

# A Personal Simple Blog System Based on Gradle and Springboot Implementation

Pan Zhang

Jincheng College, School of Computer and Software, Sichuan University, Chengdu 611731, Sichuan, China

**Abstract:** *With the development of the Internet, the emergence of blogs gives more opportunities to express their views and information to others in life. In a blog, you can freely express your thoughts, emotions, and make friends with different people. The blog system quickly became a platform for individuals to express their opinions. This blog system is mainly implemented using Gradle and Springboot in Java, with the help of the thymeleaf template engine.*

**Keywords:** Blog; Java; Gradle; Springboot.

## 1. INTRODUCTION

With the advent of the Internet, people are more and more interested in online activities, so the emergence of blogs undoubtedly increases the fun of people's lives, enriches people's lives, and makes their personal space wider. The development of the Internet has also brought a variety of technologies, and the maturity of technology has also brought a variety of different system websites, so the spring boot technology is used to implement the personal blog system.

## 2. TECHNOLOGIES USED TO IMPLEMENT THE PROJECT

### 2.1 Introduction to B/S

There are two types of software system architectures, C/S and B/S.

C/S mode: It is a client/server layout, where the client and server are directly connected.

B/S mode: It is a browser/server layout. This structural pattern unifies the client side, mainly by placing extremely important parts on the server, which is beneficial for system development, makes system development simpler, and is more conducive to system care. Compared to C/S mode applications, the biggest advantage of B/S mode is that the client can run as long as there is a browser, while C/S mode applications must be done on the client side and upgrading is not very convenient. On the other hand, B/S mode applications are always the latest version for the client side.

### 2.2 Introduction to Java

In 1995, Java, a completely object-oriented language, was born. It is a programming language that is easy to learn and platform (operating system) independent. Java language has three technical architectures: JavaEE (Java Enterprise Edition Application, which includes Servlets, JSP, etc., mainly for Java web application development), JavaSE, JavaME, so Java language has a wide range of applications. The predecessor of Java was Oak. Sun Company invented the trademark Oak in 1995 when applying for trademark registration and it was used by others. After careful consideration, they directly used the Java word that the proposer accidentally mentioned while drinking a cup of Java coffee as the trademark.

To use Java, the first step is to download the Java software development kit from the official Oracle website, which is divided into Windows and Linux versions. Generally, the Windows version is chosen. After downloading from the official website, you need to configure the environment. Step one, click on Advanced System Settings in Computer Properties, and then click on Environment Variables. Step 2, set a variable to JAVA\_HOME, where the value is your Java installation directory. Its advantage lies in the fact that some Java applications require the use of this variable, otherwise it cannot be opened, such as Eclipse. Step three, set a system variable for the class path, which is used to search for the class path when writing Java source files and packages. Step three, add the bin path% JAVA\_HOME% \ bin directly after the path; This matches the runtime of Java perfectly.

### 2.3 Gradle Introduction

As a world-class mainstream programming language, Java has a desire and goal for efficient and easy-to-use project management tools. First, in 2000, there was the project management tool Ant, which could effectively solve a series of tedious problems in project development. However, the good times did not last long as Ant exposed its own problems. Therefore, Maven tool was born in 2004 and quickly achieved success in the Java market, but both had certain limitations.

Gradle, developed in 2012 based on Ant and Maven, addresses the shortcomings of Ant and Maven. Maven has two core functions, one is to help us manage jar packages, and the other is to help us build projects, among which the most important is to manage jar packages. So the emergence of Gradle has effectively compensated for the drawback of cumbersome XML configuration. Gradle uses the new language Groovy, and all Java languages can be executed in the Groovy environment. Of course, Groovy has many features that make programming simple.

Usage: First, download the required compressed file from the Gradle official website. After downloading, unzip it and then configure the environment variables. There are many methods online that everyone can explore on their own.

When we initialize the first web project, we compile the project using gradle without consuming a lot of time http://spring. Download the Gradle project from io, compile the project with Gradle at the command prompt using the command: Gradle build, and then run the project Java - jar build/libs/initialized - start -0 0. 1-SNAPSHOT. jar. After running the command, the project generation is completed and some files will be generated.

Gradle. bat: Environment script, double-click to automatically configure the environment. Build: is a directory generated after the Gradle project is built

Gradle: There is a rapper in the form, which means that project members can run and install Gradle independently, and Gradle will be automatically downloaded. The advantage is that it can unify the version of Gradle.

Src: It is a convention where main is the source code of the program and test is the test code.

### 2.4 Introduction to Springboot

Following the trend of dynamic communication (Ruby, Groovy, Scala, Node.js), the development of Java appears to be particularly clumsy: numerous setup equipment, low development efficiency, huge setup process, and difficulty in integrating third-party technologies. So to solve this problem, the pivot team has provided a brand new framework called springboot.

Its core advantages are as follows. Firstly, quickly create a Spring project that can run independently. If you create a Spring project, you may need to configure a lot of content yourself, but SpringBoot is very easy. Secondly, using the default embedded servlet container has the advantage that if we were to develop a web application again, we would not need to package the web application as a WAR and place it on the server. It is possible that the server does not have the Tomcat environment installed yet, and we would need to install Tomcat to deploy the WAR package. After developing the springboot application, we can directly create a jar package and use Java jar commands to run it directly. Thirdly, there are many starters (also known as launchers) in springboot, which help us with automatic dependency management and version control. If we want to use a certain feature, there will be corresponding starters. For example, if we want to use the web feature, we can import its launcher. Springboot also controls the other jar packages that need to be included in the web feature, including the version of each jar package. Fourth: Ready to use out of the box, without relying on code to generate implementations, but through conditional annotations, without requiring any XML configuration to implement all Spring configurations.

Simply put, Web MVC involves the controller intercepting user requests [2], calling the model to perform read and write operations on the state [3], sending data to the view [4], and the view rendering the final result and passing it to the user [5].

### 2.5 Understanding Thymeleaf

It is a Java template engine. Capable of solving HTML, XML, JavaScript, CSS, and even plain text. It is a natural template, with the prototype being the page, following web standards, and supporting HTML5. The purpose is page rendering.

### 3. SYSTEM FUNCTIONAL REQUIREMENT ANALYSIS

The functions of blogs are quite powerful, including user management, blog management, and comment management.

(1) After entering the user management interface, perform login operations. In the code, there are users and login passwords that I default to stored in memory. After logging in, users can be added, deleted, modified, and queried.

(2) Blog management: Upon entering the system's main page, users can post blogs, edit blogs, delete unwanted blogs, set tags for blogs, sort them according to the latest information, track their views, and configure their personal homepage.

(3) Comment management: Users can post comments on the blog, delete comments, and obtain comment volume.

### 4. FUNCTION IMPLEMENTATION

#### 4.1 Development Environment Configuration

(1) Using Eclipse 2019 version to ensure the running environment of Java requires JDK 1.8. Configure tomcat-8.5.39 serves as a web server.

(2) Configure Gradle-5.4.1

(3) spring boot 1.5.2.RELEASE

(4) Navicat for MySQL serves as a database management tool.

(5) Develop using Gradle+SpringBoot.

(6) The operating system used is Windows 10 and Sogou High Speed Browser.

(7) elasticsearch-7.2.0

#### 4.2 Introduction to the Common Parts of the Entire System

The entire system consists of 6 major categories, including domain, controller, service, repository, config, and til packages. The XXController class is placed inside the controller and is responsible for the interaction between the front-end and back-end. Create an XXservice interface and implementation interface class (such as ServiceImpl implementation class) under the service package, where the service is responsible for business logic. Create a series of Java object classes and place them in the domain, such as Blog. The Repository package is used to place some interfaces that inherit from the JpaRepository class. Place configuration classes under the config package, such as security configuration classes, and use the til package to place utility classes.

The system uses the thymeleaf template to render the page as part of the view interface. Place page templates and static templates in the Templates directory. Place static resources (images, JS, CSS) under static.

#### 4.3 Specific Implementation

##### 4.3.1 Interface design for user management:

register:

```
@GetMapping("/register")
public String register() {
    return "register";
}
```

When accessing the register URL, the registration interface will be returned.

/Register: OST registration, post request is to submit personal information to the system for registration, and successful registration will redirect to the login interface. User: This parameter is the user object to be saved.

Users: User management/ Users: GET user list.

The parameters are: asynchronous: whether to request the page asynchronously. The function is in AJAX, because when we make the first request, the data is bound to the template, and our interface returns it together. When I conduct a search or execute a pagination query for the second time, I actually did not retrieve the entire page, but only retrieved the data. Therefore, it involves an asynchronous request problem, which is actually using AJAX to retrieve relevant data. Compared to retrieving the entire page, the amount of data is much smaller.

PageIndex pageSize name: The keyword for the user name.

/users/add; GET to obtain the user's newly added page and add the user's interface.

/Users: POST saves users. User: User information to be saved.

@Delete Mapping (value="{id}") deletes users based on their IDs.

/Users/edit {id}: GET to retrieve the specific user editing interface. Id: The ID of a certain user.

#### 4.3.2 Interface design of user homepage

User space: User homepage space/ US/{username}: GET the homepage of a specific user/ US/{username}/profile: GET to retrieve personal settings page. Username: is the user's account.

/US/{username}/profile: POST saves personal settings. User: Object to be saved.

/US/{username}/logs: GET to query user blogs, choose one of the following conditions.

- (1) Order: Sort type, new/hot, default is new.
- (2) Catalog: Blog category ID, default is empty.
- (3) Keyword: Search for keywords, the tags of the blog are the keywords.

/US/{username}/blogs/edit: GET to get the interface for adding a new blog.

/US/{username}/blogs/edit: POST adds and edits blogs, which is used to save blogs. Parameter Blog: is the blog object to be saved.

/US/{username}/logs/edit {id}: GET to obtain the interface for editing blogs. ID: The ID of the blog.

/US/{username}/logs/edit {id}: DELETE to delete a blog. ID: The ID of the blog.

I mainly explain the statistics of blog reading volume. The object is readSize, which initializes the reading volume to 0, private Integer readSize = 0; And setting and obtaining methods.

Defined an interface called void readingIn create (Long id) with increasing reading volume; And methods for implementing interfaces

```
public void readingIncrease(Long id) {
    Blog blog = blogRepository.findOne(id);
    blog.setReadSize(blog.getReadSize() + 1);
    this.saveBlog(blog);
}
```

When we want to increase the reading volume of a blog, we first look up the blog based on its ID and increment it by 1 on top of the original blog to achieve an increase in blog reading volume.

#### 4.3.3 Interface Design for Comment Management

For the comment module, when you want to express your opinions, you can leave comments, delete unwanted or meaningless comments, and count the number of comments. There is no editing function for comments.

Comments: Comment management.

/Comments: GET to obtain a list of user comments. BlogId: Blog ID.

Under the @ PostMapping request, publish comments through the @ RequestMapping ("/Comments") path.

CommentContent: Content published.

/Comments/{id}: Delete to delete comments. The delete method was used to delete. Id: Comment ID

## 5. SUMMARY

When studying a course, I always think about putting it into practice. Imagination is beautiful, but reality is cruel. Only after doing it myself can I know the actual situation. This time, I tried to write a simple personal blog system, learned a lot of knowledge, and also encountered many setbacks. Learn how to quickly build a project using Gradle and how to combine Gradle, SpringBoot, and SpringMVC. What is the purpose of choosing these technologies? Choosing springboot allows for minimal or no Spring configuration, making it less complex and providing a faster entry point for Spring based development. In this design process, my self-learning ability has been improved. When encountering technical difficulties, I am able to take them seriously and never give up. Whether it is checking various books, searching for various information online, or providing help to teachers, I will try my best to solve this problem. This will also be of great help to my future learning.

The personal blog system I designed, although the interface is a bit rough, has a decent level of technical content. For example, it can enable users to add, delete, modify, search, and browse their basic information. Of course, due to the short time and being a beginner, there are still shortcomings in this system that need to be improved in the future learning process.

## REFERENCES

- [1] Wang, Z., Yan, H., Wang, Y., Xu, Z., Wang, Z., & Wu, Z. (2024). Research on autonomous robots navigation based on reinforcement learning. arXiv preprint arXiv:2407.02539.
- [2] Jiang, L., Yu, C., Wu, Z., & Wang, Y. (2024). Advanced AI framework for enhanced detection and assessment of abdominal trauma: Integrating 3D segmentation with 2D CNN and RNN models. arXiv preprint arXiv:2407.16165.
- [3] Yan, H., Wang, Z., Xu, Z., Wang, Z., Wu, Z., & Lyu, R. (2024). Research on image super-resolution reconstruction mechanism based on convolutional neural network. arXiv preprint arXiv:2407.13211.
- [4] Wu, Z., Wang, X., Huang, S., Yang, H., & Ma, D. (2024). Research on Prediction Recommendation System Based on Improved Markov Model. *Advances in Computer, Signals and Systems*, 8(5), 87-97.
- [5] Wu, Z. (2024). MPGAAN: Effective and Efficient Heterogeneous Information Network Classification. *Journal of Computer Science and Technology Studies*, 6(4), 08-16.
- [6] Zhu, Z., Wang, Z., Wu, Z., Zhang, Y., & Bo, S. (2024). Adversarial for Sequential Recommendation Walking in the Multi-Latent Space. *Applied Science and Biotechnology Journal for Advanced Research*, 3(4), 1-9.
- [7] Wang, Z., Zhu, Y., He, S., Yan, H., & Zhu, Z. (2024). LLM for Sentiment Analysis in E-commerce: A Deep Dive into Customer Feedback. *Applied Science and Engineering Journal for Advanced Research*, 3(4), 8-13.
- [8] Ji, H., Xu, X., Su, G., Wang, J., & Wang, Y. (2024). Utilizing Machine Learning for Precise Audience Targeting in Data Science and Targeted Advertising. *Academic Journal of Science and Technology*, 9(2), 215-220.
- [9] Ma, Y., Shen, Z., & Shen, J. (2024). Cloud Computing and Hyperscale Data Centers: A Comparative Study of Usage Patterns. *Journal of Theory and Practice of Engineering Science*, 4(06), 11-19.
- [10] Shen, Z. (2023). Algorithm Optimization and Performance Improvement of Data Visualization Analysis Platform based on Artificial Intelligence. *Frontiers in Computing and Intelligent Systems*, 5(3), 14-17.