

基于持续集成的智慧校园开放平台的设计与实现分析

江春¹ 罗亦言² 邱素贞³ 通讯作者 麦志坚¹ 李湘棱¹

(1. 南宁学院数字经济学院 南宁 530200; 2. 南宁学院信息化处 南宁 530200; 3. 南宁学院人工智能学院 南宁 530200)

摘要: 在信息技术的快速发展下,各高校均建设有若干信息化管理平台,使广大师生从中得益。但是多数服役的信息管理平台在功能方面都是基于当时的需求调研而固化的,当想要扩展一些新功能时,就不得不淘汰旧的平台,引入具备新功能的新平台,增加了成本。鉴于此,本文提出一种持续集成的智慧校园开放平台,这种平台具有良好的开放性,当需要加入新功能时,可以在原有平台上添加新功能模块,不再需要重新设计平台,使校园信息化管理平台能够随着不同时期的功能需要进行动态适配,增强校园管理平台的使用效率。

关键词: 持续集成; 智慧校园; 校园管理平台; 开放式; 设计

近年来在信息技术发展的推动下,高校信息化管理取得了长足进步,如校园网、校园一卡通、教育信息系统、校园图书馆系统等,这些系统的出现提升了校园管理质量,增强了学生的校园新体验【1】。但是各系统的不断出现,为校园的数据统一管理带来了一定难度,不同系统形成了很多的信息孤岛。对此,可建立信息化管理平台,将各不同系统统一集成到信息管理平台上,当有新系统、新功能时,也可集成到这一平台中,这样教职工及学生就可以登录一个平台实现各项功能。

一. 基于持续集成的智慧校园开放平台的架构

本平台采用基于 Blueprint 的开源应用程序引擎为引擎作为平台基础,这可以看做是一个能够支持可移植功能的容器式平台。开放平台工包括核心系统、持续集成部分、应用部分、数据中心,所有模块都在 Docker 环境下运行,见图 1。

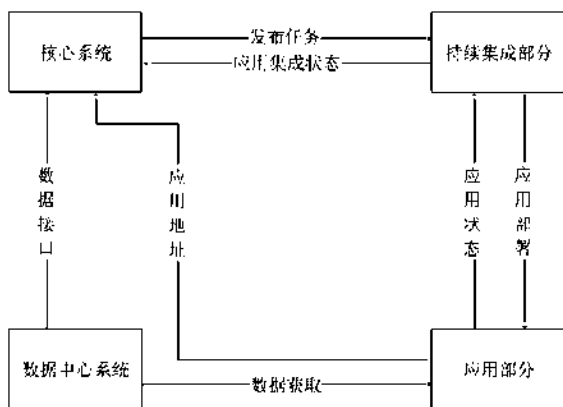


图1 智慧校园持续集成开放式管理平台结构框架

(一) 核心系统

本开放平台用 Java 程序开发,数据库为 MySQL,并使用 Redis 作为数据加速服务。系统前端为 Layui 框架,并配有供手机、平板登录的移动端,移动端通过 MUI 和 MTML5+ 进行开发。电脑端平台前端和后端各组件建立通信【2】。而移动端主要是和应用部分建立通信。核心系统的通信协议是设计的关键,本系统采用的是 HTTPS 协议,这种协议属于应用层协议,利用 TCP/IP 作为通信协议,结合 SSL+HTTP 构建的可进行加密传输、身份认证的网络协议来完成数据传递。允许客户端通过网络服务器在系统中获取各类资源。但是这类协议存在一定的安全问题,为了规避安全风险,对于系统中的内部信息需要采用加密保护协议,防止敏感信息泄露。

(二) 应用部分

纳入到平台 Blueprint 这一容器中的各应用,都会存储在应用部分。用户通过身份认证后基于权限可以访问平台中的这些应用,此外应用在运行过程中所产生的日志、运行信息等都会被平台捕获并记录。本平台支持 Java、Python、Node.JS 等语言所对应的 Springboot、Django、Express 等常用 Web 应用框架。

(三) 数据中心系统

本开放平台中装设有校园数据中心接口,使各应用系统的数据和校园学生、教职工等相关信息都存储在平台之中【3】。用户在注册时就会要求输入学籍、学号等信息,通过验证后允许登录。此外登录系统后,教职工的账户有权限登录数据中心,查询校园各方面信息,系统有检索功能,通过搜索关键词快速查询相关信息。

(四) 持续集成部分

这一部分是本开放平台实现功能持续集成的关键,利用的是 Jenkins 持续集成工具,在核心系统的后端操作,发送指令、代码能完成新功能新应用的集成置入。完成新应用的编译、测试、功能部署等功能【4】。在软件工程中,所谓持续集成就是将新的项目副本不断加入到项目主干中的开发过程,编码人员把副本代码整合到主干上,并进行执行与测试,通过测试结果评价集成质量及性能指数。如果未达到预期就需要及时调整代码,直到能够有效加入到平台之中。

二. 智慧校园开放平台的设计

(一) 核心系统的设计

核心系统的通信形式是 HTTPS,采用基于 Spring Boot 框架的 Spring MVC 代码,在 Spring MVC 之中,用户发送请求先经过 DispatcherServlet 前端的控制器,再进入到 HandlerMapping 之中,找到和用户请求匹配的 Handler,得到匹配的 Handler 后进程处理及执行,执行的方法是向 HandlerAdapter 发送基于 Handler 的 ModelAndView 信息,这个信息为属于 SpringMVC 的底层对象,其中包括有 Model 以及 VIEW 等内容,之后根据系统的视图解析器进行解析,经过视图渲染形成用户前端或移动端的最终内容,传送给前端或是移动端,见图 2。

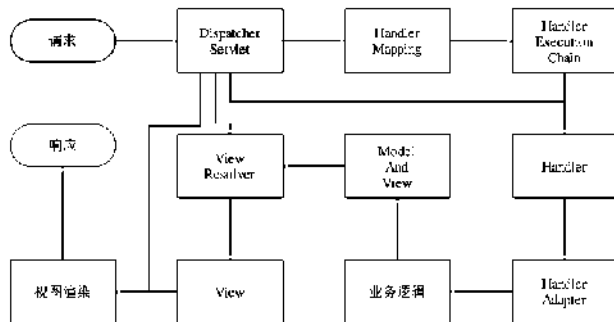


图2 基于 Spring MVC 的核心系统运作

本平台在基于 Spring MVC 的基础上,还采用了中心认证 CAS 来验证用户的登录信息,并通过 Shrio 来设置不同用户账户的权限。比如后台管理人员、教职工、学生等不同账户允许登录的空间,能够进行的功能是不同的,就通过 Shrio 来设置各自的权限。

系统的数据库部分采用的是 MySQL 数据池,此外还加入了能够起到数据容灾和日常监控作用的 Druid,对数据库的运作情况进行全过程监控,保证数据库的安全,当有安全威胁时,Druid 会向管理人员发出警报【5】,此外还结合了基于 Redis 的缓存空间,在缓存空间中能够将常用的信息存到这个空间中,当有用户需要相关信

息时,会直接从缓存空间中调取,在 Druid 大规模并行处理的协作下,综合提高了平台整体响应速度。

(二) 应用部分的设计

应用部分是存储平台中各应用程序的空间,平台支持 Java、Python、Node.JS 等多种语言的应用,各个应用之间是相隔离的,这样一个应用如果出现异常,那么也不会对其他应用以及整个平台造成影响。为了有效实现各应用的独立性,本系统就采用了 Doceker 容器结构,这种结构能够有效满足各应用的独立性,并且支持模板化应用部署。

持续集成部分是将应用从外部连入到平台中,连接的方式就是利用 Jenkins 工具,将应用处理为 doceker.sock 的文件,从而实现和平台的有效连接,连接后进入到这个 Docker 大容器中,并成为一个独立的子容器【6】。用户登录访问某一应用时,平台会通过动态监测以系统自动分配的方式,将客户所登录的应用传送给客户端,实现访问。

(三) 数据中心系统的设计

数据中心系统也采用的是 SpringBoot 来开发,和核心系统采用了相类似的 MVC 模式,另外,数据中心还和学校其他信息管理系统以及认证服务系统相连接,学校教职工只需要登录这个开放平台,就可以查询或处理日常管理和教学事务。

数据信息系统也有可以扩展的接口,通过这一接口可以将外部的信息导入到数据中心之中。此外,系统还具有加密功能,比如学生的个人信息、成绩等,这些都属于私密的信息,将这些信息设置密码,通过密码登录才可以发布及编辑这些信息【7】。学生可以在用户端通过学籍号查询个人成绩。而可以公开的信息就无需加密,登录平台可以浏览这些不加密的信息。

(四) 持续集成部分的设计

对于持续集成部分这是本平台中的一个重点内容,目前常用的集成工具包括 Jenkins、Travis CI 以及 Go CD 等,系统采用的是 Jenkins 作为持续集成的工具。本平台应用的 Jenkins 具备自动化程度高、功能强大、响应迅速的基础特性。项目主干就相当于一个仓库,这个仓库有很多分支,当提交分支代码后会触发自动化的合成构建,整个过程可以在核心系统的操作后来完成。通过这一工具实现系统的功能扩展,将各 Doceker 系统的主干相连接。Jenkins 能够有效将原本的单个或是若干个任务连接起来,主要原理是利用 Jenkins 工具,将应用处理为能够和 Doceker 相连的 doceker.sock 的文件,还能完成编译、测试以及部署等集成相关操作流程的自动化处理。除了代码合成的集成模式外,Jenkins 还具有插件扩展功能,将新系统功能以插件的方式融入到平台之中或为容器内系统增添新特性。

(五) Web 端及移动端的设计

本平台可以同时支持电脑登录和手机(平台)等移动终端登录。其中的移动端是通过 MUI 以及 HTML5+ 进行研发,利用 HBuilder 分别编译为适用安卓系统及 IOS 系统的应用,移动端利用 HTTPS 协议实现了平台核心系统的通信连接。系统选择采用 HTML5+ 进行开发是因为目前越来越多的安卓及 IOS 系统应用都是通过 HTML5+ 来开发的,能够很好地适应目前绝大部分智能手机的硬件性能,开发效果理想且成本可控,且因为应用广泛,有大量的工具类库和实例,可以有效提高平台开发进度的可控性。

适用于电脑端的 Web 端的开发框架为 Layui,也是利用 HTTPS 协议和平台核心系统相连,通过 Layui 框架中的 Layer 来构建弹层组件,在不同组件中设计相应的功能。此外,还利用 nicesroll 对首页进行优化,提高平台首页的风格统一性。

三. 智慧校园开放平台重要功能的实现

(一) 登录模块的实现

用户登录平台可选择电脑 WEB 端或是智能手机、平板的移动端,这两种登录实现模式基本一致,主要包括两个方面:①用户登录平台输入登录信息,如账户号或学籍号、身份证号,以及密码等,信息传递到数据信息模块进行认证。②认证信息无误后,通过 Shiro 框架来匹配相应的权限。经过以上两个步骤,才能登录平台。如果认证不通过,非本校师生,或者管理人员账号,那么就无法登录。

登录成功之后,会跳转到平台地址,平台会根据 Shiro 框架所返回的用户权限开放相应的功能并渲染对应的界面布局。

(二) 数据缓存功能的实现

平台中每天都会处理大量数据信息,从数据库中完成数据的调取、读取、传输等一系列操作,这些频繁的操作会对系统性能造成影响,如果同时处理海量的数据信息,会引起系统卡顿,针对这一问题,系统将结合基于 Redis 的缓存数据库,通过算法判断信息的访问频次,以提高重复信息的识别率,高频次数据将存储至 Redis 中,以提高高速综合信息检索的命中率,这将大幅度减少了平台的处理压力,通过合理调控内存利用率来提升性能,同时,在内存允许的情况下,缓存空间所占用的空间要明显低于数据库,也容易维护。实现缓存功能需要经过以下步骤:①判断用户所要提取的信息是否存储在缓存空间之中,如果没有存储,那么就要进入到数据库中调取,之后根据信息摘要赋予权重,为系统是否将该信息存储进 Redis 中提供判断依据。②如果在 Redis 缓存空间中存储了客户所要提取的信息,那么就会直接将信息传递给客户端。

(三) 应用功能的实现

用户在智慧校园平台上,需要调用某一应用时,客户端会向应用层传递一个 XML 文件,不同应用的 XML 文件都有自己对应的编码,平台会从应用层中匹配和 XML 文件对应的应用,以及持续集成部分 Jenkins 中所对应的相关插件,均传输给客户端。具体的步骤如下:①根据用户所要登录的应用形成 XML 文件,将文件发送给应用层。②应用层在接受 XML 文件后,会在代码地址库中搜索和其相对应的应用以及 Jenkins 插件,并构建对应的 Docker 镜像以及 Docker 容器,映射至应用文件夹。③检测发出应用请求的用户端,将文件夹发送给用户端。④用户端将文件下载下来,并存储到 Redis 缓存空间,方便下次操作时直接调取。用户端针对不同类型的应用配制对应的命令,以实现应用的运行。比如 Java 应用执行 Java-jar 命令,Node.js 应用执行 npm start 命令,Python 应用执行 Python 命令。

四. 结束语

综上,本文提出一种基于 Blueprint 的低代码持续开发集成平台,将其应用于校园信息管理中,推动智慧校园建设,当本系统需要扩展新功能时,可在原有平台上插入新功能模块,不再需要重新设计平台,使校园信息化管理平台能够随着不同时期的功能需要,增加功能,增强校园管理平台的使用效率,希望本次研究能够为相关领域工作者提供些许参考借鉴之用。

参考文献:

- [1] 龚晨,唐雷,向往.大数据时代高校智慧校园建设中的“模块化功能”构建研究[J].四川省干部函授学院学报,2021(03):116-120+115.
 - [2] 张彬,范佳伟,李志国,孙威.智慧校园下的网络安全防护[J].网络安全技术与应用,2022(04):95-97.
 - [3] 王录通,唐慧羽,李琴,刘莉.基于大数据的高职智慧校园信息化管理模型研究[J].中国信息技术教育,2021(22):109-112.
 - [4] 常晟,王佳雨,张洋,赵博涵.基于云数据环境的智慧校园建设与实践[J].现代教育技术,2021,31(11):55-62.
 - [5] 杨江,王强.大数据背景下高校智慧校园建设探索[J].福建电脑,2022,38(04):32-35.DOI:10.16707/j.cnki.fjpc.2022.04.008.
 - [6] 宋雪,李小庄,卢甜甜.基于虚拟现实技术在智慧校园中的建设与应用[J].网络安全技术与应用,2021(11):87-88.
 - [7] 王超,宋文君,王晓亮.大数据及云计算技术在智慧校园中的应用研究[J].中国新通信,2021,23(21):85-86.
- 基金资助: 2022 年度广西高校中青年教师科研基础能力提升项目“智慧校园视角下基于 Blueprint 的轻量级 Web 应用程序引擎设计与应用”(2022KY1789); 南宁学院教授培育工程(2021JSGC08); 南宁学院科研项目(2022KY199); 南宁学院科研项目(2022KY226)
- 通讯作者简介: 邱素贞(1982-),女,广西合浦人,硕士研究生,副教授,主要研究方向为计算机技术、信息化教育。