##### 基础配置交互注意点

目录

[行政区域 2](#_Toc483229134)

[行政单位 2](#_Toc483229135)

[用户管理 2](#_Toc483229136)

[增加修改用户 2](#_Toc483229137)

[关于人员表和用户表的保存 2](#_Toc483229138)

[关于用户时间的保存格式(包括以后所有的时间保存格式) 3](#_Toc483229139)

[服务器管理 3](#_Toc483229140)

[任务管理 3](#_Toc483229141)

[1. Task\_Type tinyint 任务类型 3](#_Toc483229142)

[2. F\_Time\_Template\_ID 任务运行模板 3](#_Toc483229143)

[3.F\_Camera\_ID抓拍相机ID 3](#_Toc483229144)

[4. F\_Audit\_User 审核人员 4](#_Toc483229145)

[5. F\_User\_ID 布控用户 4](#_Toc483229146)

[人脸库管理 4](#_Toc483229147)

[主界面展示 4](#_Toc483229148)

[增加修改操作 4](#_Toc483229149)

[人脸库存在上下级 4](#_Toc483229150)

[删除人脸库操作时 4](#_Toc483229151)

[人脸库返回的错误码 5](#_Toc483229152)

[登录操作 5](#_Toc483229153)

[公共部分 5](#_Toc483229154)

[错误码 5](#_Toc483229155)

[分页控件 6](#_Toc483229156)

# 行政区域

增改的弹出框中。初始以主界面界面树结点选中的树结点为主显示。

并且在弹出框中还可以动态选择行政区域。

# 行政单位

注意点和行政区域相同

# 用户管理

## 增加修改用户

1.增改弹出框中，初始选择的行政区域以主界面树节点上的行政区域为主，默认

2.行政单位下拉框选择 需要等行政区域下拉树加载完成之后，并选中一个行政区域结点后再去后台获取对应行政区域的行政单位(调用存储过程)

3.进行保存的AreaID和UnitID: 实际保存的AreaID以选中的UnitID的上一级的AreaID为准，而不是以界面上的下拉框中选择的行政区域结点的AreaID为准。（因为在显示过程中，列出的行政单位是查询出的当前选择的行政区域及其子区域下的所有行政单位，所以会出现选中的行政单位的直接父级不是界面上看见的行政区域的可能。）

## 关于人员表和用户表的保存

修改操作时，用户的Uid字段不能被修改

由于后台把事务放在了dao层, 所以nodejs控制不了事务。

故user表和person表保存失败的情况不考虑事务的回滚

所以对于用户信息的保存，顺序为：

增加的时候:

1. 判断Uid是否重复
2. 增加person表，获取personid
3. 增加user表
4. 返回

修改的时候:

1. 通过user表id取得user对象(没取到则出错), 通过person表id取得person对象(没取到则出错)
2. 将前端传来的值依次赋值给user表查到的数据和person表查到的数据中
3. 保存user表和person表

## 关于用户时间的保存格式(包括以后所有的时间保存格式)

前端在保存的时候就将数据进行转换, 转换后的格式为 2017-05-05 10:57:30

即 hhhh-MM-dd HH:mm:ss

# 服务器管理

不填行政单位，只填行政区域

增改弹出框中，行政单位字段暂时不需要填写(置空)。

行政区域的下拉框选择方式与行政区域相同

# 任务管理

字段交互来源：

## Task\_Type tinyint 任务类型

第一步 枚举类型（0 人脸布控 1 人像捉拍） **选择按钮** 获得

## F\_Time\_Template\_ID 任务运行模板

t\_time\_template时间模板 表中id

第一步 运行计划获取 已存在时间模板列表 供 **选择** 获得

## 3.F\_Camera\_ID抓拍相机ID

第二步 获取库资源列表 获取 摄像机资源列表 （联动选择，单选）

## F\_Audit\_User 审核人员

问题： 保存user 表中ID 还是 person 表中ID

查列表 需要 person 中的 name属性，若保存 userID 查列表需同时查user

## F\_User\_ID 布控用户

暂无登录 ，先写死

## 第三步: 联动动作

客户端接收报警: 对应保存字段 SendAlarmToClientActionExt.java 对应数据库 t\_event\_rule\_action

发送短信:

声光报警器:

推送给消息服务:

编辑 人像抓拍、人像布控 过程中 上一步 下一步的 数据切换 数据独立记忆，

# 人脸库管理

## 主界面展示

左侧区域树，右侧人脸库列表

## 增加修改操作

直接操作数据库，对应的数据库为人像比对系统中的t\_business\_lib,(文档还没更新，直接进10.0.8.62上查看数据库snfds里的信息)

### 人脸库存在上下级

所以在增加和修改操作时，添加人脸库是可以选择其上级人脸库的，若没有上级库则是选择其上级人脸区域(由于树形结构未定，故暂不管上级区域或者库的选择字段，默认以添加时候选择的上级区域或者库为准)

## 删除人脸库操作时

在删除人脸库时，必须先判断当前人脸库是否在运行。

判断条件需要从nodejs中通过人脸库id查询snfds中的t\_object\_relation表。

人脸库id在表中对应的字段为F\_Related\_Object\_ID。

判断表中的任务的ObjectType是否为StructTask(结构化任务)枚举类型

若存在RelatedObjectID等于人脸库id，且ObjectType为StructTask时，不允许删除人脸库，返回给界面删除错误的提示。

## 人脸库返回的错误码

UnknowErr("unknown error", Integer.valueOf(500)),

Success("successful", Integer.valueOf(200)),

NoSuchMethodErr("this command not define methond", Integer.valueOf(31001)),

CommandInvokeErr("commond invoke error", Integer.valueOf(31002)),

SessionTagIsNull("session tag is null", Integer.valueOf(32001)),

NotRegCallback("not register callbak", Integer.valueOf(32002)),

ParamDeserializeErr("parameter deserialize error", Integer.valueOf(32003)),

RetrievalParamIsNull("retrieval parameter is null", Integer.valueOf(32004)),

TaskIdIsNull("task id is null", Integer.valueOf(32005)),

TaskSubmitErr("task submit error", Integer.valueOf(33001)),

TaskNotExist("task not exist", Integer.valueOf(33002)),

OffLine("engine offline error", Integer.valueOf(33003));

# 登录操作

登录操作直接在前端client中直接访问java后台的tomcat，不经过nodejs了。注销和刷新心跳也是。

# 公共部分

## 错误码

错误码定义：基础配置服务错误码1开头，例如11001，11002，12001，NodeJS2开头，例如21001，21002，22001,，引擎3开头，例如31001，31002，32001，人像监控比对4开头，例如41001，41002，42001

## 分页控件



点击最左边的下拉选择pageSize，选择完成后，立即刷新页面,从第一页开始按照当前选中的pageSize来查询数据。