

Rich Internet Web Application Development using Google Web Toolkit

Niriksha Bhojaraj Kabbin*, Sharmila Sequeira**

*(Department of Computer Science, NMAM Institute of Technology, Nitte-574110)

** (Department of Computer Science, NMAM Institute of Technology, Nitte-574110)

ABSTRACT

Web applications in today's world has a great impact on businesses and are popular since they provide business benefits and hugely deployable. Developing such efficient web applications using leading edge web technologies that promise to deliver upgraded user interface, greater scalability and interoperability, improved performance and usability, among different systems is a challenge. Google Web Toolkit (GWT) is one such framework that helps to build Rich Internet Applications (RIAs) that enable fertile development of high performance web applications. This paper puts an effort to provide an effective solution to develop quality web based applications with an added layer of security.

Keywords – GWT, RIAs, Web Application, Interoperability, Performance

I. Introduction

Traditional web applications focus on client-server architecture where all the processing is done on the server side and a thin client which is only used to display the static contents. Synchronous mode is used to do request and response to the server and from the server. Server creates new page for every request and sends to the client browser and keeps the state information every time the request is made. This approach has significant drawbacks and limitations, especially due to the richness of the application interfaces and the overall sophistication of the solutions that could be built and delivered [3].

These Traditional old-fashioned web applications are being replaced by the Rich Internet Applications (RIAs) which use a Rich Client deployment model that provide richer and more interactive user interfaces. AJAX (Asynchronous JavaScript and XML) uses Rich Internet Application that follow Asynchronous mode for request and response which sends a data to the server and retrieve from a server without interfering with the display and behavior of the existing page asynchronously (in the background)

Google Web Toolkit (GWT) is a technology for creating rich AJAX applications using Java as the programming language which on compilation produces JavaScript code suitable for each browser. GWT is open source and used by hundreds of developers to build large scale and high performance web application by keeping them as easy-to-maintain. This paper attempts to provide measure for developing an application using the features of Google Web Toolkit. We have shown the procedure of integrating the different elements of GWT.

II. Conceptualization

2.1 Google Web Toolkit

GWT is an open source framework that allows a developer to develop Ajax-based applications in the Java language and provides tools to convert the Java code into JavaScript and HTML. For the compilation process a set of JavaScript libraries that emulate requires the classes to be included in the Java virtual machine (JVM) known as JRE (Java Runtime Environment emulation library [1]. This frees developer, from the burden of rewriting JavaScript Code to suit the standards of various web browsers.

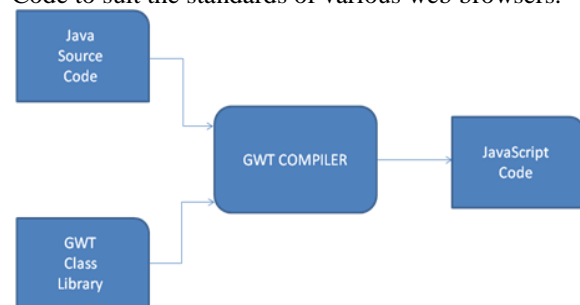


Fig 1. Architecture – GWT Web application.

With the GWT SDK, AJAX front-end is written in the Java programming language which GWT then cross-compile into optimized JavaScript that automatically works across all major browsers. It supports “edit-refresh-view” cycle having the benefit of being able to edit the java code when required and refreshing the browser to look for the changes made in the java code and viewing the page. Thus it helps to debug and step through Java code line by line. When the application is deployed to production, the GWT compiler translates user written Java application to browser-compliant JavaScript.

GWT provides Java API's and libraries to facilitate the creation of graphical user interfaces and logic for the browser client that provides variety of graphical components ranging from space handlers (layouts) to lists and tables [1]. The GWT framework provides a number of user interface widgets for client-side UI development. The widgets in GWT are divided into container widgets and user-interface widgets. The collection of all such widgets is what composes the widget library. A UI *widget* is defined as a graphical component and is used for building graphical user interfaces. A button is a UI widget, and a user can signal an event by clicking it. Thus Widgets allow a human to interact with the computer application by performing some action on the widget. In GWT Widgets are rendered using dynamically created HTML. Each widget in the GWT framework has a corresponding HTML representation, and the compiler translates the widget objects in Java into the corresponding HTML equivalent for the widget [4].

A feature of GWT is that, being based in Ajax, execution is done on the client through the Web browser and control of the GUI application is handled there and not on the server as they do in traditional Web applications, this brings the advantage of fact that user interface show only the information necessary for its operation, which in turn releases the load on the server and provides greater interactivity and usability in these applications [1].

An approach to such behavior can be observed in the Fig 2.

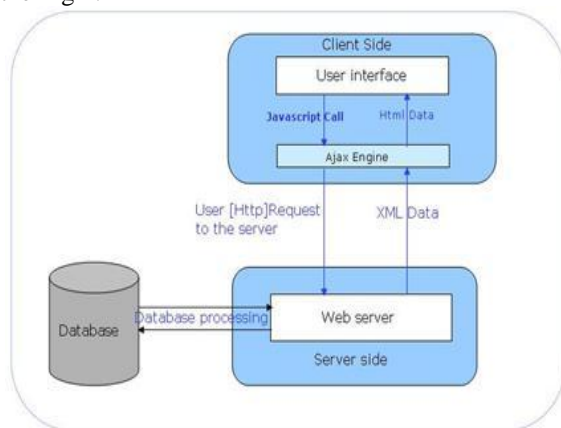


Fig 2. GWT Ajax web Application Model.

AJAX ENGINE is like intermediate layer between the user and server. When session starts instead of loading a page on each request, Ajax engine is loaded by the browser and this engine is responsible to load the part of the page data and communicating with the server, everything happens in the asynchronous mode.

III. Proposed System

The web application using GWT includes functionality between the client and server, where the

functional and major part of the application that is handling of the Graphical User Interface (GUI) like event handling, graphics creation, controlling the components will be executed and performed on the client and handling of the requests and storage of information for application's operation is done on the server.

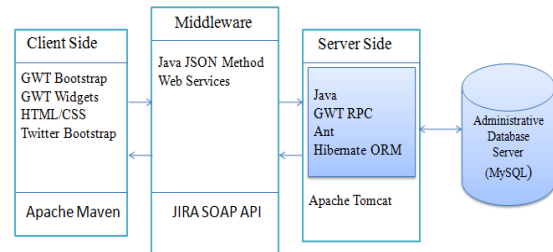


Fig 3. Model of the GWT client side and server side working process.

User Interface is built using the GWT libraries that provide attractive user interface controls like widgets, grids with grouping and paging, layouts and many more [2]. GWT Bootstrap which is a toolkit can also be used which provides flexible and simple components representing Bootstrap components, styles, plug-ins and also Twitter Bootstrap styles and widgets. It is a front-end API to develop the GWT application faster and more beautiful.

Fig 3 is a graphical representation of the model of GWT frontend and backend development and working process of the client side and the server side of the application. When a request is sent to server, GWT plays a role to process the request and sends that in JSON format to the JAVA method. It can also transfer the data using the third party API calls like JIRA to get the product build information. GWT transfers this JSON data to the JAVA method for the data processing. GWT-RPC (Remote Procedure Call) is used for communication between the client and the server, which allows transport of objects and also facilitates handling. Data is then fetched from the Database that processes the data and sends back, this happens to and fro. The fetched data is generate-d back to the java objects by the server. In the end Client receives the full fledged processed data.

IV. Conclusion

The old-fashioned Traditional Web applications have significant drawbacks in terms of usability and interactivity of their user interfaces. Rich Internet Applications provides rich assets, functionality and clients with Asynchronous communication method that allows the application to fetch the data from the server faster and without the reload of the page. RIA's overcome the drawbacks of traditional Web applications by being more user friendly. The study of this research provides a way to build Rich Internet Applications using the GWT technology which is an

AJAX framework thus providing a flexibility to integrate with other client and server Java frameworks by gaining the benefits of JAVA and improving the structure of the application.

Acknowledgements

Thanks to Versatiletech, Bangalore for support and guidance. Our special gratitude to the Department of Computer Science and Engineering, NMAM Institute of Technology, Nitte for providing us the support and cooperation.

References

- [1] Correa, J.D.Y, Ricaurte J.A.B, Web Application Development Technologies using Google Web Toolkit and Google App Engine-JAVA, *IEEE Latin America Transactions*, 12(2), 2014, 372 – 377.
- [2] Pawlak.P, Sakowicz.B, Mazur.P, Napieralski .A, Social Network Application Based On Google Web Toolkit. *CADSM'2009*, 2009, 461-464.
- [3] Melia, S. , Gomez, J. ,Perez, S. , Diaz, O. , A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA, *Eighth International Conference of Web Engineering*,2008,13-23.
- [4] Vipul Gupta, *Accelerated GWT: Building Enterprise Google* (APress ©2008).