A Comparison of the Efficiency of Using HTML over XML and JSON for the Asynchronous Communication in Rich Internet Applications

NR Dissanayake[1#], D De Silva[2] and GKA Dias[1]

[1]University of Colombo School of Computing, Colombo 7, Sri Lanka

[2]Informatics Institute of Technology, Wellawatta, Sri Lanka

#nalakadmnr@gmail.com

Abstract— *Rich Internet Applications use XML or JSON to format the data in asynchronous communication, which engaged a serialization process in the server and a de-serialization process in the client. If we use HTML to format the data it can get rid of the de-serialization process and make the development easier. In this paper we compare the efficiency of using HTML over XML and JSON, for asynchronous communication in Rich Internet Applications, by the means of time and the size of the data. Based on the results, we expect to introduce a set a facts to consider when selecting the technique / technology for the asynchronous communication in Rich Internet Applications.*

Keywords— *Rich Internet Applications, Asynchronous communication, HTML, XML, JSON*

I. INTRODUCTION

In this era of Web2, Rich Internet Applications (RIAs) have gained the demand of the users, with enhanced user experience via rich User Interfaces (UI), and faster responds (Lawton, 2008). Asynchronous communication in RIAs between the client and the server plays a major role in providing rich features, which respond faster (Busch & Koch, 2009). When the user initiates a process, the client-side RIA engine sends an asynchronous request to the server, and the server processes the request and sends only the results – instead of a complete web page – back to the client. The client-side app then processes and shows the results on the current page by updating only the necessary segments on the UI. This partial page rendering nature along with the asynchronous communication, enables developing rich UI components and rich features in web applications (Busch & Koch, 2009).

The data is sent from the client to the server as parameters along with the request, using GET or POST form methods. The respond from the server to the client may contain larger data set(s), where the client is supposed to understand and process the data; and display information on the UI. Extensible Markup Language (XML) (Bray, et al., 2006) or JavaScript Object Notation (JSON) (T. Bray, 2014) is used to ensure a good structure and semantic of the data exchanged in both request and respond of the asynchronous communication.

XML had originally been introduced to define structure(s) for data sets in storing and communication of data. It uses a nested arrangement of elements to provide structures for data; and uses Attributes to describe them further. The usage of XML spreads over a larger domain, but here we limit its scope to the use in the asynchronous communication of RIAs.

In web applications' data communication process, the time taken for the communication and the size of the data communicated are two main factors to be considered. Larger data sets introduce traffic in the network and may affect the speed of the communication too. Since the size of the XML data set is considered large, JSON has been used as a better solution for the data communication in web applications. JSON is a light weighted and text based format, with a small set of formatting rules to form a portable set of structured data. And JSON was proven that it is better than XML in data communication (Lin, et al., 2012).

Despite the technology used for the asynchronous communication, the process contains the following steps in the processing algorithm. The client sends the request; and the server processes the request and prepares the respond by serializing the data, which means preparing the XML or JSON structure of the data. Then the server sends the serialized data to the client and the client de T serialize/parse/extract data from the XML or JSON structure and re-formats them to be shown on the UI. Finally the client performs the partial update of the UI and renders the information to the user. Figure 1 shows the steps of the abstract algorithm of the asynchronous communication process.