基于微服务的高校信息化系统研究和实现*

郑海山,刘燕文
(厦门大学 信息与网络中心,福建 厦门 361005)

摘 要:高校信息化系统基于传统三层BS(Browser/Server,浏览器/服务器模式)架构设计,随着信息化系统的增加,运维难度加大,数据共享变得困难。文章分析了高校信息化系统存在的问题,认为在教育信息化2.0的背景下,高校信息化建设应当尽快进入基于API(Application Programming Interface,应用程序接口)的微服务时代。介绍了开源API网关的选择和在高校使用场景下的配置细节,给出了API开发者应当遵循的最佳实践,并且对开源API网关缺失的统计和监控功能进行了补充。通过基于微服务的开发理念,使用自动化部署手段结合开源软件和少量编程搭建API网关,编写的示例应用验证了通过解决微服务全生命周期管理问题,高校的信息化系统开发、运维将更加透明和高效,数据共享将更为安全。

关键词:信息化;微服务;应用程序接口;软件开发
中图分类号:TP393 **文献标志码**:A **文章编号**:1673-8454(2019)17-0053-06

随着信息化的建设与发展,高校信息化系统数量越来越多,系统越来越庞大,功能也越来越复杂[1],建设和运行维护模式是一个很大的挑战。高校信息化系统大部分基于传统的三层BS架构,各个系统之间类似烟囱式结构,数据共享较为困难,一般需要借助数据交换平台进行数据共享[2]。有些高校基于面向服务体系结构(Service-oriented Architecture,SOA)的理念对数字校园进行改造[3-5]。过往的SOA实现大部分依托于XML(Extensible Markup Language,可扩展标记语言)、SOAP(Simple Object Access Protocol,简单对象访问协议)、WSDL(Web Services Description Language,网络服务描述语言)等标准,系统整体较为庞大。近年来,基于API(Application Program Interface)的微服务架构开始流行起来,比如互联网公司普遍采用了基于API的微服务架构[6-7]。基于API的微服务架构[8]更为轻量级,开发维护较为简单。

信息化系统基于微服务开发存在诸多优点,例如可大批量部署,原先应用部署在同一台机器,横向扩展较为困难,使用API开发后计算可以在多台服务器内分散。核心API开发完成后,对于不同的终端比如PC、移动客户端、微信等调用均可实行一次开发,多次使用而无需改变业务功能,只需要调整前端显示逻辑即可。对于数据共享,基于API提供可以克服原先数据库共享数据只能以只读方式共享的问题。对业务的操作逻辑通过调用API,可以做权限和数据校验等安全性判断,也可以对调用进行跟踪,实时性也更高。通过API的版本化,可以提供多个版本共存,对外提供统一的接口,隐藏内部实现逻辑。通过统一的API网关,业务也无需实现重复的验证、授权、缓存、熔断、日志等功能。然而现在存量信息系统的改造不是一朝一夕可以完成的。

本文分析了高校信息化系统的现状,并对信息化系统和数据共享存在的问题提出了解决方法,认为在保留原有所有模式的情况下,高校信息化系统应采取逐步迁移升级的方法尽快向API架构迁移[9]。通过对开源API网关的测试、部署方法研究、测试应用编写,验证了基于微服务的开发模式更适应未来高校信息化的发展。

一、高校信息化系统的现状和解决方法

1.信息化业务系统和数据共享现状

高校信息化系统大部分还是基于传统的三层BS架构,各个系统由不同外包厂商开发,技术能力参差不齐,系统建设周期不同,上线时间不一,各个系统比较独立,属于烟囱式结构。而互联网公司由于自有开发团队,统一技术架构。互联网公司已经将大部分应用基于API架构进行开发。高校信息化系统可分为两种类型:一种是类似教务系统、学工系统、人事系统、科研系统、财务系统等,这些业务系统业务较为复杂,业务大部分功能由业务部门管理员使用;另一种是网盘、即时通讯、自助打印等业务系统,这些业务系统定制化程度不高,功能较为简单,普通师生使用较多,有移动端,起步较晚。第一

*基金项目:中国高等教育学会2016年度教育信息化专项课题(2016XXZD06)。

种系统一般采取高内聚的开发模式,由于管理员需要频繁查询调用数据,所以系统架构必须直接对数据库进行操作,数据在系统内流转,只有少部分结果数据需要共享给其他系统。而第二种相对独立,一般会采用技术更先进的 API 架构开发,同时支持多个不同客户端。

高校的数据共享目前主要还在采用包括离线传递、数据库视图开放、共享库同步等机制。离线传递的实时性不高,数据使用无法管控。数据库视图开放需要数据库直接面对第三方业务系统开放,安全性较低。第三方业务系统的访问也会对数据库性能造成影响,数据使用过程无法管控。共享库同步存在同步周期时间问题,实时性不高,只能共享部分数据,第三方业务系统较难对数据进行更新。

2.信息化系统改造解决方法

高校的信息化系统应当尽快向 API 迁移,但是对于第一种系统,要进行整体 API 改造难度较大,成本很高,受益也很低。很多高校依赖第三方外包公司,改造进度受制于外包公司。老牌的外包公司存在较久,业务沉淀较深,应用开发较早,技术框架较老,更新需要较长周期。

所以较为可行的方案应当是逐步迁移。对于第二种的系统,应当采购技术较为先进,有 API 开放功能的系统。对于第一种系统,如果无法整体迁移到 API 架构,也应当尝试对部分业务功能开发 API。高校目前在广泛开展一站式服务门户网站建设[10],一站式服务门户不再以业务来区分系统,而是以角色或服务来开发不同颗粒度的功能。通过管理功能和服务功能分开,面向管理人员的可采取原有模式,而面对受众较广的师生员工的需求开发出 API,在一站式门户对外提供服务。

使用 API 改造数据共享也可以解决数据拥有者对数据的授权问题。高校内数据可分为三种:第一种是公开的数据,诸如组织机构名称和代码,任何人应当均可通过 API 访问;第二种为需要权限获取的代码,诸如教务系统某位教师的任课信息,应当可授权比如人事系统调用;第三种为师生员工的个人隐私数据,诸如消费记录、成绩等,应当可通过师生员工显式授权后开放给第三方调用。通过 API 和 OAuth2(Open Authorization v2,开放式授权第二版)的授权模式可以解决这些问题,促进高校内应用生态环境的健康发展。

在提供 API 数据共享模式后,原有的包括离线传递、数据库视图开放、共享库同步也不会消失,以上三种共享方法也会存在一定的使用场景,因为 API 倾向于传递较小的数据量,对于需要获取大量数据进行实时或离线分析不存在优势。

3.API 模式存在的挑战

基于 API 的微服务架构可以改变传统烟囱式架构在部署和扩展方面的局限性。微服务各个服务可以独立部署,服务可以使用不同的语言和数据库开发,对外隐藏实现细节并独立演化。然而新技术也会带来新问题,微服务服务数量增加,服务之间关系复杂。微服务的开发、测试、部署、接入服务平台、验证、授权、访问频率控制、缓存、状态、监控、熔断、日志、版本化、服务终止等一系列全生命周期管理问题都需要研究。通过引入 API 网关[11],在一个集中的位置部署 API 网关,可以解决以上问题。

二、系统设计方案

1.基于 API 的下一代应用

基于 API 的新应用系统应当尽量采用前后端完全分离的技术开发。前端指运行在浏览器进行页面展示的 HTML、CSS、JavaScript 等代码,后端指运行在远程服务器内的业务逻辑。通过前后端完全分离,前后端工程师可以在约定好的 API 接口上独立开发。前端展示可以适配 PC 端和移动端等浏览器而无需修改后端业务逻辑代码。通过编写单页应用,前端所有代码均可独立缓存,加快渲染速度,前后端 API 数据传输量小,减少用户的流量耗用。前后端分离增强了代码的可维护性,极大地提高了开发效率。

为了让师生员工可以将自己的数据对第三方应用进行授权,原有的统一身份认证除了提供 SSO(Single Sign On,单点登录)功能外,还应当扩展出 OAuth2 的授权功能。

API 的输出格式应当支持包括 JSON、XML 等格式,并推荐使用 JSON 较为轻量级的格式。

2.基于 API 网关的架构

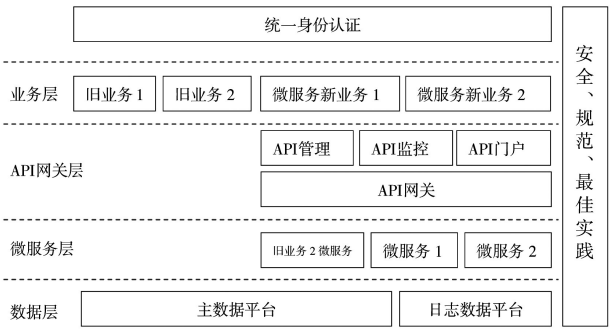

图 1 基于 API 网关开发的系统整体架构

图 1 为基于 API 网关开发的系统整体架构,在架构内,包含以下主要角色和功能:

(1)API 网关:提供 API 接入、验证、授权、访问频率

控制、缓存、状态、监控、熔断、日志、版本化、服务终止等一系列全生命周期管理功能。提供 SSL 卸载。

(2)API 服务提供者:旧业务系统根据新需求开发出 API,或者新开发的业务系统应当要求同步提供 API。

(3)API 管理:对 API 网关进行管理。

(4)认证方式提供者:高校原有的统一身份认证系统,并扩展 OAuth2 功能。

(5)API 门户:统一的位置发布 API,详细的说明文档对 API 约定、调用方法、测试流程等进行说明。

(6)API 调用者:对 API 服务提供者提供的 API 进行消费。

(7)日志分析:对系统涉及到的所有服务进行日志记录和分析。

(8)监控:对 API 的运行和性能指标进行监控。

(9)安全和规范:一系列 API 提供者和调用者需要遵守的关于安全开发和安全提供数据的最佳实践细则。

相同功能的开源软件很多,对其进行选择以满足业务功能的最大化为原则。如果无法全部满足,应当选择只需通过少量扩展即可满足业务的开源软件。一般开源软件更新较快,必须时时跟踪开源软件的开发进度,并在一段时间后重新评估并选择是否进行切换迁移或者继续保留原软件。应当选择社区活跃度高、安全更新和功能开发及时的开源软件。表 1 和表 2 为常用 API 开源软件的比较。

通过综合比较以上开源 API 网关软件,我们最终选定了 Kong。Kong 较为轻量级,功能基本上可以满足 API 网关主要需求。Kong 缺少的分析统计功能可以使用其他开源软件补充。

三、系统实现细节

1.系统搭建部署

由于系统涉及到较多开源软件,操作系统层面有 Ubuntu,API 网关为 Kong,数据库为 Cassandra、MySQL,缓存为 Redis、Memcached,监控有 Elasticsearch、Logstash、Kibana、Prometheus、Grafana、Icinga2,文档生成格式为 MkDocs、Swagger Editor,测试 API 提供者和消费者使用 Apache2、Nginx、Python、Django、Flask。需要对所有开源软件进行安装和配置,还需要对所有系统进行安全加固,所以我们使用自动化部署工具 Ansible 实现基础架构一键部署。

在部署时应当考虑安全性和各系统性能要求,对所有应用进行分析,规划好部署位置。为可横向扩展预留空间。由于各个软件之间交互复杂,在部署时应当根据最小权限原则规划严格的防火墙规则。Kong 管理端口只允许管理员 IP 访问。API 只开放 443 端口,强制要求 HTTPS。API 服务提供者只允许 API 网关和监控系统访问。所有后台监控和日志分析只允许管理员 IP 访问。数据

表 1 开源 API 网关社区活跃度比较

开源 API 网关名称	提交数量	源代码贡献人数	GitHub Watch 数量	GitHub Star 数量	GitHub Fork 数量	最后更新时间	Google 搜索结果
Kong	4 千	177	919	18000	2100	最近一个月	684 万
Tyk	2 千	37	221	3800	516	最近一个月	590 万
API Umbrella	3 千	12	99	1100	230	最近一年前	129 万
Fusio	1 千	1	42	566	65	最近一个月	18 万
Gravitee.io	1 千	12	46	329	91	最近一个月	13 万
apiman	2 千	31	61	433	214	三个月前	1 万
ApiAxle	1 千	13	N/A	N/A	N/A	一年半前	<1 万

表 2 开源 API 网关功能比较

开源 API 网关名称	主要功能	优点	缺点
Kong	OAuth2 认证、访问控制、频率控制、日志	基于 Nginx,简单高效	Admin Console 为收费,日志分析为收费。无缓存
Tyk	日志分析、API 开发者 Portal、API 文档认证、频率控制、Mock、转换数据、服务发现	开发者可以自行管理。功能强大	免费试用,License 每年申请一次,免费 License 只能单机部署。组件较多,安装复杂
API Umbrella	Admin Console、缓存、基于角色的权限控制	使用了 Nginx、Varnish、Redis、MongoDB、Elasticsearch。api.data.gov 正在使用	无 OAuth2 机制
Fusio	OAuth2 认证、频率控制、文档、开发者门户	PHP 开发,无转发 API 功能,适合单体应用	API 定义使用自有格式
Gravitee.io	基于 Java,有 Admin Console、开发者 Portal、负载均衡、日志分析、健康度检查	功能强大	基于 Java,较为庞大,组件复杂
apiman	API 管理、多层结构,可嵌入单体应用,异步调用	N/A	基于 Java,无 OAuth2,架构较重
ApiAxle	前端 Proxy 架构、API 管理、日志分析	简单,轻量级	无 OAuth2,无 Dashboard,不再活跃开发

库和缓存只允许本机访问或者特定的服务器 IP 访问。

2.Kong 的功能配置和插件扩展

Kong 提供有基本认证、密钥认证、OAuth2 认证、HMAC 认证、JWT 认证、LDAP 认证等方法,在高校内,可只使用密钥认证、OAuth2 认证和 JWT 认证。对于获取类似高校整体组织架构、课程列表、教师授课列表等,可以直接使用密钥认证的模式。如果是需要师生员工授权的数据等则使用 OAuth2 认证模式。如果开发前后端完全分离的应用则可使用 JWT 认证。Kong 的 OAuth2 认证需要我们自行实现认证和授权功能。

为了更安全,对 API 消费者要求提供访问的服务器 IP 地址,加入 Kong 的 IP 限制和 ACL 控制内,提高系统的安全性。对于 API 消费者请求的 API,限制合适的访问频率和请求大小限制,以避免一定程度的数据泄露风险。

Kong 的统计和分析功能为非开源产品,所以需要自行根据日志做分析,由于 Kong 有完善的日志,只需要配置 Kong 发送日志到 Elasticsearch 或者 Prometheus 等开源软件,则可自行对 Log 进行分析。同时日志也写入文本文件保存,方便今后使用大数据平台诸如 Hadoop 等分析。

对于 API 消费者调用后端的服务,Kong 支持从 HTTP Header 内的 Host 字段、URL 的 Path 路径字段和 HTTP 方法 3 个途径进行路由,由于 API 还涉及版本不同,版本也需加入路由。如果使用 Host 字段,调试不够方便,API 地图不够清晰。如果将 API 版本使用HTTP Header 传递,对于 API 消费者比较简单,只需要替换 HTTP Header 无需更换 URL Path 即可使用不同的 API 版本,然而也会导致 API 地图不够清晰、API 提供者适配也较为麻烦的问题。为了简化架构,我们统一只使用 Path 路径字段进行路由,同时 API 的版本直接体现在 URL 内,版本由 API 提供者自行路由。

在 API 挂接时,Kong 提供 strip_path 变量控制传递到后端 API 服务器是否带上路由的 URL Path,比如 API 消费者访问/jwc/course_list,如果 strip_path 为 True 的话,后端 API 服务器的地址则为/course_list,推荐所有 API 路由均设置 strip_path 为 True,此项配置可以对后端 API 服务器隐藏 Kong API 地图信息,方便今后挂接到不同 API 网关。

对于 API 挂接的,API 提供者可以选择只挂接一个总的服务入口,或者将服务内的所有 API 端点全部挂接到 Kong,然而这势必造成 Kong 配置数据膨胀,对于 API 的调节较为麻烦,为了简化操作,我们统一规定,每个 API 提供者建议只提供 2 类的服务,一类是使用密钥认证的 API 族群,一类是使用 OAuth2 的 API 族群,这种情况下授权 API 消费者只需对应不同的 API 消费者授权不同的 API 服务族群。当然,这样也造成了 API 消费者可以消费某个 URL Path 路径下面的所有 API。所以对于特殊的 API,也允许 API 提供者挂接多个重复的 API。表 3 为厦门大学部分 API 挂接路由示例。

表 3 API 挂接路由示例

API Endpoint	API 介绍	路由目的地
/jwc/	教务处相关 API	路由到教务处 API 提供者
/jwc/v2/	教务处相关 API,第二个版本	路由到教务处 API 提供者,由 API 提供者适配第二个 API 版本
/xsc/	学生处相关 API	路由到学生处 API 提供者
/xsc/personal/	学生处学生隐私数据	路由到学生处个人数据 API 提供者
/news/	校内新闻相关 API	路由到校内新闻 API 提供者
/ecard/	一卡通相关 API	路由到一卡通 API 提供者

对于 Kong 的使用,应当通读文档,对每个配置项均做测试,并找出符合自己学校需求的配置项。

3.开发者门户网站

API 网关实际上是 API 提供者和消费者之间的桥梁,应当搭建完善的帮助网站,API 提供者可以在门户发布 API 功能,消费者可以浏览 API 功能。同时为提供者和消费者约定建议,遵循一定的规范。在书写 API 文档时我们建议统一采用 Swagger 文档格式。通过提供 Swagger 文档,API 消费者可以无需编写代码,使用工具即可自动生成所使用编程语言的 API 调用 SDK,也可以即时在浏览器内做 API 调用测试。对于 API 的编码,由于 API 实质上也是 Web 应用程序,也会有 Web 应用程序所有存在的安全弱点,所以我们提供了 Web 应用程序安全编写的所有规范和检查点。

4.API 提供者

为了更方便让 API 可以进行消费,API 门户应指导 API 提供者编写 Swagger 格式文档,方便 API 消费者浏览和测试 API,方便了 API 提供者和 API 消费者之间进行沟通。

API 提供者到 API 网关之间的流量应当使用HTTPS 加密,为了系统安全,应当只允许 API 网关访问 API 提供者服务器的 HTTPS 端口。

API 的所有权限交由 API 网关处理,API 提供者应当信任 API 网关发送的任何信息,业务相关的 OAuth2的 ID 从 API 网关获取。API 的版本也遵守 Kong 的配置,在 URL 内硬编码版本信息,返回的数据格式以JSON 为主。

5.API 消费者

API 消费者需要知道 API 网关提供了哪些 API,通过 API 门户可以让 API 消费者阅读 API 列表，通过 Swagger 提供的 API 格式文档测试调用 API。同时提供帮助文件，指导 API 消费者对 API 进行测试以及搭建 Mock 服务器测试。指导 API 消费者申请 API 账户、申请提高调用频率等。

6.日志分析

日志分析分为以下几种：

(1)信息安全等级保护合规的系统安全加固和攻击溯源准备,必须将所有服务器的日志保存到远程。具体实现为在所有服务器上安装 Filebeat，把所有服务器相关系统日志写入 Elasticsearch,使用 Kibana 图形化界面分析查看和检索。

(2)所有业务数据,包括 API 访问日志、Nginx 日志、Kong 日志,定期 Logrotate 轮换日志,并使用 Filebeat 写入 Elasticsearch 数据库,可作为调试查错使用,同时所有业务日志每天拷贝一份到远程 NFS 服务器以方便今后可能的 Hadoop 大数据分析。

7.监控

监控分为以下几种监控：

(1)监控所有服务器的 CPU、内存、硬盘、进程、网络流量,使用 Metricbeat、Packetbeat,传输进入Elasticsearch,可以分析系统性能,网络流量是否剧增等分析是否存在大量数据 API 泄露。

(2)监控所有服务器的可用性。使用 Icinga2,监控所有服务器的 Ping、SSH 管理端口和 Web 端口,在服务器出故障之前提前得到通知。

(3)监控 API 提供者服务的可用性。通过在 API 网关添加一个监控账户,授权可只读访问所有 API,也可要求 API 提供者提供特定的可用性状态 API，使用 Icinga2 定期对 API 进行访问测试。

(4)监控各个 API 调用的频率。通过将 Kong 的日志传入到 Elasticsearch,分析 API 调用频率。可使用 Kibana 分析查看 API 调用频率与设置的调用频率的差别,可以比较不同或相同业务内不同 API 的调用频率,找出最受欢迎的 API 服务等等。

四、应用效果

厦门大学基于 Kong 建立了 API 网关，提供了 API 门户。同时也以 API 提供者和 API 消费者角色建立了应用,测试整个开发流程,实现了基于微服务开发模式的落地[12]。

以 API 提供者角色从主数据中心内读取各种数据，完成了基于 OAuth2 授权的在借图书、一卡通余额和消费记录、WIFI 上线下线提醒、教职员工工资查询、学生成绩查询、免费米饭消费查询等功能。

以 API 消费者角色开发了免费米饭消费分析和微服务微信公众号,两个系统均采用前后端完全分离的架构,调用 API 网关的数据展示给用户。在开发过程中,基于 Kong 完善的日志，原先在单体应用内较难定位的各函数执行时间等参数,在日志层即可看到各个 API 调用的频率、执行的时间,可以有针对性地对 API 进行性能调优、缓存、查错等。

通过开发实践,我们验证了基于 API 网关开发应用的流程,API 编写完成可以为各个不同应用开发者调用,提高了代码重用性,通过前后端完全分离,提高了开发速度,通过 API 门户,为 API 提供者和 API 消费者之间搭建起了桥梁。同时,由于系统有完善的日志和监控,使得系统更为安全。

五、结束语

本文通过分析高校信息化系统的现状,给出了解决方法,并通过选择 Kong API 网关,搭建了厦门大学 API 平台,建立了 API 开发门户平台,搭建起 API 提供者和 API 消费者之间的桥梁，通过引入 Elasticsearch、Prometheus 等开源软件填补 Kong 开源版本没有提供的日志统计分析功能,并编写了免费米饭消费分析和微服务微信公众号,验证了新信息化系统开发模式的合理性。开源软件变化较快，应当时时跟踪多个不同的开源软件,以便进行迁移,目前架构没有使用到服务自发现等功能,今后应当加入 Consul、etcd、Zookeeper 等服务自动发现机制,并最终过渡到基于服务网格的架构[13]。

参考文献：

[1]蒋东兴,付小龙,袁芳等.大数据背景下的高校智慧校园建设探讨[J].华东师范大学学报(自然科学版),2015(S):119-125,131.

[2]宓咏.互联网+思维的智慧校园数据治理与应用服务思考——IT 技术支撑学校管理服务变革的探索[EB/OL].http://go.wisedu.com/homePage/linePlay/82.mooc.

[3]梅立军,付小龙,刘启新等.基于 SOA 的数据交换平台研究与实现[J].计算机工程与设计,2006(19):3601-3603,3627.

[4]王明洁.基于 SOA 的复旦迎新系统的设计与实现[D].上海:复旦大学,2012.

[5]于磊.基于 SOA 的高校异构系统集成平台的研究[D].天津:天津大学,2016.

基于智能终端的移动学习系统研究与设计*

姚红静[1,2]
(1.西安市现代教育信息技术中心,陕西 西安 710018;
2.西北工业大学 自动化学院,陕西 西安 710018)

摘 要:传统课堂教学时间、地点受限,互动性差,学生自主性不强。本文在分析使用智能终端设备开展移动学习可行性的基础上,结合智慧校园建设中移动学习技术的应用现状,以英语课程教学为切入点,从需求分析、功能模块设计、学习过程控制等几个方面研究探索并设计实现了基于智能终端的移动学习系统。该系统可以随时随地满足用户的个性化学习需求,提升教学互动。本研究是对传统教学的一种补充,也是智慧校园建设中信息技术与教育教学深度融合的一个积极探索和参考案例。

关键词:移动学习;智能终端;智慧校园;Web Services;过程控制
中图分类号:TP393 **文献标志码**:A **文章编号**:1673-8454(2019)17-0058-05

Alexzander Dye[1]等人认为移动学习是一种在移动计算设备帮助下,能够在任何时间任何地点发生的学习,移动学习所使用的移动计算设备(PDA、智能手机、笔记本电脑等) 必须能够有效地呈现学习内容并且提供教师与学习者之间的双向交流。随着移动计算技术、互联网技术、移动通信技术的日趋成熟和智能终端(智能手机、iPAD 等)的普及,移动学习产业进入了快速成长期。

传统教学以教师为主体, 缺少教学互动和师生互动,教师不能够及时获得学生学习数据,不能对学生学习过程采取及时有效的调控措施和教学策略。如何使用智能终端移动学习技术弥补传统教学不足,使其在教育教学中发挥有效作用,是当前的热点话题。基于该问题,本文进行了研究和探讨。

一、使用智能终端设备开展移动学习的可行性研究

移动学习技术正逐步改变着人们的学习环境,逐渐被应用到教育教学中[2-3]。要充分考虑智能终端设备应用的可行性,相关因素分析如下:

一是用户层面。智能终端设备延伸了教室学习的空间,可以让学生的学习更加便捷[4-6]。对于校园用户来说,智能终端设备价格适中、方便携带、易扩展、相关软件比较普遍,应用广泛的终端设备使得学生随时、随地、个性化的学习成为可能。

二是学科学习特征层面。作为一种学习工具,智能

*基金项目:本文系全国教育信息技术研究“十二五”规划课题青年课题“移动学习技术在基础教育教学中的应用研究”(立项号:124440793)、西安文理学院教师教育专项研究子课题基金项目(JYZX1703)研究成果。

[6]NETFLIX.How We Build Code at Netflix[EB/OL].https://medium.com/netflix-techblog/how-we-build-code-at-netflix-c5d9bd727f15.

[7]邱小侠.电商 CRM 的微服务重构实践[EB/OL].http://www.infoq.com/cn/presentations/crm-micro-service-refactoring-practice.

[8]陈春霞.基于容器的微服务架构的浅析[J].信息系统工程,2016(3):95-96.

[9]LIN J, LIN L C, HUANG Shi-che.Migrating web applications to clouds with microservice architectures [C].2016 International Conference on Applied System Innovation (ICASI).Japan Okinawa:IEEE,2016:1-4.

[10]沈富可.华东师范大学 高校信息化应用要以融合为先[N].中国信息化周报,2018-10-08(019).

[11]何翔.从 0 到 1,苏宁 API 网关的演进之路[EB/OL].http://www.infoq.com/cn/articles/suning-11-11-api-gateway.

[12]沈富可.回顾 2017:我们不缺理念,缺“落地”[J].中国教育网络,2018(1):16-17.

[13]MORGAN W. What's a service mesh? And why do I need one?[EB/OL].https://blog.buoyant.io/2017/04/25/whats-a-service-mesh-and-why-do-i-need-one/.

(编辑:王晓明)