

WeChat Command System: Research, Development and Application Postprint

Authors: Lingfeng Yu

Date: 2023-10-08T00:00:00+00:00

Abstract

In the era of new media, this study explores and applies methodologies for effectively enhancing WeChat read counts and forwarding volumes through the WeChat command system independently developed by the Technical Center of Xiaoshan Daily. By leveraging WeChat's development interfaces and implementing a solution based on Java+MySQL, a comprehensive forwarding system was established with rational institutional design. This technical approach achieved a year-over-year increase of nearly 20% in per-post forwarding and read volumes, thereby effectively strengthening the dissemination impact of new media through technological means.

Full Text

Preamble

Title: Development and Application of a WeChat Directive System

Abstract: In the new media era, effectively increasing WeChat read and forward counts remains a critical challenge. This paper explores and applies a self-developed WeChat directive system by the Technical Center of Xiaoshan Daily. Leveraging WeChat's development APIs and implementing a solution using Java and MySQL, we established a comprehensive forwarding system with institutional design improvements. This approach achieved nearly a 20% year-over-year increase in single-article forward and read counts, demonstrating how technical measures can significantly enhance new media dissemination impact.

Keywords: WeChat; read count; Java MySQL; database connection pool

Classification Code: G202

Document Code: A

Article ID: 1671-0134(2018)02-075-03

DOI: 10.19483/j.cnki.11-4653/n.2018.02.028

Author: Yu Lingfeng

1. Background and Current Situation

With the rapid emergence of new media platforms, WeChat has become an increasingly indispensable social software. Tencent's WeChat Subscription Accounts and Service Accounts have been widely adopted by enterprises. For media professionals, the most common practice involves daily news updates through subscription accounts. However, as the number of subscription accounts grows, two significant problems have emerged. First, distribution channels remain singular and passive. Modern users follow numerous accounts, and content solely relying on official publication without proactive push mechanisms often goes unnoticed. Second, forward counts remain critically low. Low forwarding rates prevent the formation of viral dissemination patterns, limiting effective improvement in both read counts and engagement metrics.

To address these challenges, centralized forwarding by organizational employees could substantially increase WeChat article forward and read counts. However, this approach raises a critical technical question: how to efficiently distribute forwarding directives and accurately track their completion. To solve this problem, the Xiaoshan Daily Technical Center decided to utilize WeChat's public platform development SDK to design a comprehensive system for directive distribution and statistical tracking.

Since its implementation, the system has issued over 700 WeChat directives, with more than 17,000 approved completions, effectively improving forward and read counts while motivating employees to participate actively. Additionally, this system has attracted interest from industry peers, leading to replication and broader promotion efforts.

2. System Requirements and Analysis

Through systematic investigation, we established the following workflow: First, create a WeChat Service Account named "Xiaobao Forward" for all employees to follow (with an approval mechanism to exclude non-staff members). The backend system pushes messages to users, who then forward the directive to their Moments, capture screenshots, and upload them through the service account. Upon approval, the system sends a confirmation message and maintains detailed records for statistical purposes.

Based on this analysis, the forwarding system requires five core modules: **User Management** handles follow requests and user verification; **Directive Management** enables editing, pushing, and approval of directives (with automatic rejection of overdue submissions); **Department Management** facilitates organizational structuring and user grouping; **Statistical Analysis** provides multi-dimensional reporting on completion rates to reward performance and address deficiencies; and **Permission Design** establishes clear authority boundaries,

delegating partial administrative rights to secondary managers.

For the technology stack, the mobile interface adopts WeChat's native WeUI framework, while the backend employs the mature combination of Tomcat, Java, Bootstrap, and MySQL for UI development, business logic, and database management. The server infrastructure utilizes stable and secure Alibaba Cloud services for cloud deployment.

3.1 Database Development and Connection Pool Design

Through comprehensive database analysis and ER modeling, we structured seven core tables: `in_{customer}` (user table), `in_{datastatistic}` (statistics table), `in_{department}` (department table), `task` (mission table), and three additional supporting tables. A critical consideration involves WeChat nicknames now supporting emoji characters, which the legacy utf8 storage engine cannot handle. Consequently, both the database and tables must adopt the utf8mb4 engine to ensure proper emoji storage and broader compatibility.

Furthermore, because the system may receive numerous concurrent connection requests immediately after issuing directives, we implemented database connection pooling to significantly enhance connection efficiency and support higher concurrency levels.

3.2 MVC Layered Development and Optimization

The application logic is developed in Java using object-oriented design principles with strict adherence to MVC layered architecture. The primary data flow follows: View Layer → Servlet Layer (Service) → Data Access Layer → Database Connection Pool → Database. This layering substantially reduces code coupling and improves reusability.

For data access operations, all CRUD (Create, Read, Update, Delete) functions are encapsulated within a unified DbDao layer, streamlining database interactions while enhancing security checks and overall system safety. To facilitate future replication and scaling, we extracted and consolidated database connection configurations, WeChat API settings, and system parameters into centralized configuration files, enabling straightforward customization for new deployments.

During production optimization, we discovered that the default JVM memory allocation (256MB) was insufficient, occasionally causing memory overflow and system crashes. After adjusting to 2GB, the production environment's memory usage stabilized and fully met operational requirements.

3.3 WeChat Configuration and Access Token Refresh

As the system builds upon WeChat's APIs, it must adhere to platform-specific protocols. The communication flow between WeChat clients and our system is illustrated in [Figure 4: see original paper]. Basic configuration on WeChat's

platform requires setting up server forwarding to Alibaba Cloud, specifying the server address and verification Token.

Since Access Tokens are valid for only 7200 seconds and limited to 2000 daily retrieval calls, caching becomes essential to avoid exceeding rate limits. Moreover, most advanced WeChat APIs require Access Token authorization. Our system implements a central caching mechanism for Access Token storage and refresh, preventing conflicts from multiple refresh attempts and ensuring uninterrupted service even during high-volume forwarding periods.

4. Institutional Support

To further motivate employees to actively participate in WeChat forwarding initiatives, complementary institutional mechanisms have been established to reward engagement and maintain sustained participation.

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

Source: ChinaXiv — Machine translation. Verify with original.