Ziyun Wang

COMP531

November 26

Spring MVC in Web Development

In COMP531, we mainly learned about developing a web application in javascript with the assist of Node.js. Despite the fact that javascript is ubiquitous and used extensively in a web application, Java still has its benefits in many aspects as a programming language. Java provides features such as better IDE, multithreading, more libraries and strict typing. Therefore, it is also useful to examine some of the platforms for writing a web application in Java. One of the most popular development frameworks used for java Enterprise web application is Spring MVC. It takes advantages of Java's language features and achieves separation of concerns by following a light-weight Model-View-Controller design pattern.

First, it is useful to understand why java is a preferred language for backend programming by some programmers. One benefit it provides is parallelism, which is not supported by a node application. This is particularly important when the data processing the backend wants to take full advantage of multithreading. Secondly, java has more libraries than most of the programming languages. Because java has been used extensively in all kinds of applications, it has very mature and powerful libraries for the programmers to choose from, some of which depends on multi-threading for performance. Last but not least, Java has some language features that make it stand out as large-scale projects. Its strict typing, exception handling, and generic typing make it easier for teams to work on a large scale backend concurrently. That being said, there are many java frameworks that take

advantage of these features. Spring MVC, a framework that both uses java as a platform and supports easy MVC implementation makes it a good choice for backend programming for many web developers.

Then we want to understand why Spring MVC is preferred by some programmers. Spring MVC is used extensively because of its decoupling of the business logic from the view. Traditionally, the way java web application works is that the client request resources from the java server, and the java server sends back the webpage along with the data in html format. The application built this way does not have a clear separation of roles. The servlet both produces the data and renders the view. Some argue that this makes it hard for the UI developers and the logic developers to work independently from each other in this way. That said, people seek to find the solution from the traditional MVC software methodology, in which models and views are completely decoupled and the control talked to both of them through interfaces. In this way, the UI developers can work on the UI, given what is provided by/requested of by the controller without worrying about how the data is generated by the model. Similarly, the model can produce the data based on the interface without knowing anything about the view. “These layers [model, view, and controller] are relatively independent, and each has a clear function. On account of these feathers, MVC is beneficial to modularized development and greatly improves the efficiency, maintainability of system development and code reusability, which can adapt to design requirements of the increasingly complex multi-tier application system” (Zhang, Dandan, Zhiqiang Wei, and Yongquan Yang) It generally considered a good design decision to separate the system layers and let them communicate through interfaces. Because Spring MVC provides the features that make implementing the requirements

above easier, it stands out as one the most popular frameworks used for complicated enterprise applications.

There are some import concepts defined in Spring MVC defined by the Apache documentation. These concepts are critical to understanding the framework. “Front Controller” serves as the interface to the client. It processes the request sent by the client and identifies which “handler” should be used to process this request. “Controller” is a handler that produces the data/business logic to be used for rendering the page. The model is produced by the “Controller” is passed back to the “Front Controller”. “View template” is an object that uses the model to produce the actual page and return the page to “Front Controller”. This is basically the data flow of web application using Spring MVC. It guarantees that only the “Front Controller” talks to the client and that the model view and controller are properly separated. With this clear separation, individual teams can work on each part of the application separately. Nevertheless, this framework produces a lot of overhead in that a typical MVC structured software requires factories and making changes to an existing project is also complicated. Spring MVC also provides features that further reduces the overhead. “Programmers can effectively annotate the java classes with component name or alias. A configuration file can also have declarations of a class object… For instance, on a project where multiple teams are developing different versions of the same class, it is not feasible to change the code every time or write different functions for different implementations.” (Hemrajani 128) This makes testing and developing more concurrent among teams.

After examining the benefits of using spring MVC, it is also important to understand some of the limitations of Spring MVC. One observation is that modern applications do not take full advantages of MVC. For example, some web applications are using React.js and redux to manage the data flow for making dynamic pages. This substitutes the “view” part of the Spring MVC framework. Some companies use Spring MVC as the backend for producing data and hosting HTTP endpoints. The good news is, the new Spring framework 5 will be able to add the compatibility with React. But currently, this still remains a problem. If the framework is only used for the backend, it really fails to stand out much among all the Java RESTful frameworks because the view and the model are already decoupled completely. Another limitation lies in the programming overhead. There are sometimes programmers complaining about having to write configuration files carefully in order to get the project even to run. For a small application that does not require a lot of object substitution/injection, it is a big overhead to start the project. In particular, if someone wants to write both the backend and the front of a simple application, it is not necessary to completely talk through the controller. The MVC pattern also takes more in-depth understanding to be able to take its full advantage. Therefore, it is not recommended to use such framework if neither scalability nor separation of concern is a major consideration.

To conclude, Spring MVC provides a series of convenient features that enable the developers to effectively achieve the separation of concerns while take advantage of the language features of Java. However, with the rapid change of technology, such as the development of ReactJS, Java frameworks are used more often for the RESTful server rather than a whole web application. Therefore, developers sometimes are not taking full

advantage of Spring MVC's ability to separate concerns. Nevertheless, some companies are still using it as the backend because they want to use java as the backend and Spring MVC provides some nice features in object injection and testing. Additionally, the new Spring framework added more support for modern technologies like React.js. Despite the disadvantages of Spring MVC, it takes advantage of Java's language advances, the well-established design pattern and adds more support for modern technology to itself. Therefore, it is foreseeable that it would remain active in web development in the next few years. Consequently, it is a tool worth learning for any web developer who prefers Java as their development language.

Works Cited

Hemrajani, Anil. Agile Java Development: With Spring, Hibernate and Eclipse. Indianapolis, IN: Developer's Library, 2006. Print.

Zhang, Dandan, Zhiqiang Wei, and Yongquan Yang. "Research on Lightweight MVC Framework Based on Spring MVC and Mybatis." 2013 Sixth International Symposium on Computational Intelligence and Design (2013): n. pag. Web.