

Business Account Management and Statistical Report Analysis System Based on Front-end and Back-end Separation Framework

Ran An, Xiao-kui Xing, Shu-kun Li, Han Jia and Hong-shu Ye

Beijing Smartchip Microelectronics Technology Company Limited, Beijing, China

Abstract. By integrating the needs of each team, combining the previous development experience and the current mature development framework mode, this paper proposes to build a business account management and statistical report analysis system based on the front-end and back-end separation framework, which can be used by each team to improve work efficiency and enhance the reusability and practicality of various marketing data. This paper introduces the web directory structure of the whole system and the contents and related operations displayed on the corresponding management pages. For the needs and problems raised by users, the system adds corresponding functions and pages to solve them. The back-end of the marketing data management system is based on J2EE framework and developed by springboot+mybatis. The front-end is developed by react + ant design + node + echarts+ less.

Keywords. Business account; Front-end and back-end separation; Reusability; Springboot; React

© 2022 by The Authors. Published by Four Dimensions Publishing Group INC.
This work is open access and distributed under Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous development of the company's business, the volume of sales related business is several times that of a few years ago. There are a wide variety of projects, the number of salesman increases, and the business processes involved become more cumbersome with the upgrading of the company's management requirements. The original manual collection and filling of relevant project and material data are becoming more and more complex, resulting in a doubling of the workload of relevant personnel. At the same time, the accuracy of the filled relevant data is also significantly reduced with the continuous increase of data. In addition, the current progress cannot be queried in real time for the data that has been filled in and approved, which seriously affects the work efficiency of all links. Therefore, it is urgent to develop a system that can operate online and query the approval progress in real time. By sorting out the processes of each business team, the following problems exist in the circulation and statistics of various marketing documents:

(1). There is no tool to effectively connect various accounts, forming "data islands" one by one. A

large number of valuable data resources can not play a greater role and are not interconnected.

(2). There is a delay in data statistics. Business data information such as shipment, contract, payment collection and invoicing need to be delivered to support personnel for batch processing. Before the processing is completed, an account cannot be generated to generate data. If the delivery manager submits the data in the processing process in advance, the data will be inconsistent.

(3). The implementation of the new task cannot be linked to the sales situation of the bidding and formed contract in real time. Each business team needs to repeatedly capture the data statistics. However, the statistical results are inconsistent due to the inconsistent dimensions and principles captured by each person in the process.

(4). The order forms required by the products in the charge of each department are inconsistent. Different products require manual editing of a variety of order and delivery forms, which increases a lot of workload.

(5). All data in the management service account output by the delivery support personnel every week are manually edited, which consumes a lot of time and energy. Since there is no system support, data search, attachment preparation, audit and modification, and account compilation at each step (such as shipment / contract) take up a lot of time. Therefore, it is necessary to build a marketing management system to improve efficiency.

In view of the above problems, combined with the management effect and operation experience of relevant personnel, the R & D personnel put forward a design scheme of business account management and statistical report analysis system based on the separation of front-end and back-end. The front-end framework is react and the back-end framework is springboot. The functions of the client and server are separated, which can quickly find and locate problems and improve the maintainability of the overall code.

This paper is the front-end page design of the business account management and statistical report analysis system based on react architecture. This paper mainly expounds and explains the front-end and back-end separation development mode, and then analyzes the design idea of react. Finally, the function and operation of the front-end page of the system are introduced in detail.

2. Main Body

2.1. Front-end and back-end separation

With the rapid development of Internet and the upgrading of software development technology, the separation of front-end and back-end has become the industry standard way of Internet project development. In practical work, the workload of interface joint commissioning and docking at the front-end and back-ends accounts for 30-50% of the daily work of front-end personnel, or even longer. The front-end and the back-end are developed independently. They are placed on two different servers and need to be deployed independently. Two different projects, two different code bases, different developers, and front-end and back-end engineers need to agree on interactive interfaces to achieve synchronous development. After development, it needs to be deployed independently. The front-end calls the API of the back-end through the interface. The front-end only needs to pay attention to the page style and the parsing and rendering of dynamic data, while the back-end focuses on the specific business logic. The specific process is shown in the figure:

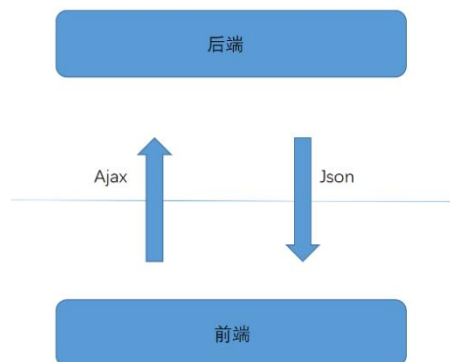


Figure 1. Front-end and back-end separation.

Advantages of front-end and back-end separation:

- (1). Clear division of labor to improve work efficiency. The front-end only cares about the work of the front-end and the background only cares about the work of the background. The development of the two can be carried out at the same time. When the background has not been written or the interface has not been provided, the front-end can first write the data or call the local JSON file. The addition of pages and the modification of routes do not have to go through the background, so that the development is more flexible.
- (2). The front-end was completely liberated. The front-end no longer needs to provide specific templates to the background or embed relevant background code in the front-end HTML in the background.
- (3). Reduce maintenance costs. Through the current mainstream front-end mvc/mvvc and other mode frameworks, we can quickly locate and find the problem. The problem of the client does not need the background personnel to participate in and debug, and the code reconfiguration and maintainability have been significantly enhanced.
- (4). Local performance improvement. Through the front-end routing configuration, we can load the page on demand, without loading all the resources of the website at the beginning of loading the home page, and the server no longer needs to parse the front-end page, which will greatly improve the page interaction and user experience.
- (5). Even if the back-end service temporarily times out or goes down, the front-end pages will be accessed normally, but the data cannot be provided to improve the user experience.
- (6). It can make the developers better pursue high concurrency, high availability and high performance, and the front-end better pursue page performance, smooth speed, compatibility, user experience, etc.
- (7). Realize high cohesion and low coupling, and reduce the concurrency / load pressure of the back-end (application) server.

2.2. React front-end frame

React framework is an internal project originally developed by Facebook. The framework adopts a declarative design idea, making it efficient and flexible to use. With its unique design ideas and outstanding performance, react has been favored by the majority of developers to become a mainstream framework for web front-end development [1].

The design idea of react is mainly reflected in the following three aspects:

- (1). Componentization. The react framework reconstructs the UI interface in the form of components.

Developers divide the UI interface into different components from the perspective of functions. Each component only needs to care about the logic inside the component and is independent of each other.

(2). Focus on the layer. React focuses on providing concise and clear view layer solutions, and also includes a library of controllers and vision. For systems with complex business logic, developers can choose their own business logic framework according to the actual situation, and use it with flux, redux, etc. according to their needs.

(3). Efficient rendering. The developers of react introduced the virtual DOM mechanism. When the current component needs to be updated, react will create a new virtual Dom and compare it with the previously stored DOM through the diff algorithm. It only modifies and loads the DOM that needs to be changed, greatly reducing the loading amount and loading time.

The front-end architecture of the business account management and statistical report analysis system adopts a three-tier architecture of view layer, behavior layer and data layer. The three-tier architecture performs its own duties, and the view layer is mainly responsible for processing the view display; The behavior layer triggers different operation behaviors according to different interaction instructions of front-end users; The data layer handles the behavior logic by listening to the behavior type and responding to the callback function. At the same time, it sends API requests to the server asynchronously to update its own data; Finally, the view layer reflects the data changes to different views for display by listening to the data layer.

When the user accesses the web interface of the management system through the browser, the proxy server will automatically send the page request to the file server in the server cluster through HTTP, and then the file server loads the JavaScript script and CSS file of the system home page.

When the user needs to access other interfaces, react will automatically re-render the display interface according to the direction of the jump page without re-establishing the HTTP request. When the user request involves back-end data, the proxy server sends the API request to the database server, performs operations such as adding, modifying, deleting and querying the database, and returns the required data to the front-end through JSON files.

2.3. Function introduction

The system is divided into five user roles. Each role has different functions and corresponding permissions and page operations:

Table 1. Roles and permissions of system.

Roles	Accounts	Permissions
Administrators	Preset in database	System management, assigning user roles, menus, etc.
Delivery Manager	Feishu account	Basic data management and business management.
Regional Manager	Feishu account	Basic data management, business management, and view the business information of personnel in the region.
Delivery supporter team manager	Feishu account	Basic data management, business management, and view the business information of all personnel.
Delivery supporter	Feishu account	Basic data management and business management.
Marketing Administrator	Feishu account	Basic data management and business management.

2.3.1. Basic data management

Table 2. Module and introduction of basic data management.

Primary function	Introduction
Role management	The system is divided into five roles: administrator, delivery manager, regional manager, team manager, delivery support and marketing service administrator; Administrators can create roles, users, and organizational structures; The delivery operation and maintenance manager and the delivery support can use their respective business functions.
User management	Access in the form of web pages to realize the role assignment of different personnel and the operations of adding, modifying and deleting users. The administrator can manage other personnel, including the functions of adding, deleting, changing, and resetting passwords.
Organization management	Based on the company's organizational structure, manage the organization, including adding, deleting, modifying and querying.
Production management	It contains all the product information, which can be imported through Excel and covered in full; All data can be modified and deleted only with delivery support.
Customer management	All product information can be imported through Excel. If the new unified credit code is the same as the previous data, it is not allowed to import and prompts. Other information can be imported normally; All data can only be modified and deleted by the creator and delivery support.
Dictionary management	The data items used in the database are stored, and all kinds of data in the database are collected together for development and maintenance.

2.3.2. Business management

The main business processes include: bidding and tendering (optional) → contract management → shipment management → invoice application → payment collection management → goods preparation → after-sales.

2.3.3. Build front-end architecture and front-end pages

The system introduces node JS, install relevant dependent packages, and generate the file structure of the front-end project, as shown in the figure:

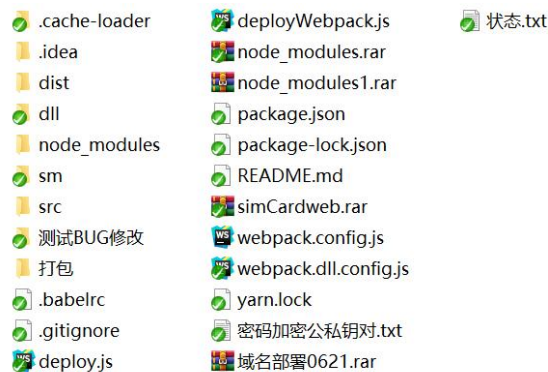


Figure 2. File structure of the front-end project.

The dist folder contains the files after packaging and deployment. The SRC folder contains the entire project page and related configuration methods. Package The startup, packaging methods and imported dependent versions are configured in the JSON file. Webpack.config.JS configures the entry of the whole project and the related front-end startup project port. The front-end local startup port of the project is 8800.

In src folder, which contains the pages of the whole system and the control logic and display style of each page; The images folder contains the picture files introduced in the project, including the background pictures of the login page, icons of some pages, etc; The components folder encapsulates public components, including basic table forms, pagination, pie charts, bar charts, line charts, bottom of pages, and selection tree; The virtual data stored in the mocker folder that convenient for the front-end to splice its own data without the background interface, so as to improve the development efficiency; The api folder is the overall configuration of the background interface, the front-end sending request interceptor, and the back-end returns data. It encapsulates a unified processing method to facilitate the unified processing and calling of data; Index.js and index.html files are the entry of front-end engineering page; The file style.less is the global style variables of the whole project that configured in less, which is convenient for unified retrieval in each page and is conducive to later maintenance; Router.js is used to configure the route of the entire front-end project and jump between pages. The directory structure is shown in the figure below:

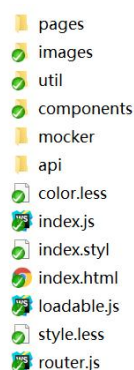


Figure 3. Structure of src folder.

2.3.4. Specific pages and functions

Log in to the system and enter the home page. The home page displays the list of unprocessed

messages of the currently logged in user, which includes bid management, contract management, agreement management, shipment management, invoicing management, stock management and other related types of information. Click the "process" button to directly jump to the operation interface of the pending items on the corresponding management page, as shown in the figure:

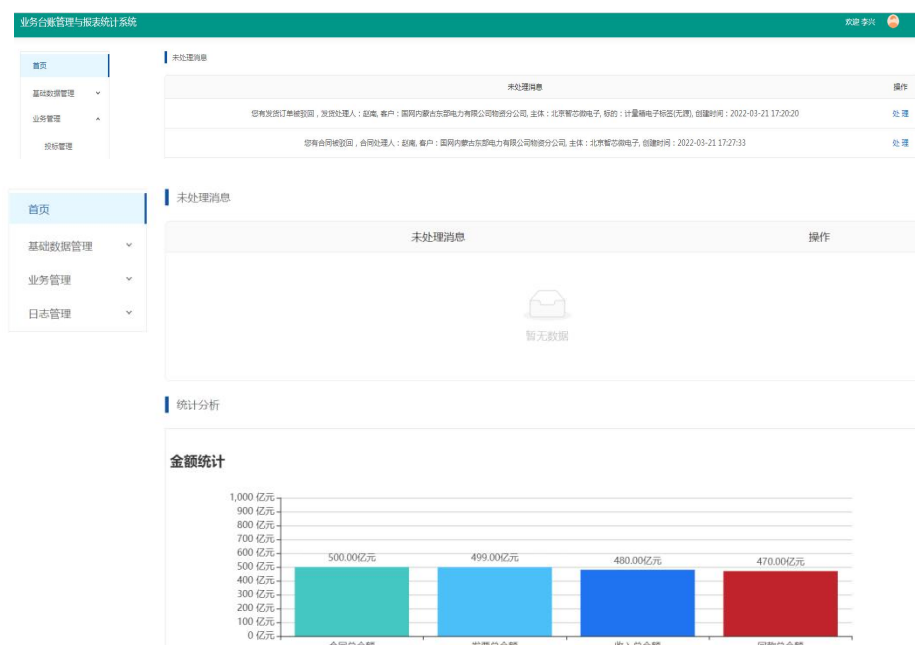


Figure 4. Home page of system.

In addition to various data operations on various management pages, you can also view the node and progress of a specific operation through the view process button, as shown in the figure:



Figure 5. The node and progress of operation.

4. Conclusion

This project is an internal R & D project of the company. The whole system integrates the product, template, customer, bidding, contract, shipment, invoice application, payment collection, after-sales, goods preparation and other relevant data information of the market delivery department, and achieves the following expected goals:

1) The accounts of all nodes in the whole process of the sales contract are digitized, the template is

unified, and the process progress can be queried.

2) Automatically generate comprehensive reports, including bid winning rate statistics, single order analysis statistics, employee / team performance indicator achievement statistics and product line sales statistics, so as to provide decision support for relevant personnel.

3) Secure access to the system through VPN, single point integrated login, access to the core + platform, and access through the core +.

The business account management and statistical report analysis system based on the front-end and back-end separation framework reduces most of the redundant work of market delivery personnel, changes the previous pure manual entry of document information, greatly improves the accuracy and efficiency of related work, and provides online operation functions for personnel in various roles to facilitate relevant personnel to examine and control related marketing data from a higher and more macro perspective, It points out the direction for the future marketing content and business classification, and perfectly solves the current problem of "separate businesses, data islands". See the figure for details:

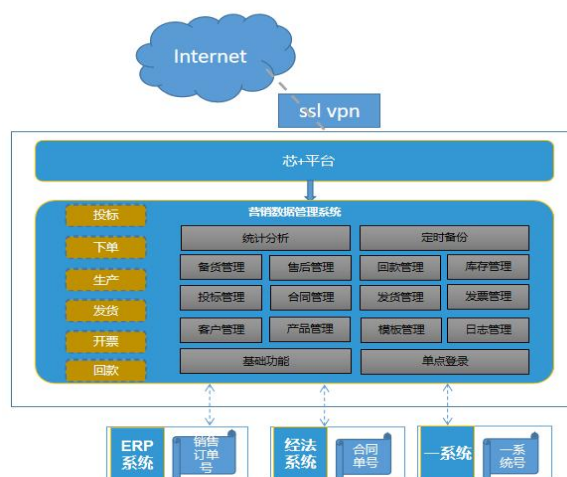


Figure 6. The business function of system.

References

- [1] Hui Qi, Xiaoqiang Di, Lin Bi, etc Research on the construction of MOOC mobile learning platform based on react [j] Modernization of education, 2016 (38): 261-262