

A Survey of WebSocket Development Techniques and Technologies

T. Wirasingha^{#1}, N.R. Dissanayake¹

¹*Informatics Institute of Technology, No.57, Ramakrishna road, Colombo 06, Sri Lanka*

[#]*torinindika@gmail.com*

Abstract—*WebSocket protocol has become a sound technology in software industry for real-time web application development, since it provides bidirectional, full-duplex communication between client and server over a single connection. There are several concepts, techniques, and technologies such as frameworks and libraries for WebSocket based development. However, the knowledge existing regarding these concepts, techniques, and technologies is scattered, thus it is not easy to obtain the information required for a specific development environment. In this paper we analyse the literature, then discuss about the existing concepts, techniques, and technologies, available for WebSocket based development. The facts delivered in this paper can be utilized to reduce knowledge search time for engineering of WebSocket based applications, and it can help developers to easily find the needful for the implementation of WebSocket. The knowledge of this paper will be utilized in our ongoing research, towards design and development of a WebSocket server-tool, which will help in rapid development.*

Keywords— **Real time web, Survey, WebSocket**

I. INTRODUCTION

The web applications were built on the client-server architecture, in which the client requests for services from the server, and then the server responds with the data or information. Client-server architecture is based on the request/response paradigm of Hyper Text Transfer Protocol (HTTP). HTTP is the fundamental protocol for the service of the web; and, one of the main limitations of HTTP is that it is half-duplex (Lubbers, Albers, & Salim, 2010). In the late 1990s, developers started to develop web pages with dynamic content, with client-side and server-side application development languages.

Whenever a web application runs on a web browser it will be communicating with a server. But if that communication takes a long time, the web page may become an empty page that has no texts and no images. If this occurs frequently, when a user requests “Reload” under heavy loaded servers, it could lead to users’ confusion because users lose sight of the operation on a web page. To resolve this issue, AJAX (Asynchronous JavaScript and XML) technology is adopted to web applications (Garrett, 2005). If a web application is constructed based on Ajax

technology, the empty web pages will be avoided because client programs communicate asynchronously with web servers. Client programs do not need to wait for the termination of communication with the server.

In 2011, WebSocket was introduced and it provides a full-duplex communication channel over a single WebSocket (WS) connection. WebSocket protocol was standardized by the Internet Engineering Task Force (IETF) as RFC 6455, and WebSocket Application Program Interface (API) is standardized by World Web Consortium (W3C). WebSocket provides a way of creating a persistent, low latency connection, which is capable of handling transactions initiated by either the client or the server, thus full-duplex (Skvorc, Horvat, & Sribljic, 2014) and support both data-push and data-pull.

A. Problem and motivation

There are different concepts, techniques, and technologies regarding WebSocket, based on different programming languages and platforms. In order to develop a WebSocket web application, the developers need to write code for both client and server components using these available concepts, techniques, and technologies. When selecting suitable concepts, techniques, and technologies for the application, they need to spend a considerable amount of time on studying and comparing existing concepts, techniques, and technologies to understand their pros and cons.

Furthermore, we identified that there is a lack of proper details as in surveys related to the WebSocket development which compare real-time web concepts, techniques, and technologies, and their implementations. This setting leads to instil erroneous impressions on software engineers and developers.

To address these issues and fill the gaps, this paper provides an analysis of a survey of the concepts, techniques, and technologies related to WebSocket. In the Background section we discuss and present the pros and cons of the asynchronous communication implementation concepts, techniques, and technologies available for the Rich Internet Application (RIA) development. Then in the next section we present the analysis of the findings of the survey.