步骤 |
| imageFileReady | string | | □公有 ■私有 □保护 | | |  | | 镜像文件就绪状态：0 未就绪 1已就绪 2 同步中 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_application | | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询应用镜像列表  SELECT  e.id,  e.name,  e.chart\_name,  e.use\_unreadable\_dataset,  e.status,  e.image\_name  e.image\_tag,  e.image\_id,  e.image\_status  e.ports,  e.icon,  e.description,  e.author,  e.type,  e.version,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.dept\_id,  e.credit,  e.platform,  e.static\_file\_dir,  e.swagger\_file,  e.output\_paths,  e.output\_samples,  e.data\_import\_method,  e.datasets\_to\_use,  e.dataset\_files,  e.usage\_steps,  e.image\_file\_ready; |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | ID |  | | name | 应用名称 |  | | chart\_name | Helm Chart 应用名称 |  | | use\_unreadable\_dataset | 安全级别：0开发级 1生产级 |  | | status | 应用状态：-1未审核 0审核中 1已发布 2审核拒绝 |  | | image\_name | 镜像名称 |  | | image\_tag | 镜像Tag |  | | image\_id | 镜像ID |  | | image\_status | 镜像状态：0commit&push中 1commit&push成功 2commit||push出错 |  | | ports | 应用服务端口 |  | | icon | 图标 |  | | description | 描述 |  | | author | 应用提供者 |  | | type | 应用类型，分为web服务类、计算类及混合类 |  | | version | 应用Helm Chart版本 |  | | creator | 创建者 |  | | create\_date | 创建时间 |  | | updater | 更新者 |  | | update\_date | 更新时间 |  | | dept\_id | 所属机构ID |  | | credit | 所需积分 |  | | platform | 所属平台 |  | | static\_file\_dir | 静态文件目录 |  | | swagger\_file | API描述文档路径 |  | | output\_paths | 输出文件路径 |  | | output\_samples | 输出文件样例路径 |  | | data\_import\_method | 数据导入方式：分为upload(上传文件)和dataset(使用已有数据集) |  | | datasets\_to\_use | 计划使用数据集 |  | | dataset\_files | 数据集文件 |  | | usage\_steps | 使用步骤 |  | | image\_file\_ready | 镜像文件就绪状态：0 未就绪 1已就绪 2 同步中 |  | |
|  |

##### 总体设计方案

###### 设计方案

基于开源的 Harbor 实现的容器镜像仓库及管理系统，实现镜像的存放、拉取、版本管理，提供隐私计算平台自身容器，应用市场容器的镜像管理功能。

###### 处理流程

服务启动后，镜像仓库作为一个标准的Docker镜像仓库，提供容器镜像的推送、存储、拉取功能。平台运维人员可以在需要时通过镜像管理系统提供的标准界面来查看系统镜像，此时调用【7.5.2.1.1.1.1.2.1获取应用列表接口】，后台执行 【SQL01】语句获取到镜像列表。

此外，隐私计算平台自身的子系统容器镜像以及平台应用商店中由平台默认提供的应用容器镜像，在安装时已经预置在镜像仓库中。用户可以调用【7.5.2.1.1.1.1.2.2应用下载接口】下载镜像。

#### 应用静态安全检查

##### 类设计

###### 控制类

ApplicationController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\application\controller\ApplicationController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| downloadDatasetSample | path | String | ■公有 □私有 □保护 | CsvFile | 下载数据集样例 |
| fileDownload | id | Long | ■公有 □私有 □保护 | JsonFile | 下载接口swagger文档 |

算法描述

下载数据集样例接口

接口描述：下载数据集样例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /downloadDatasetSample | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| body | [text/csv] | 是 | 数据集样例文件 |

下载接口swagger文档接口

接口描述：下载接口swagger文档。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /download/swagger/{id} | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| body | [application/json] | 是 | Swagger文件 |

ApplicationAuditTaskController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationAuditTaskController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用安全镜像相关接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\controller\ApplicationAuditTaskController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| changeAuditStatus | processResult | ProcessResult | ■公有 □私有 □保护 | Result | 上报任务进度 |
| staticCheck | id | Long | ■公有 □私有 □保护 | Result | 开始静态检查 |

算法描述

上报任务进度接口

接口描述：上报任务进度接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /process | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| processResult | [ProcessResult] | 是 | processResult |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 结果 |

开始静态检查接口

接口描述：开始静态检查。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /static/start | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 结果 |

###### 实体类

ApplicationAuditTaskEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ApplicationAuditTaskEntity | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 应用审核任务详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\entity\ApplicationAuditTaskEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | Long | | □公有 ■私有 □保护 | | | NNU | | 主键ID |
| appId | Long | | □公有 ■私有 □保护 | | | NNI | | 应用ID |
| appName | String | | □公有 ■私有 □保护 | | | NNS | | 应用名称 |
| appType | String | | □公有 ■私有 □保护 | | | NNS | | 应用类型 |
| applicantDeptId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请方机构ID |
| applicantDeptName | String | | □公有 ■私有 □保护 | | | NNS | | 申请方机构名称 |
| applicantUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请人ID |
| applicantUserName | String | | □公有 ■私有 □保护 | | | NNS | | 申请人名称 |
| claimUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 处理人ID |
| claimUserName | String | | □公有 ■私有 □保护 | | | NNS | | 处理人名称 |
| status | Byte | | □公有 ■私有 □保护 | | | NNU | | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |
| creator | Long | | □公有 ■私有 □保护 | | | NNI | | 创建人 |
| createDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 创建日期 |
| updater | Long | | □公有 ■私有 □保护 | | | NNI | | 更新人 |
| updateDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 更新日期 |
| taskId | String | | □公有 ■私有 □保护 | | | NNS | | 流程任务ID |
| staticStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| staticResult | String | | □公有 ■私有 □保护 | | |  | | 静态审核结果 |
| staticAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 静态审核意见 |
| dynamicStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| dynamicResult | String | | □公有 ■私有 □保护 | | |  | | 动态审核结果 |
| dynamicAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 动态审核意见 |
| artificialStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| artificialResult | String | | □公有 ■私有 □保护 | | |  | | 人工审核结果 |
| artificialAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 人工审核意见 |
| finalOpinion | String | | □公有 ■私有 □保护 | | |  | | 任务最终审核意见 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_application\_audit\_task | | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询应用审核记录列表  SELECT  e.id,  e.app\_id,  e.app\_name,  e.app\_type,  e.applicant\_dept\_id,  e.applicant\_dept\_name,  e.applicant\_user\_id,  e.applicant\_user\_name,  e.claim\_user\_id,  e.claim\_user\_name,  e.status,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.task\_id,  e.static\_status,  e.static\_result,  e.static\_audit\_opinion,  e.dynamic\_status,  e.dynamic\_result,  e.dynamic\_audit\_opinion,  e.artificial\_status,  e.artificial\_result,  e.artificial\_audit\_opinion,  e.final\_opinion  【SQL02】查询指定id的应用审核记录  SELECT  e.id,  e.app\_id,  e.app\_name,  e.app\_type,  e.applicant\_dept\_id,  e.applicant\_dept\_name,  e.applicant\_user\_id,  e.applicant\_user\_name,  e.claim\_user\_id,  e.claim\_user\_name,  e.status,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.task\_id,  e.static\_status,  e.static\_result,  e.static\_audit\_opinion,  e.dynamic\_status,  e.dynamic\_result,  e.dynamic\_audit\_opinion,  e.artificial\_status,  e.artificial\_result,  e.artificial\_audit\_opinion,  e.final\_opinion  WHERE  e.id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<插入>** |
| SQL语句：  【SQL03】插入应用审计详细信息  INSERT INTO rsdp\_application\_audit\_task(id,  app\_id,  app\_name,  app\_type,  applicant\_dept\_id,  applicant\_dept\_name,  applicant\_user\_id,  applicant\_user\_name,  claim\_user\_id,  claim\_user\_name,  status,  creator,  create\_date,  updater,  update\_date,  task\_id,  static\_status,  static\_result,  static\_audit\_opinion,  dynamic\_status,  dynamic\_result,  dynamic\_audit\_opinion,  artificial\_status,  artificial\_result,  artificial\_audit\_opinion,  final\_opinion  )  VALUES  (#{id},  #{app\_id},  #{app\_name},  #{app\_type},  #{applicant\_dept\_id},  #{applicant\_dept\_name},  #{applicant\_user\_id},  #{applicant\_user\_name},  #{claim\_user\_id},  #{claim\_user\_name},  #{status},  #{creator},  #{create\_date},  #{updater},  #{update\_date},  #{task\_id},  #{static\_status},  #{static\_result},  #{static\_audit\_opinion},  #{dynamic\_status},  #{dynamic\_result},  #{dynamic\_audit\_opinion},  #{artificial\_status},  #{artificial\_result},  #{artificial\_audit\_opinion},  #{final\_opinion}  ) |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<更新>** |
| SQL语句：  【SQL04】更新应用审计详细信息  UPDATE  rsdp\_application\_audit\_task  SET  app\_id= #{app\_id},  app\_name= #{app\_name},  app\_type= #{app\_type},  applicant\_dept\_id= #{applicant\_dept\_id},  applicant\_dept\_name= #{applicant\_dept\_name},  applicant\_user\_id= #{applicant\_user\_id},  applicant\_user\_name= #{applicant\_user\_name,  claim\_user\_id= #{claim\_user\_id},  claim\_user\_name= #{claim\_user\_name},  status= #{status},  creator= #{creator},  create\_date= #{create\_date},  updater= #{updater},  update\_date= #{update\_date},  task\_id= #{task\_id},  static\_status= #{static\_status},  static\_result= #{static\_result},  static\_audit\_opinion= #{static\_audit\_opinion},  dynamic\_status= #{dynamic\_status},  dynamic\_result= #{dynamic\_result},  dynamic\_audit\_opinion= #{dynamic\_audit\_opinion},  artificial\_status= #{artificial\_status},  artificial\_result= #{artificial\_result},  artificial\_audit\_opinion= #{artificial\_audit\_opinion},  final\_opinion= #{final\_opinion},  WHERE  id = #{id}; |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 主键ID |  | | app\_id | 应用ID |  | | app\_name | 应用名称 |  | | app\_type | 应用类型 |  | | applicant\_dept\_id | 申请方机构ID |  | | applicant\_dept\_name | 申请方机构名称 |  | | applicant\_user\_id | 申请人ID |  | | applicant\_user\_name | 申请人名称 |  | | claim\_user\_id | 处理人ID |  | | claim\_user\_name | 处理人名称 |  | | status | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |  | | creator | 创建人 |  | | create\_date | 创建日期 |  | | updater | 更新人 |  | | update\_date | 更新日期 |  | | task\_id | 流程任务ID |  | | static\_status | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | static\_result | 静态审核结果 |  | | static\_audit\_opinion | 静态审核意见 |  | | dynamic\_status | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | dynamic\_result | 动态审核结果 |  | | dynamic\_audit\_opinion | 动态审核意见 |  | | artificial\_status | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | artificial\_result | 人工审核结果 |  | | artificial\_audit\_opinion | 人工审核意见 |  | | final\_opinion | 任务最终审核意见 |  | |
|  |

##### 总体设计方案

###### 设计方案

安全检查员通过个人管理菜单处理应用上架申请，进行静态与动态安全检查并决定是否批准，确保应用安全合规。在待审核菜单中，办理按钮启动上架申请处理对话框，显示应用信息。审核页面提供静态安全检查、动态安全检查和人工审查标签页，用户根据扫描结果决定是否驳回申请。

###### 处理流程

首先，安全检查员通过“个人管理\审核与申请\待审核”菜单找到待审核的应用上架申请记录，并点击“办理”按钮，进入上架申请处理对话框。该对话框展示了申请用户信息、应用名称、类型、图标、服务端口等信息。用户点击“开始处理”按钮后，跳转到应用上架审核页面。，此过程中需要下载用户审核应用的样例数据。调用【7.5.2.2.1.1.1.2.1下载数据集样例接口】和【7.5.2.2.1.1.2.2.2下载接口swagger文档接口】下载数据集样例数据和Swagger文档，安全员可以查看到应用审计记录列表，执行【SQL01】看到列表，先执行【SQL02】获取到审核记录的信息之后返回下载文件数据。

在审核页面，安全检查员可以看到申请上架的应用信息，包括应用名称、应用类型、图标、服务端口、申请人、申请机构、审核进度等。页面包含三个标签页：“静态安全检查”、“动态安全检查”和“人工审查”。

在“静态安全检查”标签页，用户点击“开始扫描”按钮，启动后台静态安全扫描。扫描结果显示后，如果扫描不通过，将显示原因。用户可以在扫描完成后重新发起扫描。根据扫描结果，用户可以点击“驳回”按钮，输入审核意见，驳回用户的应用上架申请。以上过程需要调用【7.5.2.2.1.1.1.2.2开始静态检查接口】开启静态检查。执行【SQL03】创建应用审计记录。

若扫描通过，安全员调用【7.5.2.2.1.1.2.2.1上报任务进度接口】进行静态审核结果的上报，后台会执行【SQL04】更新应用审核的记录。之后可以进入“动态安全检查”标签页，进行动态安全检查。该设计方案为应用上架审核提供了一个详细的流程，确保应用在上架前经过严格的安全检查，保障用户的安全和应用的合规性。

#### 应用动态安全检查

##### 类设计

###### 控制类

ApplicationController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\application\controller\ApplicationController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| downloadDatasetSample | path | String | ■公有 □私有 □保护 | CsvFile | 下载数据集样例 |
| startDynamicInstance | instanceId | Long | ■公有 □私有 □保护 | Result | 启动审核实例 |
| StopDynamicInstance | instanceId | Long | ■公有 □私有 □保护 | Result | 关闭审核实例 |

算法描述

下载数据集样例接口

接口描述：下载数据集样例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /downloadDatasetSample | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| body | [text/csv] | 是 | 数据集样例文件 |

启动审核实例口

接口描述：启动审核实例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | startDynamicInstance | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| appId | [number] | 是 | 应用id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [number] | 是 | 实例id号 |

关闭审核实例口

接口描述：关闭审核实例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | stopDynamicInstance | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| instanceId | [number] | 是 | 实例id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

ApplicationAuditTaskController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationAuditTaskController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用安全镜像相关接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\controller\ApplicationAuditTaskController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| changeAuditStatus | processResult | ProcessResult | ■公有 □私有 □保护 | Result | 上报任务进度 |

算法描述

上报任务进度接口

接口描述：上报任务进度接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /process | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| processResult | [ProcessResult] | 是 | processResult |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 结果 |

###### 实体类

ApplicationAuditTaskEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ApplicationAuditTaskEntity | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 应用审核任务详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\entity\ApplicationAuditTaskEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | Long | | □公有 ■私有 □保护 | | | NNU | | 主键ID |
| appId | Long | | □公有 ■私有 □保护 | | | NNI | | 应用ID |
| appName | String | | □公有 ■私有 □保护 | | | NNS | | 应用名称 |
| appType | String | | □公有 ■私有 □保护 | | | NNS | | 应用类型 |
| applicantDeptId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请方机构ID |
| applicantDeptName | String | | □公有 ■私有 □保护 | | | NNS | | 申请方机构名称 |
| applicantUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请人ID |
| applicantUserName | String | | □公有 ■私有 □保护 | | | NNS | | 申请人名称 |
| claimUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 处理人ID |
| claimUserName | String | | □公有 ■私有 □保护 | | | NNS | | 处理人名称 |
| status | Byte | | □公有 ■私有 □保护 | | | NNU | | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |
| creator | Long | | □公有 ■私有 □保护 | | | NNI | | 创建人 |
| createDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 创建日期 |
| updater | Long | | □公有 ■私有 □保护 | | | NNI | | 更新人 |
| updateDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 更新日期 |
| taskId | String | | □公有 ■私有 □保护 | | | NNS | | 流程任务ID |
| staticStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| staticResult | String | | □公有 ■私有 □保护 | | |  | | 静态审核结果 |
| staticAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 静态审核意见 |
| dynamicStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| dynamicResult | String | | □公有 ■私有 □保护 | | |  | | 动态审核结果 |
| dynamicAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 动态审核意见 |
| artificialStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| artificialResult | String | | □公有 ■私有 □保护 | | |  | | 人工审核结果 |
| artificialAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 人工审核意见 |
| finalOpinion | String | | □公有 ■私有 □保护 | | |  | | 任务最终审核意见 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_application\_audit\_task | | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<查询>** |
| 【SQL01】查询指定id的应用审核记录  SELECT  e.id,  e.app\_id,  e.app\_name,  e.app\_type,  e.applicant\_dept\_id,  e.applicant\_dept\_name,  e.applicant\_user\_id,  e.applicant\_user\_name,  e.claim\_user\_id,  e.claim\_user\_name,  e.status,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.task\_id,  e.static\_status,  e.static\_result,  e.static\_audit\_opinion,  e.dynamic\_status,  e.dynamic\_result,  e.dynamic\_audit\_opinion,  e.artificial\_status,  e.artificial\_result,  e.artificial\_audit\_opinion,  e.final\_opinion  WHERE  e.id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<更新>** |
| SQL语句：  【SQL02】更新审核记录信息  UPDATE  rsdp\_application\_audit\_task  SET  app\_id= #{app\_id},  app\_name= #{app\_name},  app\_type= #{app\_type},  applicant\_dept\_id= #{applicant\_dept\_id},  applicant\_dept\_name= #{applicant\_dept\_name},  applicant\_user\_id= #{applicant\_user\_id},  applicant\_user\_name= #{applicant\_user\_name,  claim\_user\_id= #{claim\_user\_id},  claim\_user\_name= #{claim\_user\_name},  status= #{status},  creator= #{creator},  create\_date= #{create\_date},  updater= #{updater},  update\_date= #{update\_date},  task\_id= #{task\_id},  static\_status= #{static\_status},  static\_result= #{static\_result},  static\_audit\_opinion= #{static\_audit\_opinion},  dynamic\_status= #{dynamic\_status},  dynamic\_result= #{dynamic\_result},  dynamic\_audit\_opinion= #{dynamic\_audit\_opinion},  artificial\_status= #{artificial\_status},  artificial\_result= #{artificial\_result},  artificial\_audit\_opinion= #{artificial\_audit\_opinion},  final\_opinion= #{final\_opinion},  WHERE  id = #{id}; |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 主键ID |  | | app\_id | 应用ID |  | | app\_name | 应用名称 |  | | app\_type | 应用类型 |  | | applicant\_dept\_id | 申请方机构ID |  | | applicant\_dept\_name | 申请方机构名称 |  | | applicant\_user\_id | 申请人ID |  | | applicant\_user\_name | 申请人名称 |  | | claim\_user\_id | 处理人ID |  | | claim\_user\_name | 处理人名称 |  | | status | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |  | | creator | 创建人 |  | | create\_date | 创建日期 |  | | updater | 更新人 |  | | update\_date | 更新日期 |  | | task\_id | 流程任务ID |  | | static\_status | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | static\_result | 静态审核结果 |  | | static\_audit\_opinion | 静态审核意见 |  | | dynamic\_status | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | dynamic\_result | 动态审核结果 |  | | dynamic\_audit\_opinion | 动态审核意见 |  | | artificial\_status | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | artificial\_result | 人工审核结果 |  | | artificial\_audit\_opinion | 人工审核意见 |  | | final\_opinion | 任务最终审核意见 |  | |
|  |

##### 总体设计方案

###### 设计方案

安全检查员通过进入“待处理应用上架申请列表”进行审核。在“动态安全检查”标签页，可查看应用信息、启动实例、试用应用，并通过“检查最新状态”获取安全检查结果。若无安全问题，可进入“人工审查”进行进一步审核。安全检查员可进行暂停、继续、结束实例等操作，并可选择是否拒绝应用上架申请。

###### 处理流程

安全检查员处理应用上架申请的流程分为三个主要阶段：审核准备、动态安全检查和结果处理。

在审核准备阶段，安全检查员首先访问待处理应用上架申请列表，并选择一个应用进行审核。他们进入应用上架审核页面，并切换到“动态安全检查”标签页，在这里可以查看应用的基本信息，包括应用类型、输入输出数据、OpenAPI通信数据和应用说明等。过程中审核员随时可以调用【7.5.2.3.1.1.1.2.1下载数据集样例接口】下载数据集样例数据，执行【SQL01】获取应用审核记录的信息。

接下来是动态安全检查阶段，安全检查员启动应用实例，此时调用【7.5.2.3.1.1.1.2.2启动审核实例口】启动审核实例，并将样例数据导入安全容器内。实例启动后，安全检查员通过新浏览器窗口打开应用的web界面，并按照应用说明进行操作和试用。在此过程中，安全沙箱会记录并上报任何安全风险至系统后台。试用结束后，安全检查员检查最新状态，获取动态安全检查结果，如果有安全风险，则会以列表形式展示。检查完毕后，调用【7.5.2.3.1.1.1.2.3关闭审核实例口】关闭实例。以上过程中后台会执行【SQL02】更新审核记录的信息，将实例相关信息记录下来。

最后是结果处理阶段，安全检查员根据检查结果决定是否通过申请。如果发现安全风险，他们可以选择拒绝申请，并输入审核意见。如果没有发现问题，或者问题不严重，安全检查员可以下载输出数据和相关文件，进行人工审核，以确保数据安全和隐私保护。此外，安全检查员还可以暂停、继续或结束应用实例的运行，并重复进行实例的启动、查询和删除操作，直到满意为止。

#### 应用人工审查支撑

##### 类设计

###### 控制类

ApplicationController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\application\controller\ApplicationController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| downloadDatasetSample | path | String | ■公有 □私有 □保护 | CsvFile | 下载数据集样例 |
| startDynamicInstance | instanceId | Long | ■公有 □私有 □保护 | Result | 启动审核实例 |
| StopDynamicInstance | instanceId | Long | ■公有 □私有 □保护 | Result | 关闭审核实例 |

算法描述

下载数据集样例接口

接口描述：下载数据集样例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /downloadDatasetSample | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| body | [text/csv] | 是 | 数据集样例文件 |

启动审核实例口

接口描述：启动审核实例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | startDynamicInstance | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| appId | [number] | 是 | 应用id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [number] | 是 | 实例id号 |

关闭审核实例口

接口描述：关闭审核实例。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | stopDynamicInstance | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| instanceId | [number] | 是 | 实例id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

ApplicationAuditTaskController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationAuditTaskController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用安全镜像相关接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\controller\ApplicationAuditTaskController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| changeAuditStatus | processResult | ProcessResult | ■公有 □私有 □保护 | Result | 上报任务进度 |

算法描述

上报任务进度接口

接口描述：上报任务进度接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /process | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| processResult | [ProcessResult] | 是 | processResult |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 结果 |

###### 实体类

ApplicationAuditTaskEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ApplicationAuditTaskEntity | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | 应用审核任务详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\flow\authorize\entity\ApplicationAuditTaskEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| id | Long | | □公有 ■私有 □保护 | | | NNU | | 主键ID |
| appId | Long | | □公有 ■私有 □保护 | | | NNI | | 应用ID |
| appName | String | | □公有 ■私有 □保护 | | | NNS | | 应用名称 |
| appType | String | | □公有 ■私有 □保护 | | | NNS | | 应用类型 |
| applicantDeptId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请方机构ID |
| applicantDeptName | String | | □公有 ■私有 □保护 | | | NNS | | 申请方机构名称 |
| applicantUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 申请人ID |
| applicantUserName | String | | □公有 ■私有 □保护 | | | NNS | | 申请人名称 |
| claimUserId | Long | | □公有 ■私有 □保护 | | | NNI | | 处理人ID |
| claimUserName | String | | □公有 ■私有 □保护 | | | NNS | | 处理人名称 |
| status | Byte | | □公有 ■私有 □保护 | | | NNU | | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |
| creator | Long | | □公有 ■私有 □保护 | | | NNI | | 创建人 |
| createDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 创建日期 |
| updater | Long | | □公有 ■私有 □保护 | | | NNI | | 更新人 |
| updateDate | LocalDateTime | | □公有 ■私有 □保护 | | | NNU | | 更新日期 |
| taskId | String | | □公有 ■私有 □保护 | | | NNS | | 流程任务ID |
| staticStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| staticResult | String | | □公有 ■私有 □保护 | | |  | | 静态审核结果 |
| staticAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 静态审核意见 |
| dynamicStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| dynamicResult | String | | □公有 ■私有 □保护 | | |  | | 动态审核结果 |
| dynamicAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 动态审核意见 |
| artificialStatus | Byte | | □公有 ■私有 □保护 | | | NNU | | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |
| artificialResult | String | | □公有 ■私有 ☑保护 | | |  | | 人工审核结果 |
| artificialAuditOpinion | String | | □公有 ■私有 □保护 | | |  | | 人工审核意见 |
| finalOpinion | String | | □公有 ■私有 □保护 | | |  | | 任务最终审核意见 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_application\_audit\_task | | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询指定id的应用审核记录  SELECT  e.id,  e.app\_id,  e.app\_name,  e.app\_type,  e.applicant\_dept\_id,  e.applicant\_dept\_name,  e.applicant\_user\_id,  e.applicant\_user\_name,  e.claim\_user\_id,  e.claim\_user\_name,  e.status,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.task\_id,  e.static\_status,  e.static\_result,  e.static\_audit\_opinion,  e.dynamic\_status,  e.dynamic\_result,  e.dynamic\_audit\_opinion,  e.artificial\_status,  e.artificial\_result,  e.artificial\_audit\_opinion,  e.final\_opinion  WHERE  e.id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_application\_audit\_task**>**  **操作：<更新>** |
| SQL语句：  【SQL02】更新应用审核记录信息  UPDATE  rsdp\_application\_audit\_task  SET  app\_id= #{app\_id},  app\_name= #{app\_name},  app\_type= #{app\_type},  applicant\_dept\_id= #{applicant\_dept\_id},  applicant\_dept\_name= #{applicant\_dept\_name},  applicant\_user\_id= #{applicant\_user\_id},  applicant\_user\_name= #{applicant\_user\_name,  claim\_user\_id= #{claim\_user\_id},  claim\_user\_name= #{claim\_user\_name},  status= #{status},  creator= #{creator},  create\_date= #{create\_date},  updater= #{updater},  update\_date= #{update\_date},  task\_id= #{task\_id},  static\_status= #{static\_status},  static\_result= #{static\_result},  static\_audit\_opinion= #{static\_audit\_opinion},  dynamic\_status= #{dynamic\_status},  dynamic\_result= #{dynamic\_result},  dynamic\_audit\_opinion= #{dynamic\_audit\_opinion},  artificial\_status= #{artificial\_status},  artificial\_result= #{artificial\_result},  artificial\_audit\_opinion= #{artificial\_audit\_opinion},  final\_opinion= #{final\_opinion},  WHERE  id = #{id}; |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 主键ID |  | | app\_id | 应用ID |  | | app\_name | 应用名称 |  | | app\_type | 应用类型 |  | | applicant\_dept\_id | 申请方机构ID |  | | applicant\_dept\_name | 申请方机构名称 |  | | applicant\_user\_id | 申请人ID |  | | applicant\_user\_name | 申请人名称 |  | | claim\_user\_id | 处理人ID |  | | claim\_user\_name | 处理人名称 |  | | status | 任务状态：0 待处理 1 处理中 2 审核通过 3 审核未通过 |  | | creator | 创建人 |  | | create\_date | 创建日期 |  | | updater | 更新人 |  | | update\_date | 更新日期 |  | | task\_id | 流程任务ID |  | | static\_status | 静态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | static\_result | 静态审核结果 |  | | static\_audit\_opinion | 静态审核意见 |  | | dynamic\_status | 动态审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | dynamic\_result | 动态审核结果 |  | | dynamic\_audit\_opinion | 动态审核意见 |  | | artificial\_status | 人工审核状态：0 待处理 1 处理中 2 审核通过 3 审核拒绝 |  | | artificial\_result | 人工审核结果 |  | | artificial\_audit\_opinion | 人工审核意见 |  | | final\_opinion | 任务最终审核意见 |  | |
|  |

##### 总体设计方案

###### 设计方案

安全检查员通过进入待处理应用上架申请列表页面，进行人工审核。在系统后台安全区启动应用实例，下载并检查数据集样例，远程登录查看应用实例的运行状态和安全性。根据检查结果，可选择同意或驳回应用上架申请，并输入审核意见。通过审核的应用将在市场中展示。

###### 处理流程

安全检查员在处理应用上架申请时，首先进入待处理应用上架申请列表页面，然后点击“处理”按钮进入应用上架审核页面。在审核页面中，安全检查员需要选择进入“3. 人工审核”标签页，然后点击“启动实例”按钮，在系统后台安全区启动申请应用的实例。系统会根据用户申请时提供的信息，将所需输入数据集的样例数据导入应用实例的安全容器内。

安全检查员可以下载用户申请使用的数据集样例数据文件，并按照页面提示信息远程登录系统后台安全区。调用【7.5.2.4.1.1.1.2.1下载数据集样例接口】下载数据集样例数据，执行【SQL01】获取应用审核记录的信息。

此时调用【7.5.2.4.1.1.1.2.2启动审核实例口】启动审核实例，在安全区中，安全检查员可以以交互方式访问应用实例内部文件系统，查看进程运行情况、用户应用的端口开放情况、敏感数据的使用情况、临时目录的使用情况、资源占用情况以及通信连接情况等。同时，安全检查员可以选择点击“打开”按钮，在新的浏览器窗口中打开用户应用的web界面，按照应用说明操作、试用用户应用，并同步在交互模式下查看用户应用的运行状态。检查完毕后，调用【7.5.2.4.1.1.1.2.3关闭审核实例口】关闭实例。以上过程中后台会执行【SQL02】更新审核记录的信息，将实例相关信息记录下来。

安全检查员根据实际检查结果填写人工检查结果信息，包括是否通过检查以及不通过的具体原因。此外，安全检查员可以多次重复应用实例的启动、交互式检查和删除操作。根据是否检查到安全风险及安全风险严重程度，安全检查员可以选择点击“驳回”按钮，输入审核意见并确认，驳回用户的应用上架申请。如果安全检查员认为应用符合要求，可以选择点击“同意”按钮，输入审核意见并确认，同意用户的应用上架申请。通过上架申请的应用将在应用市场中展示。

### 安全沙箱容器

#### 主动数据安全防御

主动数据安全防御提供对实例的主动防御。主要流程如下：



##### 类设计

###### 控制类

ActiveDefenseController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ActiveDefenseController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/ActiveDefenseController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<ActiveDefenseDTO > | 分页获取主动防御策略列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<ActiveDefenseDTO> | 根据ID获取防御策略 |
| save | dto | ActiveDefenseDTO | ■公有 □私有 □保护 | Result | 添加主动防御策略 |
| update | dto | ActiveDefenseDTO | ■公有 □私有 □保护 | Result | 更新主动防御策略 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除主动防御策略 |

算法描述

分页获取主动防御策略列表

接口描述：分页获取主动防御策略列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/activedefense/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
|  |  |  |  |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取防御策略

接口描述：根据ID获取防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/activedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加主动防御策略

接口描述：添加主动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/activedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | ActiveDefenseDTO | 是 | 主动防御策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除主动防御策略

接口描述：删除主动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/acitvedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新主动防御策略

接口描述：更新主动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/activedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | ActiveDefenseDTO | 是 | 主动防御策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

ActiveDefenseEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | ActiveDefenseEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 主动防御实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/ActiveDefenseEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| interactive | Boolean | | □公有 ■私有 □保护 | | |  | | 是否允许容器交互 |
| fileMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 是否启用文件监控 |
| alarm | Boolean | | □公有 ■私有 □保护 | | |  | | 是否自动发送报警 |
| evaluable | Boolean | | □公有 ■私有 □保护 | | |  | | 是否启动评价子系统 |
| transmission | Boolean | | □公有 ■私有 □保护 | | |  | | 是否启动传输子系统 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | active\_defense | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

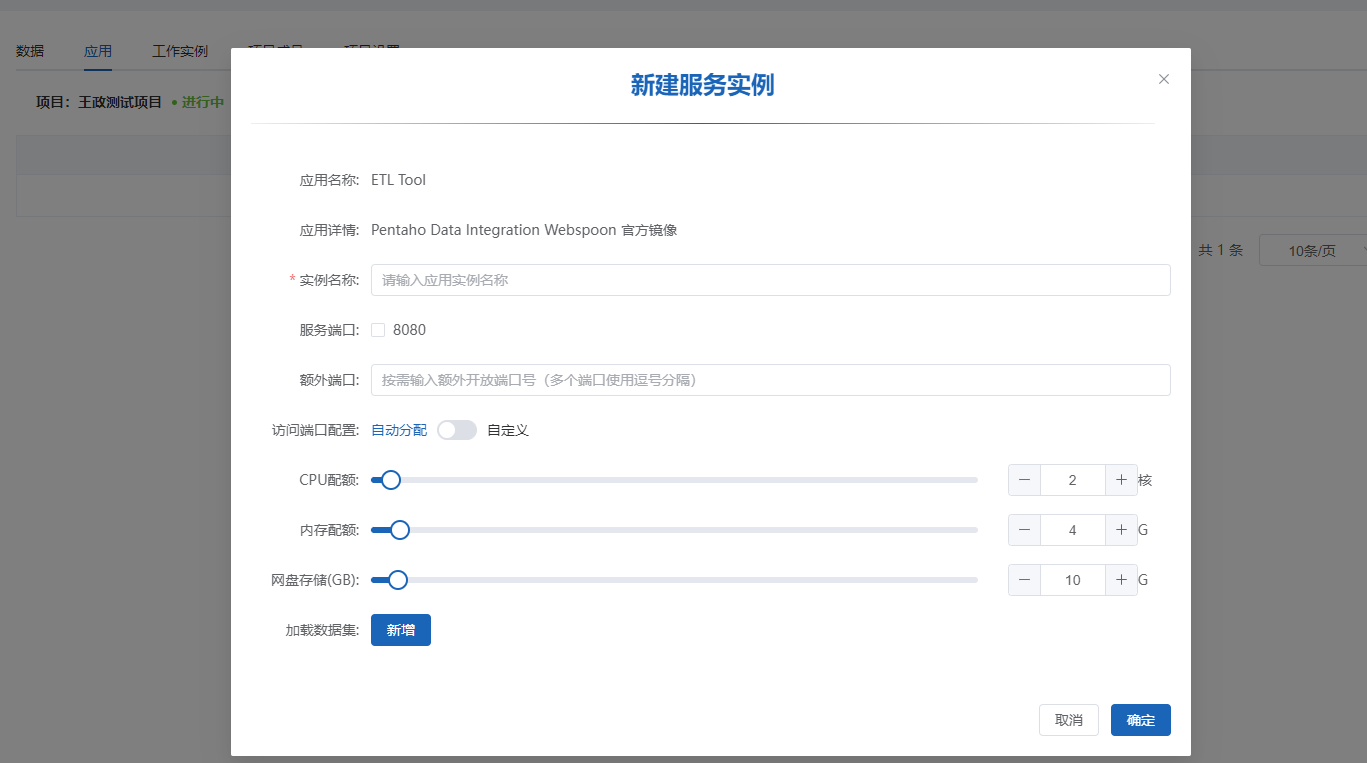
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**active\_defense **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取主动防御策略  select name, project\_id, instance\_id, app\_id, interactive, file\_monitor, alarm, evaluable, transmission from active\_defense limit ? offset ?  【SQL02】根据ID获取主动防御策略  select name, project\_id, instance\_id, app\_id, interactive, file\_monitor, alarm, evaluable, transmission from active\_defense where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**active\_defense **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增主动防御策略  insert into active\_defense (name, project\_id, instance\_id, app\_id, interactive, file\_monitor, alarm, evaluable, transmission) values (?, ?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**active\_defense **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除主动防御策略  delete from active\_defense where id=? |
| **程序流程中涉及的数据库表：<表名：**active\_defense **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新主动防御策略  update active\_defense set name=?, project\_id=?, instance\_id=?, app\_id=?, interactive=?, file\_monitor=?, alarm=?, evaluable=?, transmission=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | projectId | 项目ID |  | | instanceId | 实例ID |  | | appId | 应用ID |  | | interactive | 是否允许容器交互 |  | | fileMonitor | 是否启用文件监控 |  | | alarm | 是否自动发送报警 |  | | evaluable | 是否启动评价子系统 |  | | transmission | 是否启动传输子系统 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动应用主动安全防御策略。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动防御策略

切换到工作实例标签页，可以看到刚刚创建的实例，防御策略在实例启动后会自动启动。容器交互被封闭，文件监控启动，安全评价子系统启动，安全传输子系统启动。系统调用【7.5.3.1.1.1.1.2.1-分页获取主动防御策略列表】执行【SQL01】分页获取主动防御策略列表。调用【7.5.3.1.1.1.1.2.3-添加主动防御策略】执行【SQL03】添加主动防御策略。调用【7.5.3.1.1.1.1.2.4-删除主动防御策略】执行【SQL04】删除主动防御策略。调用【7.5.3.1.1.1.1.2.5-更新主动防御策略】执行【SQL05】更新主动防御策略。

#### 被动数据安全防御

被动数据安全防御提供对实例的被动防御。主要流程如下：

****

##### 类设计

###### 控制类

PassiveDefenseController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | PassiveDefenseController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/PassiveDefenseController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PassiveDefenseDTO > | 分页获取被动防御策略列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<PassiveDefenseDTO> | 根据ID获取防御策略 |
| save | dto | PassiveDefenseDTO | ■公有 □私有 □保护 | Result | 添加被动防御策略 |
| update | dto | PassiveDefenseDTO | ■公有 □私有 □保护 | Result | 更新被动防御策略 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除被动防御策略 |

算法描述

分页获取被动防御策略列表

接口描述：分页获取被动防御策略列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/passivedefense/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | ActiveDefenseDTO | 是 | 被动防御策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取防御策略

接口描述：根据ID获取防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/passivedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加被动防御策略

接口描述：添加被动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/passivedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | PassiveDefenseDTO | 是 | 被动防御策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除被动防御策略

接口描述：删除被动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/passivedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新被动防御策略

接口描述：更新被动防御策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/passivedefense | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | PassiveDefenseDTO | 是 | 被动防御策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

PassiveDefenseEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | PassiveDefenseEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 被动防御实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/PassiveDefenseEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| tlsEnable | Boolean | | □公有 ■私有 □保护 | | |  | | 是否启用https |
| sampleInput | String | | □公有 ■私有 □保护 | | |  | | 样例输入 |
| sampleOutput | String | | □公有 ■私有 □保护 | | |  | | 样例输出 |
| apiMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 是否启用api监控 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | passive\_defense | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

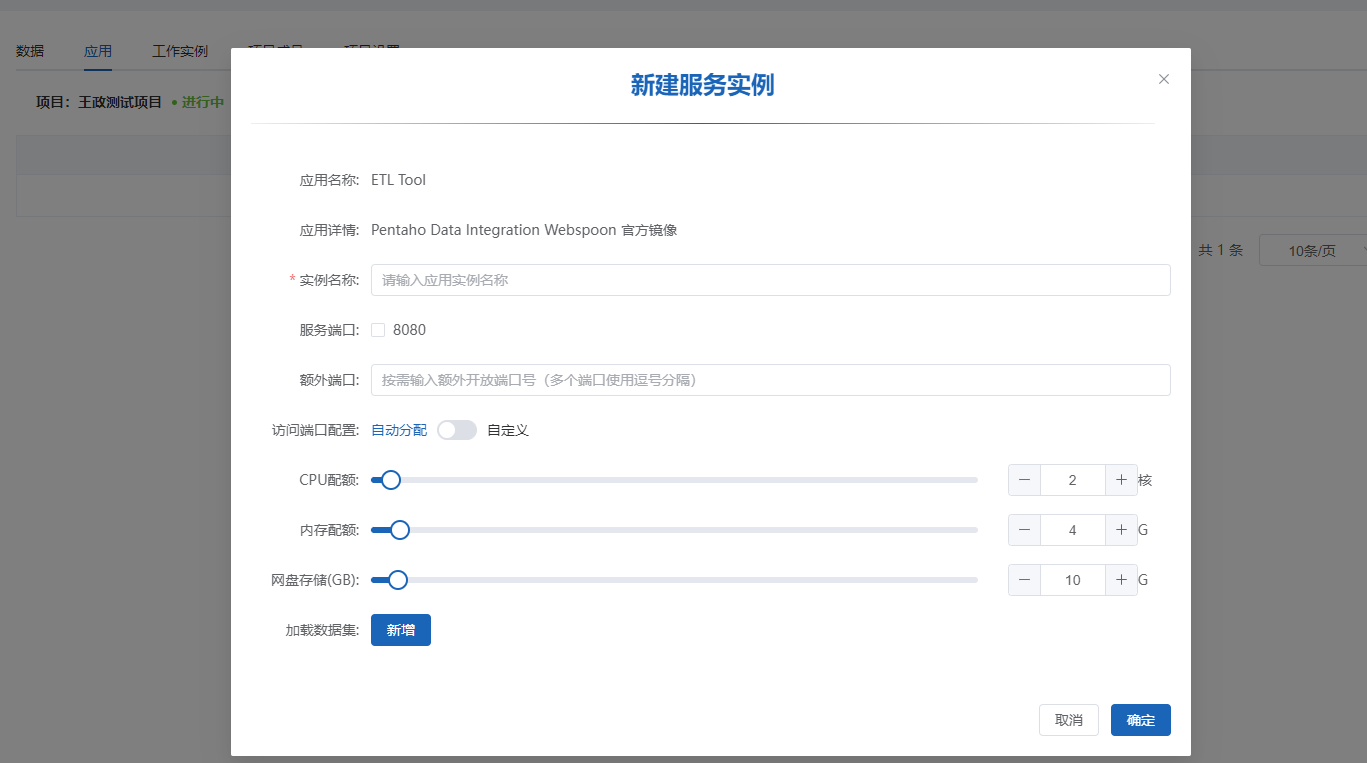
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**passive\_defense **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取被动防御策略  select name, project\_id, instance\_id, app\_id, tls\_enable, sample\_input, sample\_output, api\_monitor from passive\_defense limit ? offset ?  【SQL02】根据ID获取被动防御策略  select name, project\_id, instance\_id, app\_id, tls\_enable, sample\_input, sample\_output, api\_monitor from passive\_defense where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**passive\_defense **>**  **操作：<新增>** |
| SQL语句：  【SQL01】新增主动防御策略  insert into passive\_defense (name, project\_id, instance\_id, app\_id, tls\_enable, sample\_input, sample\_output, api\_monitor) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**passive\_defense **>**  **操作：<删除>** |
| SQL语句：  【SQL01】删除主动防御策略  delete from passive\_defense where id=? |
| **程序流程中涉及的数据库表：<表名：**passive\_defense **>**  **操作：<更新>** |
| SQL语句：  【SQL01】更新主动防御策略  update passive\_defense set name=?, project\_id=?, instance\_id=?, app\_id=?, tls\_enable=?, sample\_input=?, sample\_output=?, api\_monitor=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | projectId | 项目ID |  | | instanceId | 实例ID |  | | appId | 应用ID |  | | tlsEnable | 是否启用https |  | | sampleInput | 样例输入 |  | | sampleOutput | 样例输出 |  | | apiMonitor | 是否启用api监控 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动应用被动安全防御策略。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动防御策略

切换到工作实例标签页，可以看到刚刚创建的实例，被动防御策略在实例启动后会自动启动。样例输入自动导入到容器，检测实例输出与样例输出格式是否一致，检测请求api是否符合规范，未经验证的请求和未识别的返回字段会被拦截并触发报警。系统调用【7.5.3.2.1.1.1.2.1-分页获取被动防御策略列表】执行【SQL01】分页获取被动防御策略列表。调用【7.5.3.2.1.1.1.2.3-添加被动防御策略】执行【SQL03】添加被动防御策略。调用【7.5.3.2.1.1.1.2.4-删除被动防御策略】执行【SQL04】删除被动防御策略。调用【7.5.3.2.1.1.1.2.5-更新被动防御策略】执行【SQL05】更新被动防御策略。

#### 动态安全运行评价

动态安全运行评价根据规则及预设分值对容器内可疑的数据操作及传输行为提供从告警到拦截的多种操作。主要流程如下：



##### 类设计

###### 控制类

OperationEvaluationController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | OperationEvaluationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 运行安全评价处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/OperationEvaluationController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<OperationEvaluationDTO > | 分页获取安全运行评价列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<OperationEvaluationDTO > | 根据ID获取安全运行评价 |
| save | dto | OperationEvaluationDTO | ■公有 □私有 □保护 | Result | 添加安全运行评价 |
| update | dto | OperationEvaluationDTO | ■公有 □私有 □保护 | Result | 更新安全运行评价 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除安全运行评价 |

算法描述

分页获取安全运行评价列表

接口描述：分页获取安全运行评价列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/operationevaluation/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | ActiveDefenseDTO | 是 | 动态安全运行 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取安全运行评价

接口描述：根据ID获取安全运行评价。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/operationevaluation | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加安全运行评价

接口描述：添加安全运行评价。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/operationevaluation | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | OperationEvaluationDTO | 是 | 安全运行评价信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除安全运行评价

接口描述：删除安全运行评价。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/operationevaluation | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新安全运行评价

接口描述：更新安全运行评价。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/ | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | OperationEvaluationDTO | 是 | 安全运行评价信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

OperationEvaluationEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | OperationEvaluationEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 被动防御实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/OperationEvaluationEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| score | Interger | | □公有 ■私有 □保护 | | |  | | 当前分数 |
| rule | String | | □公有 ■私有 □保护 | | |  | | 扣分规则 |
| dataFilter | Boolean | | □公有 ■私有 □保护 | | |  | | 是否开启数据过滤 |
| killApp | Boolean | | □公有 ■私有 □保护 | | |  | | 是否停止应用程序 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | operation\_evaluation | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

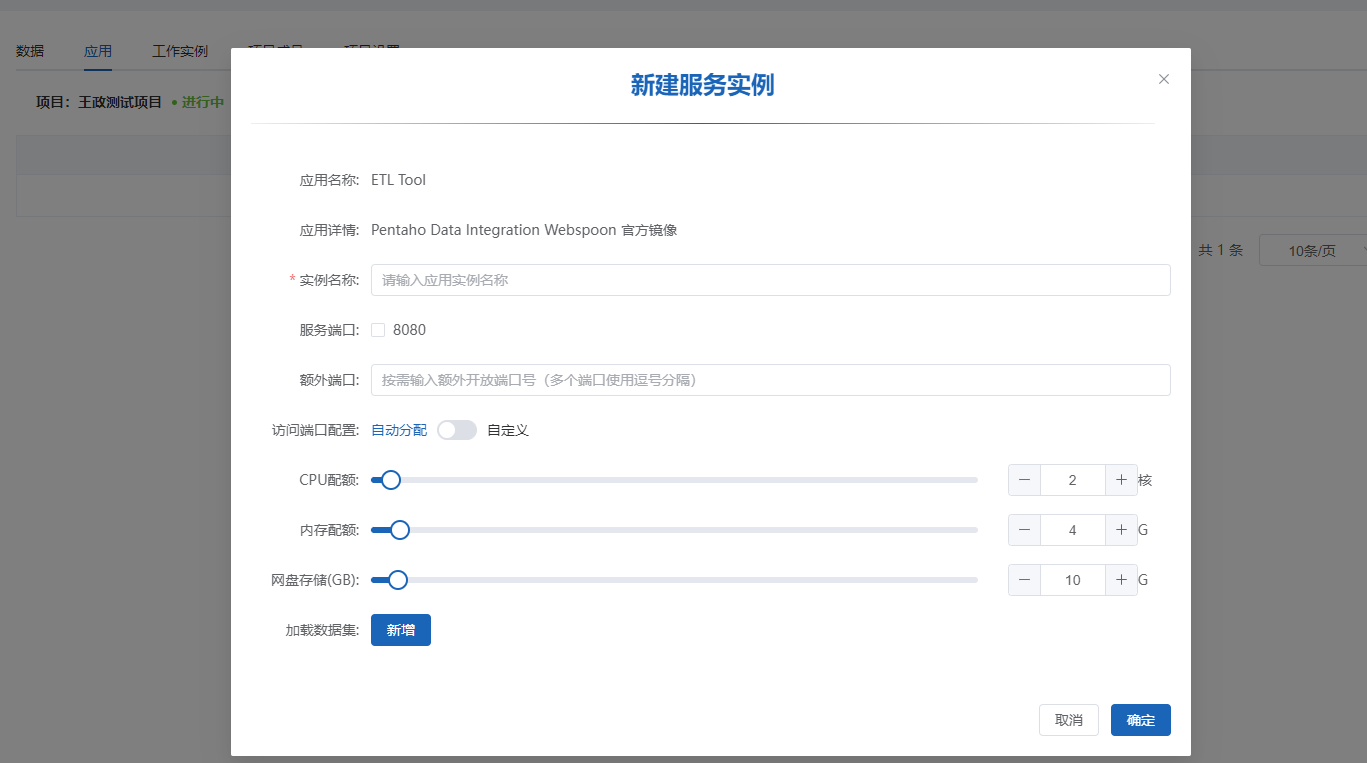
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取安全运行评价  select name, project\_id, instance\_id, app\_id, score, rule, data\_filter, kill\_app from operation\_evaluation ? offset ?  【SQL02】根据ID获取安全运行评价  select name, project\_id, instance\_id, app\_id, score, rule, data\_filter, kill\_app from operation\_evaluation where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增动态安全评价  insert into operation\_evaluation (name, project\_id, instance\_id, app\_id, score, rule, data\_filter, kill\_app) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除动态安全评价  delete from operation\_evaluation where id=? |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新动态安全评价  update operation\_evaluation set name=?, project\_id=?, instance\_id=?, app\_id=?, score=?, rule=?, data\_filter=?, kill\_app=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | projectId | 项目ID |  | | instanceId | 实例ID |  | | appId | 应用ID |  | | score | 当前分数 |  | | rule | 扣分规则 |  | | dataFilter | 是否开启数据过滤 |  | | killApp | 是否停止应用程序 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动应用动态安全运行评价。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动动态安全运行评价

切换到工作实例标签页，可以看到刚刚创建的实例，安全运行评价在实例启动后会自动启动。动态安全运行评价子系统从后台获取平台预设的安全违规程序行为定义及其对应需要扣减的分值。为其他容器内的安全防御及安全监控子系统上报的违规行为进行综合打分并计算扣分总额，当达到规则定义的分值时，控制安全防御子过程对违规应用程序进行处理，包括滤除违规数据，阻断通信直至停止用户应用程序等操作。将违规处理情况发送给系统后台，并触发安全告警。系统调用【7.5.3.3.1.1.1.2.1-分页获取安全运行评价列表】执行【SQL01】分页获取安全运行评价列表。调用【7.5.3.3.1.1.1.2.3-添加安全运行评价】执行【SQL03】添加安全运行评价。调用【7.5.3.3.1.1.1.2.4-删除安全运行评价】执行【SQL04】删除安全运行评价。调用【7.5.3.3.1.1.1.2.5-更新安全运行评价】执行【SQL05】删除安全运行评价。

#### 数据安全传输

数据安全传输为加密数据进出安全沙箱容器提供支持。主要流程如下：



##### 类设计

###### 控制类

DataTransmissionController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | DataTransmissionController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 数据安全传输处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/DataTransmissionController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<DataTransmissionDTO > | 分页获取数据安全传输信息列表 |
| get | id | Long | ■公有 □私有 □保护 | Result< DataTransmissionDTO > | 根据ID获取数据安全传输信息 |
| save | dto | DataTransmissionDTO | ■公有 □私有 □保护 | Result | 添加数据安全传输信息 |
| update | dto | DataTransmissionDTO | ■公有 □私有 □保护 | Result | 更新数据安全传输信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除数据安全传输信息 |

算法描述

分页获取数据安全传输信息列表

接口描述：分页获取数据安全传输信息列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/datatransmission/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| DataTransmission | Object | 是 | 数据安全传输 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取数据安全传输信息

接口描述：根据ID获取数据安全传输信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/ datatransmission | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加数据安全传输信息

接口描述：添加数据安全传输信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/ datatransmission | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | DataTransmission DTO | 是 | 数据安全传输信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除数据安全传输信息

接口描述：删除数据安全传输信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/datatransmission | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新数据安全传输信息

接口描述：更新数据安全传输信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/datatransmission | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | DataTransmissionDTO | 是 | 数据安全传输信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

DataTransmissionEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | DataTransmissionEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 数据传输实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/ DataTransmissionEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| encrypted | Boolean | | □公有 ■私有 □保护 | | |  | | 是否加密 |
| datasetKey | String | | □公有 ■私有 □保护 | | |  | | 数据集密钥 |
| projectKey | String | | □公有 ■私有 □保护 | | |  | | 项目密钥 |
| apiPath | Boolean | | □公有 ■私有 □保护 | | |  | | api地址 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | data\_transmission | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

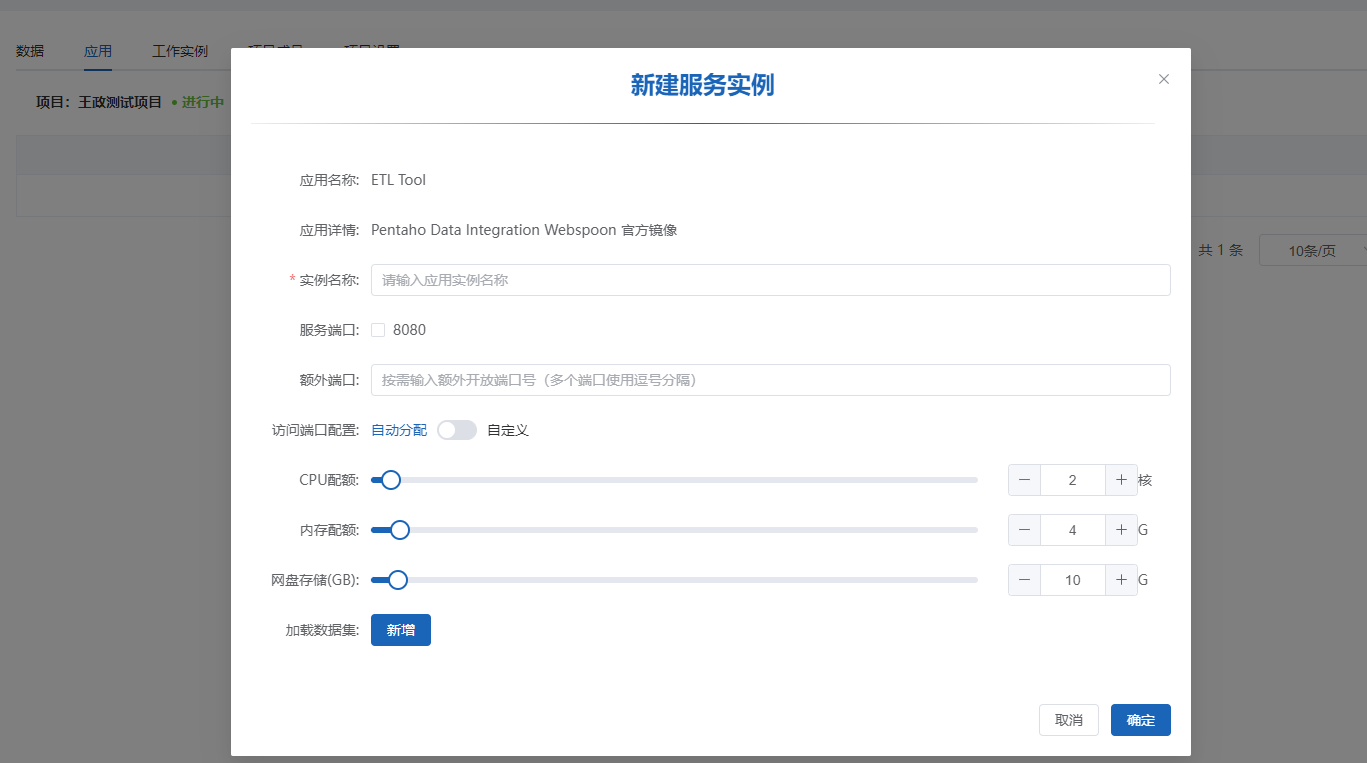
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取安全运行评价  select name, project\_id, instance\_id, app\_id, encrypted, dataset\_key, project\_key, api\_path from data\_transmission limit ? offset ?  【SQL02】根据ID获取安全运行评价  select name, project\_id, instance\_id, app\_id, encrypted, dataset\_key, project\_key, api\_path from data\_transmission where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**operation\_evaluation **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增动态安全评价  insert into operation\_evaluation (name, project\_id, instance\_id, app\_id, encrypted, dataset\_key, project\_key, api\_path) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**data\_transmission **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除动态安全评价  delete from data\_transmission where id=? |
| **程序流程中涉及的数据库表：<表名：**data\_transmission **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新动态安全评价  update data\_transmission set name=?, project\_id=?, instance\_id=?, app\_id=?, encrypted=?, dataset\_key=?, project\_key=?, api\_path=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | project\_id | 项目ID |  | | instance\_id | 实例ID |  | | app\_id | 应用ID |  | | encrypted | 是否加密 |  | | dataset\_key | 数据集密钥 |  | | project\_key | 项目密钥 |  | | api\_path | api地址 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动启动数据安全传输功能。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动数据安全传输功能

切换到工作实例标签页，可以看到刚刚创建的实例，数据安全传输系统在实例启动后会自动启动。数据安全传输子系统从后台获取应用内需要使用的授权加密数据集存储信息。调用加密文件存储系统提供的API经由加密传输信道获取数据集文件。通过密匙管理系统获取数据集文件密匙解密文件并保存到容器内的持久存储空间\data\1目录下。多个数据集的情况自动将每个数据集内个文件放入\data下对应自增数字的目录中。通过密匙管理系统获取数据集文件密匙解密文件并保存到容器内的持久存储空间\data\1目录下。多个数据集的情况自动将每个数据集内个文件放入\data下对应自增数字的目录中。在进行数据导出时，数据安全传输子系统按数据集敏感信息设定和输出数据格式说明对需要导出的数据进行内容扫描。如果发现违规数据（源数据集文件内容泄露到输出数据文件中）时查询动态安全运行评价子系统中断数据导出的请求并触发告警。当输出文件的扫描结果正常，满足安全要求的情况下，通过密匙管理系统获取项目空间文件密匙加密数据文件，并调用加密文件存储系统提供的API经由加密传输信道将加密数据文件保存到项目存储空间的指定目录（workspace目录下）。系统调用【7.5.3.4.1.1.1.2.1-分页获取数据安全传输信息列表】执行【SQL01】分页获取数据安全传输信息。调用【7.5.3.4.1.1.1.2.3-添加数据安全传输信息】执行【SQL03】添加数据安全传输信息。调用【7.5.3.4.1.1.1.2.4-删除数据安全传输信息】执行【SQL04】删除数据安全传输信息。调用【7.5.3.4.1.1.1.2.5-更新数据安全传输信息】执行【SQL05】更新数据安全传输信息。

#### 沙箱内安全监测

沙箱内安全监测监测沙箱授权使用数据集文件的明文内容的存储和使用。主要流程如下：



##### 类设计

###### 控制类

MonitorController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | MonitorController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 沙箱内安全监测处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/MonitorController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<MonitorDTO > | 分页获取安全监测列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<MonitorDTO > | 根据ID获取安全检测信息 |
| save | dto | MonitorDTO | ■公有 □私有 □保护 | Result | 添加安全监测信息 |
| update | dto | MonitorDTO | ■公有 □私有 □保护 | Result | 更新安全监测信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除安全监测信息 |

算法描述

分页获取安全监测信息列表

接口描述：分页获取安全监测信息列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/monitor/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| Monitor | DataTransmission | 是 | 沙箱安全 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取安全监测信息

接口描述：根据ID获取安全监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/monitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加安全监测信息

接口描述：添加安全监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/monitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | MonitorDTO | 是 | 安全监测信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除安全监测信息

接口描述：删除安全监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/monitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新安全监测信息

接口描述：更新安全监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/monitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | MonitorDTO | 是 | 安全监测信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

MonitorEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | MonitorEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 安全监测实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/ MonitorEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| cycle | Integer | | □公有 ■私有 □保护 | | |  | | 监控周期 |
| strategy | String | | □公有 ■私有 □保护 | | |  | | 探查策略 |
| childProcess | Boolean | | □公有 ■私有 □保护 | | |  | | 是否监控子进程 |
| infoDic | String | | □公有 ■私有 □保护 | | |  | | 敏感信息字典 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | monitor | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

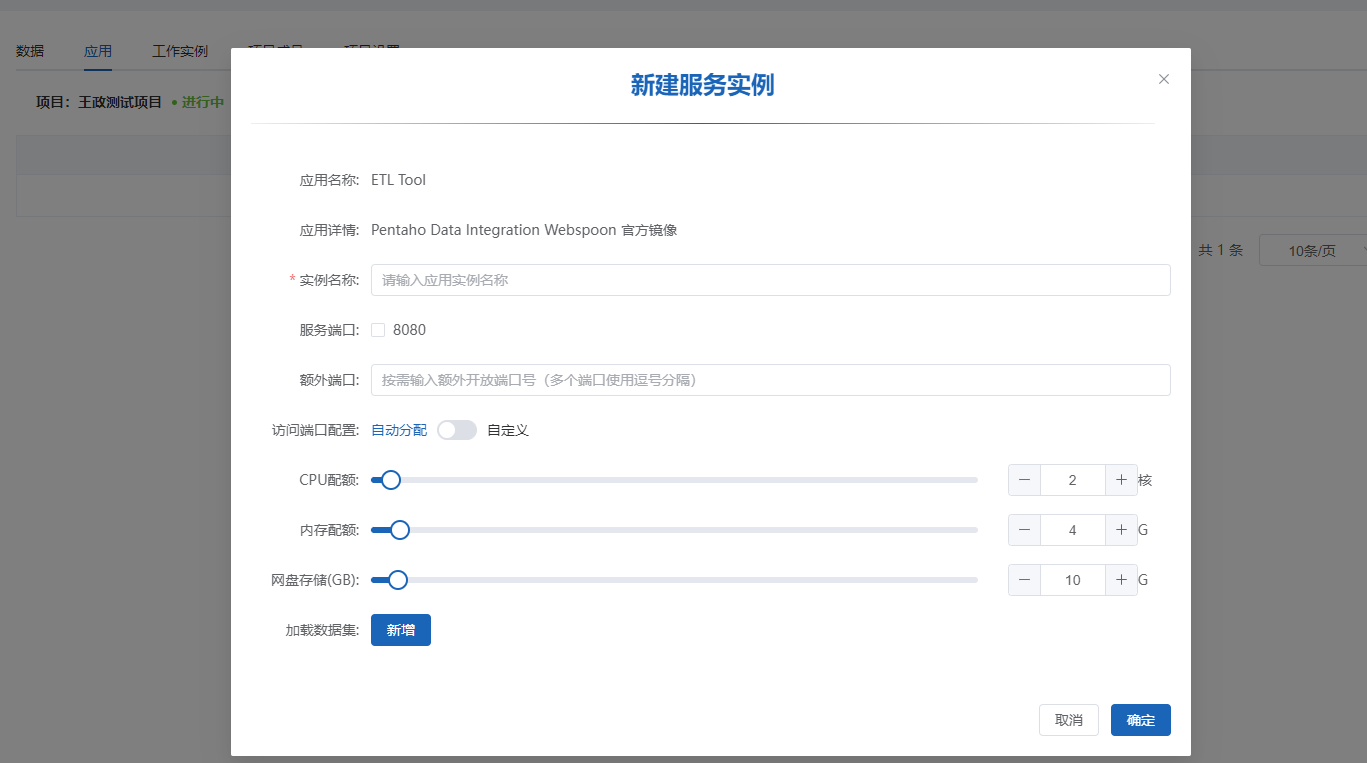
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**monitor **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取安全运行评价  select name, project\_id, instance\_id, app\_id, cycle, strategy, child\_process, info\_dic from monitor limit ? offset ?  【SQL02】根据ID获取安全运行评价  select name, project\_id, instance\_id, app\_id, cycle, strategy, child\_process, info\_dic from monitor where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**monitor **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增动态安全评价  insert into monitor (name, project\_id, instance\_id, app\_id, cycle, strategy, child\_process, info\_dic) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**monitor **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除动态安全评价  delete from monitor where id=? |
| **程序流程中涉及的数据库表：<表名：monitor** **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新动态安全评价  update monitor set name=?, project\_id=?, instance\_id=?, app\_id=?, cycle=?, strategy=?, child\_process=?, info\_dic=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | project\_id | 项目ID |  | | instance\_id | 实例ID |  | | app\_id | 应用ID |  | | cycle | 监控周期 |  | | strategy | 探查策略 |  | | child\_process | 是否监控子进程 |  | | info\_dic | 敏感信息字典 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动启动沙箱内安全监测。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动数据安全传输功能

切换到工作实例标签页，可以看到刚刚创建的实例，安全监测系统在实例启动后会自动启动。沙箱内安全监测子系统根据输入数据集敏感信息定义并结合数据集大小制定探查策略（全量或采样），按数据字段读取并索引敏感信息，构建敏感信息字典。沙箱内安全监测子系统监控沙箱的用户进程及其子进程的启停过程，对于被监控进程的开关行为及开关频度进行判断，监控沙箱内的程序目录、临时目录，对新生成文件及文件内容变化进行追踪扫描。将扫描结果在敏感信息字典中进行交叉比较查询，如果发现敏感信息在沙箱内发生存盘等操作，查询动态安全运行评价子系统并在需要时发送低风险安全警告。定期对容器内进行例行安全检查，检查活跃连接数量及变化频度等。系统调用【7.5.3.5.1.1.1.2.1-分页获取安全监测信息列表】执行【SQL01】分页获取安全监测信息列表。调用【7.5.3.5.1.1.1.2.3-添加安全监测信息】执行【SQL03】添加安全监测信息。调用【7.5.3.5.1.1.1.2.4-删除安全监测信息】执行【SQL04】删除安全监测信息。调用【7.5.3.5.1.1.1.2.5-更新安全监测信息】执行【SQL05】更新安全监测信息。

#### 容器启停、运行监控鉴别

容器启停、运行监控鉴别对用户应用容器实例的启动、停止行为进行监测，对于不是用户主动操作的容器启停事件进行告警，防止运维人员直接启动、停止用户应用容器，尝试获取容器内的数据。主要流程如下：

****

##### 类设计

###### 控制类

OperationMonitorController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | OperationMonitorController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/OperationMonitorController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<OperationMonitorDTO > | 分页获取运行监控列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<OperationMonitorDTO > | 根据ID获取运行监控信息 |
| save | dto | OperationMonitorDTO | ■公有 □私有 □保护 | Result | 添加运行监控信息 |
| update | dto | OperationMonitorDTO | ■公有 □私有 □保护 | Result | 更新运行监控信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除运行监控信息 |

算法描述

分页获取运行监控列表

接口描述：分页获取运行监控列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/operationmonitor/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | 容器监控 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取运行监控信息

接口描述：根据ID获取运行监控信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/operationmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加运行监控信息

接口描述：添加运行监控。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/operationmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | OperationMonitorDTO | 是 | 运行监控信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除运行监控信息

接口描述：删除运行监控信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/operationmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新运行监控信息

接口描述：更新运行监控信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/operationmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | OperationMonitorDTO | 是 | 运行监控信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

OperationMonitorEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | OperationMonitorEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 安全监测实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/OperationMonitorEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| startMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 容器启动监控 |
| stopMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 容器停止监控 |
| volumeMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 挂卷监控 |
| match | Boolean | | □公有 ■私有 □保护 | | |  | | 是否匹配 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | operation\_monitor | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

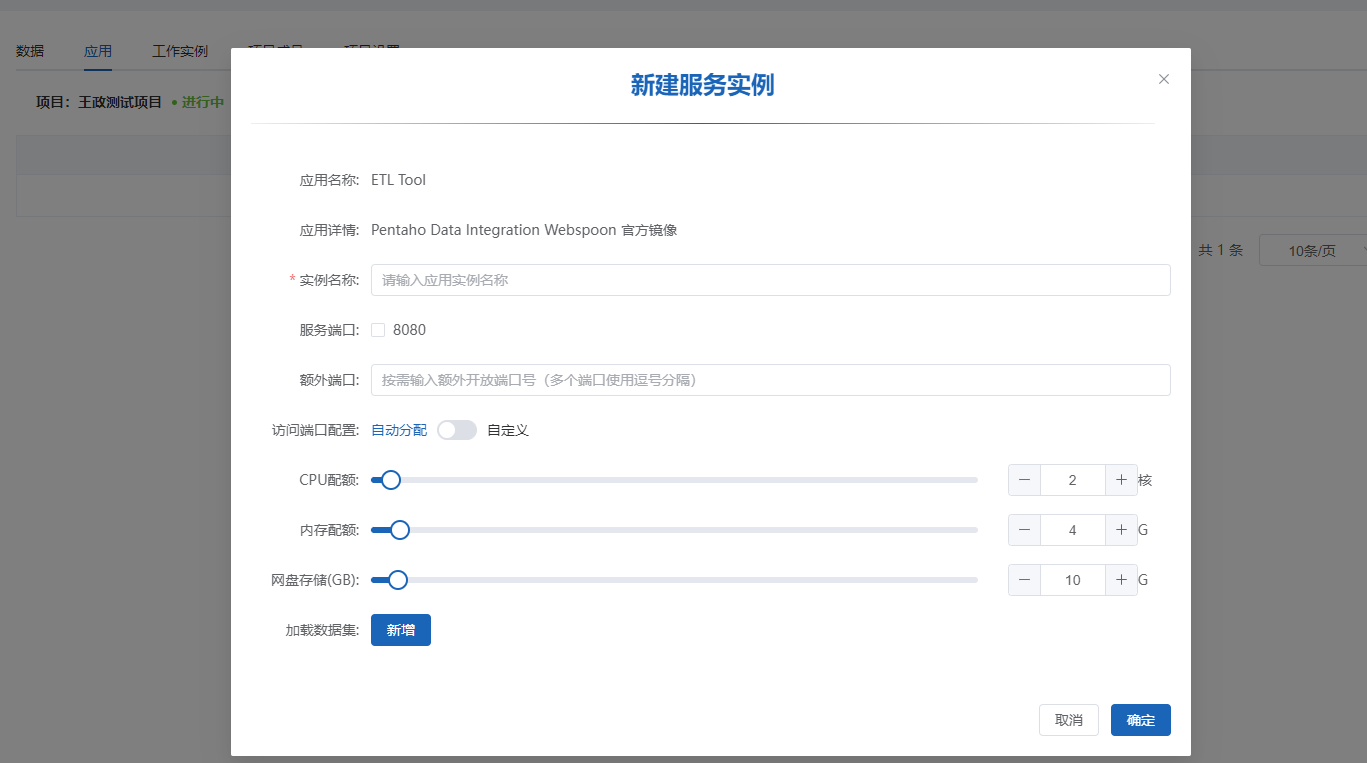
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**operation\_monitor **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取安全运行评价  select name, project\_id, instance\_id, app\_id, start\_monitor, stop\_monitor, volume\_monitor, match from operation\_monitor limit ? offset ?  【SQL02】根据ID获取安全运行评价  select name, project\_id, instance\_id, app\_id, start\_monitor, stop\_monitor, volume\_monitor, match from operation\_monitor where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**operation\_monitor **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增动态安全评价  insert into operation\_monitor (name, project\_id, instance\_id, app\_id, start\_monitor, stop\_monitor, volume\_monitor, match) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**operation\_monitor **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除动态安全评价  delete from operation\_monitor where id=? |
| **程序流程中涉及的数据库表：<表名：**operation\_monitor **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新动态安全评价  update operation\_monitor set name=?, project\_id=?, instance\_id=?, app\_id=?, start\_monitor=?, stop\_monitor=?, volume\_monitor =?, match=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | project\_id | 项目ID |  | | instance\_id | 实例ID |  | | app\_id | 应用ID |  | | start\_monitor | 容器启动监控 |  | | stop\_monitor | 容器停止监控 |  | | volume\_monitor | 挂卷监控 |  | | match | 是否匹配 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动启动容器启停、运行监控。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动运行监控功能

切换到工作实例标签页，可以看到刚刚创建的实例，运行监控系统在实例启动后会自动启动。系统启动后，后台进程启动容器监控子系统，通过K8S提供的标准API对注册容器启停、挂卷等操作的回调事件。用户在项目详情页的工作实例标签页通过按钮和菜单操作容器实例启动、暂停、恢复及删除。这些操作都会被记录的系统后台。容器监控子系统在接收到K8S发送的容器生命周期变化的事件时与用户操作行为记录进行交叉对比，包括实例名称，操作，时间等信息，当无法匹配正常用户操作行为时对可疑的容器实例操作行为进行记录并发送安全告警。系统调用【7.5.3.6.1.1.1.2.1-分页获取运行监控列表】执行【SQL01】分页获取运行监控列表。调用【7.5.3.6.1.1.1.2.3-添加运行监控信息】执行【SQL03】添加运行监控信息。调用【7.5.3.6.1.1.1.2.4-删除运行监控信息】执行【SQL04】删除运行监控信息。调用【7.5.3.6.1.1.1.2.5-更新运行监控信息】执行【SQL05】更新运行监控信息。

#### 系统通信安全监测

系统通信安全监测对系统模块间通信监测及异常流量告警。主要流程如下：



##### 类设计

###### 控制类

CommMonitorController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | CommMonitorController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/CommMonitorController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<CommMonitorDTO > | 分页获取通信监测列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<CommMonitorDTO > | 根据ID获取通信监测信息 |
| save | dto | CommMonitorDTO | ■公有 □私有 □保护 | Result | 添加通信监测信息 |
| update | dto | CommMonitorDTO | ■公有 □私有 □保护 | Result | 更新通信监测信息 |
| delete | ids | Long | ■公有 □私有 □保护 | Result | 删除通信监测信息 |

算法描述

分页获取通信监测列表

接口描述：分页获取通信监测列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/commmonitor/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | 系统通信 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取通信监测信息

接口描述：根据ID获取通信监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/commmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加通信监测信息

接口描述：添加通信监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/commmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | CommMonitorDTO | 是 | 通信监测信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除通信监测信息

接口描述：删除通信监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/commmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新通信监测信息

接口描述：更新通信监测信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/operationmonitor | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | CommMonitorDTO | 是 | 通信监测信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

CommMonitorEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | CommMonitorEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 安全监测实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/CommMonitorEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | String | | □公有 ■私有 □保护 | | |  | | 防御策略名称 |
| projectId | Long | | □公有 ■私有 □保护 | | |  | | 项目ID |
| instanceId | Long | | □公有 ■私有 □保护 | | |  | | 实例ID |
| appId | Long | | □公有 ■私有 □保护 | | |  | | 应用ID |
| fronendMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 前端监控 |
| backendMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 后端监控 |
| keyMonitor | Boolean | | □公有 ■私有 □保护 | | |  | | 密钥监控 |
| cache | Boolean | | □公有 ■私有 □保护 | | |  | | 缓存告警 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | comm\_monitor | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

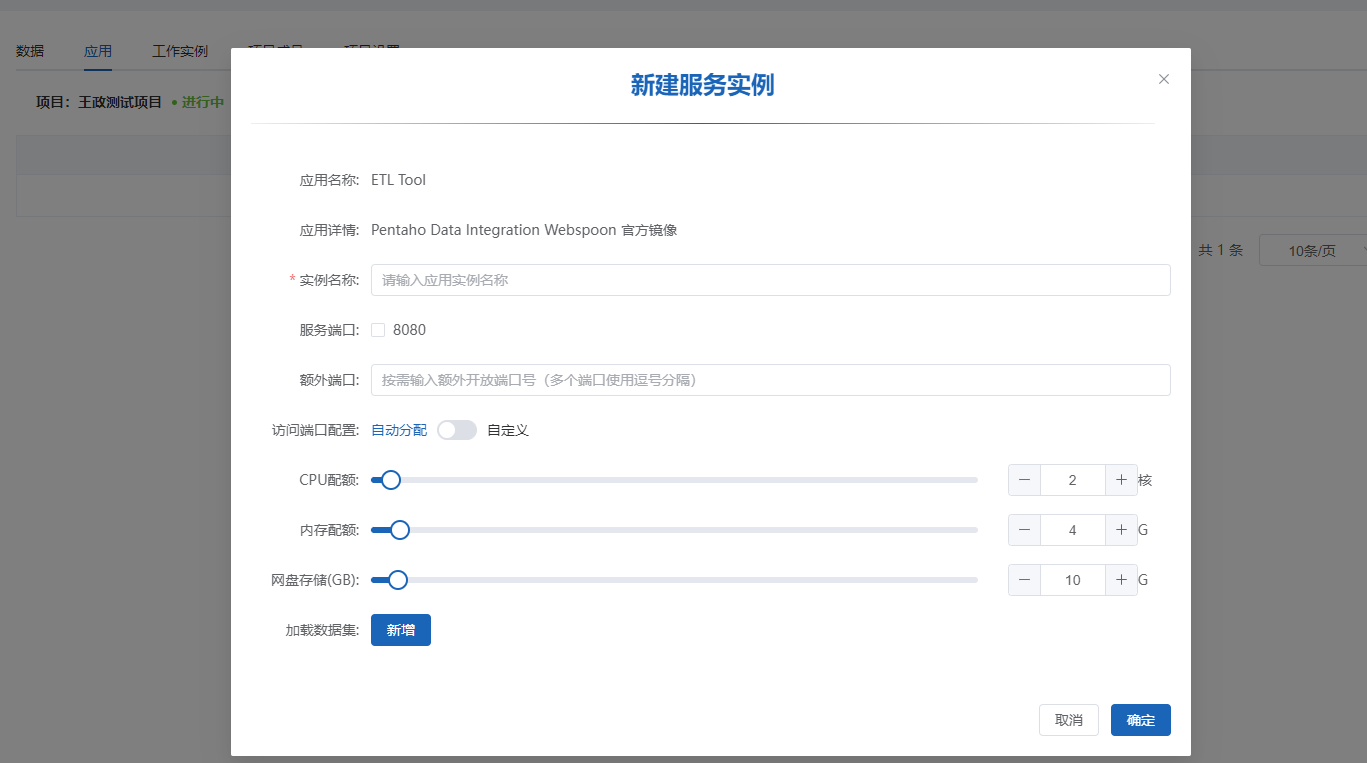
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**comm\_monitor **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取安全运行评价  select name, project\_id, instance\_id, app\_id, frontend\_monitor, backend\_monitor, key\_monitor, cache from comm\_monitor limit ? offset ?  【SQL02】根据ID获取安全运行评价  select name, project\_id, instance\_id, app\_id, frontend\_monitor, backend\_monitor, key\_monitor, cache from comm\_monitor where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**comm\_monitor **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增动态安全评价  insert into comm\_monitor (name, project\_id, instance\_id, app\_id, frontend\_monitor, backend\_monitor, key\_monitor, cache) values (?, ?, ?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**comm\_monitor **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除动态安全评价  delete from comm\_monitor where id=? |
| **程序流程中涉及的数据库表：<表名：**comm\_monitor **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新动态安全评价  update comm\_monitor set name=?, project\_id=?, instance\_id=?, app\_id=?, frontend\_monitor =?, backend\_monitor=?, key\_monitor =?, cache=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | name | 防御策略名称 |  | | project\_id | 项目ID |  | | instance\_id | 实例ID |  | | app\_id | 应用ID |  | | frontend\_monitor | 前端监控 |  | | backend\_monitor | 后端监控 |  | | key\_monitor | 密钥监控 |  | | cache | 缓存告警 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户进入项目空间创建实例，实例会自动启动容器启停、运行监控。

###### 处理流程



1.启动实例

用户点击项目进入项目空间，点击应用标签页，选择一个应用点击启动，输入实例参数后点击确定。

2.启动运行监控功能

切换到工作实例标签页，可以看到刚刚创建的实例，通信监测系统在实例启动后会自动启动。通信安全监测子系统对前台网页通信，后台API接口通信，安全容器内的守护主程序的通信，加密文件存储系统API通信，密匙管理系统API的通信都通过进行统计与识别，对于异常流量进行记录并触发告警。通信安全监测子系统定期轮训各个安全沙箱内的主安全守护进场状态，发现多次通信失败后进行告警并通知容器平台重新启动相关容器。安全沙箱内主安全守护进程平台定期轮训平台状态在平台不能正常通信时根据规则配置决定自行停止主进程，等待容器管理平台自动重启实例或缓冲安全告警并不断尝试恢复与平台的的通信，上报缓冲的告警信息。系统调用【7.5.3.6.1.1.1.2.1-分页获取运行监控列表】执行【SQL01】分页获取运行监控列表。调用【7.5.3.6.1.1.1.2.3-添加运行监控信息】执行【SQL03】添加运行监控信息。调用【7.5.3.6.1.1.1.2.4-删除运行监控信息】执行【SQL04】删除运行监控信息。调用【7.5.3.6.1.1.1.2.5-更新运行监控信息】执行【SQL05】更新运行监控信息。

### 数据安全服务

#### 密钥管理

密钥管理提供全平台范围的密匙和证书的管理。主要流程如下：



##### 类设计

###### 控制类

RsdpKmsController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpKmsController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 密钥处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpkms/controller/RsdpKmsController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<RsdpKmsDTO > | 分页获取密钥列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<RsdpKmsDTO > | 根据ID获取密钥 |
| save | dto | RsdpKmsDTO | ■公有 □私有 □保护 | Result | 添加密钥 |
| update | dto | RsdpKmsDTO | ■公有 □私有 □保护 | Result | 更新密钥 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除密钥 |

算法描述

分页获取密钥列表

接口描述：分页获取密钥列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdpkms/rsdpkms/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | 密钥管理 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取密钥

接口描述：根据ID获取密钥。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdpkms/rsdpkms | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加密钥

接口描述：添加密钥。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /rsdpkms/rsdpkms | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpKmsDTO | 是 | 密钥信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除密钥

接口描述：删除密钥。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /rsdpkms/rsdpkms | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新密钥

接口描述：更新密钥。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /rsdpkms/rsdpkms | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpKmsDTO | 是 | 密钥信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpKmsEntity

摘要

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | | |
| 英文名称 | RsdpKmsEntity | | | | | | | |
| 父类 | BaseEntity | | | | | | | |
| 类职责简述 | 主动防御实体类 | | | | | | | |
| 持久化 | 是 | | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpkms/entity/RsdpKmsEntity.java | | | | | | | |
| 属性 | | | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | 数据库校验\* | | | 说明 |
| targetName | String | □公有 ■私有 □保护 | | |  | | | 目标名称 |
| category | String | □公有 ■私有 □保护 | | |  | | | 类别 |
| targetId | Long | □公有 ■私有 □保护 | | |  | | | 目标ID |
| ip | String | □公有 ■私有 □保护 | | |  | | | IP |
| origin | String | □公有 ■私有 □保护 | | |  | | | 源 |
| operation | String | □公有 ■私有 □保护 | | |  | | | 操作 |
| operator | String | □公有 ■私有 □保护 | | |  | | | 操作者 |
| description | String | □公有 ■私有 □保护 | | |  | | | 描述 |
| 方法 | | | | | | | | | |
| 方法名称 | 参数名称 | | 参数类型 | 可见性 | 返回值 | | 说明 | |
| 无 | 无 | | 无 | 无 | 无 | | 无 | |
| 对应数据库表 | | | | | | rsdp\_kms | | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_kms **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取密钥列表  select target\_name, category, target\_id, ip, origin, operation, operator, description from rsdp\_kms limit ? offset ?  【SQL02】根据ID获取密钥  select target\_name, category, target\_id, ip, origin, operation, operator, description from rsdp\_kms where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_kms **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增密钥  insert into rsdp\_kms (target\_name, category, target\_id, ip, origin, operation, operator, description) values (?, ?, ?, ?, ?, ?, ?, ?,) |
| **程序流程中涉及的数据库表：<表名：**rsdp\_kms **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除密钥  delete from rsdp\_kms where id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_kms **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新密钥  update rsdp\_kms set target\_name=?, category=?, target\_id=?, ip=?, origin=?, operation=?, operator=?, description=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | target\_name | 目标名称 |  | | category | 类别 |  | | target\_id | 目标ID |  | | ip | IP |  | | origin | 源 |  | | operation | 操作 |  | | operator | 操作者 |  | | description | 描述 |  | |
|  |

##### 总体设计方案

###### 设计方案

系统基础安全机制，随系统自动启动。

###### 处理流程

1. 系统启动时启动密匙管理系统子系统。

2. 密匙管理子系统提供高安全等级的密匙管理功能，用于存放加密文件存储系统的密匙和系统内加密通信使用的证书。

3. 密匙管理子系统通过密秘分享的方式开启，必须同时提供5个主密匙中的3个才能开启密匙管理系统。根据部署环境的安全等级要求可以人工输入启动密匙（系统无法自动启动，需要人工操作，但提供最高安全等级），可以可以自动预置启动密匙（系统可以自动启动，安全等级不如人工输入密匙）。

4. 密匙管理子系统提供通用的加密算法和国密算法。

5. 算法采用主流CPU的专用指令集进行优化，保证算法执行效率。

6. 出于安全考虑，密匙管理系统不提供配置界面，全部功能通过加密通信API方式提供。通过API可以完成密匙的生成与读取、证书的生成与存取。

7. 同样出于安全考虑，密匙管理系统作为加密文件存储系统的子系统提供，不直接对外提供密匙存取的API访问，但可以通过查询接口获取密匙读取的访问记录。

8. 系统调用【7.5.4.1.1.1.1.2.1-分页获取密钥列表】执行【SQL01】分页获取密钥列表。调用【7.5.4.1.1.1.1.2.3-添加密钥】执行【SQL03】添加密钥。调用【7.5.4.1.1.1.1.2.4-删除密钥】执行【SQL04】删除密钥。调用【7.5.4.1.1.1.1.2.5-更新密钥】执行【SQL05】更新密钥。

#### 加密文件存储

文件加密存储系统，每个数据集一个独立密码，每个项目存储空间一个独立密码。主要流程如下：



##### 类设计

###### 控制类

MinioController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | MinioController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 加密处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/minio/controller/MinioController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| endpoint |  |  | ■公有 □私有 □保护 | Result<String> | 获取minio endpoint |
| webhookHead | header | Map<String, Object> | ■公有 □私有 □保护 | Result<Object> | webhook HEAD接口 |
| webhook | body | String | ■公有 □私有 □保护 | Result<Object> | webhook POST接口 |
| projectWebhookHead | header | Map<String, Object> | ■公有 □私有 □保护 | Result<Object> | project webhook HEAD接口 |
| projectWebhook | body | String | ■公有 □私有 □保护 | Result<Object> | project webhook POST接口 |

算法描述

获取minio endpoint

接口描述：获取minio endpoint。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /minio/minio/endpoint | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | 加密文件传输 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | String | 是 | minio endpoint |

webhook HEAD接口

接口描述：webhook HEAD接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| Head | /minio/minio/webhook | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| header | Map<String, Object> | 是 | 头信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

webhook POST接口

接口描述：webhook POST接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /minio/minio/webhook | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| body | String | 是 | 请求体 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

Project webhook HEAD接口

接口描述：project webhook HEAD接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| Head | /minio/minio/projectWebhook | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| header | Map<String, Object> | 是 | 头部信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

Project webhook POST接口

接口描述：project webhook POST接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /minio/minio/projectWebhook | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| body | String | 是 | 请求体 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

MinioEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | MinioEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | |  | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/minio/entity/MinioEntity | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| objectName | String | | □公有 ■私有 □保护 | | |  | | 名称 |
| isDir | Boolean | | □公有 ■私有 □保护 | | |  | | 是否是文件夹 |
| size | Long | | □公有 ■私有 □保护 | | |  | | 大小 |
| lastModified | Date | | □公有 ■私有 □保护 | | |  | | 最后修改时间 |
| key | String | | □公有 ■私有 □保护 | | |  | | key |
| etag | Boolean | | □公有 ■私有 □保护 | | |  | | etag |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | minio | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**minio **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取minio信息  select object\_name, is\_dir, size, last\_modified, key, etag from minio limit ? offset ?  【SQL02】根据ID获取minio信息  select object\_name, is\_dir, size, last\_modified, key, etag from minio where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**minio **>**  **操作：<新增>** |
| SQL语句：  【SQL01】新增minio信息  insert into minio (object\_name, is\_dir, size, last\_modified, key, etag) values (?, ?, ?, ?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**minio **>**  **操作：<删除>** |
| SQL语句：  【SQL01】删除minio信息  delete from minio where id=? |
| **程序流程中涉及的数据库表：<表名：**minio **>**  **操作：<更新>** |
| SQL语句：  【SQL01】更新minio信息  update minio set object\_name=?, is\_dir=?, size=?, last\_modified=?, key=?, etag=? Where id=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | object\_name | 名称 |  | | is\_dir | 是否是文件夹 |  | | size | 大小 |  | | last\_modified | 最后修改时间 |  | | key | key |  | | etag | etag |  | |
|  |

##### 总体设计方案

###### 设计方案

系统基础安全机制，随系统自动启动。

###### 处理流程

1. 系统启动时启动加密文件存储系统子系统。

2. 加密文件存储子系统通过块存储方式保存文件内容。加密文件存储子系统与密匙管理子系统高度集成，对外提供存储单元创建、文件保存、读取、创建目录、检索文件及目录等API。API只能通过HTTPS的安全传输方式调用。

3. 创建存储单元时，加密文件存储子系统自动调用密匙管理系统的API创建一个新的密匙，并使用存储单元的标识符作为密匙标识。

4. 保存文件时，加密文件存储子系统通过存储单元标识符从密匙管理子系统获取之前创建的加密密匙，将文件内容按存储块划分后对每个存储块进行加密后保存。

5. 读取文件时，加密文件存储子系统通过存储单元标识符从密匙管理子系统获取之前创建的加密密匙，获取文件所属的全部数据块并使用密匙逐块解密后通过加密传输方式发送给接受端。

6. 由于所有存储数据块全部加密后保存，可以保证物理硬盘丢失或被复制的情况下，攻击者无法恢复原始明文数据。

7. 机构用户在网页上上传数据集文件或向项目空间上传文件时，系统之间从前端调用加密文件存储子系统提供的加密信道上的数据存取API，因此系统任何web服务器或代理服务器都不会接触到用户的明文数据。即数据一旦存入系统就已加密。

8. 文件加密存储系统提供加密传输API在保证数据完整性的前提下完成数据文件在系统内的传输。

9. 系统采用一事一密的方式存储用户数据。即每个数据集和项目网络空间都拥有独立的复杂密码。防止一个密码意外泄漏造成大量数据被盗取。

10. 系统调用【7.5.4.2.1.1.1.2.1-获取minio endpoint】执行【SQL01】获取minio endpoint。调用【7.5.4.2.1.1.1.2.2-webhook HEAD接口】执行【SQL02】接收webhook HEAD请求。调用【7.5.4.2.1.1.1.2.3-webhook POST接口】执行【SQL03】接收webhook POST请求。调用【7.5.4.2.1.1.1.2.4-Project webhook HEAD接口】执行【SQL04】接收project webhook HEAD请求。调用【7.5.4.2.1.1.1.2.5-Project webhook POST接口】执行【SQL05】接收project webhook POST请求。

#### 数据访问授权

数据访问授权提供对数据集和项目空间数据的访问授权检查。主要流程如下：



##### 类设计

###### 控制类

DatasetUseAuthorizeController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | DatasetUseAuthorizeController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/flow/authorize/controller/DatasetUseAuthorizeController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| start | dto | DatasetUseAuthorizeDTO | ■公有 □私有 □保护 | Result | 启动流程 |
| info | dto | DatasetUseAuthorizeDTO | ■公有 □私有 □保护 | Result<DatasetUseAuthorizeDTO> | 根据数据集ID获取授权信息 |
| get | id | Long | ■公有 □私有 □保护 | Result | 表单信息 |

算法描述

启动流程

接口描述：启动流程。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /flow/authorize/dataset-use/start | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| datasetId | Long | 是 | 数据集ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据数据集ID获取授权信息

接口描述：根据数据集ID获取授权信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /flow/authorize/dataset-use/info | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| datasetId | Long | 是 | 数据集ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID查询授权信息

接口描述：根据ID查询授权信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /flow/authorize/dataset-use/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

DatasetUseAuthorizeEntity

摘要

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | |
| 英文名称 | DatasetUseAuthorizeEntity | | | | | |
| 父类 | BaseEntity | | | | | |
| 类职责简述 | 主动防御实体类 | | | | | |
| 持久化 | 是 | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/flow/authorize/entity/DatasetUseAuthorizeEntity.java | | | | | |
| 属性 | | | | | | |
| 属性名称 | 类型 | 可见性 | | 数据库校验\* | | 说明 |
| datasetId | Long | □公有 ■私有 □保护 | |  | | 数据集ID |
| applicantDeptId | Long | □公有 ■私有 □保护 | |  | | 申请人机构ID |
| creditChangeId | Long | □公有 ■私有 □保护 | |  | | 积分消费记录ID |
| instanceId | String | □公有 ■私有 □保护 | |  | | 审核流程id |
| status | Integer | □公有 ■私有 □保护 | |  | | 申请状态 0：审核中 1：通过 2：拒绝 |
| datasetName | String | □公有 ■私有 □保护 | |  | | 数据集名称 |
| credit | Integer | □公有 ■私有 □保护 | |  | | 所需积分 |
| originDeptName | String | □公有 ■私有 □保护 | |  | | 数据源所属机构名称 |
| datasetReadability | Integer | □公有 ■私有 □保护 | |  | | 数据集授权类型 0：可用不可见 1：可用可见 |
| 方法 | | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | rsdp\_authorize\_dataset\_use | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_authorize\_dataset\_use **>**  **操作：<查询>** |
| SQL语句：  【SQL01】根据datasetId查询授权信息  select \* from rsdp\_authorize\_dataset\_use where applicant\_dept\_id = #{applicantDeptId} and dataset\_id = #{datasetId} order by create\_date desc limit 1  【SQL02】根据instanceId查询授权信息  select \* from rsdp\_authorize\_dataset\_use where instance\_id = #{instanceId} |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_authorize\_dataset\_use **>**  **操作：<更新>** |
| SQL语句：  【SQL03】更新实例  update rsdp\_authorize\_dataset\_use set instance\_id = #{instanceId} where id = #{id}  【SQL04】更新积分消费记录  update rsdp\_authorize\_dataset\_use set credit\_change\_id = #{creditChangeId} where id = #{id}  【SQL05】更新状态  update rsdp\_authorize\_dataset\_use set status = #{status} where id = #{id}  【SQL06】通过实例ID更新状态  update rsdp\_authorize\_dataset\_use set status = #{status} where instance\_id = #{instanceId} |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | applicantDeptId | 申请人机构ID |  | | creditChangeId | 积分消费记录ID |  | | instanceId | 审核流程id |  | | status | 申请状态 0：审核中 1：通过 2：拒绝 |  | |
|  |

##### 总体设计方案

###### 设计方案

用户使用非公开数据集时需先获取授权，数据集授权之后才能使用，未授权的数据集不允许使用。

###### 处理流程



1. 系统提供不同的数据集访问类型，机构用户在发布数据集时可以选择“公开”，“需要授权”，“机构私有”3种类型。

2. “公开”型的数据集，可以不经发布机构授权直接在应用实例中使用；“需要授权”的数据集，需要向发布机构申请使用，在通过发布机构具有“数据集授权审核主管”身份的用户的批准后才能在应用实例中使用；“机构私有”型的数据集只能被机构内部用户在应用实例中使用。

3. 系统为每个项目提供加密的网络存储空间，只有项目成员才可以使用项目空间存取文件。系统允许通过容器内的API将应用容器实例内的数据导出的项目存储空间，导出时需要进行数据安全检查防止源数据内容泄露。

4. 系统应用实例中使用数据集进行访问控制，只有生产型的应用实例才可以添加可用不可见型的数据集。系统在前端界面、后端API和容器启动获取数据集信息时都会进行数据访问检查，确保所以获取数据集的入口都不能绕过数据访问授权的控制机制。

5. 系统调用【7.5.4.3.1.1.1.2.1-启动流程】执行【SQL01】启动流程。系统调用【7.5.4.3.1.1.1.2.2-根据数据集ID获取授权信息】执行【SQL02】根据数据集ID获取授权信息。系统调用【7.5.4.3.1.1.1.2.3-根据ID查询授权信息】执行【SQL03】根据ID查询授权信息。

#### 数据传输完整性校验

数据传输完整性校验通过证书和传输校验机制确保数据传输的完整性并防止被篡改。主要流程如下：



##### 类设计

###### 控制类

IntegrityCheckController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | IntegrityCheckController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/IntegrityCheckController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<IntegrityCheckDTO > | 分页获取完整性校验策略列表 |
| get | id | Long | ■公有 □私有 □保护 | Result<IntegrityCheckDTO > | 根据ID获取完整性校验策略 |
| save | dto | IntegrityCheckDTO | ■公有 □私有 □保护 | Result | 添加完整性校验策略 |
| update | dto | IntegrityCheckDTO | ■公有 □私有 □保护 | Result | 更新完整性校验策略 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除完整性校验策略 |

算法描述

分页获取完整性校验策略列表

接口描述：分页获取完整性校验策略列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/integritycheck/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | 数据传输校验 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

根据ID获取完整性校验策略

接口描述：根据ID获取完整性校验。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /security/integritycheck | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | Long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

添加完整性校验策略

接口描述：添加完整性校验策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /security/integritycheck | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | IntegrityCheckDTO | 是 | 完整性校验策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

删除完整性校验策略

接口描述：删除完整性校验策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /security/ integritycheck | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long | 是 | id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

更新完整性校验策略

接口描述：更新完整性校验策略。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /security/integritycheck | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | IntegrityCheckDTO | 是 | 完整性校验策略信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

IntegrityCheckEntity

摘要

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | | | | | |
| 英文名称 | IntegrityCheckEntity | | | | | | | | |
| 父类 | BaseEntity | | | | | | | | |
| 类职责简述 | 主动防御实体类 | | | | | | | | |
| 持久化 | 是 | | | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/entity/IntegrityCheckEntity.java | | | | | | | | |
| 属性 | | | | | | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | 数据库校验\* | | | | 说明 | |
| datasetId | Long | □公有 ■私有 □保护 | | |  | | | | 数据集ID | |
| detailId | Long | □公有 ■私有 □保护 | | |  | | | | 数据集详情ID | |
| md5 | String | □公有 ■私有 □保护 | | |  | | | | 文件md5 | |
| 方法 | | | | | | | | | | | | |
| 方法名称 | 参数名称 | | 参数类型 | 可见性 | | 返回值 | | 说明 | | | |
| 无 | 无 | | 无 | 无 | | 无 | | 无 | | | |
| 对应数据库表 | | | | | | | integrity\_check | | | | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**integrity\_check **>**  **操作：<查询>** |
| SQL语句：  【SQL01】分页获取完整性校验策略列表  select dataset\_id, detail\_id, md5 from integrity\_check limit ? offset ?  【SQL02】根据ID获取完整性校验策略  select dataset\_id, detail\_id, md5 from integrity\_check where id=? |
|  |
| **程序流程中涉及的数据库表：<表名：**integrity\_check **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增完整性校验策略  insert into integrity\_check (dataset\_id, detail\_id, md5) values (?, ?, ?) |
| **程序流程中涉及的数据库表：<表名：**integrity\_check **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除完整性校验策略  delete from integrity\_check where id=? |
| **程序流程中涉及的数据库表：<表名：**integrity\_check **>**  **操作：<更新>** |
| SQL语句：  【SQL05】更新完整性校验策略  update integrity\_check set dataset\_id=?, detail\_id=?, md5=? |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | datasetId | 数据集ID |  | | detailId | 数据集详情ID |  | | md5 | 文件md5 |  | |
|  |

##### 总体设计方案

###### 设计方案

通过文件加密系统API存取加密文件时，会触发数据完整性校验，当校验通过时，数据才被保存或读取，校验失败则会发送报警。

###### 处理流程



1. 加密文件存储子系统提供数据传输完整性校验以确保数据文件在各个系统内各个子系统间传递时是完整且未经篡改的。当调用API通过加密信道存取加密文件存储子系统内的文件时，系统提供额外的数据传输完整性签名校验。

2. 加密文件存储子系统在以下使用场景中提供传输完整性校验：

 机构用户在数据集详情页上操作数据集文件；

 机构用户在项目详情页的项目存储空间操作文件；

 沙箱容器内安全主守护主进程按要求向容器内传入授权使用的数据集；

 沙箱容器内安全主进程相应用户应用的请求从容器内将计算结果的数据文件导出到项目存储空间中；

3. 系统调用【7.5.4.4.1.1.1.2.1-分页获取完整性校验策略列表】执行【SQL01】分页获取完整性校验策略列表。调用【7.5.4.4.1.1.1.2.3-添加完整性校验策略】执行【SQL03】添加完整性校验策略。调用【7.5.4.4.1.1.1.2.4-删除完整性校验策略】执行【SQL04】删除完整性校验策略。调用【7.5.4.4.1.1.1.2.5-更新完整性校验策略】执行【SQL05】更新完整性校验策略。

#### 数据防拷贝告警

数据防拷贝告警通过监控加密文件系统及记录用户文件访问操作来识别未授权的可疑访问并发出报警信息主要流程如下：



##### 类设计

###### 控制类

RsdpOperationRecordController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpOperationRecordController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/security/controller/RsdpOperationRecordController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| createOperationRecord | body | Map<String, Object> | ■公有 □私有 □保护 | Result<Object> | 创建操作记录 |

算法描述

创建操作记录

接口描述：创建操作记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /rsdpoperationrecord/rsdpoperationrecord/createOperationRecord | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | operationRecord | 是 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpOperationRecordEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | RsdpOperationRecordEntity | | | | | | | |
| 父类 |  | | | | | | | |
| 类职责简述 | 操作记录实体类 | | | | | | | |
| 持久化 | 是 | | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdpoperationrecord/entity/RsdpOperationRecordEntity.java | | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | | 数据库校验\* | | 说明 | |
| id | Long | □公有 ■私有 □保护 | | | |  | | ID | |
| s3 | String | □公有 ■私有 □保护 | | | |  | | s3信息 | |
| target | String | □公有 ■私有 □保护 | | | |  | | 目标名称/ID | |
| type | String | □公有 ■私有 □保护 | | | |  | | 类型 | |
| event | String | □公有 ■私有 □保护 | | | |  | | 事件 | |
| tag | String | □公有 ■私有 □保护 | | | |  | | 标签 | |
| time | Date | □公有 ■私有 □保护 | | | |  | | 时间 | |
| flag | Integer | □公有 ■私有 □保护 | | | |  | | 标记 | |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | | 返回值 | | 说明 | | |
| 无 | 无 | 无 | 无 | | 无 | | 无 | | |
| 对应数据库表 | | | | rsdp\_operation\_record | | | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_operation\_record **>**  **操作：<新增>** |
| SQL语句：  【SQL01】新增操作记录  insert into rsdp\_operation\_record (id, s3, target, type, event, tag, time flag) values (?, ?, ?, ?, ?, ?, ?, ?, ?) |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | ID |  | | s3 | s3信息 |  | | target | 目标名称/ID |  | | type | 类型 |  | | event | 事件 |  | | tag | 标签 |  | | time | 时间 |  | | flag | 标记 |  | | id | ID |  | |
|  |

##### 总体设计方案

###### 设计方案

系统基础安全机制，随系统自动启动。会启动一个队列，用于对比用户操作和webhook操作。在正常情况下，这两种操作是一一对应的关系。若存在不对应的情况，这说明用户进行了某些非法操作，会触发报警。

###### 处理流程

1. 系统启动时启动数据防拷贝子系统。

2. 加密文件存储系统提供API，可以对文件创建、删除、读写等操作行为触发通知。

3. 机构用户点击“资产管理\数据集”菜单进入数据集列表页，从列表中点击数据集名称进入数据就详情页面。在数据集详情页面的数据标签页内进行数据集文件的上传、覆盖、新建目录、下载、重命名、删除等操作时系统会记录用户菜单操作行为包括：数据集、操作人、行为、时间等。

4. 机构用户点击“项目管理”菜单进入项目列表页，从列表中点击项目进程项目详情页面。在项目详情页面的数据标签页内进行项目文件的上传、覆盖、新建目录、下载、重命名、删除等操作时系统会记录用户菜单操作行为包括：项目、操作人、行为、时间等。

5. 系统接收到加密文件存储系统触发的通知时与用户在界面上的文件操作记录进行比对。当发现未记录的违规数据文件操作时向平台发送安全报警。

6. 系统调用【7.5.4.5.1.1.1.2.1-创建操作记录】执行【SQL01】创建操作记录。

### 安全审计

对平台内所有程序或个人访问加密密匙的行为提供全记录和查找服务，对可疑操作提供告警。



#### 密钥访问审计

##### 类设计

###### 控制类

RsdpAuditSecretRecordsController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAuditSecretRecordsController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 密钥审计控制类 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/controller/RsdpAuditSecretRecordsController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAuditSecretRecordsService | RsdpAuditSecretRecordsService | private | | | 无 |
| sysDictTypeService | SysDictTypeService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAuditSecretRecordsDTO>> | 分页获取应用审计信息 |
| get | id | Long | ■公有 □私有 □保护 | Result<RsdpAuditSecretRecordsDTO> | 根据ID获取应用审计信息 |
| save | dto | RsdpAuditSecretRecordsDTO | ■公有 □私有 □保护 | Result | 新建应用审计信息 |
| update | dto | RsdpAuditSecretRecordsDTO | ■公有 □私有 □保护 | Result | 更新应用审计信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除应用审计信息 |
| export | params  response | Map<String, Object>  HttpServletResponse | ■公有 □私有 □保护 | 无 | 导出应用审计信息 |

算法描述

分页获取密钥审计记录

接口描述：分页获取密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditsecretrecords/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | int | 是 | 当前页码，从1开始 |
| limit | int | 是 | 每页显示记录数 |
| orderField | String | 否 | 排序字段 |
| order | String | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 密钥审计分页记录 |

根据ID获取应用审计记录

接口描述：根据ID获取密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditsecretrecords/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 密钥审计记录 |

新建密钥审计记录

接口描述：新建密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | audit/rsdpauditsecretrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditSecretRecordsDTO | 是 | 密钥审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

更新密钥审计记录

接口描述：更新密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | audit/rsdpauditsecretrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditSecretRecordsDTO | 是 | 密钥审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

删除密钥审计记录

接口描述：删除密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | audit/rsdpauditsecretrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | ID列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

导出密钥审计记录

接口描述：导出密钥审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditsecretrecords/export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | [stream] | 是 | 文件流 |

###### 实体类

RsdpAuditSecretRecordsEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | RsdpAuditSecretRecordsEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 密钥审计记录表 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $com/capinfo/healthcare/mhsdip/ppc/modules/audit/entity/RsdpAuditSecretRecordsEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| secretId | long | | □公有 ■私有 □保护 | | |  | | 密钥id |
| secretName | string | | □公有 ■私有 □保护 | | |  | | 密钥名称 |
| actionType | int | | □公有 ■私有 □保护 | | |  | | 操作类型 0-创建密钥 1-发布密钥 2-删除密钥 3-授权成功 4-取消授权 5-项目引用 6-取消引用 |
| userId | long | | □公有 ■私有 □保护 | | |  | | 操作人员id |
| userName | string | | □公有 ■私有 □保护 | | |  | | 操作人员姓名 |
| info | string | | □公有 ■私有 □保护 | | |  | | 操作详情 |
| evidence | string | | □公有 ■私有 □保护 | | |  | | 存证数据 |
| evidenceSignature | string | | □公有 ■私有 □保护 | | |  | | 存证数据签名 |
| evidenceHash | string | | □公有 ■私有 □保护 | | |  | | 存证数据哈希 |
| transactionHash | string | | □公有 ■私有 □保护 | | |  | | 交易哈希 |
| blockNumber | string | | □公有 ■私有 □保护 | | |  | | 区块高度 |
| blockHash | string | | □公有 ■私有 □保护 | | |  | | 区块哈希 |
| isBlockchainRecord | boolean | | □公有 ■私有 □保护 | | | NN | | 是否是区块链存证数据 0-否 1-是 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_audit\_secret\_records | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_secret\_records **>**  **操作：<查询>** |
| SQL语句：  【SQL01】获取密钥审计列表  SELECT id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_secret\_records ORDER BY create\_date DESC LIMIT ?  【SQL02】根据ID获取密钥审计记录  SELECT id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_secret\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_secret\_records **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增密钥审计记录  INSERT into rsdp\_audit\_secret\_records (id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_secret\_records **>**  **操作：<删除>** |
| 【SQL04】删除密钥审计记录  DELETE from rsdp\_audit\_secret\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_secret\_records **>**  **操作：<更新>** |
| 【SQL05】更新密钥审计记录  UPDATE rsdp\_audit\_secret\_records SET secret\_id=?, evidence=?, evidence\_signature=?, evidence\_hash=?, transaction\_hash=?, block\_number=?, block\_hash=?, secret\_name=?, action\_type=?, user\_id=?, user\_name=?, info=?, is\_blockchain\_record=? WHERE id=? |

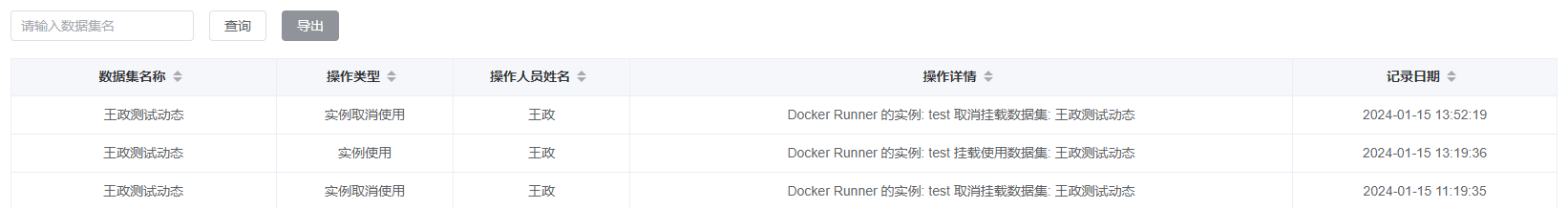
##### 总体设计方案

###### 设计方案

左侧选择审计报表菜单，点击密钥审计菜单，查看审计记录。

###### 处理流程

1、密钥审计记录



2、密钥审计表

调用【7.5.5.1.1.1.1.2.1. 分页获取密钥审计记录】执行【SQL01】，查询密钥审计信息，用户可以在此查看密钥审计记录。

3、密钥审计信息

调用【7.5.5.1.1.1.1.2.2. 根据ID获取应用审计记录】执行【SQL02】，根据ID查询密钥审计记录详情。

4、删除密钥审计信息

调用【7.5.5.1.1.1.1.2.5. 删除密钥审计记录】执行【SQL04】，根据ID删除指定的密钥审计记录。

#### 数据集审计

提供数据集创建、修改、发布、交易、使用的全流程记录和审计服务，主要流程如下：



##### 类设计

###### 控制类

RsdpAuditDatasetRecordsController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAuditDatasetRecordsController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 数据集审计控制类 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/controller/RsdpAuditDatasetRecordsController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAuditDatasetRecordsService | RsdpAuditDatasetRecordsService | private | | | 无 |
| sysDictTypeService | SysDictTypeService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAuditDatasetRecordsDTO>> | 分页获取数据集审计信息 |
| get | id | Long | ■公有 □私有 □保护 | Result<RsdpAuditDatasetRecordsDTO> | 根据ID获取数据集审计信息 |
| save | dto | RsdpAuditDatasetRecordsDTO | ■公有 □私有 □保护 | Result | 新建数据集审计信息 |
| update | dto | RsdpAuditDatasetRecordsDTO | ■公有 □私有 □保护 | Result | 更新数据集审计信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除数据集审计信息 |
| export | params  response | Map<String, Object>  HttpServletResponse | ■公有 □私有 □保护 | 无 | 导出数据集审计信息 |

算法描述

分页获取数据集审计记录

接口描述：分页获取数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditdatasetrecords/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | int | 是 | 当前页码，从1开始 |
| limit | int | 是 | 每页显示记录数 |
| orderField | String | 否 | 排序字段 |
| order | String | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 数据集审计分页记录 |

根据ID获取数据集审计记录

接口描述：根据ID获取数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditdatasetrecords/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 数据集审计记录 |

新建数据集审计记录

接口描述：新建数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | audit/rsdpauditdatasetrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditDatasetRecordsDTO | 是 | 数据集审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

更新数据集审计记录

接口描述：更新数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | audit/rsdpauditdatasetrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditDatasetRecordsDTO | 是 | 数据集审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

删除数据集审计记录

接口描述：删除数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | audit/rsdpauditdatasetrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | ID列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

导出数据集审计记录

接口描述：导出数据集审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditdatasetrecords/export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | [stream] | 是 | 文件流 |

状态转换描述

无。

###### 实体类

RsdpAuditDatasetRecordsEntity

摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | RsdpAuditDatasetRecordsEntity | | | | | | |
| 父类 | BaseEntity | | | | | | |
| 类职责简述 | 数据集审计信息记录 | | | | | | |
| 持久化 | 是 | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/entity/RsdpAuditDatasetRecordsEntity.java | | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | 数据库校验\* | | 说明 |
| datasetId | long | □公有 ■私有 □保护 | | |  | | 存证主体ID |
| evidence | string | □公有 ■私有 □保护 | | |  | | 存证数据 |
| evidenceSignature | string | □公有 ■私有 □保护 | | |  | | 存证数据签名 |
| evidenceHash | string | □公有 ■私有 □保护 | | |  | | 存证数据哈希 |
| transactionHash | string | □公有 ■私有 □保护 | | |  | | 交易哈希 |
| blockNumber | string | □公有 ■私有 □保护 | | |  | | 区块高度 |
| blockHash | string | □公有 ■私有 □保护 | | |  | | 区块哈希 |
| datasetName | string | □公有 ■私有 □保护 | | |  | | 数据集名称 |
| actionType | int | □公有 ■私有 □保护 | | |  | | 操作类型  0-创建数据集  1-发布数据集  2-删除数据集  3-添加数据文件  4-删除数据文件  5-授权成功  6-取消授权  7-项目引用  8-项目取消  9-实例使用  10-实例取消 |
| userId | long | □公有 ■私有 □保护 | | |  | | 操作人员id |
| userName | string | □公有 ■私有 □保护 | | |  | | 操作人员姓名 |
| info | string | □公有 ■私有 □保护 | | |  | | 操作详情 |
| isBlockchainRecord | string | □公有 ■私有 □保护 | | |  | | 是否是区块链存证数据 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | rsdp\_audit\_dataset\_records | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_dataset\_records**>**  **操作：<查询>** |
| SQL语句：  【SQL01】获取数据集审计列表  SELECT id, dataset\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, dataset\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_dataset\_records ORDER BY create\_date DESC LIMIT ?  【SQL02】根据ID获取数据集审计记录  SELECT id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_dataset\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_dataset\_records **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增数据集审计记录  INSERT into rsdp\_audit\_dataset\_records (id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_dataset\_records **>**  **操作：<删除>** |
| 【SQL04】删除数据集审计记录  DELETE from rsdp\_audit\_dataset\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_dataset\_records **>**  **操作：<更新>** |
| 【SQL05】更新数据集审计记录  UPDATE rsdp\_audit\_dataset\_records SET secret\_id=?, evidence=?, evidence\_signature=?, evidence\_hash=?, transaction\_hash=?, block\_number=?, block\_hash=?, secret\_name=?, action\_type=?, user\_id=?, user\_name=?, info=?, is\_blockchain\_record=? WHERE id=? |

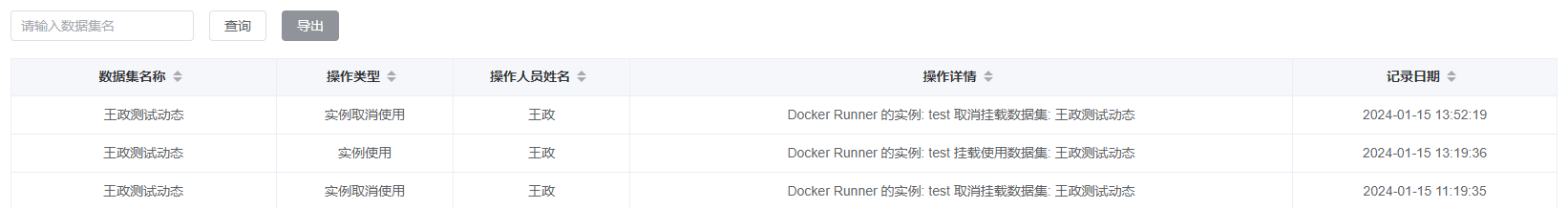
##### 总体设计方案

###### 设计方案

左侧选择审计报表菜单，点击数据集审计菜单，查看数据集审计记录。

###### 处理流程

1.数据集审计记录

2、数据集审计表

调用【7.5.5.2.1.1.1.2.1. 分页获取数据集审计记录】执行【SQL01】，查询数据集审计信息，用户可以在此查看数据集审计记录。

3、数据集审计信息

调用【7.5.5.2.1.1.1.2.2. 根据ID获取数据集审计记录】执行【SQL02】，根据ID查询数据集审计记录详情。

4、删除数据集审计信息

调用【7.5.5.2.1.1.1.2.5. 删除数据集审计记录】执行【SQL04】，根据ID删除指定的数据集审计记录。

#### 应用、实例审计

提供应用审批、上架、实例化、实例启动、停止的全流程记录和审计服务。



##### 类设计

###### 控制类

RsdpAuditApplicationRecordsController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAuditApplicationRecordsController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用审计控制类 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/controller/RsdpAuditApplicationRecordsController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAuditApplicationRecordsService | RsdpAuditApplicationRecordsService | private | | | 无 |
| sysDictTypeService | SysDictTypeService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAuditApplicationRecordsDTO>> | 分页获取应用审计信息 |
| get | id | Long | ■公有 □私有 □保护 | Result<RsdpAuditApplicationRecordsDTO> | 根据ID获取应用审计信息 |
| save | dto | RsdpAuditApplicationRecordsDTO | ■公有 □私有 □保护 | Result | 新建应用审计信息 |
| update | dto | RsdpAuditApplicationRecordsDTO | ■公有 □私有 □保护 | Result | 更新应用审计信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除应用审计信息 |
| export | params  response | Map<String, Object>  HttpServletResponse | ■公有 □私有 □保护 | 无 | 导出应用审计信息 |

算法描述

分页获取应用审计记录

接口描述：分页获取应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditapplicationrecords/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | int | 是 | 当前页码，从1开始 |
| limit | int | 是 | 每页显示记录数 |
| orderField | String | 否 | 排序字段 |
| order | String | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 应用审计分页记录 |

根据ID获取应用审计记录

接口描述：根据ID获取应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditapplicationrecords/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 应用审计记录 |

新建应用审计记录

接口描述：新建应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | audit/rsdpauditapplicationrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditApplicationRecordsDTO | 是 | 应用审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

更新应用审计记录

接口描述：更新应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | audit/rsdpauditapplicationrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditApplicationRecordsDTO | 是 | 应用审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

删除应用审计记录

接口描述：删除应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | audit/rsdpauditapplicationrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | ID列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

导出应用审计记录

接口描述：导出应用审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditapplicationrecords/export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | [stream] | 是 | 文件流 |

状态转换描述

无。

RsdpAuditInstanceRecordsController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAuditInstanceRecordsController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 实例审计控制类 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/controller/RsdpAuditInstanceRecordsController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAuditInstanceRecordsService | RsdpAuditInstanceRecordsService | private | | | 无 |
| sysDictTypeService | SysDictTypeService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAuditInstanceRecordsDTO>> | 分页获取实例审计信息 |
| get | id | Long | ■公有 □私有 □保护 | Result< RsdpAuditInstanceRecordsDTO> | 根据ID获取实例审计信息 |
| save | dto | RsdpAuditInstanceRecordsDTO | ■公有 □私有 □保护 | Result | 新建实例审计信息 |
| update | dto | RsdpAuditInstanceRecordsDTO | ■公有 □私有 □保护 | Result | 更新实例审计信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除实例审计信息 |
| export | params  response | Map<String, Object>  HttpServletResponse | ■公有 □私有 □保护 | 无 | 导出实例审计信息 |

算法描述

分页获取实例审计记录

接口描述：分页获取实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditinstancerecords/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | int | 是 | 当前页码，从1开始 |
| limit | int | 是 | 每页显示记录数 |
| orderField | String | 否 | 排序字段 |
| order | String | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 实例审计分页记录 |

根据ID获取实例审计记录

接口描述：根据ID获取实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditinstancerecords/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 应用审计记录 |

新建实例审计记录

接口描述：新建实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | audit/rsdpauditinstancerecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditInstanceRecordsDTO | 是 | 实例审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

更新实例审计记录

接口描述：更新实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | audit/rsdpauditinstancerecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditInstanceRecordsDTO | 是 | 实例审计信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

删除实例审计记录

接口描述：删除实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | audit/rsdpauditinstancerecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | ID列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

导出实例审计记录

接口描述：导出实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditinstancerecords/export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | [stream] | 是 | 文件流 |

状态转换描述

无。

###### 实体类

RsdpAuditApplicationRecordsEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | RsdpAuditApplicationRecordsEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 应用审计记录表 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $com/capinfo/healthcare/mhsdip/ppc/modules/audit/entity/RsdpAuditApplicationRecordsEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| applicationId | long | | □公有 ■私有 □保护 | | |  | | 应用id |
| applicationName | string | | □公有 ■私有 □保护 | | |  | | 应用名称 |
| actionType | int | | □公有 ■私有 □保护 | | |  | | 操作类型 0-创建应用 1-发布应用 2-删除应用 3-授权成功 4-取消授权 5-项目引用 6-项目取消 |
| userId | long | | □公有 ■私有 □保护 | | |  | | 操作人员id |
| userName | string | | □公有 ■私有 □保护 | | |  | | 操作人员姓名 |
| info | string | | □公有 ■私有 □保护 | | |  | | 操作详情 |
| evidence | string | | □公有 ■私有 □保护 | | |  | | 存证数据 |
| evidenceSignature | string | | □公有 ■私有 □保护 | | |  | | 存证数据签名 |
| evidenceHash | string | | □公有 ■私有 □保护 | | |  | | 存证数据哈希 |
| transactionHash | string | | □公有 ■私有 □保护 | | |  | | 交易哈希 |
| blockNumber | string | | □公有 ■私有 □保护 | | |  | | 区块高度 |
| blockHash | string | | □公有 ■私有 □保护 | | |  | | 区块哈希 |
| isBlockchainRecord | boolean | | □公有 ■私有 □保护 | | | NN | | 是否是区块链存证数据 0-否 1-是 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_audit\_application\_records | | |

算法描述

无。

状态转换描述

无。

RsdpAuditInstanceRecordsEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | RsdpAuditInstanceRecordsEntity | | | | | | | |
| 父类 | BaseEntity | | | | | | | |
| 类职责简述 | 应用实例审计记录表 | | | | | | | |
| 持久化 | 是 | | | | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/entity/RsdpAuditDatasetRecordsEntity.java | | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | | 类型 | | 可见性 | | 数据库校验\* | | 说明 |
| projectId | | long | | □公有 ■私有 □保护 | |  | | 项目ID |
| projectName | | string | | □公有 ■私有 □保护 | |  | | 项目名称 |
| applicationId | | long | | □公有 ■私有 □保护 | |  | | 应用ID |
| applicationName | | string | | □公有 ■私有 □保护 | |  | | 应用名称 |
| instanceId | | long | | □公有 ■私有 □保护 | |  | | 实例ID |
| instanceName | | string | | □公有 ■私有 □保护 | |  | | 实例名称 |
| actionType | | int | | □公有 ■私有 □保护 | |  | | 操作类型 0-创建实例 1-删除实例 2-暂停实例 3-恢复实例 |
| userId | | long | | □公有 ■私有 □保护 | |  | | 操作人员id |
| userName | | string | | □公有 ■私有 □保护 | |  | | 操作人员姓名 |
| info | | string | | □公有 ■私有 □保护 | |  | | 操作详情 |
| evidence | | string | | □公有 ■私有 □保护 | |  | | 存证数据 |
| evidenceSignature | | string | | □公有 ■私有 □保护 | |  | | 存证数据签名 |
| evidenceHash | | string | | □公有 ■私有 □保护 | |  | | 存证数据哈希 |
| transactionHash | | string | | □公有 ■私有 □保护 | |  | | 交易哈希 |
| blockNumber | | string | | □公有 ■私有 □保护 | |  | | 区块高度 |
| isBlockchainRecord | | string | | □公有 ■私有 □保护 | |  | | 是否是区块链存证数据 |
| 方法 | | | | | | | | |
| 方法名称 | | 参数名称 | 参数类型 | | 可见性 | 返回值 | 说明 | |
| 无 | | 无 | 无 | | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_audit\_instance\_records | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_application\_records、rsdp\_audit\_instance\_records **>**  **操作：<查询>** |
| SQL语句：  【SQL01】获取应用审计列表  SELECT id, application\_id, application\_name, action\_type, user\_id, user\_name, info, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_application\_records ORDER BY create\_date DESC LIMIT ?  【SQL02】获取实例审计列表  SELECT id, project\_id, project\_name, application\_id, application\_name, instance\_id, instance\_name, action\_type, user\_id, user\_name, info, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, is\_blockchain\_record, creator, create\_date FROM rsdp\_audit\_instance\_records ORDER BY create\_date DESC LIMIT ?,? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_application\_records、rsdp\_audit\_instance\_records **>**  **操作：<新增>** |
| SQL语句：  【SQL03】新增应用审计记录  INSERT into rsdp\_audit\_application\_records (id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)  【SQL04】新增实例审计记录  INSERT into rsdp\_audit\_instance\_records (id, secret\_id, evidence, evidence\_signature, evidence\_hash, transaction\_hash, block\_number, block\_hash, secret\_name, action\_type, user\_id, user\_name, info, is\_blockchain\_record, creator, create\_date) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_application\_records、rsdp\_audit\_instance\_records **>**  **操作：<删除>** |
| 【SQL05】删除应用审计记录  DELETE from rsdp\_audit\_application\_records WHERE id=?  【SQL06】删除实例审计记录  DELETE from rsdp\_audit\_instance\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_application\_records、rsdp\_audit\_instance\_records **>**  **操作：<更新>** |
| 【SQL07】更新应用审计记录  UPDATE rsdp\_audit\_application\_records SET secret\_id=?, evidence=?, evidence\_signature=?, evidence\_hash=?, transaction\_hash=?, block\_number=?, block\_hash=?, secret\_name=?, action\_type=?, user\_id=?, user\_name=?, info=?, is\_blockchain\_record=? WHERE id=?  【SQL08】更新实例审计记录  UPDATE rsdp\_audit\_instance\_records SET secret\_id=?, evidence=?, evidence\_signature=?, evidence\_hash=?, transaction\_hash=?, block\_number=?, block\_hash=?, secret\_name=?, action\_type=?, user\_id=?, user\_name=?, info=?, is\_blockchain\_record=? WHERE id=? |

##### 总体设计方案

###### 设计方案

左侧选择审计报表菜单，点击应用审计菜单，查看应用审计记录。

###### 处理流程

1.应用审计记录



应用审计表

调用【7.5.5.3.1.1.1.2.1. 分页获取应用审计记录】执行【SQL01】，查询应用审计信息，用户可以在此查看应用审计记录。

2.实例审计记录



调用【7.5.5.3.1.1.2.2.1. 分页获取实例审计记录】执行【SQL02】，查询实例审计信息，用户可以在此查看实例审计记录。

3、删除应用审计信息

调用【7.5.5.3.1.1.1.2.5.删除应用审计记录】执行【SQL05】，根据ID删除应用审计记录详情。

4、删除实例审计信息

调用【7.5.5.3.1.1.2.2.5. 删除实例审计记录】执行【SQL06】，根据ID删除指定的实例审计记录。

#### 审批流程可视化

对于数据、应用及项目使用流程提供流程审批服务并对流程处理提供可视化展示。



##### 类设计

###### 控制类

RsdpAuditFlowController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAuditFlowController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 审计流程控制类 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/audit/controller/RsdpAuditFlowController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAuditFlowService | RsdpAuditFlowService | private | | | 无 |
| sysDictTypeService | SysDictTypeService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAuditFlowRecordsDTO>> | 分页获取审计信息 |
| get | id | Long | ■公有 □私有 □保护 | Result< RsdpAuditFlowRecordsDTO> | 根据ID获取审计信息 |
| save | dto | RsdpAuditFlowRecordsDTO | ■公有 □私有 □保护 | Result | 新建审计信息 |
| update | dto | RsdpAuditFlowRecordsDTO | ■公有 □私有 □保护 | Result | 更新审计信息 |
| delete | ids | Long[] | ■公有 □私有 □保护 | Result | 删除审计信息 |
| export | params  response | Map<String, Object>  HttpServletResponse | ■公有 □私有 □保护 | 无 | 导出审计信息 |

算法描述

分页获取审计流程记录

接口描述：分页获取审计流程记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditflowrecords/page | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | int | 是 | 当前页码，从1开始 |
| limit | int | 是 | 每页显示记录数 |
| orderField | String | 否 | 排序字段 |
| order | String | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 审计流程分页记录 |

根据ID获取审计流程记录

接口描述：根据ID获取审计流程记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpaudiflowrecords/{id} | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | long | 是 | ID |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |
| data | [object] | 是 | 应用审计记录 |

新建审计流程

接口描述：新建审计流程。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | audit/rsdpauditflowrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditFlowRecordsDTO | 是 | 审计流程信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

更新审计流程

接口描述：更新审计流程。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | audit/rsdpauditflowrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| dto | RsdpAuditFlowRecordsDTO | 是 | 审计流程信息 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

删除审计流程记录

接口描述：删除审计流程记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | audit/rsdpauditflowrecords | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | Long[] | 是 | ID列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 编码：0表示成功，其他值表示失败 |
| msg | [string] | 是 | 消息内容 |

导出实例审计记录

接口描述：导出实例审计记录。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | audit/rsdpauditflowrecords/export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | [stream] | 是 | 文件流 |

状态转换描述

无。

###### 实体类

RsdpAuditFlowRecordsEntity

摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | RsdpAuditFlowRecordsEntity | | | | | | |
| 父类 | BaseEntity | | | | | | |
| 类职责简述 | 审计流程记录表 | | | | | | |
| 持久化 | 是 | | | | | | |
| 代码发布路径 | $com/capinfo/healthcare/mhsdip/ppc/modules/audit/entity/RsdpAuditFlowRecordsEntity.java | | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | 可见性 | | | 数据库校验\* | | 说明 |
| flowId | long | □公有 ■私有 □保护 | | |  | | 流程id |
| flowName | string | □公有 ■私有 □保护 | | |  | | 流程名称 |
| actionType | int | □公有 ■私有 □保护 | | |  | | 操作类型 0-创建流程 1-发布流程 2-删除流程 3-授权成功 4-取消授权 5-项目引用 6-取消引用 |
| userId | long | □公有 ■私有 □保护 | | |  | | 操作人员id |
| userName | string | □公有 ■私有 □保护 | | |  | | 操作人员姓名 |
| info | string | □公有 ■私有 □保护 | | |  | | 操作详情 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | rsdp\_audit\_flow\_records | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_flow\_records **>**  **操作：<查询>** |
| SQL语句：  【SQL01】获取审计流程列表  SELECT id, flow\_id, flow\_name, action\_type, user\_id, user\_name, info, creator, create\_date FROM rsdp\_audit\_flow\_records ORDER BY create\_date DESC LIMIT ? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_flow\_records**>**  **操作：<新增>** |
| SQL语句：  【SQL02】新增审计流程记录  INSERT into rsdp\_audit\_dataset\_records (id, flow\_id, flow\_name, action\_type, user\_id, user\_name, info, creator, create\_date) values(?,?,?,?,?,?,?,?,?) |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_flow\_records**>**  **操作：<删除>** |
| 【SQL03】删除审计流程记录  DELETE from rsdp\_audit\_flow\_records WHERE id=? |
| **程序流程中涉及的数据库表：<表名：**rsdp\_audit\_flow\_records**>**  **操作：<更新>** |
| 【SQL04】更新审计流程记录  UPDATE rsdp\_audit\_flow\_records SET flow\_id=?, flow\_name=?, action\_type=?, user\_id=?, user\_name=?, info=? WHERE id=? |

##### 总体设计方案

###### 设计方案

左侧选择审计报表菜单，点击我的审核菜单，查看审计流程记录。

###### 处理流程

1.审计流程记录



2、审计流程列表

调用【7.5.5.4.1.1.1.2.1. 分页获取审计流程记录】执行【SQL01】，查询审计流程信息，用户可以在此查看审计流程记录。

3、删除审计流程信息

调用【7.5.5.4.1.1.1.2.5. 删除审计流程记录】执行【SQL03】，根据ID删除指定的审计流程记录详情。

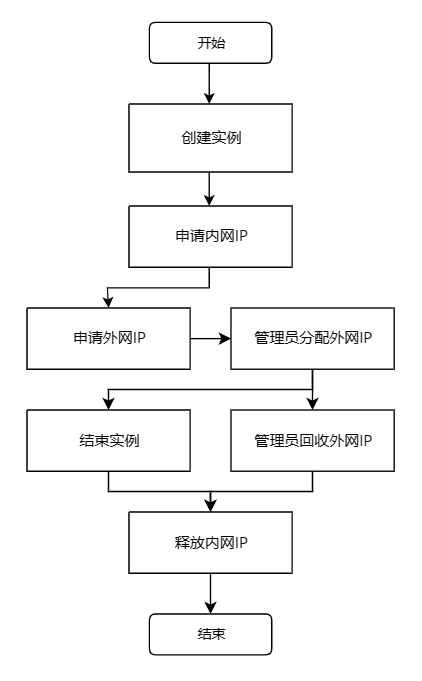
4、删除数据集审计信息

调用【7.5.5.4.1.1.1.2.4. 更新审计流程】执行【SQL04】，根据ID更新指定的审计流程记录。

### 运营管理

#### 外网访问控制

外网访问控制主要实现管理系统的公网IP资源并分配给应用实例使用。主要负责对网络中的设备分配IP地址，并确保这些IP地址的有效使用和合理分配以及回收。IP管理为用户实例分配内网和外网IP，并统一管理调配。主要流程如下：



##### 类设计

###### 控制类

RsdpIpController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpIpController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | IP相关接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\rsdpip\controller\RsdpIpController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | page  paseSize | int  int | ■公有 □私有 □保护 | Result<PageData<RsdpIpDTO>> | 获取异常日志列表 |
| getIdleList | 无 | 无 | ■公有 □私有 □保护 | Result<List<RsdpIpDTO>> | 获取空闲IP列表 |
| getInstanceIpPorts | id | Long | ■公有 □私有 □保护 | Result<RsdpIpEntity | 获取instance绑定的ip信息 |
| get | id | Long | ■公有 □私有 □保护 | Result<RsdpIpEntity | 根据ID获取信息 |
| mapIPAddress | data | MapData | ■公有 □私有 □保护 | Result<> | 添加IP映射关系 |
| unmapIPAddress | data | MapData | ■公有 □私有 □保护 | Result<> | 移除IP映射关系 |
| applyIPAddress | data | ApplyData | ■公有 □私有 □保护 | Result<> | 为实例申请内网IP |
| freeIPAddress | data | ApplyData | ■公有 □私有 □保护 | Result<> | 释放内网IP |
| save | entity | RsdpIpEntity | ■公有 □私有 □保护 | Result<> | 新增IP信息 |
| update | entity | RsdpIpEntity | ■公有 □私有 □保护 | Result<> | 更新IP信息 |
| delete | entity | RsdpIpEntity | ■公有 □私有 □保护 | Result<> | 删除IP信息 |
| export | 无 | 无 | ■公有 □私有 □保护 | XlsxFile | 导出异常日志 |

算法描述

获取IP列表接口

接口描述：查询IP列表，接口返回IP结构体列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /page | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | [number] | 否 | 页码 |
| pageSize | [number] | 否 | 页码显示条数 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| list | [array] | 是 | IP列表 |

获取空闲IP列表接口

接口描述：获取当前未分配的空闲IP列表。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /getIdleList | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| list | [array] | 是 | 空闲IP列表 |

获取实例绑定的ip信息接口

接口描述：请求中携带实例id，返回此id绑定的ip的详细信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /getInstanceIpPorts | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | 实例id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [ipEntity] | 是 | ip实体类型 |

根据ID获取IP信息接口

接口描述：根据IP的id，获取此ip的详细信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /{id} | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | IP的id号 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [ipEntity] | 是 | ip实体类型 |

添加IP映射关系接口

接口描述：添加内网ip地址和外网ip地址之间的映射关系。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /map | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | [mapData] | 是 | 映射关系对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

移除IP映射关系接口

接口描述：移除内网ip地址和外网ip地址之间的映射关系。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /unmap | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | [mapData] | 是 | 映射关系对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

为实例申请内网IP接口

接口描述：选择ip，将实例与此ip进行绑定，将其分配给实例并修改状态。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /apply | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | [applyData] | 是 | 映射关系对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

释放内网IP接口

接口描述：将ip与绑定的实例执行解绑操作，并修改ip的状态。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /free | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| page | [applyData] | 是 | 映射关系对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

新增IP信息接口

接口描述：添加IP记录并写入数据库。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | / | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| entity | [ipEntity] | 是 | IP信息对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

更新IP信息接口

接口描述：用请求中携带的IP信息更新原IP信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | / | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| entity | [ipEntity] | 是 | IP信息对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

删除IP信息接口

接口描述：删除指定id的IP信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | / | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| ids | [array] | 是 | 需删除的id列表 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [string] | 是 | 结果信息 |

IP信息导出接口

接口描述：请求导出IP列表，返回IP列表的xlsx文件。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /export | application/vnd.ms-excel |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| body | [application/vnd.ms-excel] | 是 | 导出文件 |

###### 实体类

RsdpIpEntity

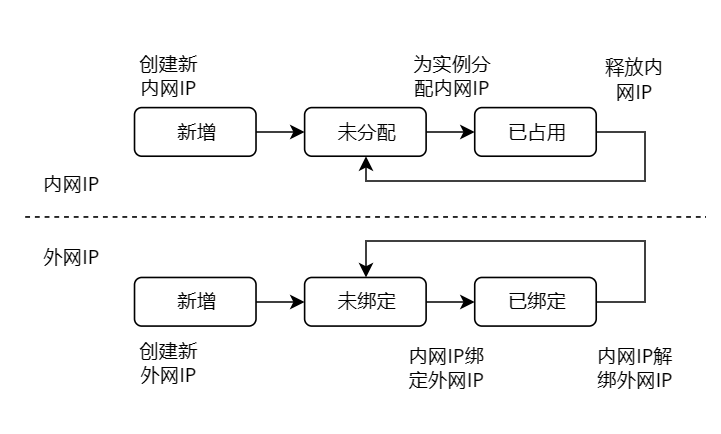
摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | RsdpIpEntity | | | | | | |
| 父类 | | 无 | | | | | | |
| 类职责简述 | | IP地址详细信息 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\rsdpip\entity\RsdpIpEntity.java | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | | 说明 |
| name | string | | □公有 ■私有 □保护 | | |  | | 名称 |
| address | string | | □公有 ■私有 □保护 | | | NNS | | 地址 |
| status | integer | | □公有 ■私有 □保护 | | | NNIC | | 状态 |
| portsStatus | string | | □公有 ■私有 □保护 | | |  | | 端口状态 |
| ports | string | | □公有 ■私有 □保护 | | |  | | 端口 |
| target | string | | □公有 ■私有 □保护 | | |  | | 分配对象 |
| targetId | long | | □公有 ■私有 □保护 | | |  | | 分配对象id |
| description | string | | □公有 ■私有 □保护 | | |  | | 描述 |
| updater | long | | □公有 ■私有 □保护 | | |  | | 更新人员id |
| updateDate | date | | □公有 ■私有 □保护 | | |  | | 更新时间 |
| deptId | long | | □公有 ■私有 □保护 | | |  | | 所属机构 |
| ipType | string | | □公有 ■私有 □保护 | | | NNS | | IP类型 |
| map | string | | □公有 ■私有 □保护 | | |  | | IP映射 |
| version | Integer | | □公有 ■私有 □保护 | | | NNIC | | 版本 |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 | |
| 无 | 无 | | | 无 | 无 | 无 | 无 | |
| 对应数据库表 | | | | | | rsdp\_ip | | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_ip **>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询IP列表  SELECT  e.id,  e.name,  e.address status,  e.ports\_status,  e.ports,  e.description,  e.target\_id,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.dept\_id,  e.version,  e.ip\_type,  e.map,  e.target  LIMIT  (page - 1) \* pageSize, pageSize;  【SQL02】查询指定id的IP详细信息  SELECT  e.id,  e.name,  e.address status,  e.ports\_status,  e.ports,  e.description,  e.target\_id,  e.creator,  e.create\_date,  e.updater,  e.update\_date,  e.dept\_id,  e.version,  e.ip\_type,  e.map,  e.target  WHERE  e.id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_ip **>**  **操作：<插入>** |
| SQL语句：  【SQL03】插入IP详细信息  INSERT INTO rsdp\_ip(id, name, address, status, ports\_status, ports, description, target\_id, creator, create\_date, updater, update\_date, dept\_id, version, ip\_type, map, target  )  VALUES  (#{id},  #{name},  #{address status},  #{ports\_status ports},  #{description},  #{target\_id},  #{creator},  #{create\_date},  #{updater },  #{update\_date},  #{dept\_id },  #{version },  #{ip\_type },  #{map},  #{target}  ) |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_ip **>**  **操作：<更新>** |
| SQL语句：  【SQL04】更新IP详细信息  UPDATE  rsdp\_ip  SET  name = #{name},  address = #{address},  status = #{status},  ports\_status = #{ports\_status},  ports = #{ports},  description = #{description},  target\_id = #{target\_id},  creator = #{creator},  create\_date = #{create\_date},  updater = #{updater},  update\_date = #{update\_date},  dept\_id = #{dept\_id},  version = #{version},  ip\_type = #{ip\_type},  map = #{map},  target = #{target}  WHERE  id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_ip **>**  **操作：<删除>** |
| SQL语句：  【SQL05】删除IP详细信息  DELETE FROM  rsdp\_ip  WHERE  id IN #{ids}; |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | id |  | | name | 名称 |  | | address | ip地址 |  | | status | 状态 |  | | ports\_status | 端口状态 |  | | ports | 端口 |  | | description | 备注 |  | | target\_id | 目标id |  | | creator | 操作人员 |  | | create\_date | 创建时间 |  | | updater | 更新人员 |  | | update\_date | 更新时间 |  | | dept\_id | 部门id |  | | version | 版本 |  | | ip\_type | Ip类型 |  | | map | 绑定信息 |  | | target | 目标名称 |  | |
|  |

##### 总体设计方案

###### 设计方案

旨在实现IP地址的高效管理，通过一系列的接口提供全面的IP管理功能。主要功能包括分页查询IP列表、获取空闲IP列表、查看单个IP详细信息、添加和移除IP映射关系、为实例申请和释放内网IP、保存和修改IP信息、删除IP信息以及导出IP信息等。系统采用了权限控制，确保只有授权用户才能进行操作。此外，系统还提供了异常处理机制，确保在出现错误时能够给出相应的提示信息。

###### 处理流程

1.IP创建

管理员以及具备操作权限的用户可以在系统界面IP管理中，在添加 IP 地址对话框里可以选择添加 IP 范围并输入开始 IP 和结束 IP 或选择单一 IP 并输入 IP 地址，点击“确定”按钮将 IP 或 IP 范围对应的多个IP 加入 IP 列表。创建内网和外网IP， 通过调用【7.5.6.1.1.1.1.2.9.新增IP信息接口】，系统接收到请求后执行【SQL03】执行创建IP的操作。

2.IP编辑

管理员以及具备操作权限的用户可以在系统界面IP管理中，调用【7.5.6.6.1.1.1.2.1-获取IP列表接口】执行【SQL01】获取IP列表，编辑和修改内网和外网IP信息，通过调用【7.5.6.6.1.1.1.2.10-更新IP信息接口】更新修改IP信息接口，系统接收到请求后执行【SQL04】执行更新IP的操作。

3.IP删除

管理员以及具备操作权限的用户可以在系统界面IP管理中，删除执行内网和外网IP信息，通过调用【7.5.6.6.1.1.1.2.11-删除IP信息接口删除IP信息接口，系统接收到请求后执行【SQL05】执行删除IP的操作。

4.IP管理与分配





“内网 IP”标签页以列表方式分页显示内网 IP 及使用情况，包括：IP、

端口、可用性、外网 IP、外网端口、占用实例及操作，提供基于内网 IP 和外

网 IP 的条件查询；点击“添加”按钮弹出添加 IP 地址对话框。“外网 IP”标签页以列表方式分页显示外网 IP 及使用情况，包括：IP、端口、可用性、内网 IP、内网端口、占用实例及操作，提供基于内网 IP 和外网 IP 的条件查询。

在内网 IP列表的操作区点击“分配”按钮弹出外网 IP 选择对话框，在对话框内选择外网 IP 并输入端口，点击“确定”按钮完成内网及外网 IP 的映射。此过程中，系统首先会调用【7.5.6.6.1.1.1.2.5-添加IP映射关系接口】添加IP映射关系结构，将内网和外网IP绑定，执行【SQL04】更新IP的状态和关联关系。在内网IP列表的操作区点击“解除”按钮弹出确认对话框，点击“确认”按钮释放公网 IP。此过程中，系统首先会调用【7.5.6.6.1.1.1.2.6-移除IP映射关系接口】解除IP映射关系结构，将内网和外网IP解绑，执行【SQL04】更新IP的状态和关联关系。

在用户创建实例时，可以选择分配一个内网IP，此时系统首先会调用【7.5.6.6.1.1.1.2.2-获取空闲IP列表接口】获取空闲IP列表，供用户选择，选择完成后系统调用【7.5.6.6.1.1.1.2.7-为实例申请内网IP接口】为实例申请内网IP接口，并执行【SQL04】更新IP的状态和IP的目标实例信息。在实例删除后，系统会调用【7.5.6.6.1.1.1.2.6-移除IP映射关系接口】解除实例和内网IP关联接口，并执行【SQL04】更新IP的状态和移除IP的目标实例信息。

#### 入侵监测

##### 类设计

###### 控制类

WebhookRecordController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | ApplicationController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 应用接口实现 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\webhook\controller\WebhookRecordController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| recordWebhook | entity | WebhookRecordEntity | ■公有 □私有 □保护 | Result | 产生webhook后信息上报 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData> | 分页获取webhook记录信息 |
| get | id | Long | ■公有 □私有 □保护 | Result | 根据ID获取webhook记录信息 |

算法描述

记录webhook接口

接口描述：记录webhook接口

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /recordWebhook | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| entity | [WebhookRecordEntity] | 是 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 消息 |

分页获取webhook记录信息

接口描述：分页获取webhook记录信息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | page | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| params | [map] | 是 | 参数 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| list | [array] | 是 | webhook纪录列表 |

获取webhook接口

接口描述：获取webhook接口。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | get | 无 |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| id | [number] | 是 | id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| data | [entity] | 是 | 结果信息 |

###### 实体类

WebhookRecordEntity

摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | | WebhookRecordEntity | | | | | |
| 父类 | | 无 | | | | | |
| 类职责简述 | | Webhook记录信息 | | | | | |
| 持久化 | | 是 | | | | | |
| 代码发布路径 | | $\com\capinfo\healthcare\mhsdip\ppc\modules\webhook\entity\WebhookRecordEntity.java | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | 说明 |
| id | Long | | □公有 ■私有 □保护 | | | NNU | 主键ID |
| target | String | | □公有 ■私有 □保护 | | | NNS | 目标名称/ID |
| type | String | | □公有 ■私有 □保护 | | | NNS | 类型 |
| event | String | | □公有 ■私有 □保护 | | | NNS | 事件 |
| time | Date | | □公有 ■私有 □保护 | | | NNU | 时间 |
| targetKey | String | | □公有 ■私有 □保护 | | | NNS | 操作key |
| size | Long | | □公有 ■私有 □保护 | | | NNI | 大小 |
| contentType | String | | □公有 ■私有 □保护 | | | NNS | 内容类型 |
| tag | String | | □公有 ■私有 □保护 | | | NNS | 标签 |
| flag | Integer | | □公有 ■私有 □保护 | | | NNIC | 标记 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | 说明 |
| 无 | 无 | | | 无 | 无 | 无 | 无 |
| 对应数据库表 | | | | | | rsdp\_webhook\_record | |

算法描述

无。

状态转换描述



##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_webhook\_record**>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询webhook列表  SELECT  id,  target,  type,  event,  time,  target\_key,  size,  content\_type,  tag,  flag  【SQL02】查询指定id的webhook记录  SELECT  target,  type,  event,  time,  target\_key,  size,  content\_type,  tag,  flag  WHERE  e.id = #{id}; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_webhook\_record**>**  **操作：<插入>** |
| SQL语句：  【SQL03】插入webhook信息  INSERT INTO rsdp\_application\_audit\_task(  target,  type,  event,  time,  target\_key,  size,  content\_type,  tag,  flag  )  VALUES  (  #{target},  #{type},  #{event},  #{time},  #{target\_key},  #{size},  #{content\_type},  #{tag},  #{flag}  ) |
|  |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 主键ID |  | | target | 目标名称/ID |  | | type | 类型 |  | | event | 事件 |  | | time | 时间 |  | | targetKey | 操作key |  | | size | 大小 |  | | contentType | 内容类型 |  | | tag | 标签 |  | | flag | 标记 |  | |
|  |

##### 总体设计方案

###### 设计方案

平台管理员通过菜单进入在线用户列表页，查看和管理用户登录状态。管理员可查看用户登录历史和操作记录，支持条件查询和日志导出功能。异常日志列表页显示系统内部异常信息，提供异常详情查看和日志导出功能。

###### 处理流程

管理员可以点击“日志管理\登录日志”菜单，进入用户登录日志列表页。该列表页同样以分页方式显示用户登录的历史信息，包括用户名、操作类型、状态、操作IP、User-Agent、时间等。管理员可以根据用户名和状态进行条件查询，并通过点击“导出”按钮，将日志导出到Excel文件并下载。

在用户使用系统时，进行操作时会记录并调用【7.5.6.2.1.1.1.2.1记录webhook接口】，执行【SQL03】产生webhook记录。后台程序查询【7.5.6.2.1.1.1.2.3获取webhook接口】并执行【SQL02】查询记录，并进行比较。

管理员还可以点击“日志管理\操作日志”菜单，进入操作日志列表页。该列表页分页显示用户访问系统网页进行的增、删、改、查等操作，包括用户名、用户操作、访问地址、1、请求方式、2、请求参数、请求时长、状态、操作IP、User-Agent、时间等信息。管理员可以根据状态进行条件查询，并通过点击“导出”按钮，将日志导出到Excel文件并下载。此时会调用【7.5.6.2.1.1.1.2.2分页获取webhook记录信息】并执行【SQL01】查询列表。

管理员点击“日志管理\异常日志”菜单，可以进入异常日志列表页。该列表页分页显示系统内部发生的异常信息，包括访问地址、请求方式、请求参数、请求时长、状态、操作IP、User-Agent、时间、操作等。管理员可以点击操作区的“异常详情”按钮，查看此次异常的全部调用堆栈信息。同样，管理员也可以通过点击“导出”按钮，将日志导出到Excel文件并下载。

#### 系统可视化监控

##### 类设计

###### 控制类

RsdpController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 可视化监控处理 | | | | |
| 代码发布路径 | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdp/controller/RsdpController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| 无 | 无 | 无 | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| statistics | params | Map<String, Object> | ■公有 □私有 □保护 | Result<RsdpStatisticsDTO> | 获取平台的统计数据 |
| instanceStatistics | params | Map<String, Object> | ■公有 □私有 □保护 | Result<Map<String, Integer>> | 获取实例统计数据 |
| datasetStatistics | params | Map<String, Object> | ■公有 □私有 □保护 | Result<Map<String, RsdpDatasetStatisticsDTO>> | 获取数据集统计数据 |
| projectStatistics | params | Map<String, Object> | ■公有 □私有 □保护 | Result<Map<String, RsdpProjectStatisticsDTO>> | 获取项目统计数据 |
| tradeStatistics | params | Map<String, Object> | ■公有 □私有 □保护 | Result<Map<String, Long>> | 查询交易统计数据 |

算法描述

获取平台的统计数据

接口描述：获取平台的统计数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdp/rsdp/statistics | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| deptId | Long | 是 | 部门id |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

获取实例统计数据

接口描述：获取实例统计数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdp/rsdp/instanceStatistics | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

获取数据集统计数据

接口描述：获取数据集统计数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdp/rsdp/datasetStatistics | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

获取项目统计数据

接口描述：获取项目统计数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdp/rsdp/projectStatistics | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

获取交易统计数据

接口描述：获取交易统计数据。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdp/rsdp/tradeStatistics | application/json |

2、请求参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| 无 | 无 | 无 | 无 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | int | 是 | 编码：0表示成功，其他值表示失败 |
| msg | String | 是 | 消息内容 |
| data | Object | 是 | 响应数据 |

###### 实体类

RsdpEntity

摘要

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | | |
| 英文名称 | | RsdpEntity | | | | | | |
| 父类 | | BaseEntity | | | | | | |
| 类职责简述 | | 数据集详情实体类 | | | | | | |
| 持久化 | | 是 | | | | | | |
| 代码发布路径 | | $/com/capinfo/healthcare/mhsdip/ppc/modules/rsdp/entity/RsdpEntity | | | | | | |
| 属性 | | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | | 数据库校验\* | 说明 | |
| totalProject | Interger | | □公有 ■私有 □保护 | | |  | 总项目数 | |
| ongoingProject | Interger | | □公有 ■私有 □保护 | | |  | 进行中的项目数 | |
| totalUser | Interger | | □公有 ■私有 □保护 | | |  | 总用户数 | |
| activeUser | Interger | | □公有 ■私有 □保护 | | |  | 活跃用户数 | |
| totalDataset | Interger | | □公有 ■私有 □保护 | | |  | 总数据集数 | |
| totalInstance | Long | | □公有 ■私有 □保护 | | |  | 总实例数 | |
| 方法 | | | | | | | | |
| 方法名称 | 参数名称 | | | 参数类型 | 可见性 | 返回值 | | 说明 |
| 无 | 无 | | | 无 | 无 | 无 | | 无 |
| 对应数据库表 | | | | | | rsdp | | |

算法描述

无。

状态转换描述

无。

##### SQL文一览

|  |
| --- |
| **程序流程中涉及的数据库表：<表名：**rsdp\_authorize\_dataset\_use、rsdp\_dataset、sys\_dept、sys\_user **>**  **操作：<查询>** |
| SQL语句：  【SQL01】获取项目总数  select \* from rsdp\_project where dept\_id=?  【SQL02】获取总数据集数  select count(\*) from rsdp\_dataset where dept\_id=?  【SQL03】获取总用户数  select \* from sys\_user where dept\_id=?  【SQL04】获取总实例数  select count(\*) from rsdp\_project\_instance where project\_id IN (?) |

##### 总体设计方案

###### 设计方案

用户登录系统，在主页上会显示系统的各种数据，由此可进行系统可视化监控。

###### 处理流程

1.登录系统

输入用户名和密码登录系统。

2.进入主页



进入主页后，执行【7.5.6.3.1.1.1.2.1-获取平台的统计数据】调用【SQL03】获取总用户数目，执行【7.5.6.3.1.1.1.2.2-获取实例统计数据】调用【SQL04】获取实例数目，执行【7.5.6.3.1.1.1.2.3-获取数据集统计数据】调用【SQL02】获取数据集数目，执行【7.5.6.3.1.1.1.2.4-获取项目统计数据】调用【SQL01】获取项目数目，然后在主页上可以查看到各种信息，包括项目总数、进行中的项目数、用户总数、活跃用户数、数据集总数及实例总数。

在下方图表中可以看到实例类型分布、数据集大小及私有数据集和公有数据集数量随时间分布、项目总数及参与项目数量和拥有项目数量随时间分布、卖出和买入的数据集数量等信息。

#### 安全报警通知

提供实时网页报警功能，根据配置规则发送多维度的安全报警邮件或短信。



##### 类设计

###### 控制类

RsdpAlarmController

摘要

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 控制器类 | | | | | |
| 英文名称 | RsdpAlarmController | | | | |
| 父类 | 无 | | | | |
| 类职责简述 | 报警管理 | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\rsdpalarm\controller\ RsdpAlarmController.java | | | | |
| 属性 | | | | | |
| 属性名称 | 类型 | 可见性 | | | 说明 |
| rsdpAlarmService | RsdpAlarmService | private | | | 无 |
| 方法 | | | | | |
| 方法名称 | 参数名称 | 参数类型 | 可见性 | 返回值 | 说明 |
| page | params | Map<String, Object> | ■公有 □私有 □保护 | Result<PageData<RsdpAlarmDTO>> | 获取报警消息列表 |
| save | dto | RsdpAlarmDTO | ■公有 □私有 □保护 | Result | 新建报警消息 |
| update | dto | RsdpAlarmDTO | ■公有 □私有 □保护 | Result | 修改报警消息 |
| delete | ids | Long | ■公有 □私有 □保护 | Result | 删除所选报警消息 |

算法描述

获取报警消息列表

接口描述：查询当前时间系统接收到的所有历史报警消息，并将消息详情整理后反回到前端页面。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| GET | /rsdpalarm/rsdpalarm/page | application/json |

2、请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | | 说明 |
| page | int | | 是 | 当前页码，从1开始 |
| limit | int | | 是 | 每页显示记录数 |
| orderField | String | | 否 | 排序字段 |
| order | String | | 否 | 排序方式，可选值(asc、desc) |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 请求返回信息 |
| data | [object] | 是 | 报警信息数据 |

报警消息保存

接口描述：解析报警消息结构内容，保存报警消息到数据库，并且根据报警消息级别发送告警通知。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| POST | /rsdpalarm/rsdpalarm | application/json |

2、请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | | 说明 |
| dto | RsdpAlarmDTO | | 是 | 报警消息对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 请求返回信息 |
| data | [object] | 否 | 报警信息 |

更新报警消息

接口描述：查找报警消息，解析报警消息结构内容，更新为最新的报警内容。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| PUT | /rsdpalarm/rsdpalarm | application/json |

2、请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | | 说明 |
| dto | RsdpAlarmDTO | | 是 | 报警消息对象 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 请求返回信息 |
| data | [object] | 否 | 报警信息 |

删除报警消息

接口描述：接收报警消息的id列表，遍历删除所有报警消息。

1、请求方式

|  |  |  |
| --- | --- | --- |
| HTTP方法 | URI | 消息体 |
| DELETE | /rsdpalarm/rsdpalarm | application/json |

2、请求参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | | 说明 |
| ids | long[] | | 是 | 当前页码，从1开始 |

3、返回参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必填 | 说明 |
| code | [number] | 是 | 状态码 |
| msg | [string] | 是 | 请求返回信息 |
| data | [object] | 否 | 无 |

###### 实体类

RsdpAlarmEntity

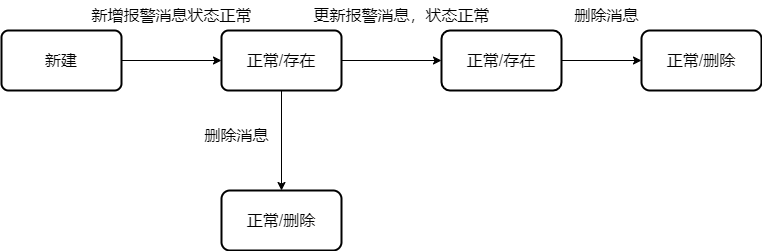
摘要

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 实体类 | | | | | | | |
| 英文名称 | RsdpAlarmEntity | | | | | | |
| 父类 | BaseEntity | | | | | | |
| 类职责简述 | 保存报警消息内容 | | | | | | |
| 持久化 | 是 | | | | | | |
| 代码发布路径 | $\com\capinfo\healthcare\mhsdip\ppc\modules\rsdpalarm\entity\RsdpAlarmEntity | | | | | | |
| 属性 | | | | | | | |
| 属性名称 | 类型 | | 可见性 | | 数据库校验\* | | 说明 |
| content | string | | □公有 ■私有 □保护 | |  | | 报警内容 |
| level | string | | □公有 ■私有 □保护 | |  | | 报警级别 |
| category | string | | □公有 ■私有 □保护 | |  | | 威胁类型 |
| projectName | string | | □公有 ■私有 □保护 | |  | | 项目名称 |
| appName | string | | □公有 ■私有 □保护 | |  | | 应用名称 |
| instanceId | long | | □公有 ■私有 □保护 | |  | | 实例ID |
| instanceName | string | | □公有 ■私有 □保护 | |  | | 实例名称 |
| source | string | | □公有 ■私有 □保护 | |  | | 报警来源 |
| updator | long | | □公有 ■私有 □保护 | |  | | 更新者 |
| updateDate | date | | □公有 ■私有 □保护 | |  | | 更新时间 |
| 方法 | | | | | | | |
| 方法名称 | 参数名称 | 参数类型 | | 可见性 | 返回值 | 说明 | |
| 无 | 无 | 无 | | 无 | 无 | 无 | |
| 对应数据库表 | | | | | rsdp\_alarm | | |

算法描述

无。

状态转换描述



##### SQL文一览

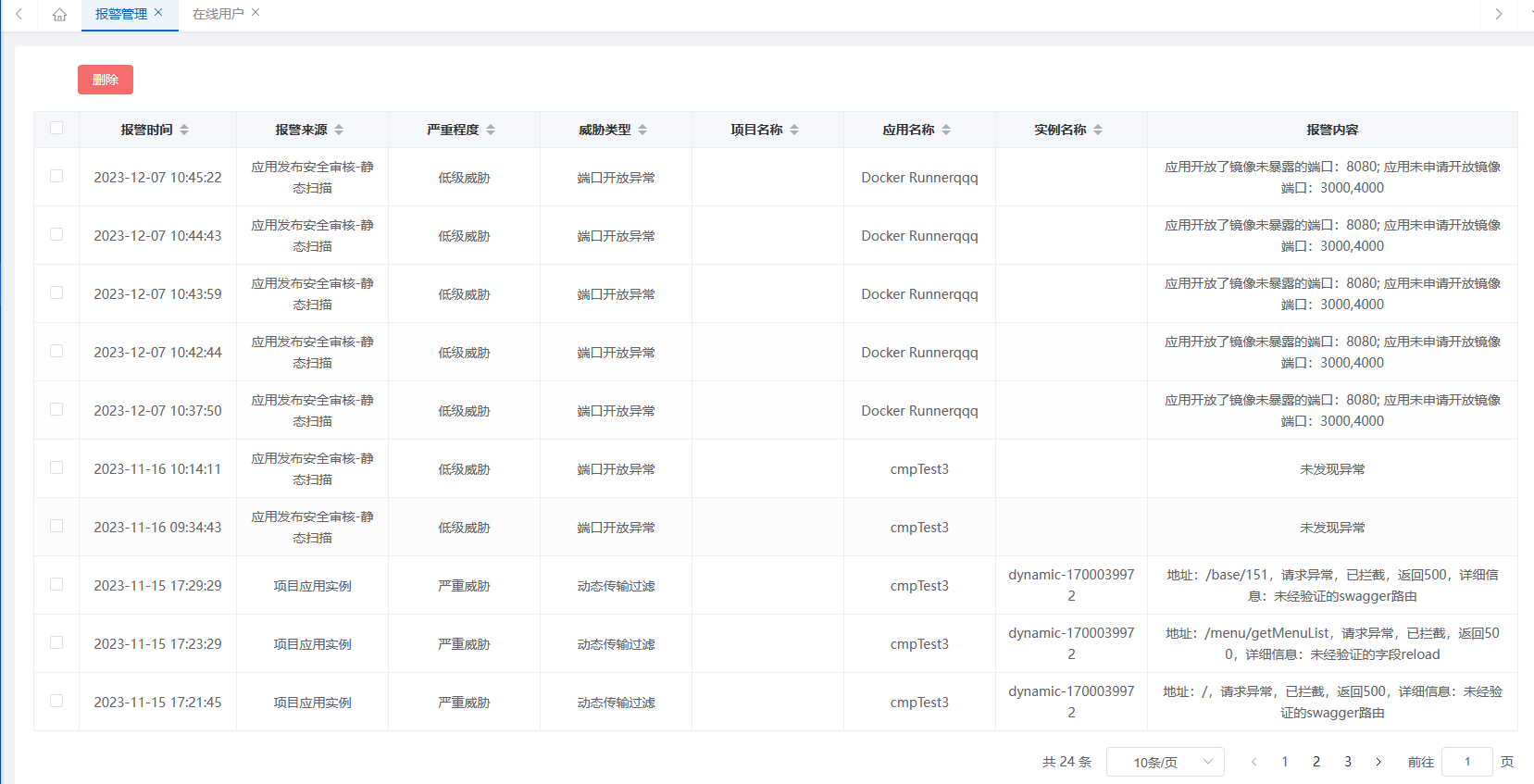
|  |
| --- |
| **程序流程中涉及的数据库表：<表名：rsdp\_alarm** **>**  **操作：<查询>** |
| SQL语句：  【SQL01】查询报警信息列表  SELECT  id,  content,  level,  category,  project\_name,  app\_name,  instance\_id,  instance\_name,  source,  creator,  create\_date,  updator,  update\_date  FROM rsdp\_alarm; |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_alarm **>**  **操作：<插入>** |
| SQL语句：  【SQL02】插入报警消息  INSERT INTO rsdp\_alarm (id, content, level, category, project\_name, app\_name, instance\_id, instance\_name, source)  VALUES  (#{id},  #{content},  #{level},  #{category},  #{projectName},  #{appName},  #{instanceId},  #{instanceName},  #{source}  );  【SQL03】更新报警消息  UPDATE rsdp\_alarm SET  content=#{content},  level=#{level},  category=#{category},  project\_name=#{projectName},  app\_name=#{appName},  instance\_id=#{instanceId},  instance\_name=#{instanceName},  source=#{source}  WHERE id=#{id}; |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 报警id |  | | content | 报警内容 |  | | level | 报警级别 |  | | category | 威胁类型 |  | | project\_name | 项目名称 |  | | app\_name | 应用名称 |  | | instance\_id | 实例id |  | | instance\_name | 实例名称 |  | | source | 报警来源 |  | |
|  |
| **程序流程中涉及的数据库表：<表名：**rsdp\_alarm **>**  **操作：<删除>** |
| SQL语句：  【SQL04】删除报警消息  DELETE FROM rsdp\_alarm WHERE id=#{id}; |
| 关键字段赋值说明：   |  |  |  | | --- | --- | --- | | 字段名 | 赋值 | 备注 | | id | 报警id |  | |

##### 总体设计方案

###### 设计方案

安全监控子系统识别到异常情况后，通过调用报警接口将构建好的报警信息内容保存到系统数据库中，报警模块将根据报警内容的严重级别推送不同的报警通知。

###### 处理流程



1.获取报警列表

用户调用报警管理页面，页面调用【7.5.6.4.1.1.1.2.1.获取报警消息列表】接口执行【SQL01】获取报警消息。接口生成当前历史所有报警消息数据返回前端页面。

2.删除报警消息

用户勾选需要移除的报警消息，点击页面左上角【删除】按钮，在弹出的确认框中点击【确定】后，页面将调用【7.5.6.4.1.1.1.2.4.删除报警消息】接口执行【SQL04】尝试删除所选的所有报警消息。

报警消息状态切换逻辑：

1）安全监控系统识别到威胁，并生成报警消息内容，调用【7.5.6.4.1.1.1.2.2. 报警消息保存】接口执行【SQL02】保存报警消息到系统中；

2）用户查看页面，调用【7.5.6.4.1.1.1.2.1.获取报警消息列表】接口执行【SQL01】获取所有的报警信息；

3）用户勾选报警消息，调用【7.5.6.4.1.1.1.2.4.删除报警消息】接口执行【SQL04】删除所选中的报警消息。