Design and Implementation of a WeChat Service Platform for University Libraries: A Case Study of Zhejiang University of Technology Library (Postprint)

Authors: She Jingtao

Date: 2017-11-08T00:00:00+00:00

Abstract

Objective: To provide convenient and efficient personalized services for library users and facilitate system administrators through the design and development of a WeChat service platform.

Background: Currently, most university library WeChat service platforms require source code changes for custom menu generation and modification, and lack web-based system management functionality.

Methods: We designed and developed a WeChat service platform for university libraries by innovatively leveraging Java reflection mechanism, utilizing the WeChat Official Platform API, and employing the Java programming language and Hibernate database framework.

Results: The platform enables authorized administrators to manage the web-based system and modify in real-time the names, number, arrangement order, response action types, and bound functions of WeChat custom menus.

Conclusion: Practical use and testing demonstrate that the application significantly enhances user satisfaction and improves the work efficiency of WeChat administrators.

Full Text

Preamble

ChinaXiv Partner Journal, Issues 272/273, 2016, Nos. 7/8

Design and Implementation of a WeChat Service Platform for University Libraries: A Case Study of Zhejiang University of Technology

Library
(Library of Zhejiang University of Technology, Hangzhou 310032, China)

Abstract

[**Objective**] This study aims to design and develop a WeChat service platform that provides convenient and personalized services for library users while improving operational efficiency for system administrators. [**Context**] Currently, most university library WeChat service platforms require modification of source code to generate or alter custom menus, and lack network system management capabilities. [**Methods**] We innovatively utilized Java' s reflection mechanism, API interfaces from the WeChat public platform, Java programming language, and the Hibernate database framework to design and develop a WeChat service platform tailored for university libraries. [**Results**] The platform enables authorized administrators to manage network systems and modify WeChat custom menu names, quantities, ordering, response action types, and bound functions in real time. [**Conclusions**] Usage and testing demonstrate that this application significantly improves user satisfaction and administrative efficiency for WeChat platform management.

Keywords: WeChat Public Platform; API Interface; Hibernate; Java Reflection Mechanism
Classification Number: G250.7

The WeChat public platform is a functional module added by Tencent to its WeChat application, enabling individuals and enterprises to create official accounts for comprehensive communication and interaction with specific user groups through text, images, and voice messages [**?**]. WeChat users can engage in real-time exchanges via voice, text, images, and video, while the public platform facilitates mass messaging and integration with business service systems, thereby enhancing user experience.

The WeChat public platform offers several advantages for university library services. First, WeChat has a substantial user base, particularly among university populations, allowing libraries to expand their service influence through this platform [**?**]. Second, it provides libraries with a stable and authentic user community, serving as a bridge between libraries and patrons to strengthen word-of-mouth promotion. As a bidirectional personalized communication tool, libraries can improve their services based on user feedback to achieve service optimization [**?**]. Third, WeChat offers excellent cost-effectiveness for both users and libraries. Fourth, it enables convenient and timely information dissemination, which is crucial in the information age. Fifth, the platform provides robust API interfaces [**?**] that facilitate secondary development, allowing budget-constrained university libraries to innovatively and effectively apply the platform for user convenience.

Many domestic university libraries have established WeChat official accounts and utilized the platform' s API interfaces for secondary development to build

their own service platforms. However, most lack network system management functions, requiring source code modifications for each business system integration or custom menu change, which substantially increases administrators' workload. Regarding research on library WeChat platforms, numerous studies analyze current situations and service innovations. For example, Yang Yanni et al. [**?**] selected six university libraries to compare their Weibo and WeChat platform service functions and content resources, discussing their micro-service dissemination effects and optimization strategies. Du Hui et al. [**?**] investigated WeChat platforms of "211 Project" university libraries to examine the current state of disciplinary services. Research on platform design and development is also abundant; for instance, Luo Tao [**?**] built a local library server to enable queries for library information, personal borrowing records, FAQs, bibliographic data, and documents through the WeChat public platform. However, research on network system management services and visual configuration of WeChat custom menus is virtually nonexistent, making the design and development of a more rational and efficient WeChat service platform a worthwhile research topic for university libraries.

2. Design of the WeChat Service Platform

This study uses Zhejiang University of Technology Library as a practical case, leveraging its existing business service systems and data resources to design a WeChat service platform that fully considers the needs of both library users and system administrators.

2.1 System Design Approach

WeChat official accounts operate in two modes: editor mode and development mode. Development mode's primary advantage lies in its ability to seamlessly integrate websites, databases, and other business service systems with the WeChat public platform through code implementation and server deployment, truly enabling a powerful platform. Therefore, this study established a technical route of "implementing a library WeChat service platform through development mode."

The Zhejiang University of Technology Library WeChat service platform integrates with the library information management system, course reservation system, locker system, web server monitoring system, and library web system based on user requirements. Beyond providing personalized services for business systems, the platform implements visual configuration of WeChat custom menus. The core design concepts are as follows:

(1) Personalized Services: To improve developer efficiency, the platform uses Java' s reflection mechanism for rapid addition of personalized services. Developers only need to complete two steps: first, add response functions in the system' s core business class, which handles interface docking and communication with business service systems (systems without interfaces require secondary development to provide compliant APIs); second, bind the response function in

the platform' s backend WeChat custom menu settings. Through these steps, the platform can dynamically invoke response functions from the core business class based on user messages using Java reflection.

(2) Network System Management: The platform achieves network system management by deploying Agent programs on business service systems. An Agent is a customized web application for different business systems that provides API interfaces and runs automatically after the business system starts. After WeChat user authentication, administrators with appropriate permissions can manage business service systems by calling Agent interfaces through the WeChat service platform.

(3) Visual Configuration of WeChat Custom Menus: Through the platform backend, administrators can configure custom menus, including names, quantities, ordering, response action types, and bound menu resources (which are either URL links or response function names depending on the action type). Menu configurations are saved to the database and simultaneously update the custom menu static variable in the system (a HashMap data structure used for dynamically loading response functions).

The system architecture of the WeChat service platform is shown in Figure 1 [Figure 1: see original paper]. The interaction process follows these steps: (1) Mobile device users send messages to the WeChat server via HTTP protocol; (2) The WeChat server forwards the processed message in XML or JSON format to the WeChat service platform; (3) The platform parses the message and calls the corresponding business service system interface or Agent program interface, then packages the returned message into XML or JSON format; (4) The WeChat server forwards the parsed message to the user.

2.2 Functional Requirements Design

The Zhejiang University of Technology Library WeChat service platform aims to provide convenient and personalized services for faculty and students while offering operational convenience for system administrators. The functional modules are illustrated in Figure 2 [Figure 2: see original paper].

The platform' s frontend and backend are connected through WeChat custom menus. Through visual configuration in the backend, response function names correspond with custom menu names, ordering, and keywords, which are stored in database tables. Using Java reflection, the system dynamically looks up and invokes response functions based on keywords from user input during runtime. The network system management and WeChat management modules are described below:

(1) Network System Management - Locker Management: Administrators can restart the locker system server and query user locker records. - **Website Server Management:** Administrators can restart Apache Tomcat and MySQL services on the library' s homepage server. - **Server Status Query:**

Administrators can quickly identify abnormal web servers; server names and addresses can be added through the platform backend.

(2) WeChat Management - Visual Configuration of WeChat Custom Menus: This feature allows direct modification of custom menu names, ordering, quantities, response action types (either redirecting to a link or triggering a response function), and bound functions without source code changes. To add new custom menu response functions, developers simply implement methods in the core business class and bind them in the backend management system, greatly improving development efficiency.

3. Implementation of the WeChat Service Platform

Considering server operating system compatibility, the platform was developed using the WeChat public platform API interfaces, Java programming language, and Hibernate database framework. The platform employs a MySQL relational database and Ajax (Asynchronous JavaScript and XML) technology for web pages.

3.1 Key Technologies

(1) Hibernate Database Framework: Hibernate is a Java-based open-source persistence middleware that provides lightweight encapsulation of JDBC. It uses ORM (Object-Relational Mapping) to map Java objects to relational databases, passing SQL statements to the database and returning results as objects. Hibernate encapsulates JDBC database operations and provides object-oriented database access APIs [**?**]. By operating on data as objects, developers need not concern themselves with database types; changing databases only requires modifying configuration files, thereby improving development efficiency.

(2) Java Reflection Mechanism: Java reflection enables runtime access to all properties and methods of any class and allows invocation of any method on any object. This dynamic information retrieval and method invocation capability is called Java' s reflection mechanism [**?**]. Its power lies in creating flexible code that can be assembled at runtime without source code linking between components. Reflection is implemented through the java.lang.Class class, which stores runtime type information. Once a Class object is obtained, all properties and methods can be accessed, enabling dynamic method invocation on any object.

3.2 Key Frontend Function Implementation

The frontend business process of the Zhejiang University of Technology Library WeChat service platform is shown in Figure 3 [Figure 3: see original paper]. The platform automatically looks up corresponding response functions based on user query keywords, with functions determining whether to operate on the

database or call business service system interfaces or Agent program interfaces. If no matching response function is found, the platform returns an error message to the user.

Key frontend functions include:

(1) Personalized Service Message Processing: Different message types in personalized services are handled by different functions. For example, text messages trigger the `private static String processRequestTextMessage(ReqTextMessage reqtextMessage)` function, which uses Java reflection to retrieve response function information from a static HashMap variable. The core code is as follows:

```
// Receive text message and generate reply
private static String processRequestTextMessage(ReqTextMessage reqtextMessage) {
    // CoreMethod contains all response functions that communicate with business system inte
    // Menus is a static HashMap variable loaded from database during initialization
    Iterator iter = CoreMethod.Menus.entrySet().iterator();
    String key = "";
    boolean flag = false;
    while (iter.hasNext()) {
        Map.Entry entry = (Map.Entry) iter.next();
        key = (String) entry.getKey();
        // Check if custom menu keyword matches user message text
        if (choose.equals(key)) {
            flag = true;
            break;
        }
    }
    if (flag) {
        // Use Java reflection to invoke response function corresponding to keyword
        Class<?> cls = Class.forName("lib.zjut.weixin.service.CoreMethod");
        Method method = cls.getMethod(CoreMethod.Menus.get(choose).getResource(), ReqBaseMes
        // Package response function result as XML data
        tmpStr = MessageUtil.MessageToXML((RespBaseMessage) method.invoke(cls, (ReqBaseMessa
    } else {
        // Return help information
        tmpStr = MessageUtil.MessageToXML(CoreMethod.getHelp(reqtextMessage));
    }
    return tmpStr;
}
```

Other message type processing functions follow a similar structure.

(2) Business Service System Interfaces: Response functions connect to business service systems with Web Services interfaces using the Apache Axis2 framework. For systems without open interfaces, custom APIs are developed to read relevant data and convert it to JSON format for response function access.

(3) WeChat Information Pagination: This function handles cases where business service systems return large amounts of information that cannot fit on mobile screens. It uses a `rule` table to store current keywords and page numbers. The `rule` table structure is shown in Table 1 . For example, Zhejiang University of Technology Library has two locker systems (CH and PF). When users query locker information, the system records their unique identifier with `open_{id}`, the system type ("CH" or "PF") in the `keyword` field, and the current page number in the `page` field.

(4) Network System Management: Frontend network system management is implemented through Agent programs deployed on business service systems, enabling functions such as querying user locker records, restarting business system services, and checking web server status. The implementation steps are: first, authenticate WeChat users to ensure only administrators can access network management; the Agent program parses user query messages to obtain the OpenID and queries the WeChat service platform database to complete user authentication. Second, authorized administrators can call Agent programs and receive processing results: (1) Query user locker records: Agent calls the locker system query interface and returns data to the platform, which parses and forwards it to users; (2) Restart business system services: Agent executes a batch file to restart services and returns results; (3) Query web server status: Agent returns a web application address link (implementation detailed in reference [**?**]).

The frontend implementation effect is shown in Figure 4 [Figure 4: see original paper].

3.3 Backend Implementation

The backend includes modules for administrator management, database management, WeChat user management, bound user management, WeChat custom menu management, message content management, online consultation management, user message management, and FAQ management. The most complex module is WeChat custom menu management, which implements visual configuration. The solution includes:

(1) Database Design: Create a `Menu` table in MySQL with the structure shown in Table 2 .

(2) Visual Configuration Interface: Develop backend pages using Ajax technology. The primary menu setting interface is shown in Figure 5 [Figure 5: see original paper], and the secondary menu setting page in Figure 6 [Figure 6: see original paper]. Menu type defines the response action type, menu resource is either a response function name obtained through Java reflection or a redirect link address, and keywords are text inputs users enter to trigger corresponding response functions.

(3) Menu Generation: After adding primary and secondary menus and saving

to the database, the `createMenu` function is called. This function uses the JSONObject plugin to format `Menu` table data into the JSON structure required by WeChat for custom menu generation, loads the custom menu, returns error codes, and updates the system' s custom menu static variables.

The Zhejiang University of Technology Library WeChat service platform launched a beta version in October 2013 and officially deployed in August 2014. The platform has been well-received by faculty and students, attracting attention from student organizations. As of January 2016, the library' s WeChat official account had over 10,000 followers with more than 5,000 bound users.

This study, using Zhejiang University of Technology Library as an example, designs and develops a library WeChat service platform by leveraging existing information management systems, databases, and web applications. The goal is to provide personalized and efficient services for library users and administrators through mobile communication convenience, while sharing platform construction methods and experiences with other university libraries to enable broader adoption of WeChat public platforms.

References

[1] Chen Ying. The Application and Research of WeChat Public Platform in Library Mobile Service [J]. Research on Library Science, 2013(20): 71-75.

[2] Deng Wei. The Budding Library WeChat Service [J]. Library & Information Science Tribune, 2013(5): 38-41.

[3] Guo Chunling, Zheng Yue. Reflections on Developing Micro Message Marketing in Library [J]. Public Library Journal, 2013(3): 35-37.

[4] WeChat Public Platform [EB/OL]. [2016-01-15]. http://mp.weixin.qq.com/wiki/home/index.html.

[5] Yang Yanni, Ming Junren. The Current Situation and Optimization of the Construction of Academic Library in Micro-services [J]. Research on Library Science, 2015(19): 8-14.

[6] Du Hui, Liu Xiao, Yuan Baicheng. The Innovation of Subject Service Based on WeChat Public Platform in University Libraries [J]. Library and Information Service, 2015, 59(6): 41-45.

[7] Luo Tao. Construction and Research of Library WeChat Public Platform [J]. New Technology of Library and Information Service, 2015(1): 96-100.

[8] Hibernate [EB/OL]. [2016-01-17]. http://www.howzhi.com/course/2055/.

[9] Java Reflection Mechanism [EB/OL]. [2016-01-18]. http://baike.baidu.com/view/1865203.htm.

[10] She Jingtao, Wang Xingyu. Failure Detection of Digital Resources Web Server [J]. Computer Systems & Applications, 2014, 23(7): 218-222.

Conflict of Interest Statement: The authors declare no conflict of interest.

Supporting Data: Supporting data is self-archived by the authors, E-mail: sjtxp@zjut.edu.cn.

[1] She Jingtao. Manage_{menu}.jsp. Code for visual configuration of WeChat custom menu. [2] She Jingtao. InterfaceUtil.java. WeChat custom menu generation function createMenu.

Received Date: 2016-03-25
Revised Date: 2016-04-27

WeChat Service Platform for Academic Library: Case Study of Zhejiang University of Technology Library

She Jingtao (Library of Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: [Objective] This study tries to provide better services to academic library users and improve the efficiency of library system administrations with the help of a new WeChat service platform. [Context] At present, generating and modifying customized menu of the WeChat service platform for academic libraries require us to edit the source codes, which contain no network administration functions. [Methods] The proposed model used Java' s reflection mechanism, and utilized API technologies, Java programing and Hibernate database framework to develop a WeChat service platform for academic libraries. [Results] The new platform helped administrators manage network system, and edit customized menu' s features in real time. [Conclusions] The new WeChat service platform improves user experience and administration efficiency significantly.

Keywords: WeChat; API interface; Hibernate; Java reflection mechanism

Note: Figure translations are in progress. See original paper for figures.