

文章编号: 2095-2163(2020)07-0082-05

中图分类号: TP311.5

文献标志码: A

基于前后端分离的防伪溯源系统

万燕, 张博锐

(东华大学 计算机科学与技术学院, 上海 201620)

摘要: 针对产品在生产、运输、市场监察和销售过程中假货猖獗、溯源困难的问题, 本文研究和设计了基于前后端分离的防伪溯源系统。系统以动态二维码技术为基础, 结合当前的主流技术 Vue 与 Spring Cloud 框架, 使用前后端分离的开发模式, 降低了系统的耦合度, 提升系统的扩展性, 并利用 RESTful 接口进行数据交互和微服务模块化的设计, 为系统的性能和扩展提供了保障。该系统为性能要求较高、安全性要求较好的企业提供了一个长期可用、可维护的防伪溯源平台。

关键词: 防伪溯源; 动态二维码; Spring Cloud; 前后端分离; 微服务

Anti-counterfeiting traceability system based on front-back stage decoupling

WAN Yan, ZHANG Borui

(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

[Abstract] In order to solve the problem of rampant fakes and difficult traceability in the process of the production, transportation, market supervision and sales of products, an anti-counterfeiting traceability system based on front-back stage decoupling is studied and designed. This system is based on the dynamic two-dimensional code technology, combined with the current mainstream technology Vue and Spring Cloud framework, and uses a development model of front-back stage decoupling to reduce the coupling of the system, and improves the scalability of the system. The system uses RESTful interface for data interaction and micro-service modular design, which provides guarantee for the performance and expansibility of the system. The system provides a long-term and maintainable anti-counterfeiting traceability platform for enterprises with high performance requirements and good security requirements.

[Key words] Anti-counterfeiting traceability; Dynamic two-dimensional code; Spring Cloud; Front-back stage decoupling; Micro-service

0 引言

产品防伪溯源体系是为了打击假冒伪劣产品, 并在防伪的同时能够监控产品的物料、生产、仓储、运输信息的一个整体的质量安全体系^[1]。随着科学技术的发展, 产品质量问题已经成为影响人民生命财产安全的一大因素。从前些年的三鹿奶粉事件, 到近些年的假疫苗事件, 再到目前的疫情事件中, 可以看出: 如果能建立一个安全可靠的防伪溯源体系, 那么从出问题的产品事件中就能快速定位解决问题的关键环节, 节省大量的资源 and 时间。2016年国务院发布了《国务院办公厅关于加快推进重要产品追溯体系建设的意见》, 提出了“加快应用现代信息技术建设重要产品追溯体系”。

目前的防伪溯源系统虽然能实现产品的防伪溯源, 但其系统扩展性较差, 耦合度较高, 企业后续的使用与维护成本较高, 在规模较大的企业级应用中的性能和安全性较差。本文设计的防伪溯源体系基

于 Spring Cloud 和 Vue 结合的技术进行微服务系统开发, 并使用前后端分离开发模式, 很好地解决了这些问题。

1 技术分析

1.1 动态二维码技术

二维码/二维条码是在平面上有特定规律的, 记录数据信息的黑白相间的图形, 在信息的采集、存储、查询方面有广泛的应用。与传统的 NFC、一维条码相比, 二维码具有容错性高、成本低廉、信息容量大等技术特点, 在防伪溯源体系中有着独特的优势^[2]。本系统首先随机生成一串唯一编码, 作为查询更新产品信息的入口, 采用 RSA 非对称加密的技术, 由公钥加密编码, 并生成二维码, 客户端扫描并上传二维码携带的信息, 访问后台服务器, 由企业持有的私钥进行解码, 保证数据的安全性, 达到验证防伪的功能^[3]。

静态二维码只能存储静态不变的信息, 而客户端用户可以通过扫描动态二维码, 动态地查看产品

作者简介: 万燕(1970-), 女, 博士, 教授, 主要研究方向: 图像处理与模式识别; 张博锐(1995-), 男, 硕士研究生, 主要研究方向: 图像处理。

通讯作者: 张博锐 Email: 928892363@qq.com

收稿日期: 2020-04-12

当前的最新状态, 确保从生产源头到当前阶段的所有流程信息可以追溯查看。

1.2 后端 Spring Cloud 技术

本系统后端采用成熟的微服务架构方案 Spring Cloud 进行项目开发与实施, 可提高开发效率, 降低模块间的耦合度, 降低后期维护成本。在实际应用场景下, 基于项目整体性能针对性地调整服务层机器的数量, 达到经济与效率的平衡。项目整体架构清晰, 多个微服务之间相对独立, 且每个微服务存在备用服务, 风险可控。

微服务是一种新兴的软件架构模式, 是将一个大型的单体应用拆分为多个小的服务^[4]。而 Spring Cloud 是基于 Java 的快速开发框架 Spring Boot 的一个框架, 可以快速构建分布式系统, 降低微服务系统实现的要求^[5]。例如: 配置管理、服务注册与发现、服务熔断、智能路由、服务代理等都可以通过 Spring Cloud 简单快速地完成配置、实现与部署^[6]。本系统主要使用了其中的 Config 配置中心、Eureka 注册中心、Zuul 网关、Feign 动态代理、Ribbon 负载均衡以及 Hystrix 服务熔断等功能, 如图 1 所示。

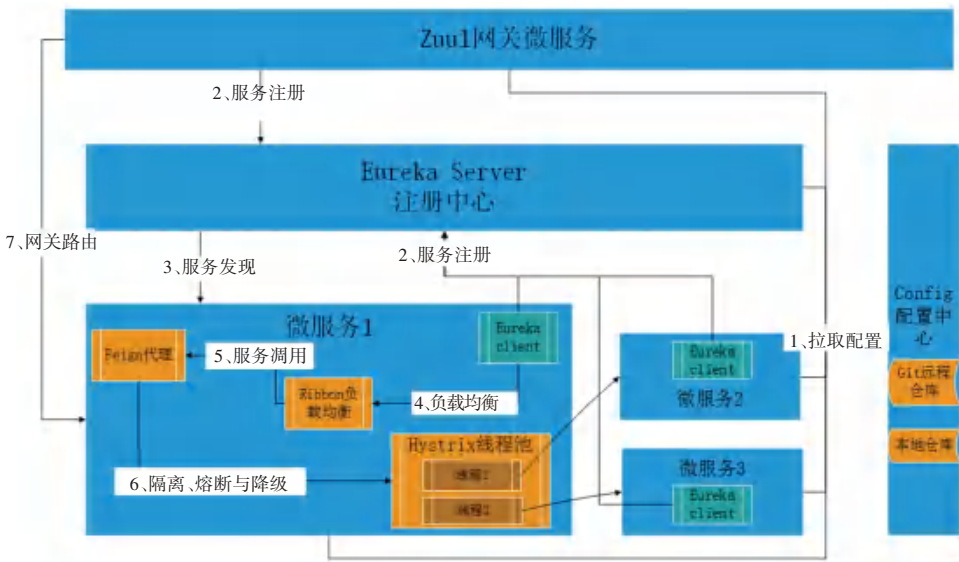


图 1 Spring Cloud 各个组件

Fig. 1 Components of Spring Cloud

配置中心 Config 服务端模块, 主要为分布式系统中不同的微服务模块的配置提供支持。从系统开发到测试, 再到生产部署, 都可以统一管理。无论是从远程仓库, 如 Git 端拉取配置实时更新, 还是将配置放在本地的服务端配置中, 在开发以及后期的维护中都是十分便利的。

Spring Cloud Eureka 是微服务架构中最为基础的模块, 主要功能是服务治理, 用来实现各个微服务的自动注册和服务发现^[7]。由于在微服务项目中, 各个服务需要降低模块之间的耦合度, 相对独立, 服务间的调用并不友好。在各个服务作为 Eureka 客户端注册到 Eureka 服务器注册中心后, 当某一个服务需要调用其他服务时, 只需要向注册中心发送请求, 获取需要调用的服务地址再去访问即可。不需要每个服务静态都保留其他服务的地址, 只由服务中心 Eureka 监听每个服务的地址, 可以进一步降低耦合。

在从注册中心获得需要调用的服务地址之后, Spring Cloud 另一核心组件 Feign 会为每个微服务创建动态代理。当其他请求发起调用时, Feign 的动态代理则会根据具体的代码构造出需要访问的服务地址, 解析响应。

Ribbon 是一个基于 HTTP 和 TCP 的客户端, 负载均衡工具, 提供了适应于不同应用场景的不同负载均衡算法。Ribbon 首先访问 Eureka 注册中心, 获取需要访问的服务的地址, 采取预设的负载均衡算法, 挑选出一个地址, 再由 Feign 动态代理构造请求路径。

在微服务场景中, 一个系统会有很多层的服务调用, 如果某些常用服务崩溃, 不能及时处理, 就会造成服务雪崩的问题。当某个服务不可用时, Spring Cloud 中的断路器 Hystrix 能够阻断雪崩的传递, 保障整个系统稳定性。它为每个微服务都创建一个处理请求的线程池, 当出现问题时, 可以及时做到服务

降级、熔断和隔离,减少或者阻断对出现问题的微服务的请求。

考虑到系统的安全性以及对外暴露的统一管理,系统选择微服务网关 Zuul 组件负责服务路由。不管是来自于前端或移动端的请求,还是服务间的相互调用,一切对服务的请求都会经过 Zuul 网关,由其实现权限验证、限流、动态路由等^[8]。Zuul 网关是微服务后台对外暴露的统一入口,对保护系统的安全起到至关重要的作用。

1.3 前端 Vue 与后端分离

传统的开发模式是由模型(Model,数据)、视图(View,页面)、控制层(Controller,事件交互,承上启下)组成的 MVC 模式,如图 2 所示。而现在的开发模式则多是由模型(Model,数据)、视图(View,页面)、视图模型(View Model,主要负责数据绑定和事件监听)组成的 MVVM 模式,如图 3 所示。相比较而言,MVC 中的控制层 Controller 换为视图模型 View Model,但视图模型并没有完全取代控制层,它的目的主要是抽取控制层中的业务逻辑,使业务模块可复用。Vue 正是基于这种开发模式设计的一套框架。



图 2 MVC 模式

Fig. 2 MVC mode



图 3 MVVM 模式

Fig. 3 MVVM mode

Vue 是一个渐进式的 JavaScript 框架,随着业务需求的增加,它可以满足不同场景的需求。需要多页面的开发时,可以将 Vue 作为一个模板来使用,也支持作为一个组件化的开发系统。当需要开发一个单页面的应用时,Vue 也提供了一套复杂应用程序所需的高级功能,例如:路由、状态管理和构建工具等。

前后端分离就是将一个应用的前端代码和后端代码分开,独立开发。传统的项目中,前端使用 JSP 开发,而 JSP 不是由后端开发者独立完成的,由前端开发 HTML 静态页面后,交由后端开发者,再改为 JSP 动态页面,这种形式一旦出现问题,需要前后端开发成员一起解决,耦合度很高,开发效率极低。使

用前后端分离的方式进行开发,可以完美的解决这一问题。前后端开发者只需要提前约定好接口文档(URL、参数、数据类型等),分别独立开发,前端可以用假数据进行测试,完全不依赖于后端。后端可以用 Postman 等工具进行测试,前端通过 Ajax 请求访问后端的数据接口,将 Model 展示到 View 中,真正实现了前后端应用的解耦合,极大地提升了开发效率。

2 系统架构设计与实现

通过对防伪溯源系统的分析,开发过程大致分为两部分:后端系统开发与前端页面开发。本文着重描述后端系统设计部分。后端系统架构分为三层:系统安全层、业务服务层和系统服务层。系统架构如图 4 所示。

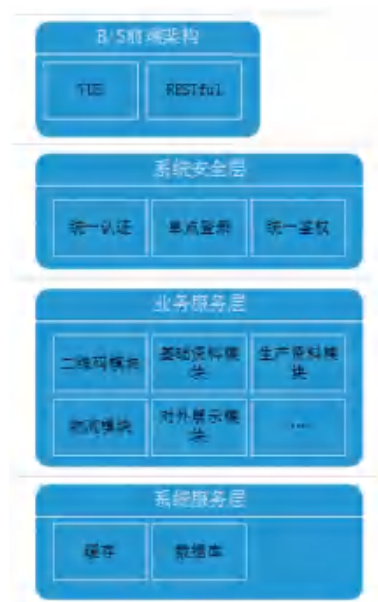


图 4 系统架构

Fig. 4 System architecture

系统安全层对外部接口的调用统一管理,同时集成服务于外部访问的单点登录服务,用户的登录认证时的鉴权认证处理,其他一些包括限流等安全业务也在这一层完成。

业务服务层主要包括各个模块的具体业务代码。最为核心的二维码模块,贯穿在整个防伪溯源系统中,从生产到运输再到监管、消费过程中作为数据交互的入口而存在。数据交互所需要的成员管理、角色分配、权限管理等功能在基础资料模块中实现。生产资料模块提供产品的生产基础信息的记录接口。物流模块负责产品运输过程中的供应商管理、经销商管理、客户管理、库存仓储管理、出入库管理以及订单管理。

系统服务层是由数据库服务以及缓存中间层等组成。数据库选择传统的关系型数据库 MySQL, 能安全持久地存储系统中的所有数据信息, 包括常用与不常用的信息。缓存则是采用非关系型数据库 Redis。业务层不直接与数据库交互, 由缓存 Redis 与数据库交互, 同时会把热点信息保留在缓存中, 不必每次都访问数据库, 减少对数据库系统的压力; 同时, Redis 的访问速度比数据库快, 能够提升整个系统的效率。

通过以上三层架构和几个业务模块实现了防伪溯源系统, 系统功能模块如图 5 所示。

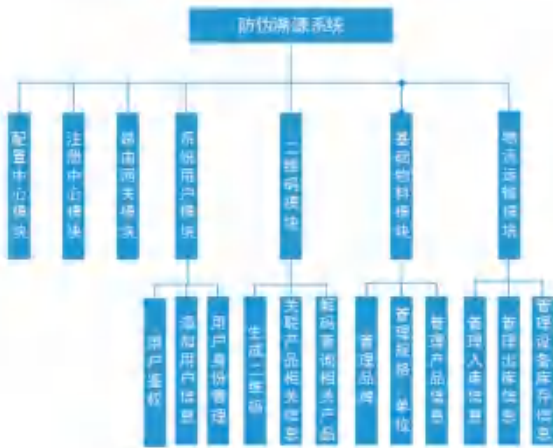


图 5 系统功能模块

Fig. 5 System function module

防伪溯源系统的流程, 如图 6 所示:

- (1) 系统用户在基础资料模块中获取授权, 登录到后台系统;
- (2) 系统用户向系统发起请求, 由系统二维码模块随机产生指定数量的二维码编码;
- (3) 系统用户在生产资料模块中录入产品基本信息, 并与(2)中生成的二维码编码相关联, 生成二维码, 喷涂在产品上;

(4) 在产品存储运输过程中, 由终端的 PDA 扫描二维码, 并由系统用户上传出入库和仓储信息;

(5) 市场监管部门和消费者可以通过扫描产品上的二维码验证产品的真伪, 同时可以了解产品从生产到运输, 再到消费的整个生命周期。

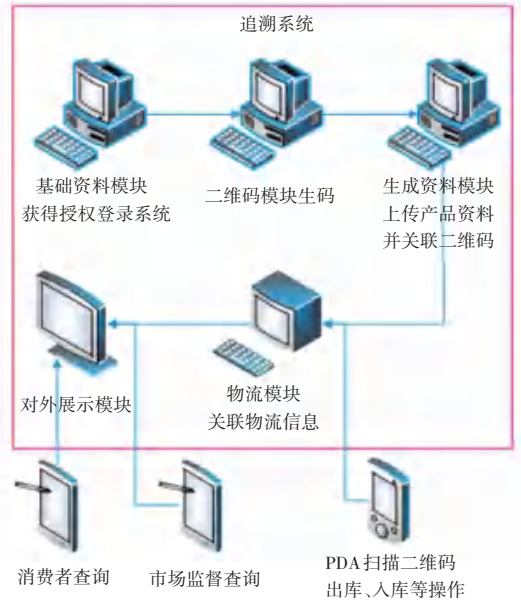


图 6 系统流程图

Fig. 6 System flow chart

3 系统实现

防伪溯源系统由以二维码模块为核心的多个模块组成, 实现产品从生产到运输, 再到市场的整个流程的监管。

在系统中最为关键的是二维码模块, 二维码必须是由系统随机产生不可更改的, 由系统用户生成批量的二维码待用, 生成二维码记录、编码以及图片, 如图 7 所示。生成的二维码在与具体产品关联时, 需要上传具体的产品信息, 再进行关联, 如图 8 所示。



图 7 生成的二维码记录、编码以及图片

Fig. 7 Records, codes and pictures of generated QR code



图8 上传的基本信息与关联的二维码

Fig. 8 Uploaded basic information and associated QR code

产品运输过程中,需要对产品的来源与目的地进行核验,在出库时指明出库代理与接收代理,在入库时从二维码中获得相关信息并审核成功后才能入

库,如图9所示。最后在市场稽查终端或消费者终端扫描二维码,可以确保产品为正品,并获取到整个产品的生产运输流程,如图10所示。

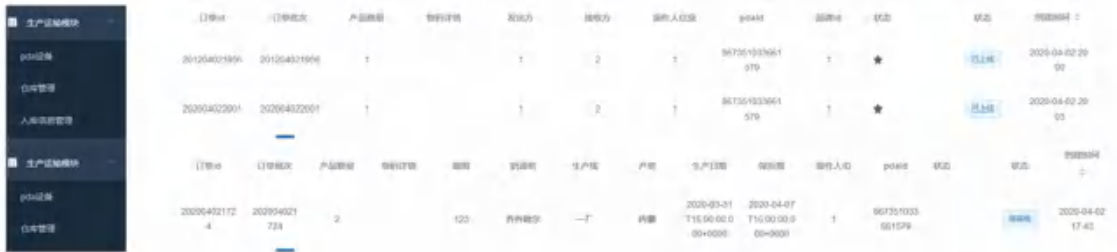


图9 入库、出库信息

Fig. 9 Information storage outbound

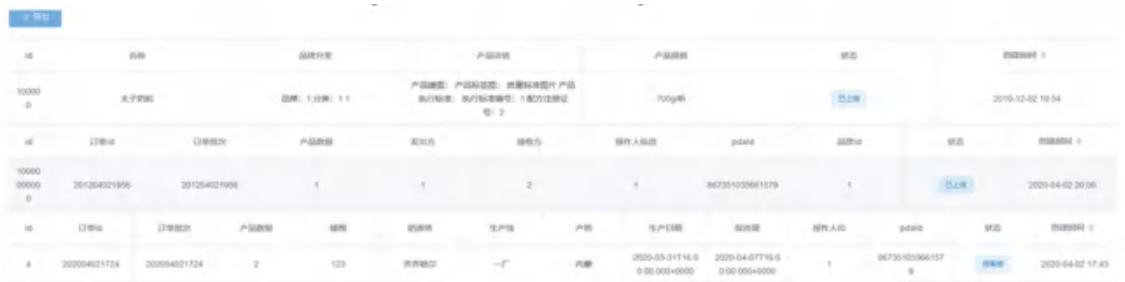


图10 通过二维码查询到的产品相关信息

Fig. 10 Information of the product found by QR code

4 结束语

加快现代信息技术应用建设,重要产品追溯体系是国家一直在推进的目标。本系统是在现有体系的基础上,结合业界前沿技术 Vue 和 Spring Cloud 及设计理念,前后端分离开发的一套从产品生产到运输,再到稽查,市场销售的全流程防伪溯源体系。系统中采用的动态二维码技术和 RSA 非对称加密技术提高了系统信息的安全性;微服务架构的应用与前后端分离的开发模式提高了系统性能的上限;模块化开发提升了系统的扩展性,使得系统的维护与更新更加便利。本系统为性能要求较高、安全性要求较好的企业提供了一个长期可用、可维护的防伪溯源平台。

参考文献

- [1] 王丹. 二维码防伪溯源专利技术综述[J]. 山东工业技术, 2019(9):170.
- [2] 兰龙辉,邱荣祖. 二维码技术在农产品物流追溯系统中的应用[J]. 物流工程与管理, 2013, 35(9): 86-89.
- [3] 章海荣,潘庆雯,程雯. 基于 SSH 框架的农产品溯源系统的设计与实现[J]. 电子设计工程, 2019, 27(9): 47-50.
- [4] 李娜. 基于 Spring Cloud 微服务架构的应用[J]. 电子技术与软件工程, 2019(12): 142.
- [5] 陈建海,陈森,浦云明. 基于微服务架构 B/S 系统的性能分析[J]. 计算机系统应用, 2020, 29(2): 233-237.
- [6] GÖTZ B, SCHEL D, BAUER D, et al. Challenges of Production Microservices[J]. Elsevier B.V., 2018, 67.
- [7] 熊益益. 基于微服务架构的电商平台的研究与实现[D]. 北京邮电大学, 2019.
- [8] 张斌,任富彬,沈炜. 基于 SpringCloud 的食品安全溯源系统的设计与实现[J]. 软件工程, 2019, 22(8): 27-30.