

Study of Multi-Tier Main-Detail Table Data Persistence Scheme

Yifei Li, Lixia He, Botao Liu*, Kangsheng Yin, Rui Tang, Yuhui Zhang, Yu Hu

Yangtze University, Jingzhou 434000, China

* Corresponding Author: liubotao920@163.com

Abstract

In the context of big data, both the government and large enterprises have established their own employees, users or business data centers. With the development of the Internet, more and more people began to contact the Internet, and the business of the platform has become more and more complex. In order to avoid the coupling of data input in each layer, the scheme abstracts the data input in each layer into different modules to reduce the coupling degree of the scheme to a certain extent, and only open the data interface between each module to the public to integrate the data from the bottom up, finally forming multi-level data. In this paper, we introduce four kinds of Vue component communication technologies to realize data communication between each component module, and use localStorage technology to realize data merging solution in each level. In this paper, we introduce four Vue component communication technologies to achieve data communication among component modules, and use localStorage technology to realize the data merging solution at each level. The solution has been applied to the "milk loading platform" system, and it has been proved to be feasible and convenient for administrators to input data from multiple levels of main and detail tables.

Keywords

Database; Vue; Front and Back-End Separation; Main-Detail Table.

1. Background

With the advent of the Internet era, the number of e-commerce and online sales is increasing, not only the traditional retail industry has both opportunities and challenges, the logistics industry has also ushered in a golden period of development, and the scale of logistics is expanding [1]. For this reason, the state and major logistics giants are promoting the development of urban smart logistics technology, which is a modern logistics model that applies the Internet of Things (IoT) technology to implement refined, visualized and dynamic management methods for all aspects of logistics. In smart logistics, it involves the problem of transporting and loading of goods, and in order to realize smart loading and automated loading, it is necessary to have vehicle information data, goods data, and goods coordinate data. The vehicle is the main, the items loaded in the vehicle are the detail, and the coordinates of the items stacked in the carriage are the detail table of the items loaded in the vehicle, forming a three-layer main-detail table data. Finally, according to the vehicle, and the loaded goods, goods coordinates for the robot arm automatic packing, reduce human error, increase logistics efficiency, reduce logistics congestion time.

The logistics industry is developing towards comprehensive intelligence, and in the future intelligence will also be applied to the whole system of logistics to achieve intelligent unmanned logistics supply chain [1]. Thus, for the case of multi-layer main-detail table, it has an important and common application in intelligent logistics.

2. Vue Component Data Communication Technology

2.1. Localstorage

localStorage is a way of browser local caching provided by HTML5, extending the 4K limit of cookies, with a size limit of 5M, which is inconsistent across browsers [11], and using a simple key/value pair storage format, providing developers with more manipulable space. localStorage local storage method of application eliminates the tedious operation of connecting to a database. [4].

localStorage can be effective for a long time, neither active clear will not disappear, not because the browser is closed and data loss, open again, as long as meet still in the same scope (the same protocol, host number, port number) can again access / modify the localStorage. independent storage of each domain, so that data storage, management brings good convenience. Due to the convenient data storage function of localStorage, the data acquisition module of this solution mainly stores and aggregates data of multiple main and detail tables through localStorage, and finally completes data submission in a unified manner.

2.2. Vuex

Vuex is a state management pattern developed for Vue. When building a large single-page application, Vuex can help developers better manage state outside of the component in a uniform way [5] and ensure that state changes in a predictable way with corresponding rules, but when using global state management, we also need to distinguish between component local state and application-level state. However, when using global state management, we also need to distinguish between component-local state and application-level state to avoid the possibility of putting the component's original state into application-level state [6].

Vuex can use Getter to get state, but the process of modifying state is more complicated. As shown in Figure 1, modifying state requires the use of Actions, which call Mutations asynchronously, and Mutations are the only way to modify state inside Vuex. The reason why Vuex is so complicated to modify state is to ensure that modifying global variables can be rendered more efficiently and in a more timely manner than using global variables directly [3]. However, in this scenario, which involves data exchange in only some of the components of a module, using Vuex would be an overkill.

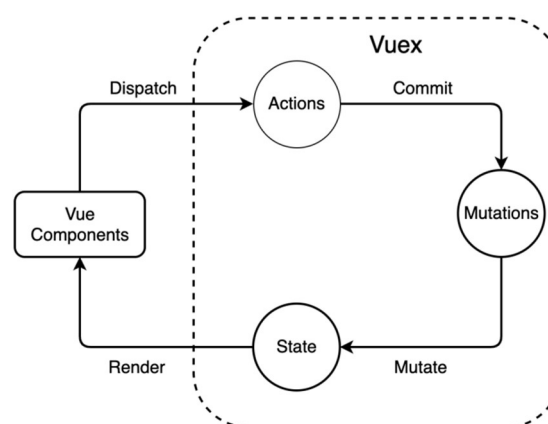


Figure 1. Vuex call flow

2.3. Emit/On

2.3.1. Emit

In Vue, `vm.$emit(event Name, [... .args])` triggers the event on the current instance. Additional arguments are passed to the listener callback. Used for child components to pass data to the

parent component, as in this scenario, the data from the second level main-detail table is invoked via \$emit for the main table with custom events with additional parameters.

2.3.2. on

In Vue, `vm.$on(event,callback)` listens for a custom event on the current instance. Events can be triggered by `vm.$emit`. The callback function takes all the additional arguments passed to the event trigger function.

3. Data Storage Solutions

3.1. Database Design

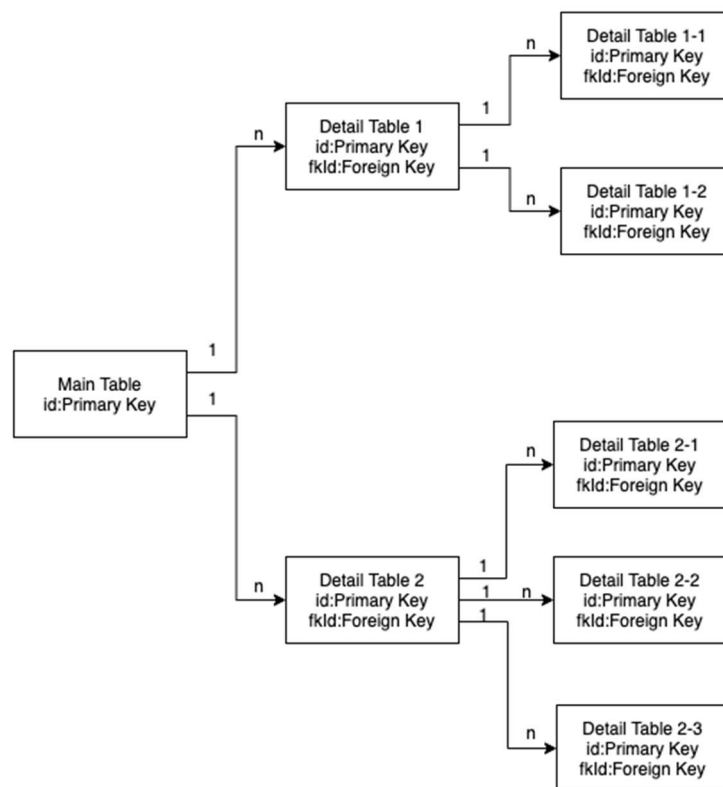


Figure 2. Database main table and detail table association

In the design of multi-tier main-detail table database tables, each tier only needs to focus on whether there is a table on the previous tier, which is a bottom-up design approach. The key to main-detail table technology is how to display the corresponding detail table information by selecting a record of the main table, which requires the foreign key of the database corresponding to this record to be read out at the moment the main table record is selected [7]. Each level of the table will have an id as the primary key, which is the identifier of this piece of data. In this way, both the detail table and the detail of the detail table can use this unique identifier as their own fkId to determine which level the data belongs to. Take the above table 2-1, table 2-4 and table 2-5 for example, these three tables are the same level of detail tables, all belong to the detail table of table 2, then table 2-3, table 2-4 and table 2-5 do not need to pay attention to the table above table 2. Only need to associate their own fkId to the id of the detail table 2, so as to achieve the entire multi-layer main-detail table data association, the final only through a main table id, you can find multiple fkId and its equivalent data in the detail table 1, detail table 2, and then through multiple detail table id, in each detail table to find the same fkId data, and so on, and finally can successfully query a set of multi-layer main-detail table data table data.

3.2. Data Persistence Methods

General main-detail table entity class design, each table of the database corresponds to an entity class. All the fields under each table correspond to the member variables of the entity class. This paper uses MyBatis for persistence and MVC architecture for data persistence. MyBatis is an excellent persistence layer framework [8], MyBatis was formerly known as iBatis [9] and was renamed to MyBatis in 2010, MyBatis is a Java-based persistence layer framework that supports SQL statement queries, MyBatis solves the problem of manual configuration of parameters in the JDBC code in Java classes and the problem of manual retrieval of result sets. manual retrieval problem by using simple XML configuration or annotations for configuration and mapping, MyBatis can use XML configuration files to create database connection pools so as to avoid the hard coding problem of database connection pools for JDBC. [10] For the common case, the backend needs to save the json data from the front-end in the entity class corresponding to the database to manipulate the data. But for the special case of multiple main-detail tables, we will design two Page classes to receive the data from the front-end, and the relationship between the receiving data classes is as follows.

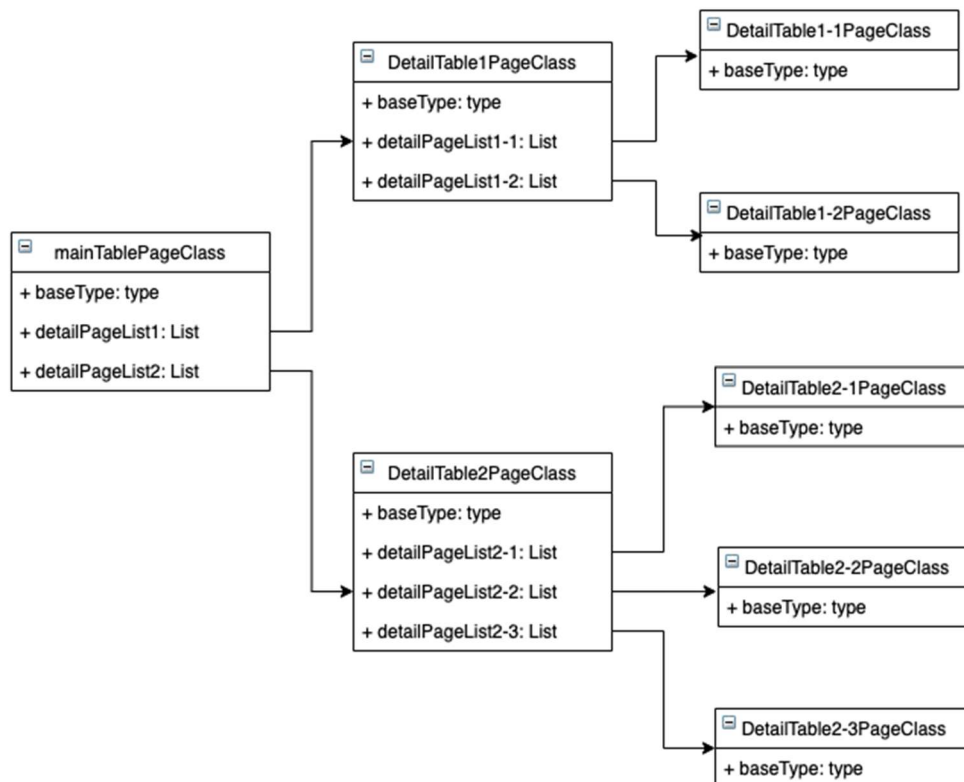


Figure 3. Receiving data class relationship diagram

Taking Fig. 3 as a reference, the final data format can be obtained as follows:

```

{
    Main table database basic fields,
    detailPageList1: [{
        Detail table database basic fields,
        detailPageList1-2: [{Detail Table 1-2 Database Basic Fields}
    detailPageList1-2: [{Detail Table 1-2 Database Basic Fields}],
    }],
}
  
```

```

detailPageList2: [{
    Detail table database basic fields,
    detailPageList2-1: [{Detail Table 2-1 Database Basic Fields }],
detailPageList2-2: [{Detail Table 2-2 Database Basic Fields }],
detailPageList2-3: [{Detail Table 2-3 Database Basic Fields }],
}],
}.

```

To do this, all accepted data can be stored in an entity class, because the main table Page object is not directly invoked method directly into the library. So we need to abstract the data to the entity class for the storage.

We use BeanUtils.copyProperties to encapsulate the data from the Page class into the respective entity class so that we can directly store the data from the table. Similarly, for from the table PageList, each object from the table PageList will be taken out separately, each from the table Page object with the database corresponding to the member variables and from the table List, the object in the member variables encapsulated in the corresponding entity class with the database. But here we should note that the json data given by the front-end does not have the value of foreign keys from the table. So we need to use the primary key id of the main table as the value of the foreign key of the detail entity class object. Then the detail table data can be imported. Finally, we deal with the detail list, because each object in the detail list is actually an entity class object corresponding to the database, so we can directly into the library, but note that the json data given by the front end does not give the value of the foreign key in the detail table. So we need to take the id from the table as the value of the foreign key corresponding to the entity class object from the table. Then we can call the method into the library.

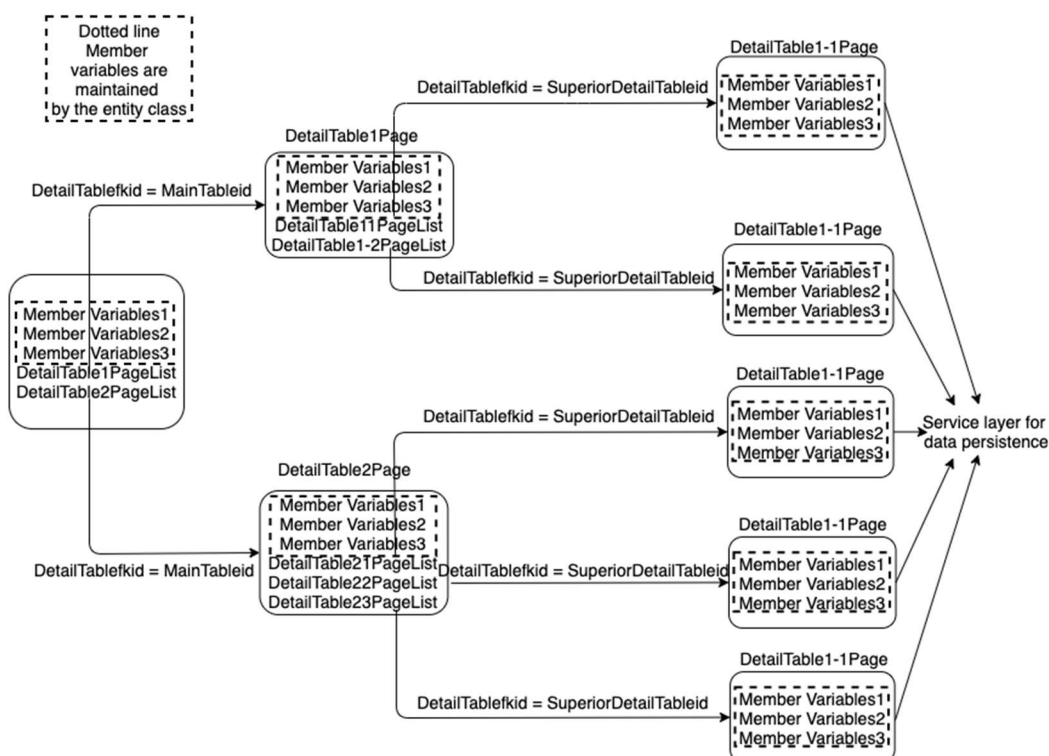


Figure 4. Data persistence detailed diagram

For the operation part of multi-layer relational data, the delete function can facilitate top-down through the main table id to delete the data whose fkid is equal to its main table id, realizing

multi-level iterative deletion. Update function, you can compare the data to be in the library with the data already in the library, delete or add. This way of violent comparison is more energy-intensive. Therefore, for the update function, we use the way of deleting before inserting. First delete all the data according to the id, and then add the data again. Query function, directly through the main table id, iterative search from the table, from the table fkid, and in the iterative process can be inserted one by one into the PageList data, forming a multi-layer object.

4. Test

Input Data:

Add

* Planned loading time: 2021-04-14 * number plate: EA-7688

* Loading plan type: 1

Available space in carriage Waybill Loading plan coordinates

+ Add

Selected 0 Xiang cleared

	#	high	width	depth	operate
☐	1	10	20	30	edit More v

Available space in carriage Waybill Loading plan coordinates

+ Add Batch operation v

Selected 0 Xiang cleared

	#	Shipment number	Sold-to party	delivery date	Delivery party	Delivery party address	Contact person	contact num
☒	1	95251	Rhyme	2021-09-02	Test Delivery party	Test Delivery party address	Test Contact person	Test contact r

1-1 A total of 1 < 1 > 10 items/page v

Order delivery

+ Add

Selected 0 Xiang cleared

	Delivery note number	storehouse	Product Specifications	Print date	order number	unit	quantity	Remark	operate
☐	Test Delivery note number 1	Test storehouse1	Test Product Specifications1	2021-09-23	Test order number 1	Test unit1	20		edit More v
☐	Test Delivery note number2	Test storehouse2	Test Product Specifications2	2021-09-18	Test order number2	Test unit2	30		edit More v

Available space in carriage Waybill Loading plan coordinates

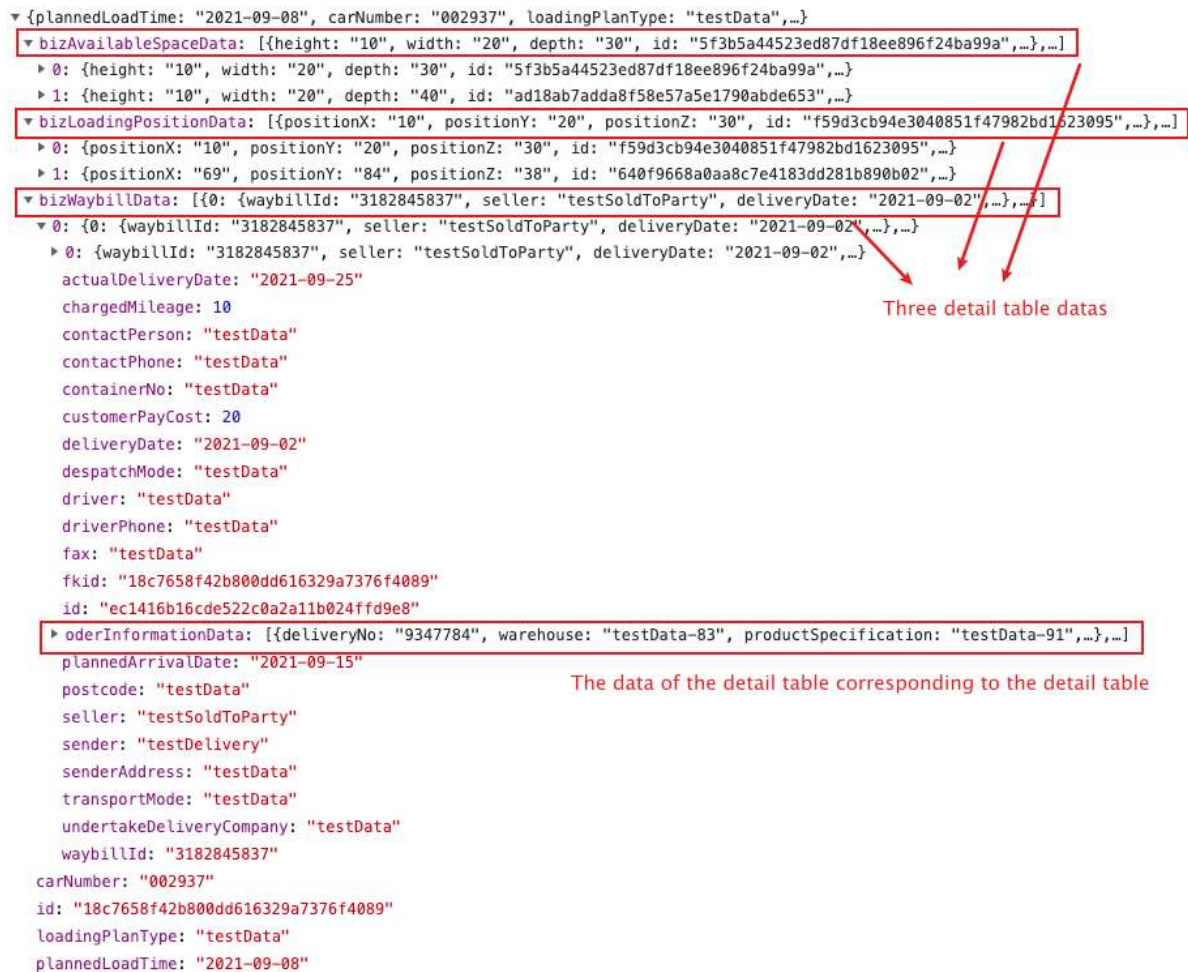
+ Add

Selected 0 Xiang cleared

	#	x coordinate	y coordinate	z coordinate	operate
☐	1	10	20	30	edit More v
☐	2	20	30	40	edit More v

Figure 5. Data entry diagram

The final submission results, the final results conform to the expected form of data.



```

{
  "plannedLoadTime": "2021-09-08",
  "carNumber": "002937",
  "loadingPlanType": "testData",
  "bizAvailableSpaceData": [
    {
      "height": "10",
      "width": "20",
      "depth": "30",
      "id": "5f3b5a44523ed87df18ee896f24ba99a",
      "..."
    },
    {
      "height": "10",
      "width": "20",
      "depth": "30",
      "id": "5f3b5a44523ed87df18ee896f24ba99a",
      "..."
    },
    {
      "height": "10",
      "width": "20",
      "depth": "40",
      "id": "ad18ab7adda8f58e57a5e1790abde653",
      "..."
    }
  ],
  "bizLoadingPositionData": [
    {
      "positionX": "10",
      "positionY": "20",
      "positionZ": "30",
      "id": "f59d3cb94e3040851f47982bd1623095",
      "..."
    },
    {
      "positionX": "10",
      "positionY": "20",
      "positionZ": "30",
      "id": "f59d3cb94e3040851f47982bd1623095",
      "..."
    },
    {
      "positionX": "69",
      "positionY": "84",
      "positionZ": "38",
      "id": "640f9668a0aa8c7e4183dd281b890b02",
      "..."
    }
  ],
  "bizWaybillData": [
    {
      "waybillId": "3182845837",
      "seller": "testSoldToParty",
      "deliveryDate": "2021-09-02",
      "..."
    },
    {
      "waybillId": "3182845837",
      "seller": "testSoldToParty",
      "deliveryDate": "2021-09-02",
      "..."
    },
    {
      "waybillId": "3182845837",
      "seller": "testSoldToParty",
      "deliveryDate": "2021-09-02",
      "..."
    }
  ],
  "actualDeliveryDate": "2021-09-25",
  "chargedMileage": 10,
  "contactPerson": "testData",
  "contactPhone": "testData",
  "containerNo": "testData",
  "customerPayCost": 20,
  "deliveryDate": "2021-09-02",
  "despatchMode": "testData",
  "driver": "testData",
  "driverPhone": "testData",
  "fax": "testData",
  "fkid": "18c7658f42b800dd616329a7376f4089",
  "id": "ec1416b16cde522c0a2a11b024ffd9e8",
  "oderInformationData": [
    {
      "deliveryNo": "9347784",
      "warehouse": "testData-83",
      "productSpecification": "testData-91",
      "..."
    },
    {
      "deliveryNo": "9347784",
      "warehouse": "testData-83",
      "productSpecification": "testData-91",
      "..."
    },
    {
      "deliveryNo": "9347784",
      "warehouse": "testData-83",
      "productSpecification": "testData-91",
      "..."
    }
  ],
  "plannedArrivalDate": "2021-09-15",
  "postcode": "testData",
  "seller": "testSoldToParty",
  "sender": "testDelivery",
  "senderAddress": "testData",
  "transportMode": "testData",
  "undertakeDeliveryCompany": "testData",
  "waybillId": "3182845837",
  "carNumber": "002937",
  "id": "18c7658f42b800dd616329a7376f4089",
  "loadingPlanType": "testData",
  "plannedLoadTime": "2021-09-08"
}

```

Three detail table datas

The data of the detail table corresponding to the detail table

Figure 6. Transformed JSON data

5. Conclusion

The multi-tier main-detail table form of data can effectively reduce the separation of business modules and facilitate the simultaneous input of highly interrelated data. With the development of new national infrastructure, more and more government departments and enterprises are customizing their own information center platforms, which include many complex and highly interrelated businesses. The multi-layer main-detail table data persistence scheme proposed in this paper can effectively solve this part of the problem.

For subsequent work, this module can be extracted into highly reusable components to cope with more complex scenarios, so that it can really be written once and used everywhere.

Acknowledgments

This research is supported by five projects: The demonstration construction projects of "Ideological and Political Course and Ideological and Political Course" in Yangtze University in 2021(No.41); The project of young people in the Education Hall of Hubei(No.Q20161311);The Yangtze Youth Fund(No.2015cqn53); The Yangtze University Students' Innovation and Entrepreneurship Training Program Project(No. Yz2020157, Yz2020150).

References

- [1] M.J.Zhang: Characteristics, current situation and strategies of Smart logistics in China under the background of new retail[J]. Business economics research. 2021(04):43-45(In Chinese).
- [2] X.Q.Xie: The Application of Internet of Things Technology in the Construction of Smart City, Management & Technology of SME(Mid-day). 2021(03):189-190(In Chinese).
- [3] D.Y.Wu: Product Lifecycle Management System based on Industrial Internet of Things Technology [D]. Zhejiang University, 2020. (In Chinese).
- [4] C.X.Zhou: Research on the application of JavaScript in message board teaching case[D]. Education Forum, 2020(08):165-166. (In Chinese).
- [5] Z.P.Zhang, S.J.Huang, L.Y.Jiang, J.M.Ma: Design and Implementation of Vehicle Insurance Quotation System Based on Vue.js[A]. Microcomputer Applications, 2020:70-74. (In Chinese).
- [6] L.Wang, B.L.Cui, H.X.Pan, L.Zhao, Y.Tian: Research and implementation of online design open platform based on vue.js[A]. Information technology and informatization, 2019:168-170. (In Chinese).
- [7] Q.W, H.L.Qi, S.X.Su: The Application and Application of Master-Slave Table Technology Based on the Research of Network Structure[A]. China Electric Power Education, 2007:198-201. (In Chinese).
- [8] H.L, M.L: Design and Implementation of Customer Management System Based on SSM Framework [J]. Software Guide, 2017, 16(7):87-89.(In Chinese).
- [9] Y.Z.Chen, Z.L.Zhang: JavaEE framework technology (SpringMVC+Spring+MyBatis) [M]. Xi'an: Xidian University Press, 2018:1-3.(In Chinese).
- [10] Y.Z.Chen, Z.L.Zhang: The Application of MyBatis and Spring in the Link Management of Library Digital Resources[A]. Xi'an: Inner Mongolia Science Technology and Economy, 2021:78-79. (In Chinese).
- [11] H.L.Huang: Talking about the local storage of mobile projects [A]. Computer Knowledge and Technology, 2015:49-51.(In Chinese).