

# 涂鳌飞的个人简历

CSDN 1,017,004 总访问量

Stars

25

个人网站 笔记

## 个人信息

姓名:	涂鳌飞	性别:	男 	
出生年月:	2000.1.11	现居地:	湖北-武汉	
电话  :	17683866724	邮箱  :	tu_aofei@163.com	

## 教育经历

湖北工程学院 计算机科学与技术 本科

全日制 计算机学院 孝感

## 专业技能

- 熟悉 Java 基础知识，如基本数据类型、异常处理、集合、多线程等，能使用 Java8 的 Stream API、Lambda表达式操作集合类。
- 熟悉 常用并实践过多种 Java 设计模式（如单例、工厂、生成器、门面、注册器、适配器）。
- 熟悉 HTML、JavaScript、jQuery、Vue、Ajax等前台技术，Ajax、axios 进行同步或异步数据交互，掌握前端开发框架，如 Vant, element-ui 的使用。
- 熟悉 Spring MVC、Spring Boot、MyBatis、Hibernate 等主流后端开发框架，熟悉 AOP、IOC 编程、PageHelper 分页插件，能独立开发项目。
- 熟悉 Redis ，掌握 Redission 分布式锁，分布式session 的使用。
- 熟悉 Quartz 作业调度框架。
- 熟悉 Mysql, Oracle 数据库及库表设计，能够通过创建索引、Explain 分析、SQL语句优化等方式优化性能。
- 熟悉 常见业务开发场景：比如nginx反向代理、全局跨域解决、多环境问题解决，api签名认证，springboot starter SDK等。

- 熟悉 Linux 运维基本命令使用，如 find、tar、ps、netstat、top 等命令，有排查 cpu 占用过高、磁盘空间不足导致应用无法正常访问的经验。
- 掌握 Git、SVN 版本控制工具。
- 掌握 常用消息队列的基本使用（Kafka）。
- 了解 Spring Cloud、Dubbo 和 ZooKeeper 等分布式技术。熟悉各大组件的使用，如ZooKeeper注册中心，Dubbo远程服务调用、Gateway网关。
- 了解 Elasticsearch 搜索引擎，能用ElasticSearch实现分词搜索，使用Logstash、定时任务实现ElasticSearch和数据库进行数据同步。

## 工作经历

### 上海威士顿信息科技股份有限公司

2021.12.01 - 2024.12.20

智能制造事业部 Java开发工程师 武汉

负责制造执行系统（MES）的开发与维护，专注于生产消耗、库存和质量管控等关键业务流程。

负责统计过程控制（SPC）的开发与维护，监控生产过程中的质量参数，实现实时质量控制。

负责厂区智能安防项目的开发与维护，主要对接海康威视平台，实现门禁控制、防区报警和入侵检测等功能。

根据需求文档进行分析设计，完成业务功能开发、单元测试、缺陷修复和维护工作。

## 项目经历

### 武汉卷烟厂 MES 系统

#### 软件架构

spring + Hibernate + oracle + mybatis + redis + Kafka + Quartz + Stimulsoft报表

#### 项目描述

用于在制造过程中监控和控制车间的生产流程的制造执行管理系统，以提高生产效率和产品质量。MES系统通过集成生产设备、收集生产数据、监控生产过程，实现了生产流程的自动化和智能化。

## 责任描述

主要参与生产消耗、库存和质量管控等关键业务流程的开发和运维。

参与项目的需求分析和功能设计，确保系统功能满足生产需求。

与客户保持沟通，收集反馈并持续优化系统功能。

## 技术描述

- 防止多个用户操作同一数据造成数据不一致的问题，使用 `dataVersion` 实现乐观锁机制，更新数据库时对比版本，保证数据一致性。
- 多数据源实现，通过实现 `spring AbstractRoutingDataSource`，不同业务模块使用独立数据源，降低系统耦合度。
- `webservice`和`restful`接口日志，统一接收接口路径前缀，使用设计模式设计抽象类接收器，抽取公共方法，不同类型的接口具体实现不同的处理器，提升了接口的可维护性和扩展性，防止接口失败可以重放请求，重新发送或接收。
- 通过扩展 `service` 方法，实现可配置的 `callback` 机制，支持在方法执行前后动态配置和调用业务逻辑（如发送接口、消息提醒等），并可灵活设置 `callback` 的执行方式（同步/异步）和异常处理策略，降低系统耦合度，提升了程序可扩展性。
- 定时任务配置,项目启动时，获取定时任务配置信息（包含执行方法，上下文信息...），构建`Quartz JobDetail`,添加到`scheduler`任务调度器，执行定时任务,并通过分布式锁保证多机部署时定时任务不会重复执行。
- 系统中用到的魔法值，可维护`code`和`name`的对应关系，不用在代码里写死；自定义注解，在`po`实体转`dto`时，通过反射自动填充编码对应的名称，集中管理，提高程序的可维护性和扩展性，提高了接近1倍的开发效率。
- 结合`Stimulsoft`报表工具，使用生成器模式构建复杂的报表对象，使用线程池 `ThreadPoolExecutor`并行查询数据，优化生成速度，生成速度提升了30%以上。
- 为实现可搜索上传文件内容，使用`elasticsearch`替代数据库的模糊搜索，使用动静分离策略创建文章索引，只在`es`中存储要搜索的字段、修改不频繁的字段（如文件名，文件内容、文件实体`id`），其它信息可关联数据库查询；并使用`ik`分词器实现更灵活的分词查询，减少`es`数据更新同步的成本、保证数据一致性，相比于传统模糊搜索搜索效率提升70%以上。

## 武汉卷烟厂 SPC 系统

### 软件架构

spring + oracle + redis + Kafka + Wonderware时序数据库 + websocket

### 项目描述

利用统计分析方法监控生产过程，确保产品质量的过程质量控制系统。用于实时监控生产过程中的各类与质量管控相关的数据，并在线进行统计、分析和展示，及时准确发现生产过程的异常波动，并指导生产及时采取纠正和调整措施，保障生产过程稳态。

### 责任描述

根据需求文档，对系统进行优化和开发

对系统进行运维，保证系统的正常运行

### 技术描述

- 通过配置需要采集的数据信息，根据工单或批次来通过查询实时数据库来获取数据采集信息并实时计算控制图，后端通过kafka获取数据进行消费处理，解决大规模实时数据的计算和处理问题，通过消息队列实现了数据处理的解耦和削峰，缓解了高并发情况下的系统资源短缺问题。
- 实时监控页面通过websocket和后端建立长链接，后台处理完成的业务数据实时推送给web端，解决了生产数据实时展示的及时性问题，提供了更好的用户体验和实时交互能力。
- 自定义线程+处理队列实现单个数据采集点位的数据处理，可通过线程标识控制线程启停和监控，解决了数据处理的并发控制，实现了采集任务的管理和监控，支持动态调整和异常处理，实现对单个数采点的管控。

## 武汉卷烟厂智能安防平台

### 软件架构

spring boot + Feigin + nacos + redis + oracle + Hibernate + mybatis + apache poi

### 项目描述

基于海康威视设备的智能安防管理平台，实现报警信息实时接收、智能分级、自动处置等功能。采用微服务架构，支持百万级报警数据处理。

### 责任描述

负责报警信息接收模块开发，实现毫秒级报警数据接入和处理

设计并实现规则引擎，支持灵活配置报警规则、多级报警分发和升级和联动策略

开发设备管理模块，实现设备状态监控和远程控制

优化系统性能，提升系统并发处理能力

## 技术描述

- 接收海康报警信息采用设计模式订阅-消费模式、模板方法模式和 `LinkedBlockingQueue` 阻塞队列实现事件调度，可灵活扩展接入不同类型的报警信息，提高程序可扩展性。
- 基于海康 `openapi` 实现监控点实时监控，通过视频流地址动态获取和用户鉴权实现各个监控点位的实时监控。
- 基于责任链模式的规则匹配，对各个场景如重点人员识别、离岗检测、玩手机等场景根据不同的报警规则进行插件化的动作处理机制，支持规则热更新。
- 线程池管理优化并发处理，对数据量大的报警信息表等采用分表处理，优化数据查询速度。

## API接口开放平台

### 软件架构

`spring boot + Dubbo + Spring Cloud Gateway`

### 项目描述

基于 `Vue + Spring Boot + Dubbo + Gateway` 的 API 接口开放调用平台。管理员可以接入并发布接口，可视化各接口调用情况；用户可以开通接口调用权限、浏览接口及在线调试，并通过客户端 SDK 轻松调用接口。

### 技术描述

- 基于 `MyBatis Plus` 框架的 `QueryWrapper` 实现对 `MySQL` 数据库的灵活查询，并配合 `MyBatis X` 插件自动生成后端 `CRUD` 基础代码，减少重复工作。
- 为防止接口被恶意调用，设计API签名认证算法，为用户分配唯一AK/SK用于鉴权，保障接口调用的安全性。
- 前后端使用 `Swagger + Knife4j` 自动生成 `OpenAPI` 规范的接口文档，前端在此基础上使用插件自动生成接口请求代码，降低前后端协作成本。
- 为解决开发者调用成本过高的问题，基于 `Spring Boot Starter` 开发了客户端 SDK，一行代码即可调用接口，提高开发体验。
- 选用 `Spring Cloud Gateway` 作为 API 网关，实现了路由转发、访问控制、流量染色，并集中处理签名校验、请求参数校验、接口调用统计等业务逻辑，提高安全性的同时、便于系统开发维护。

# 自我评价

- 注重代码可读性，可维护性。
- 喜欢写技术博客，介绍自己在日常编写代码时遇到的问题，善于帮助他人。
- 良好的学习能力，对学习有一份执着与恒心，敢于尝试新鲜事物，并从中提升自己。
- 有较强的问的解决的能力，能够利用GitHub Issues区、AI工具、搜索引擎、StackOverflow等自主解决问题。
- 具有较强的团队精神，有强烈的集体荣誉感。
- 喜欢钻研新技术，敢于面对和克服困难。