

基于 Spring Cloud 的企业级智慧后勤保障平台^①



李川, 吴钢, 利润霖

(国网信息通信产业集团有限公司, 北京 100052)
通信作者: 李川, E-mail: 18612306230@163.com

摘要: 立足于新时代后勤管理机制与服务保障模式实际需求, 为打造具有业务集约化、应用智能化、数据融合化特征的智慧后勤, 推进后勤工作提质增效, 强化后勤精益管理水平, 实现后勤管理和服务的信息化、数字化、智能化转型升级, 本文基于微服务的实现技术 Spring Cloud 开发了企业级智慧后勤保障平台。此平台通过整合各类后勤资源, 涵盖后勤管理中心、服务保障中心、物业监控中心 3 大中心, 面向企业后勤管理人员、物业人员、员工提供智慧食堂、便捷出入、物业管理等全方位后勤保障服务, 打破传统单体式后勤业务应用系统间壁垒, 实现各类后勤业务应用系统数据贯通, 进一步提升后勤保障能力和管理效率。

关键词: 智慧后勤; 微服务架构; Spring Cloud; 平台化

引用格式: 李川, 吴钢, 利润霖. 基于 Spring Cloud 的企业级智慧后勤保障平台. 计算机系统应用, 2022, 31(6): 141-148. <http://www.c-s-a.org.cn/1003-3254/8500.html>

Enterprise Intelligent Logistics Support Platform Based on Spring Cloud

LI Chuan, WU Gang, LI Run-Lin

(State Grid Information & Telecommunication Group Co. Ltd., Beijing 100052, China)

Abstract: To meet the actual needs of logistics management mechanism and service support mode in the new era, this study develops an enterprise intelligent logistics support platform with Spring Cloud, the implementation technology of microservices. It helps build intelligent logistics with the characteristics of intensive business, intelligent application and data integration, promote the quality and efficiency improvement of logistics work, strengthen the lean management level of logistics, and realize the transformation and upgrading of logistics management and services toward informatization, digitization and intellectualization. By integrating various logistics resources, this platform covers three centers: logistics management center, service support center, and property monitoring center, and provides all-round logistics support services such as smart canteen, convenient access, and property management for logistics managers, property personnel and employees of enterprises. It breaks the barriers between traditional single logistics business application systems, realizes the data connection of various logistics business application systems, and further improves the logistics support capability and management efficiency.

Key words: Intelligent logistics; microservice architecture; Spring Cloud; platformization

后勤保障服务水平是企业后勤工作的有机组成部分, 在保障企业核心业务正常开展、改善员工生产生活条件、展示企业良好社会形象等方面发挥重要作用。

随着企业发展方式转变, 后勤服务领域越来越宽, 工作标准越来越高, 保障作用越来越重要, 对深入推进后勤保障服务标准化、专业化、精益化管理提出新的更高

^① 收稿时间: 2021-08-26; 修改时间: 2021-09-26; 采用时间: 2021-09-29; csa 在线出版时间: 2022-05-26

要求^[1]。

新技术的发展为提高后勤保障服务水平提供了有力支撑,加速新兴业态发展壮大,同时也为传统的服务型企业提升保障能力、改善服务品质、增强客户体验提供了技术支撑。目前企业后勤业务涉及出入管理、车辆管理、餐饮管理等多个业务领域,各业务领域传统单体式应用系统在单位广泛分布、数量众多、独立运行,数据孤岛严重,影响了跨业务应用、跨单位应用业务的有效开展,制约后勤统一运管及保障能力提升。本文基于微服务的实现技术 Spring Cloud 开发企业级智慧后勤保障平台,打破企业各类单体式后勤业务应用系统间壁垒,实现各类后勤业务应用系统数据贯通,为企业提供统一平台、统一入口、统一管理、统一展现的立体后勤服务新模式,助力提升后勤管理、保障、服务的多元化支撑能力。

1 系统设计

1.1 系统架构

系统基于微服务架构,以分层的方式进行系统的设计^[2,3]。系统架构共分为支撑层、数据层、服务层、应用层、客户端,如图1所示。支撑层封装平台公共逻辑组件,提供标准 JSON 格式服务接口,包括消息组件、缓存组件、短信组件等;数据层负责保存后台关键业务数据和数据持久化,数据库按业务垂直拆分,不同业务模块的数据存储到不同的数据库,存储文件数据、结构化数据和历史统计数据等;服务层分为服务治理层、聚合服务层、业务服务层、基础服务层^[4,5]。服务治理层统一入口,完成服务路由、流量管控、监控报警等,接收内部应用层和外部第三方应用所有请求,根据类别转发到对应的聚合服务或业务服务。聚合服务层对业务服务进行封装,汇聚后端众多的微服务逻辑,把服务治理层的 HTTP 请求转发到业务服务层进行业务逻辑处理,以保证服务的低耦合和微服务特性。业务服务层处理后台众多的业务逻辑。基础服务层提供原子性核心业务数据处理能力,为业务服务层提供基础能力支持;应用层包括房产管理、工程管理等 11 个应用,负责客户端接入和访问;客户端分为 PC 端和移动端。

1.2 技术架构

系统技术架构自底向上分为支撑层、存储层、数据层、应用层、客户端,如图2所示。支撑层包括分布

式任务调度 XXL-JOB、消息队列 RocketMQ、工作流 Activiti 等,为整个平台提供共性组件;存储层提供以数据存储为主的能力,关系型数据库 MySQL 存储结构化数据,内存数据库 Redis 提供缓存服务, FastDFS 存储非结构化数据;服务层包括服务治理层、聚合服务层、业务服务层、持久化层。服务治理层服务网关 Gateway 是一种基于 MVC 模式的响应式 Web 框架,提供外部访问的统一入口,外部请求全部通过 Gateway 接入微服务,简化前端调用逻辑。Eureka 用于服务注册与发现,房产管理等微服务作为服务提供者注册到 Eureka Server 中,服务调用者使用 REST 接口调用服务提供者。Hystrix 提供资源隔离、降级机制等功能,避免服务不可用时产生雪崩效应。聚合服务和业务服务利用 Spring Boot 快速构建微服务,通过 Feign 方式进行交互,并向上层提供接口服务。应用层与服务层基于 HTTP 协议通信、JSON 格式数据进行交互。



图1 系统架构

1.3 功能设计

平台通过整合后勤资源,构建涵盖后勤管理中心、服务保障中心、物业监控中心 3 大中心,面对后勤管理人员、物业人员、内部员工等人员提供智慧食堂、便捷出入、出行服务、物业管理等全方位的智慧

后勤服务. 本文设计与开发的企业级智慧后勤保障平台功能结构图如图3所示.

1.3.1 后勤管理中心

(1) 房产管理

对企业办公用房、周转房等后勤房产资源进行可视化、图形化、数据化管理, 基于GIS地图构建企业后勤房产地图, 实现“以图管房”.

(2) 工程管理

通过三维建模和高清视频摄像头等各类物联网感知设备, 对工程项目现场及活动进行识别跟踪, 实现重点项目施工现场实时可视化监控. 建设智慧工地, 对现场数据进行智能化采集、智慧化管理, 推广绿色施工、文明施工, 实现标准化、在线化、可视化、流程化管控.

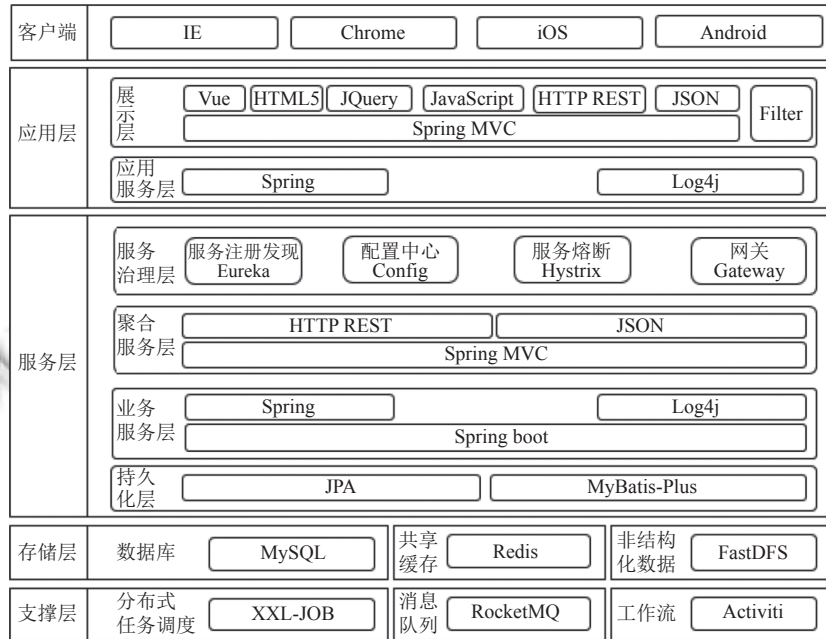


图2 技术架构

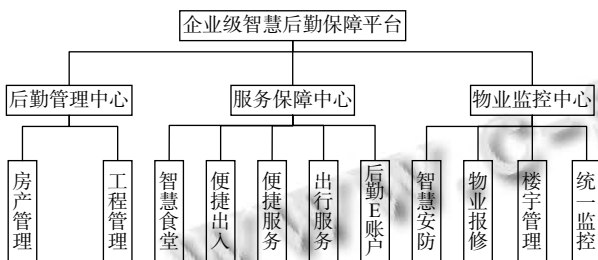


图3 系统功能结构图

1.3.2 服务保障中心

(1) 智慧食堂

借助“互联网+”技术, 实现菜谱制定、原料采购、线上订餐、智能超市、就餐结算等食堂核心业务的系统支撑, 全面提升食堂的管理水平和服务能力. 搭建跨区域智慧结算中心, 实行区域内异地就餐、野外配餐结算等, 解决外出办公、工程施工、事故抢修、节日值班等领域长期存在的地域瓶颈、管理瓶颈、员工服

务瓶颈等问题.

(2) 便捷出入

结合人脸识别技术建立出入管理功能. 员工以便捷为主, 通过E账号实现人脸识别、刷卡等方式通行; 访客以安全为主, 通过现场自助登记、移动预约、被访人邀约、现场办理等方式进行访客业务办理. 实现停车场管理, 包括车位管理、车位申请等功能, 实现人员车辆无感通行、进出安全管理.

(3) 便捷服务

整合企业内外部优质资源, 为员工在线提供报修、理发、洗衣、订票、送水等便利服务, 提升服务质效.

(4) 出行服务

出行服务融合用车申请、审批、派车、接单和评价功能, 形成闭环管理, 规范公务出行. 通过集成社会化车辆服务资源, 实现公务车辆、社会车辆统一受

理、按需分配,为员工提供更加多样便捷出行服务。

(5) 后勤 E 账户

在现有实体卡基础上,引入员工人脸识别及虚拟卡,建设统一后勤 E 账户功能,实现员工“实体卡、虚拟卡、面部信息”3 种身份验证,满足员工门禁、支付等多种快捷身份验证需求。支持人员信息管理、人脸照片管理、身份验证授权管理、电子账户信息管理和实体卡管理等功能,实现跨单位通行及就餐等服务。

1.3.3 物业监控中心

(1) 智能安防

应用智能图像识别技术、物联网技术,设置访问范围电子围栏,建立来访人员管理及非授权入侵告警,实现办公区域的动态安全监管,第一时间发现、定位异常人员及问题,实现安防事件的全过程追踪、安防设备的互联应用,提高事件响应速度。

(2) 物业管理

开展统一物业管理,通过工单形式,实现物业人员和服务的数字化管理,实现对秩序维护、工程维修等物业各部门人员的可管、可查和可追踪,以及实现楼宇日常安全防护、设备故障维修与日常巡查等服务管控。

(3) 楼宇管理

通过接入智能化楼宇运行系统,对建筑物内的各种水、电、气等设备进行集中可视化监控。

(4) 统一监控

采用全局化策略,统一设计和规划基于设备运行、楼宇运行、人员出入、车辆运营、物业报修、食堂餐饮等统一物业活动信息监控体系。

2 系统实现

系统基于 Spring Cloud 微服务架构,采用前后端分离模式进行业务系统开发、部署和运维^[6,7]。该系统后端框架采用 Spring Boot、MyBatis-Plus 等核心技术,前端框架基于 Vue、ElementUI 等进行构建。Spring Cloud 是基于 Spring Boot 进行构建服务的,它主要特点就是提高了开发和部署上的效率,用来构建分布式系统。

2.1 Spring Cloud 微服务架构构建

Spring Cloud 的主要目标就是通过提供一系列内部组件和框架,迅速搭建一个分布式的微服务系统,还提供了分布式微服务系统的一些常用组件,如服务注册和发现、配置中心、熔断器、路由网关等核心组件^[8-10]。

(1) 服务注册与发现

Eureka 作为服务注册发现组件,是微服务架构的核心功能之一。微服务部署之后,一定要有服务注册和发现的能力。Eureka 分为客户端和服务端。微服务工程的启动类上添加@EnableEurekaServer 注解,则该微服务成为 Eureka 服务端,通过在配置文件 application.yml 配置服务端地址、端口等,这样服务端启动后就能通过访问 Eureka 查询到注册到注册中心的服务。微服务工程的启动类上添加@EnableEurekaClient 注解,则该微服务成为 Eureka 客户端并作为服务提供者注册到 Eureka,供服务调用者调用,服务调用者在调用服务提供者时,无须关心服务提供者的地址、端口信息,只需要配置服务提供者暴露在注册中心服务名即可。

(2) 网关与负载均衡

Gateway 网关是一个服务,用来管理授权、访问控制和流量限制等,是系统的唯一出口。Gateway 网关封装了系统内部的微服务,这样 REST API 接口服务就被 API 网关保护起来,隐藏在 API 网关后面的业务系统就可以专注于创建和管理服务,客户端只需要调用网关接口,就可以调用到实际的微服务。Gateway 网关可以结合 Ribbon 完成负载均衡的功能,可以自动检查微服务的状况,及时剔除或者加入某个微服务到可用服务列表。

(3) 微服务熔断机制

微服务架构会有多个服务层调用,基础服务的故障可能会导致级联故障,进而造成整个系统不可用的情况,服务提供者的不可用会导致服务调用者的不可用,并将不可用范围逐渐放大。Hystrix 能够在被调用的微服务出现异常时及时熔断,触发调用方的异常处理流程,防止调用方产生异常。Feign 中集成了 Hystrix 组件相关的依赖,但是 Feign 中默认是关闭了 Hystrix 功能的,需要在配置文件中添加配置 feign.hystrix.enabled: true 开启。注解@FeignClient 上添加 fallback 属性配置快速失败处理类,该处理类是 Feign Hystrix 的逻辑处理类,必须实现被@FeignClient 注解修饰的接口。

2.2 基于元数据的前后端统一数据校验

当前不同技术下的前后端校验方法相对成熟,比如前端 Vue 的表单 rules 组件、后端 Spring 的基于 JSR303 的注解校验方法,使用这些方法进行校验时,往往需要在对应功能代码上进行分散、复杂的配置,在前后端规则一致的情况下需要配置两份,造成前后端数据校验规则维护难、复用性差的问题,本文提出

了一种基于元数据的前后端统一校验技术,通过构造基础元数据模型和功能元数据模型,定义灵活、可拓展、可集中管理的校验规则.利用数据库表结构信息作为初始化数据以减少配置工作量,通过功能元数据与基础元数据相结合的形式,达到复用和拓展校验信息的目的.

(1) 建立元数据模型

元数据模型如图4所示,涵盖数据的元数据信息,如字段类型、字符串长度、是否为空、最大值、最小

值、正则表达式、数据描述等.元数据模型包括基础元数据模型和功能元数据模型.基础元数据模型与功能元数据模型通过基础元数据主键进行关联.基础元数据模型包含数据标识字段、元数据字段和扩展字段;功能元数据包含模块编码、功能编码、功能元数据字段标识、基础元数据模型数据标识等.在基础元数据模型中定义完整、通用的元数据信息,功能元数据模型则复用基础元数据模型的元数据信息,并可以在此基础上进行修改、拓展.

基础元数据模型		功能元数据模型	
PK	Element_code [数据元编码]	PK	Module_function_code [模块功能编码]
	Element_name [数据元名称]	FK	Func_element_code [功能数据元编码]
	Element_data_type [数据元数据类型]		Element_code [数据元编码]
	Element_empty_flag [数据元可空标志]		Element_name [数据元名称]
	Element_min_value [数据元最小值]		Element_data_type [数据元数据类型]
	Element_max_value [数据元最大值]		Element_min_value [数据元最小值]
	Element_max_size [collection 数据元最大长度]		Element_max_value [数据元最大值]
	Element_regular [数据元正则表达式]		Element_max_size [collection 数据元最大长度]
	Element_desc [数据元描述]		Element_regular [数据元正则表达式]
	... [基础元数据模型扩展模型字段]		Element_desc [数据元描述]
	Remark [备注]		... [基础元数据模型扩展模型字段]
			... [功能元数据模型扩展模型字段]
			Remark [备注]

图4 元数据模型

(2) 初始化基础元数据表和功能元数据表

元数据管理工具抽取数据库表结构相关信息并对基础元数据进行初始化.对功能元数据进行初始化,每一张数据库表作为一个模块分别生成新增、更新等多个功能的元数据信息,通过主键关联,复用基础元数据模型中的元数据信息.使用元数据管理工具将从数据库表结构抽取的元数据信息中与实际校验需求不符的,通过界面操作的方式对元数据信息内容进行修改、补充,将多张业务表包含同一名称且校验规则一致的字段,去掉重复的元数据信息.将基础元数据信息与功能元数据信息整合为真正的校验规则,如图5所示,通过基础元数据主键将两个模型进行关联,功能元数据中不为空的元数据信息覆盖基础元数据信息.

(3) 校验规则存入 Redis 共享缓存

为了提升校验规则读取性能,将校验规则存入 Redis 共享缓存中.前端通过接口调用形式,以模块编码和功能编码作为查询条件,来获取校验规则进行数据校验.服务端通过自定义注解方式进行校验,并通过读取缓存,以模块编码和功能编码作为查询条件,来获取校验规则进行数据校验.

结合以上步骤,当前后端校验规则发生变化时,只需要改变元数据表的校验规则,并刷新元数据缓存,从而就保证了前后端校验的一致性,也达到了维护简单的效果.

2.3 内外网穿透传输技术

企业采用物理安全隔离装置实现内外网隔离,内外网只允许 SQL 语句穿透传输,导致内外网服务部署复杂且交互困难,本文提出了内外网穿透传输技术,该技术通过将数据交换需求转化为规范化配置,使用 SQL 对数据库中间表操作,实现内外网多种通信方式的数据交换.房产管理、便捷出入等业务服务在外网只需要部署前端服务以及其他必要组件即可支持业务系统的正常运行.

内外网穿透传输技术示意图如图6所示.调用方通过网关调用穿透传输服务接口,调用接口后需要经过安全校验,包括防重放、权限校验以及加密校验,并对参数进行解密.穿透传输服务解析出 HTTP 等请求的请求类型、请求参数、请求地址等信息,并通过 SQL 语句将解析出的请求信息存入内网数据库,同时开启轮询等待响应.对端穿透传输服务轮询模块定时批量

获取数据库中的请求数据,使用线程池开启多条线程分别处理请求,获取请求的请求类型、请求参数、请求地址等数据并根据请求数据调用对应内网业务服务.完成调用后,穿透传输配置模块对返回数据进行转换

为标准配置存入数据库.请求端穿透传输服务开启的轮询发现请求已有返回,从数据库中获取返回数据,并删除数据库中该条转发记录,对返回数据进行解析后返回调用方.

基础元数据

Element_code	Element_name	Element_data_type	Element_empty_flag	Element_min_value	Element_max_value	Element_max_size	Element_regular	Element_desc
Account	账号	String	True			50	[a-zA-Z0-9_]+	账号基础元数据
Age	年龄	Number	True	1	100			年龄基础元数据



功能元数据

Module_function_code	Func_element_code	Element_code	Element_name	Element_data_type	Element_empty_flag	Element_min_value	Element_max_value	Element_max_size	Element_regular	Element_desc
User_add	Account	Account	账号							用户模块新增功能下的账号元数据
User_add	Age	Age	年龄		False		200			用户模块新增功能下的年龄元数据



最终校验规则

Module_function_code	Func_element_code	Element_name	Element_data_type	Element_empty_flag	Element_min_value	Element_max_value	Element_max_size	Element_regular	Element_desc
User_add	Account	账号	String	True			50	[a-zA-Z0-9_]+	用户模块新增功能下的账号元数据
User_add	Age	年龄	Number	True	1	200			用户模块新增功能下的年龄元数据

图5 校验规则

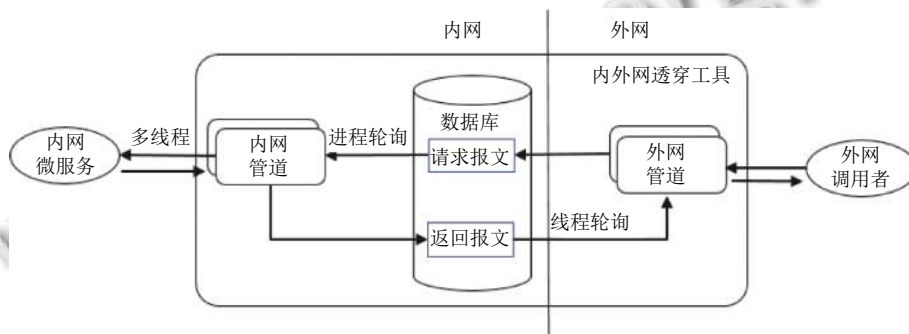


图6 内外网穿透传输技术示意图

3 系统测试

3.1 测试环境

测试机操作系统 Windows 10, 硬盘容量 500 GB, 内存 32 GB, CPU 为 8 核. 应用服务器配置 Tomcat 8.5 和 JDK 1.8, 操作系统 CentOS 7, 硬盘容量 512 GB, 内存 64 GB, CPU 为 16 核. 数据库服务器配置 MySQL

5.7, 操作系统 CentOS 7, 硬盘容量 512 GB, 内存 64 GB, CPU 为 16 核.

3.2 功能测试

本文以便捷出入模块的内部拜访为例,对内部拜访申请创建、审批、查看详情、消息等功能进行测试.内部拜访申请创建如图7所示,申请人填写被访人单

位、姓名、电话、拜访事由等信息。

内部拜访审批如图 8 所示, 被访人可以在待办列表里通过或驳回申请人的申请, 已审批过的申请可以在已经列表中查看。

内部拜访申请详情如图 9 所示。



图 7 内部拜访申请创建



图 8 内部拜访审批

被访人已经审批过的申请, 申请人可以在我的消息里查看相关消息, 如图 10 所示。



图 9 内部拜访申请详情



图 10 消息提醒

3.3 性能测试

性能测试选择用户登录、线上点餐、客饭申请、出行申请 4 个 HTTP 接口进行测试, 并发用户数为 100 个, 对其平均响应时间进行了测试, 测试结果如表 1 所示, 系统具有较好的并发处理能力。

4 成果应用

智慧后勤保障平台目前已在国网宁夏电力、中国电力科学研究院试运行,为2万多名员工提供服务保障,对员工出行、就餐等生活方式带来了实实在在的变化,特别是在疫情期间,访客线上预约,被访人线上审批,减少访客登记人员聚集,提高访客通行效率。同时,平台为企业各后勤业务子系统提供统一入口,大大降低了应用层各子系统的业务协同,使数据在各个应用子系统之间快速流转。PC应用端和移动端首页如图11和图12所示。

表1 性能测试

接口名称	平均响应时间(ms)	出错率
用户登录	495	0
线上点餐	376	0
客饭申请	280	0
出行申请	312	0

5 结语

本文基于微服务的实现技术 Spring Cloud 开发企业级智慧后勤保障平台,打破企业各类单体式后勤业务应用系统间壁垒,贯通后勤协同数据共享渠道,为企业提供统一平台、统一入口的立体后勤服务新模式,助力提升后勤管理、保障、服务的多元化支撑能力,推进后勤工作提质增效,强化后勤精益管理水平,实现后勤管理和服务的信息化、数字化、智能化转型升级。



图12 PC应用端首页



图11 移动端首页

参考文献

- 李希喆. 依托“兴服 APP”打造智慧型后勤班组体系的创新与实践. 中国国际财经(中英文), 2018, (3): 167-168.
- 杨强, 张钧鸣. 基于微服务架构的大数据应用开发创新实践. 电力大数据, 2019, 22(3): 71-76.
- 白蓉, 王东新. 基于 JavaEE 的网络舆情监控系统设计. 电子设计工程, 2020, 28(2): 40-43, 47.
- 杨秦. 基于微服务架构的云平台服务端的设计与实现 [硕士学位论文]. 成都: 电子科技大学, 2020.
- 龙新征, 彭一明, 李若森. 基于微服务框架的信息服务平台. 东南大学学报(自然科学版), 2017, 47(S1): 48-52.
- 洪传旺. 基于 Spring Cloud 的“微课堂”知识考核系统的设计与实现 [硕士学位论文]. 南京: 南京大学, 2019.
- 杜英魁, 王杨, 关屏, 等. 基于 Spring Boot 的云端数据监控管理与可视化应用系统. 计算机系统应用, 2020, 29(5): 123-127. [doi: 10.3969/j.issn.1003-3254.2020.05.018]
- 袁小凯, 李果, 黄世平. 基于微服务架构的电力云服务平台设计研究. 电子设计工程, 2021, 29(11): 185-188.
- 陈酌灼, 张梦清, 茹春平. 基于微服务的电力信息系统架构研究与设计. 南方能源建设, 2020, 7(S2): 18-24.
- 王琰洁, 王孟强, 刘泽三, 等. 基于微服务的统一应用开发平台应用上云关键技术研究. 电力信息与通信技术, 2021, 19(4): 64-69.