2016 International Conference on Electrical Engineering and Automation (ICEEA 2016)
ISBN: 978-1-60595-407-3

The Design and Application of Front End MVVC Framework Based on Management Platform

Jia-ying WANG and Fang DENG*

School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding author

Keywords: Front-end, MVVC, JSON, Session storage.

Abstract. With the continuous development of computer technology, which makes mutual experience become increasingly popular and the network platform with various applications has become a trend, resulting in more complex logic of front-end business. Therefore, the standardization of front-end code becomes more than important. Under the current industry background and development pattern of MVC in front-end, this paper is to propose a MVC-like development model based on the front end, combined with the needs of the portal site, in order to clear the complex logic of front-end code. Then summarize the application scenarios and applied in the management system.

Introduction

With the increasingly sophisticated site design, the logic of front-end code is more than complex, which lead to the rendering of front-end with UI plugin in a large scale and a huge amount of information interaction [2]. But the corresponding development model is that the data is often interacted through AJAX between the front and server ends after receiving the CDN request of static resources. However, in this mode, the pressure in server end is transferred to the front end partly, making a part of JavaScript complicated [3]. So when developing the application module, we try to make the JavaScript code separated according to the functions, in order to enhance the logic of the code.

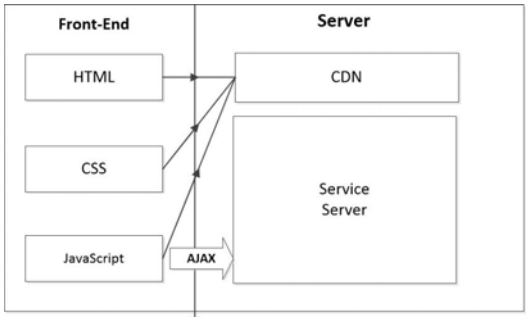

Figure 1. Traditional Development Model.

Brief Instruction of MVC Design Pattern

MVC is a design pattern that divides applications into three parts: data layer (model), presentation layer (view), and user interaction layer (controller) [4]. As we know, the occurrence of an event is such a process as below:

1. User and application interaction.
2. Trigger the controller's event handler.
3. The controller requests data from the model and passes it to the view.
4. The view presents the data to the user.

The key to achieve the MVC module is to divide the code into different parts according to their responsibilities; the code is clearly divided into several parts to maintain a good decoupling. Thus each part could be independently developed, tested and maintained [5].

MVVC Design Pattern

Based on the characteristics of MVC development model, this paper is to combine the MVVM development framework and puts forward MVVC design pattern.

MVVC Framework Design

In the traditional management platform development process, JQuery is the most frequently used representative of the packaging library - the event-driven as the core, it focuses on the DOM layer changes and user interaction with the platform, most energy of it is spent in the element operation [6]. As for data processing is weak, with the complexity of business logic, using the JQuery to control the view becomes more and more outdated. Therefore, it is necessary to integrate the characteristics of management system to design different development frameworks other than JQuery.

The framework of the MVVC module based on front-end is shown as following:

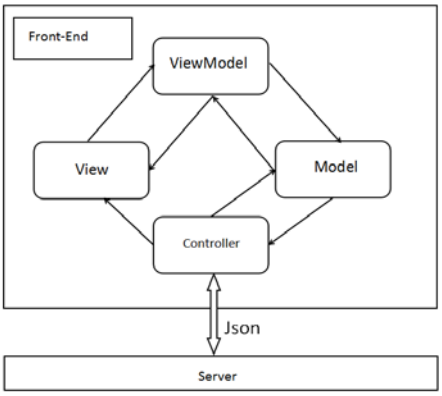

Figure 2. MVVC Model.

Model: Responsible for presenting the subject;
View (UI): Responsible for the page HTML;
View-Model: Responsible for achieving the two-way binding between model and view;
Controller: Responsible for the interaction between backgrounds and processing Model data.

MVVC Mode Features

1. Two-way data binding

Based on the dirty data checking and JS template rendering, the connectivity of the data layer and the view layer could be realized so that we can achieve CURD operation, avoiding a large mount of DOM operations and overstaffed binding and callback, which could promote the front-end function.

2. Modularization

In the traditional front-end mode, all kinds of web module codes are often mixed together, which makes it difficult to maintain and debug the latter part of the code. With the MVVC model, we can divide codes according to the functions into independent modules and in the Controller layer, information exchange could be achieved by function module, so that each module is not affected by others, and would be more conducive to the future code; At the same time, the modular design ideas can also be abstracted out of the public module, which is the concept of JQuery plugin.

3. Separation between front end and server end

Data was usually displayed in Traditional management platform by use of EL expressions through the daemon rendering. However, it is very inefficient. As a background language, EL expression is embedded in the front-end layer, and it will be measured through the background language analysis and then converted to front-end code in the HTML environment, resulting in unnecessary time-consuming [4]. In the MVVC model, based on the idea of data-driven, the JSON data with unitive style is to be obtained through the asynchronous request, then processed in the module layer, and rendered by JS template. In this way the traditional background template could be abandoned, achieving the separation of the front and server ends in data layer.

4. Control of route in front-end

Daemon routes could be transmitted to corresponding controllers in server end through different URL, and then rendered to the HTML page. In MVVC, routing rules in the front-end is been regulated, and the correspondence between the URL and the page is saved by Hash Routing table in platform maintenance, through which the purpose of rendering the view could be achieved. The route in the front-end control is to facilitate the interactive data format, which is beneficial for the collaborative development in both front and server ends. What's more, in this way we can pull out the repeating contents of the page, and just render the different parts of the two page during the process of each route jump, thereby the efficiency of front-end rendering could be improved.

MVVC Application Scenarios

1. SPA Applications

The front-end routing mechanism in MVVC allows us to develop multi-functional SPA applications, which means that it is unnecessary to request again and again to refresh the page. Front-end routing mechanism brings the efficient rendering to the HTML page, including a large number of static resources, because the fact that background does not need to reload the page, there is no need re-rendering the static files, and it is a huge convenience for HTTP requests.

2. Data-Driven Platform

MVVC provides a two-way data binding mechanism, which allows us to put more attention to the development process on the data level, largely convenient for the platform management that concludes more about data operations [7]. Compared to event-driven mode, MVVC possesses less embarrassing operations with DOM, and the only focus is the data cache and persistence.

3. Multi-languages Development Module

In MVVC mode, the agent between the front-end and the server end during the transmission of information is the data object, after the front-end sends the back-end request by use of a cross-language to RESTful interface, the purpose of rendering without server-end module could be realized, which is a kind of separation, thus allowing us to use a variety of development languages.

Application on the Management Platform

Overall Architecture Design

According to the requirement of low coupling of business module, the system is divided into three modules: resource management, user management and work order management. Among them, the resource management module is used by the background personnel to monitor the user's use of resources and deploy various virtual services. The main concern is the data display and data visualization; user management module mainly records the user's basic information, statistical user resources to use, in order to facilitate further information analysis; work order management is the user work order feedback. The three modules are independent of each other, and each module is layered as an MVVC unit.

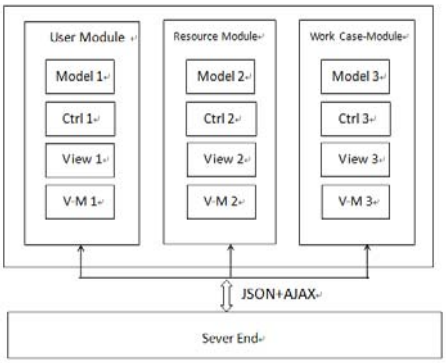

Figure 3. Management Platform Framework Design.

Feature Design

1. Information Display Module

Information display module is loaded in the page, Controller layer to the background request data, the server will return the data in JSON format, through the View Model layer command to render the data to the View layer.

Code configuration is as follows:

```
<td><a href="#/monitor/view/{{item.hostid}}">{{item.name}}</a></td>
<td ng-show="isA({{item.available}})" style="color:mediumseagreen">
   <i class=" glyphicon glyphicon-ok-circle"></i>Available
</td>
<td ng-show="isN({{item.available}})" style="color:indianred">
   <i class=" glyphicon glyphicon-remove-circle"></i>Unavailable
</td>
```

In this module, the data through the asynchronous request to return and front-end rendering, so as to avoid the coupling with the server; and for the return of the JSON data we can see it as an array of objects, JS template embedded in the HTML file, the list of data shows the module.

2. Data Visualization Module

The data visualization module is responsible for extracting and persisting the data obtained from the back-end server and then displaying the data through the third-party data visualization plug-in on the front page. The process is as follows: the server returns an array of objects to the control layer, the control layer to the data transfer model layer in the Model layer, we extract the data, cleaning and conversion, through the Controller layer as a medium to ViewModel dependency injection, Layer rendering.

3. Cache Data Module

The module is mainly for the background server POST request, as the background management module, we modify the information is often a lot of time, if every time to modify the background to send a post request, will produce a large number of duplicate URL HTTP Request; in this system, we will each time the information is modified to cache HTML5 sessionStorage, in the last page when the jump, the post request will be sent to the server.

Set the seesionStorge code configuration is as follows:

```
define(function() {//
   var cacheObj = window.sessionStorage || {//if the sessionStorage is null then new one
with getItem and setItem method
      getItem: function(key) {
         return this[key];
      },
      setItem: function(key, value) {
         this[key] = value;
      }
   };
//return clousure function with set and get mothed
   return {
      get: function(key) {return this.isFresh(key);}
      set: function(key, value, minutes) {
         var expDate = new Date();
         expDate.setMinutes(expDate.getMinutes() + (minutes || 0));
                  cacheObj.setItem(key, JSON.stringify({
            value: value,
            expires: expDate.getTime()
         }));
```

Summary

With the rapid development of the Internet, the logic of the background system would become more and more complex. We deliver the codes according to their functions based on the front-end MVC development module, reflecting the idea of management for different layers, which is separating the logical processing from UI rendering. With this idea, we can develop a contractile and expandable system, which is easy to maintain. This paper starts from the currently popular development mode, points out the shortcomings of the existing model, and puts forward the development model based on the MVC-like framework, which solves the problems of large front-end coupling that is unfavorable for expansion and maintenance.

References

[1] Ren Y., Ma Y. Construction of SSI Framework Based on MVC Software Design Model [C]// International Conference on Mechatronics, Materials, Chemistry and Computer Engineering. 2015.

[2] Huafeng Yu. The Design Frout End Based on the Framework. Journal of Science and Technology Vision, 2016 (23): 410-410.

[3] Yang Z., Hao K., Cong K., et al. Scalable Certification Framework for Behavioral Synthesis Front-End [C]// Design Automation Conference. IEEE, 2014: 1 - 6.

[4] Nik Ahmad N.A., Syed Zamri S.A. The cross platform application development adapted Spring framework to support front-end tendering services[C]// International Conference on Computer, Communications, and Control Technology. 2014:58-62.

[5] Feng Y., Chen N.J. Scheme of Mobile Web Front-end Display Based on MVVM Framework [J]. Computer & Modernization, 2014.

[6] Kim K., Lee K. Real-time processing of spatial attribute information for mobile web based on standard web framework and HTML5 [J]. Spatial Information Research, 2016, 24(2): 93-101.

[7] Li X, Chang D.L., Hui P., et al. Application of MVVM design pattern in MES [C]// IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems. IEEE, 2015.

[8] Lin Y.M. Application of MVVM design pattern and MVP design pattern based on ZK [J]. Journal of Chongqing University of Arts & Sciences, 2012.