4.3 Implementation

4.3.1 Content

4.3.3.1 Environments and tools

Running Environment: Node Server in Linux Centos 7

Tools:

1. npm: Node package management, a tool used to install or uninstall some common packages easily.

2. Vue-cli: Vue cli is a complete system based on vue.js for rapid development. There are three components:

* CLI providing Vue commands in the terminal
* CLI Services providing serve, build, and inspect commands
* CLI plugin, provideing optional functions for Vue projects.

3. Webpack: A tool for packaging, resolving dependencies and optimizing structure of code.

4. Eslint: A static code checking tool that checks whether our JavaScript code meets the specified rules.

5. Babel: A ES6 transcoder tool, which is widely used, and can convert ES6 code to ES5 code and execute in existing environment

4.3.3.2 Basic Architecture

Framework: MVVM

1. Model: The part of application program used to process application business logic and data, which is mainly responsible for network request, database processing, I/O and other operations.

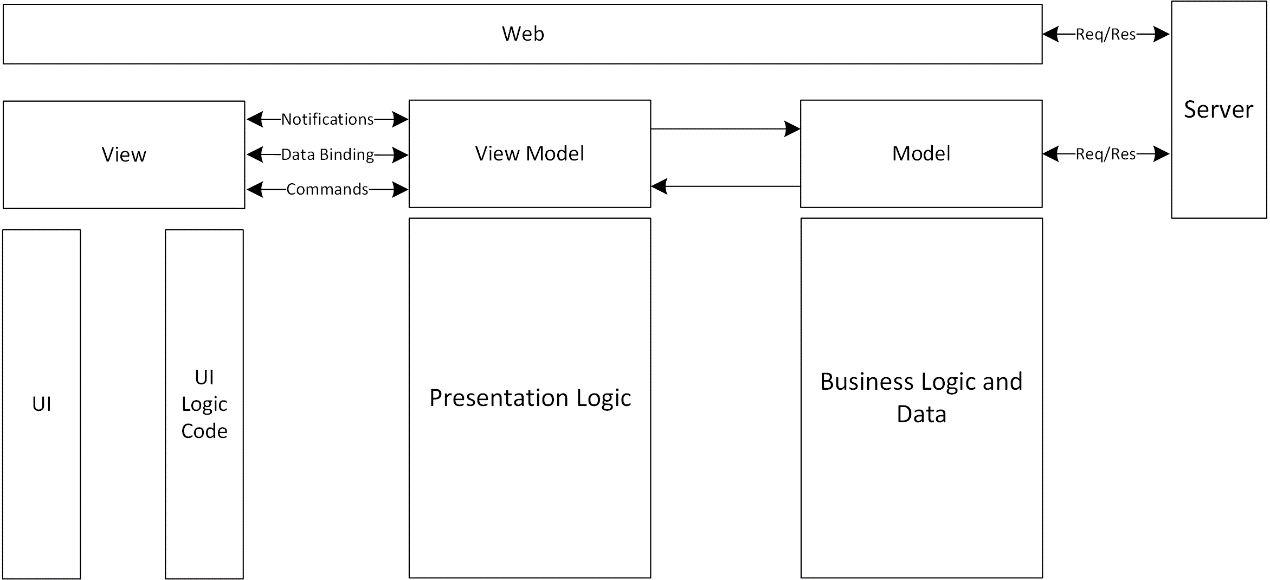
* For common data used in several modules, it is stored in main store.
* For different submodules, the data is stored in different namespaces stores. Such as “core.normal”, “core.basic” and so forth.

2. View: The part of an application that processes the display of data. In web development, it generally corresponds to HTML and JS files, activity and events.

* Apply Vue JS syntax and template in our project.
* Apply Bootstrap 3 code and style in our project.
* Some parts applied with ES6 or ES7 code in our project.

3. View Model: This part of an application creates an association to bind the model and view. In this way, once the model changes, the View Model will immediately feedback to view to automatically refresh the interface. View Model is only responsible for business logic and does nothing to do with UI.

* For displaying the data, the code for UI logic implements with UI views.
* For send the request or resolve the response, the code for business logic implements with Models.



**Figure 4.3.x** MVVM - Content Basic Architecture

4.3.3.3 Advantage of MVVM:

1. Data driven

In MVVM, data and business logic are in a separate View Model. As long as the View Model focuses on data and business logic, it does not need to deal with UI or control. The data automatically drives the UI to automatically update the UI, and the change of UI automatically feeds back to the data at the same time. The data becomes the dominant factor, so as long as the data is concerned in the business logic processing, it is convenient and much simpler.

Low coupling degree

In MVVM, data is independent of UI. View Model is only responsible for processing and providing data. How the UI wants to process data is up to the UI itself. View Model does not involve anything related to UI and does not hold the reference of UI control

Reusability

A View Model can be reused in multiple views. The same data is displayed in different UIs. For frequent version iteration UI changes, just change the view layer.

Unit testing

The data and business logic are in the View Model, and the UI is the focus of the view. So it is very convenient to do this test, and there is no dependence on each other at all. Both the unit test of the UI and the unit test of the business logic are low coupling.