

生产线信息管理系统

基于数据库对条码数据进行管理与产线配套 条码数据管理系统

系统概述

本系统为BS模式,客户端无需部署,直接使用浏览器操作.与其他模块的通讯使用api接口进行,可以和其他平台/模块进行网络通讯.部署的服务器平台为Ubuntu.平台本身由Python编写,配合MongoDB+数据库.开发团队经验丰富: 有过数十个Python+MongoDB的项目开发经验. 总结出一套专用的开发模式,并为此开发了自主知识产权的数据库持久化套件(ORM).

为什么要选择Python语言的作为系统开发语言?

- **可以很方便的调用其他语言书写的代码:** Python语言被称为胶水语言.由于Python很容易的可以调用其他语言的代码/程序,所以可以使用Python语言把不同语言写的不同的代码段粘和起来运行.这个非常有用,比如可以在把代码种对性能要求比较高的地方用C/C++语言书写代码,把需要统计数据的模块用R语言书写,把做文本处理的功能用Perl语言书写.然后使用Python把这些代码段组装起来(基本上都是一两行代码就解决了).
- **Python语言有广泛的软硬件支持:** Python可以运行在Windows, Linux, Unix, Mac, Android, 嵌入式等各种系统环境.并可对系统进行各种比如定时关机/重启,操作键盘/鼠标,打开/关闭某个功能,调整系统等自动化的操作.硬件支持方面,无论是网络模块, 还是智能硬件, 树莓派, 甚至各种专用的工业设备,Python都可以很便捷的进行操作.下面是一个Python通过以太网获取PLC设备的数据的例子.

```
# 设置PLC的连接地址
ip = '192.168.18.17' # PLC的ip地址
rack = 0 # 机架号
slot = 2 # 插槽号
tcpport = 102 # TCP端口号

# 建立连接
client = snap7.client.Client()
client.connect(ip, rack, slot, tcpport)

# 断开连接
client.disconnect()
client.destroy()

areas = ADict({
    'PE': 0x81, # Inputs
    'PA': 0x82, # Outputs
    'MK': 0x83, # Merkers
    'DB': 0x84, # Timers
    'CT': 0x1C, # Counters
    'TM': 0x1D, # Timers
})
```

- **Python在人工智能领域处于领导地位,** 随着时代的发展,AI技术越来越深的渗入到人们的生活和企业的生产环境中.选择Python作为开发语言,可以最大限度的为将来植入人工智能的模块打下环境支持的基础.
- **开发速度快,** Python语法很简洁,这也代表着实现同样的功能.Python代码量少.开发速度快.面对需求的变更能够快速的作出修改.
- **我们非常熟悉Python,** 我们的团队的技术人员,在国内属于对Python语言接触的最早的那一批.都有多年的实践经验,使用Python语言开发过大小60+个项目.Python开发经验非常丰富.

我们为什么选择MongoDB作为数据库?

和绝大多数开发团队一样,我们早期也是依赖于关系型数据库(MySQL/MSSQL等)进行开发.但随着时代的发展.数据量越来越大(上亿条记录的表很常见).数据类型越来越复杂(GEO类型,嵌套字典类型,纯对象类型等等),处理方式越来越多(各种聚合,分析算法层出不穷).为了满足不断变化的客户需求,从2012年,甲骨文开始一步一步将MySQL转向闭源开始,团队就开始尝试新的数据

库.并在2014年开始选中MongoDB作为MySQL的补充.此后的项目,但凡涉及到需要保存大量数据或者快速查询的场合,都会使用MongoDB数据库.性能提升的非常明显.在2018年,MongoDB开始支持事务操作以后.在新的项目中,使用MongoDB担任全部数据的存储功能.和传统的关系型数据库相比,MongoDB带来的改变包括:

- **更快的数据读取/写入.**
- **更多的数据类型** 支持.无论是GEO(地理位置数据),还是复杂的多重嵌套的字典和数组.都能不做类型转换的完美保存(不会像sql数据库那样随意转换数据类型,在强类型语言中,这个问题很容易导致莫名其妙的类型错误)
- **更全面的功能** 支持使用函数, 聚合, 管道等各种技术进行查询.原生为大数据数据和数据分析做好了准备
- **更容易扩展**到大规模的集群.传统的数据库集群要么收费昂贵,要么结构复杂,要么需要额外的框架/中间件支持.MongoDB自带的副本集和分片机制.可以无需额外成本的无限制的提升数据库容量(副本集群)和读写性能(分片机制).并且这个过程完全不用停机.
- **完美备份**,副本集中每一个副本都是数据库的备份. 你可以让数据库一边工作一边备份.这个备份工作完全是透明的,多个备份也不会给系统造成负担,并且这个过程无需人工干预的.无需额外的备份设备.数据备份成本低.安全可靠.
- **完美事务支持** 2018年,MongoDB4+版本已支持跨副本集(群)的事务.这本来是MongoDB的弱项.现在短板也被补齐.以前使用sql+MongoDB工作环境,现在单一的MongoDB数据库就已经完全可以应付了.
- **数据聚合** MongoDB4+自带强大的数据聚合和分析功能.强大的aggregate和map-reduce查询完全可以满足一般的数据聚合和分析的需求.

功能实现

系统所包含的功能:

- **基础信息管理模块**

- 基础信息管理

- **任务管理模块**

- 任务管理

- **条码管理模块**

- 导入导出
- 统计和查询

- **用户管理模块**

- 用户管理
- 权限管理

- **日志模块**

- 系统日志
- 用户日志

基础信息管理

基础信息包含2类

- 企业的基本信息
- 产品的基本信息

两类信息分别使用不同的表.使用外键关联.两者时间是一对多的关系.

基础信息的录入

基础信息的录入可以支持多种方式包括:

1. 管理页面录入: 由操作人员在录入页面手动录入.
2. 文件导入: 使用系统提供的导入功能.利用excel, csv,txt,json,xml等格式的文件导入基础信息
3. 数据接口: 系统可以按照用户情况定制数据的导入接口,直接从用户指定的系统/数据库导入基本信息

基础信息的编辑

用户可以在基础信息的管理页面,查看系统中的基础信息条目.按照权限分配,也可以对基础信息进行编辑,保存和删除工作.

tips: 基础信息的增加,删除和修改的操作都会被记录在用户日志中

任务管理

生产任务的管理模块

任务信息管理

1. 用户可以在任务管理页面添加, 修改,删除任务(需要权限支持)
2. 用户可以在任务管理页查看任务列表的状态和完成情况.双击可显示任务详情.
3. 用户使用组合的条件查询任务信息

任务统计信息

用户这里可以根据预设的条件(日期区间,产品批次等),进行查询.并对查询出来的任务信息进行聚合计算(统计).统计的结果以表格和图表(柱状图,线图,热力图等)两种形式展示出来.

tips: 任务信息的增加,删除和修改的操作都会被记录在系统日志中

条码管理

条码数据的导入和导出

- 导入数据时可以在系统条码数据导入页面设置导入数据文件的格式和字段
- 导出数据的时候,可以按照条件选择需要导出的数据,设定导出的格式和字段.字段可以重新命名,即可以导出为文件格式(转储),也可以指定数据库进行直接导入.

条码比对

通过网络接口/其他方式,接收嵌入式条码比对请求.并将比对的结果返回给嵌入式设备.如果比对成功.则标识比对成功的条码.否则把本次记录的信息写入异常日志.

- 通讯方式目前考虑的是采用网络通讯的方式
- 条码对比的规则可自行定义.
- 返回结果的字段格式可在管理页面自定义

统计和查询

- 已用/未用条码 统计信息,可以用图表(柱状图等)和表格的方式同时显示
- 根据指定的条件,查询相关条码的状态,包括已用的和未用的条码

入库文件处理

嵌入式设备管理

- 扫描或者手动添加嵌入式设备.
- 设定系统和嵌入式设备的连接时的密钥,密钥可以简单的采用固定的密钥对或者复杂的动态密码
- 其他的查看/修改参数的操作.

处理入库文件

接收嵌入式设备推送过来的入库文件. 并按照指定的格式写入数据库.

收据接收方式 可以考虑采用:

1. 嵌入式设备直接向系统推送入库文件.
2. 系统向嵌入式设备发起获取入库文件的请求,嵌入式文件再把入库文件返回给系统

无论那一种方式,都需要先进行安全验证,通过后才能进行通讯.

获取的入库文件,通过解析后,保存在系统中.用户可以指定一个格式对解析后的入库文件进行格式化处理.然后再进行数据库的写入或者转储工作.

用户管理

用户管理包含用户信息的增删改查,和权限的修改2大部分

权限管理

用户的权限是RBAC(Role-Based Access Control 基于角色/权限组的权限管理)模式的设计.用户需要先建立不同的权限组(比如管理员, 操作员等),然后把系统不同模块的访问权限赋予这些权限组.最后把用户加入权限组即可

用户信息管理

用户基本信息的增删改查和所在权限组的编辑.可以在这里把用户加入对应的权限组,这样用户就会拥有权限组对应的权限

tips: 任务信息的增加,删除和修改的操作都会被记录在用户日志中

日志

日志模块负责:

1. 记录系统运行时的信息,包括系统环境, 设置参数, 运行错误信息等.
2. 记录用户的操作信息

日志信息分**系统日志**和**用户/操作日志**两大部分,用户可以按照指定的条件去查询相关信息.比如使用 用户名+时间 的信息可以在用户日志中查看某用户的某个时间进行过哪些操作.