Design and Implementation of Collaborative Development Platform in Cloud Environment

Wenxuan Liu[1], Nietianhe Liu[2], Weiji Yu[3], Ying Zeng[4], Shipeng Chen[5], Yi Han[6]

[1] Shandong University of Science and Technology, Department of Electrical and Information Engineering, China

[2] Guizhou University, Electrical Engineering, China

[3] Changzhou University Huaide College, Department of Information Engineering, China

[4] University of Jinan, Department of Mechanical Engineering, China

[5] Shandong University of Science and Technology, Department of Computer Science and Engineering, China

[6] Shandong University of Science and Technology, Department of Finance and Economics, China

Abstract

In order to realize a team collaborative development platform on the cloud, this paper studies cloud computing related technologies and Web related technologies to ensure that the basic functions of collaborative development can be realized based on browsers. Realize the resource management module, provide user management so that the team can better manage user members, and provide virtual resource management so as to better monitor the cloud resource pool and schedule resources; Research WebRTC technology, build signaling server, design and implement communication demonstration module based on WebRTC, and provide communication tools based on Web, including audio and video call, shared screen and online demonstration tools. The project management is designed and implemented. The creator of the repository can download the released version of the repository and support the automatic deployment of the project. Integrate the above functional modules, and design the front-end page as a single-page application to enhance the development experience of developers, and realize a browser-based collaborative development platform on the cloud, so that the team can carry out collaborative development without restriction.

Keywords

Cloud Computing; Communication and Demonstration; Development Online; Resource Management.

1. Requirement Analysis of Collaborative Development Platform

According to the needs of the team in collaborative development and the personal development needs of developers, this paper designs a set of online collaborative development platform for team developers, which helps the team to communicate efficiently and help developers to devote themselves to business logic development anytime and anywhere without interference from other things.

2. System Architecture Design

Based on the above functional and non-functional requirements analysis, from the perspective of user experience, combined with cloud technology and Web technology, the overall system architecture of collaborative development platform is designed [26]. The collaborative development platform is divided into the following parts: application layer, user authentication layer, business layer, data layer, system layer and physical layer. See Figure 3-1 for the system architecture diagram.

Application layer: responsible for showing all functions of collaborative development platform to users. The collaborative development platform adopts B/S structure, and the browser receives the user's input and requests the server to respond. As the front-end part of Web application, the application layer is responsible for showing the visitor page and login registration page when the user is not logged in. After the user logs in and authenticates, the server interface returns the user's personal data, including the workspace address and the user's personal information. The application layer gets the data returned by the server and renders it into a page to display to the user.

User authentication layer: The collaborative development platform is designed into several modules, and each module is independent of each other by modular design, so users can log in once without authentication among each module.

Business layer: Business layer is divided into application development layer and application deployment layer. Business layer is the core module of collaborative development platform, which involves code editing, file management, version control, collaborative development and application deployment. As the server of Web application, the business layer adopts Eggjs framework based on Node.js to integrate business logic, and restful style interface is used for communication between modules.

3. Front-end Architecture Analysis and Design

Because the users of this platform will interact on Web pages for a long time, in order to improve the interactive experience of users, this paper uses the front-end framework Vue to design a single-page application. Single-page applications have the following advantages:

User experience is better than multi-page application. Single-page application does not need to reload the page, but uses AJAX to dynamically obtain the server data and load it locally. Compared with reloading the web page, it saves time and can add transition animation to improve the user experience;

Good front and rear end separation. The server can focus on business logic without splicing HTML, and the server is API. The front end can render only by requesting the server to get the corresponding data, which can reduce the pressure on the server;

Based on the separation of the front and back ends [27], the server-side code can be applied to a variety of clients without modification.

Single page brings great performance optimization, but at the same time, we also need to modularize the front end to meet the requirements of single page, and do some processing on routing and status.

Because of the single page design, when users interact on the page, there is no real route jump, but the corresponding module is rendered again. This brings performance improvement and inconvenience. The URL of a Web page will remain inconvenient, and users cannot know exactly which module they have jumped to from the URL. At the same time, when dealing with logic including page forward and backward, because the page does not really jump, it may directly retreat to the interface before entering the system, which brings inconvenience to users. Based on these shortcomings, this paper introduces Vue-router to the front-end page routing management. It realizes the routing function by mapping different URLs to different modules, and adds the URL of the current page to the stack of browser's access history through the method in HTML5 History to realize the correct logic when users move forward and backward. In addition to designing routes, we can also add authentication mechanism when the page jumps, and only users who pass the permission control can enter the corresponding interface.

It is also the logic that needs special treatment brought by single-page design. Because only module-specific data is obtained during interaction, most of the data needs to be shared among modules. At this time, a global state management is needed to realize shared data. In this paper, VueX is used to realize state management. Besides sharing data globally, VueX also provides specific methods to modify data to avoid neglected data modification. At the same time, VueX also brings in the responsive data characteristics of Vue. Developers don't need to consider the rendering problem after data changes, but only need to correspond the data to the corresponding rendering module. When the data changes, the corresponding module will be updated automatically. In addition to being global data, VueX can also create a separate data warehouse for each module, and support synchronous and asynchronous requests to modify data to meet the needs of front-end development.

Front-end authentication logic in the front-end interaction, it is very necessary to judge whether the source of the web page is a normal user request, unreasonable authentication mechanism will lead to the risk of user data leakage. In the early front-end interaction, the authentication information will be put in the cookie, and the back-end will parse the cookie and verify the cookie for authentication. However, this mode will bring problems. Browsers will send requests with cookies by default, which will bring the risk of fraudulent user identity in cross-site request forgery (CSRF).

This paper uses JWT (Json Web Token) to avoid the risk of cookie. JWT consists of three parts: Header, Payload and Signatureo, where Header is used to declare the type of Token and the type of encryption algorithm, Payload is used to carry some unimportant information related to users, and the issuance time and expiration time of Token. The first two parts are plaintext, so they should not be used to store particularly important data. The last part is the ciphertext obtained by encrypting the first two parts and the key with encryption algorithm. An interceptor is added to the request. After passing through the interceptor, each request will carry Token information on the request header for back-end authentication. When the back end receives the token, it validates the token to ensure that the token is signed by the server. If the user has login-free requirements, the Token will be saved in the browser, and whether the Token expires can be verified directly when the user logs in to the web page.

4. Analysis and Design of Server Architecture

Because the system adopts the separation design of front and back ends, the back end mainly provides interfaces for the front end to call data. The back end is based on Eggjs, which is an asynchronous message queue when designed, so it has advantages in high concurrency, so there is no need to over-design high concurrency requirements. The main design of back-end is authentication design and cache design.

1) The server-side authentication design system needs to do authentication processing for every request, so the back-end uses authentication middleware to achieve.

Every request goes through all middleware to reach the business tier when it arrives. This paper designs an authentication middleware. When the request comes, first verify whether there is a agreed request header in the request. If there is, verify the value of the request header. If the verification is successful, it will enter the specific business module. If the verification fails, it will return the status code with invalid Token to the front end, and the front end page will jump accordingly. If there is no request header, it will check whether there is a transmission user name and password. If there is, it will verify the user. After the verification is successful, it will issue a Token and allow the request to reach the corresponding business logic. If there is no request header, it will return the status code with insufficient authority to the front end, and the front end page will jump to guide the user to log in.

2) Cache design

Using Redis to store some data in memory can greatly improve the speed of reading data, thus improving the concurrent processing ability of the back-end system. At the same time, the addition of cache can effectively protect the database and avoid excessive database traffic. The cached data mainly includes two aspects:

On the one hand, it is the cache of user login status. Because of the characteristics of JWT, a JWT will be valid until the expiration time after it is issued. In business, we need to delete the Token and force it to expire before the expiration date. Therefore, the Token information will be cached in Redis after the user logs in, and it needs to be verified in Redis every time after verifying the Token.

On the other hand, it is the cache of data generated by the user's business logic. When the business logic needs some data, it will find it from the cache first. If it finds it, it will not send a request to the database. If it cannot find it, it will continue to find it in the database. In addition to reading data faster, the cache is optimized by adding Bloom filter, which stores the existing Key and makes a layer of interception. A Key that passes through a Bloom filter does not necessarily exist, but a Key that does not pass through a Bloom filter does not necessarily exist. Through this layer of interception, we can effectively avoid the impact of many malicious requests on system performance.

5. Tag

According to the problems and needs of team developers under the epidemic situation, combined with the development status of the corresponding online development platform, this paper explains the necessity and value of the development of this system. Then the cloud computing technology and web development technology are introduced, and the technical feasibility of collaborative development platform based on browser is determined. Then, the requirements of the system are analyzed, and the system is designed and implemented. The goal of team collaborative development can be achieved anytime and anywhere. The main results of this paper are as follows.

References

[1] Xu Baomin, Ni Xuguang. Development trend and key technology progress of cloud computing J]. Proceedings of Chinese Academy of Sciences, 2015, (02): 170-180.

[2] C. Prithwiraj, K. Crowston, D. Linus, et al. GitLab: work where you want, when you wantJ]. Journal of Organization Design, 2020, 9 (1): 23.

[3] C. Pahl. Containerization and the paas cloud [J]. IEEE Cloud Computing, 2015, 2 (3): 24-31.

[4] M. Malawski, M. Kuzniar, P Wojcik, et al. How to use Google app engine for free computingJ]. IEEE Internet Computing, 2013, 17 (1): 50-59.

[5] C. Newcombe, T. Rath, F. Zhang, et al. How amazon web services uses formal methodsJ]. Communications of the ACM, 2015, 58 (4): 66-73.

[6] J. Solin. Technology that educators of computing hail (tech) using cloud 9, a powerful cloudbased ide in the classroom [J]. ACM Inroads, 2017, 8 (1): 29-30.

[7] Xu Ziyan, Guo Mengjie. Overview of Microsoft Cloud Computing Platform Azure (I) [J]. Programmer, 2010, (3): 109-112.

[8] B. S. Dordevic, S. P Jovanovic, V V. Timcenko. Cloud computing in amazon and microsoft azure platforms: performance and service comparison [C] Telecommunications Forum Telfor (TELFOR). IEEE, 2014, 931-934.

[9] Zhu Erhua. Research on Web Front-end Application Based on Vue. js [J]. Science and Technology and Innovation, 2017, (20): 119-121.

[10]J. Loeliger, M. McCullough. Version control with git: powerful tools and techniques for collaborative software development [M]. O 'Reilly Media, Inc., 2012, 31-46.