Stock Watcher Application using Google Web Toolkit

Revanth Reddy Kontham
student Id: 1107178
Masters of Applied Computer Science
Lakehead University
Thunder Bay, Ontario, canada
email: rkontham@lakeheadu.ca

Akhilesh Kumar Kondoju
student Id: 1107334
Masters of Applied Computer Science
Lakehead University
Thunder Bay, Ontario, canada
email: akondoju@lakeheadu.ca

Abstract: This article describes simple stock watcher based on Google Web Toolkit (GWT) as the new technology in the creation of rich AJAX applications using only Java as the programming language, which is later on compiled into pure JavaScript and deployed as a regular web site. The user is able to know and manage stock watcher to be updated with the latest changes in stock. We used random data as input as it is a prototype model.

I. Introduction

GWT is an improvement toolbox for building and advancing complex program based applications. It will probably empower the gainful advancement of elite web applications without the engineer is a specialist in program peculiarities, XMLHttpRequest, and JavaScript. It's open-source, totally free, and utilized by a large number of designers around the globe. Google Web Toolkit, or GWT Web Toolkit, is an open-source set of apparatuses that permits web designers to make and keep up JavaScript front-end applications in Java. Other than a couple of local libraries, everything is Java source that can be based on any upheld stage with the included GWT Ant manufacture records. It lets you compose customer side applications in Java and convey them as JavaScript [3]. The actual definition of Stock Watcher is A modernized assistance that screens and explores exchanging action on the NYSE request to distinguish any strange action or security development that may be brought about by bits of gossip or criminal operations.

Developing Google web Toolkit:

Write: The GWT SDK gives a lot of center Java APIs and Widgets. These permit you to compose AJAX applications in Java and afterward order the source to profoundly advanced JavaScript that stumbles into all programs, including versatile programs for Android and the iPhone. Developing AJAX applications right now increasingly beneficial gratitude to a more significant level of reflection on basic ideas like DOM control and XHR correspondence. You aren't restricted to pre-canned gadgets either. Anything you can do with the program's DOM and JavaScript should be possible in GWT, incorporating connecting with manually written JavaScript.

Debug: You can debug AJAX applications in your most loved IDE simply like you would a work area application, and in your preferred program simply like you would on the off chance that you were coding JavaScript. The GWT engineer module traverses the hole between Java bytecode in the debugger and the program's JavaScript. Because of the GWT engineer module, there's no gathering of code to JavaScript to see it in the program. You can utilize the equivalent alter invigorate see cycle you're utilized to with JavaScript, while simultaneously review factors, set breakpoints, and use the various debugger instruments accessible to you with Java. What's more, because GWT's advancement mode is currently in the program itself, you can utilize apparatuses like Firebug and Inspector as you code in Java.

Optimize: GWT contains two integral assets for making improved web applications. The GWT compiler performs far reaching improvements over your codebase — in-covering techniques, expelling dead code, advancing strings, and that's just the beginning. By setting split-focuses in the code, it can likewise section your download into numerous JavaScript pieces, separating huge applications for quicker startup time. Execution bottlenecks aren't restricted to JavaScript. Program format and CSS regularly carry on in weird manners that are difficult to analyze. Speed Tracer is another Chrome Extension in GWT that empowers you to analyze execution issues in the program.

Run: At the point when you're prepared to send, GWT arranges your Java source code into upgraded, independent JavaScript documents that naturally run on every single significant program, just as versatile programs.

As referenced, GWT comprises of libraries liable for explicit functionalities that could possibly be utilized. The most significant components are shown in Figure 1. In this paper using all the elements of GWT we made a Stock Watcher which resembles a copy of genuine stock watcher yet with irregular information [5]. GWT development cycle has the accompanying advances:

1. Utilizing the developer's preferred Java IDE to compose and troubleshoot an application in the Java language, utilizing as many (or as few) GWT libraries as it is considered helpful.
2. Utilizing GWT's Java-to-JavaScript compiler to distill

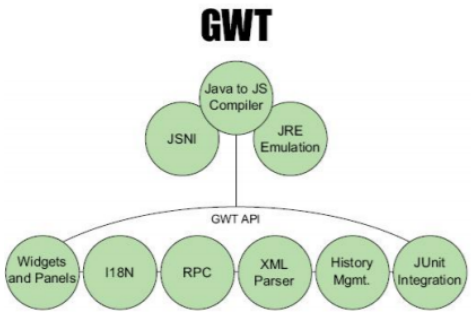

Fig. 1. GWT component Overview [2]

application into a lot of JavaScript and HTML documents that can be presented with any web server.

3. Affirming that application works in every program that is to be bolstered, which as a rule takes no extra work.

The most important features in GWT is an alignment of the data which are to be displayed, for this we use the below panels [4].

Horizontal Panel: The two components used to include a stock—the information enclose for composing another stock image and the Add button—are firmly related practically and you need to keep them together outwardly. To spread them outside-by-side, you'll put the TextBox gadget and a Button gadget in an even board. In the Java code, you'll make another occasion of Horizontal Panel and name it addPanel.

Root Panel: There is one more board you need that can't in the UI: a Root board. A Root board is a compartment for the dynamic components of your application. It is at the highest point of any GWT UI progressive system. There are two different ways you can utilize a Root board, either to create the whole body of the page or to produce explicit components installed in the body.

Vertical Panel: You need to design the rest of the components vertically. The FlexTable gadget for the stock table. The Add Stock board, which contains the information box and Adds button. The Label gadget for the timestamp. You'll do this with a vertical board. In the Java code, you'll make another occasion of VerticalPanel and name it mainPanel.

II. Related Work

J. D. Y. Correa and J. A. B. Ricaurte1 are firmly identified with our methodology however they are utilizing google application motor too [1]. In their paper, they depict a few measurements to consider for web application advancement with structures like Google Web Toolkit and Google App Engine, innovations with a specific method of activity and with certain limitations that influence the plan and the usefulness of these applications yet additionally offer extraordinary advantages, for example, improved UI, better ease of use, improved execution, more prominent adaptability and as the capacity to utilize certain administrations, which permit application interoperability with various frameworks.

Piotr Pawlak, Bartosz Sakowicz, Piotr Mazur, Andrzej Napieralski are firmly identified with our methodology yet they are utilizing google application motor too [2]. In their paper, it depicts the informal organization application dependent on Google Web Toolkit (GWT) as the innovation in the formation of rich AJAX applications utilizing just Java as the programming language, which is, later on, accumulated into unadulterated JavaScript and conveyed as a customary site. A model application is a basic social gateway as these days they are the most widely recognized delegates of Web 2.0 applications that require quick, rich interfaces like the ones made utilizing GWT.

III. Application Goals

A described application named Stock watcher which helps them to any operations on any stock. Stock Watcher is a product program used to track and report basic data to operators concerning an organization's stock. This keeps the specialist from having to continually watch the stock to decide each value modification or vacillation from current market patterns. This product just covers those stocks recorded on the New York Stock Exchange (NYSE). This product likewise screens for deceitful events and advises the Securities and Exchange Commission should they happen. Using this user can able to know any stock price of any object they want.

IV. Implementation

GWT Environment Requirements:

1. Eclipse IDE2019
2. Java JDK-8u241
3. Apache Tomcat-9.0.30
4. GWT SDK plugin-3.0.0

This Build a Sample GWT Application instructional exercise is partitioned into 5 segments following an average application improvement cycle. Each area expands on the past areas. Right now of StockWatcher, all usefulness is coded on the customer side.

A. Create:

Gwt utilities accomplish crafted by creating the venture subdirectories and records you have to begin. The Google Plugin for Eclipse contains a wizard for making GWT applications. Here are the means to make:

1. Enter the task name "StockWatcher".

2.Enter the bundle "com.google.gwt.sample.stockwatcher".

3. Ensure Use Google Web Toolkit is checked and that Use default SDK (GWT) is chosen.

4. On the off chance that you didn't introduce the SDKs when you introduced the Google Plugin for Eclipse, you should click Configure SDKs... to determine the catalog where GWT (and the App Engine SDK if fundamental) was unfastened.

5. Snap the Finish button.ing a starter application.

B. Design:

At first, you need the StockWatcher application to complete six things.

1. Give clients the capacity to include stocks.

2. Show the accompanying data for each stock: image, value, change since the last revive.

3. Give clients the capacity to erase a stock from the rundown.

4. Refresh the stock cost.

5. Figure the change since the last revive as both a number and a rate.

6. Show a timestamp demonstrating the last update.

In the wake of considering StockWatcher's useful necessities, you choose you to need these UI components:

a. A table to hold the stock information

b. Two catches, one to add stocks and one to expel them

c. An information box to enter the stock code

d. A timestamp to show the time and date of the last data.

C. Building UI:

We need UI to show StockWatcher Starts up, we will execute them in the onModuleLoad technique. Right now, will:

1. Instantiate every gadget and board: Instantiate every gadget and board utilizing class field initializers. Display recommended rectifications by tapping on the principal red "x". Resolve the various mistakes by proclaiming the import announcements similarly.

2. Create a table that holds stock information: Create a table for stock information, we can see that adding to a table can be cultivated with a call to the setText strategy. The main parameter demonstrates the line, the second the segment, and the last parameter is the content that will be shown in the table cell.

3. Layout the gadgets utilizing Add stock board and Mainboard: To spread out the gadgets, we'll collect two boards, the Add Stock board and the Main Panel. First collect the Add Stock board, an even board that wraps the information box and Add button. At that point amass the Mainboard, a vertical board that determines the format of the stock rundown table, the Add Stock board, and the timestamp

4. Associate the Main Panel with Root board: Associate the Main Panel with the host page through the Root board, In the request for any GWT gadget or board to be installed in the HTML have the page, it must be contained inside a Root board. Partner the Root board with the Vertical board allocated to mainPanel. The Root board wraps the HTML component (in StockWatcher's host page) that has an id of "stocklist".

5. Move the cursor center to the information box: Finally, move the cursor center to the information box along these lines, when StockWatcher loads, the client can start including stocks.

D. Managing Events:

Right now, wire up gadgets to tune in for and handle mouse and console

Audit Functional necessities:

1. Dealing with Mouse Events

2. Dealing with Keyboard Events

Reacting to occasions:

1. Adding the stock to the stock table

2. Approving contribution to a book box

E. Coding Client Side:

Include and Remove stocks from the stock table. Make an information structure. Add lines to the stock table. Add a catch to expel stocks from the stock table. The test being developed mode. Revive the costs and change fields for each stock in the table. Naturally, revive the Price and Change fields by executing a clock and indicating an invigorate rate. Embody the stock value information by making a class, StockPrice. Produce the stock information at the Cost and Change fields by executing the refreshWatchList strategy. Burden the Price and Change fields with the stock information by executing the updateTable technique. Test the irregular age of stock costs and change esteems. Actualize the timestamp indicating the hour of the last update

V. Results and Discussion

The web application utilizing GWT incorporates usefulness between the customer and server, where the functional and significant piece of the application that is handling of the Graphical User Interface (GUI) like event handling, 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. UI is assembled utilizing the GWT libraries that provide attractive user interface controls like widgets, grids with grouping and paging, layouts and many more. Right now, assembled the example AJAX application, Stock Watcher. Stock Watcher is an application for checking stock varieties. We created it by utilizing arbitrary costs and by utilizing the GWT timer.

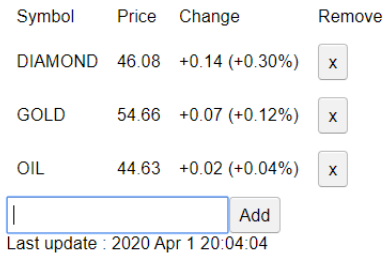

Fig. 2. Sample Result 1

fig 2 shows the sample result of the application. It contains the fields like Symbol, Price, Change, Remove. The symbol indicates the content we are writing in the content box, the value field demonstrates the sum it is experiencing as of now. essentially, the change field shows, how the cost fluctuates for at regular intervals. Remove field contains the cross alternative utilized for erasing the selected row

Fig 3 presumes that it just takes the characters and numbers, it doesn't take any specials symbols.

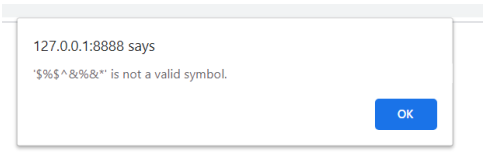

Fig. 3. Sample result 2

VI. Conclusion

The main aim of this project was to show how AJAX technology is promising if approached with the right tools, especially for the applications in creating a stock watcher. the good old Traditional Web applications have critical disadvantages as far as convenience and intelligence of their UIs. The survey of this examination gives an approach to constructing Rich Internet Applications utilizing the GWT innovation which is an AJAX system along these lines giving adaptability to coordinate other customer and server Java structures by picking up the advantages of JAVA and improving the structure of the application.

References

[1] J. David Yanquen Correa and J. Antonio Ballesteros Ricaurte, "Web Application Deveplomen Technologies Using Google Web Toolkit And Google App Engine-Java," in IEEE Latin America Transactions, vol. 12, no. 2, pp. 372-377, March 2014.
[2] P. Pawlak, B. Sakowicz, P. Mazur and A. Napieralski, "Social network application based on Google Web Toolkit," 2009 10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics, Lviv-Polyana, 2009, pp. 461-464.
[3] http://www.gwtproject.org/doc/latest/tutorial/gettingstarted.html
[4] http://tutorialspoint.com/gwt/index.html
[5] Niriksha Bhojaraj Kabbin, Sharmila Sequeira, "Rich Internet Web Application Development using Google Web Toolkit" May 2009 Int. Journal of Engineering Research and Applications