

AJAX web application for basketball statistics

P. Vasiljević*, Z. Ivanković**, Z. Milošević***, P. Pecev**** and B. Markoski**

* Faculty of Technical Sciences, Novi Sad, Serbia, petarv@uns.ac.rs

** Technical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia, ivankovic.zdravko@gmail.com, markonins@yahoo.com

*** Faculty of Sport and Physical Education, Novi Sad, Serbia, zoranais@eunet.rs

**** Dept. of Mathematics and Computers Sciences, Faculty of Sciences, Novi Sad, Serbia, predrag.pecev@gmail.com

Abstract - AJAX is a group of interrelated web development techniques used on the client-side to create interactive web applications or rich Internet applications (RIA). With AJAX, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. This paper describes web application that shows live basketball statistics of a certain game to viewers over the internet (netcasting). It is programmed using PHP, HTML and JavaScript. Presented solution is, with small changes, applicable to other sports.

I. INTRODUCTION

The domain where desktop applications beat web applications is user interaction. In desktop applications, user do not have to wait and stare at a blank screen (most of the time) until an action completes. The user interface is integrated with the actual application and allows instantaneous response. In classic web applications, when some information is submitted or requested user has to wait for an entire new page to load. AJAX is changing all of that and bringing desktop response times to the Web.

This paper describes web application that shows live basketball statistics of a certain game to viewers on the internet (netcasting). It is programmed using PHP, HTML and JavaScript.

II. AJAX

AJAX is a group of interrelated web development techniques used on the client-side to create interactive web applications or rich Internet applications (RIA) [1]. With AJAX, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. The use of AJAX has led to an increase in interactive animation on web pages and better quality of web services due to the asynchronous mode.

The name itself is an acronym for Asynchronous JavaScript and XML, and it was coined by Jesse James Garrett from the Adaptive Path in 2005. Despite the name, the use of XML is not actually required, nor do the requests need to be asynchronous.

AJAX uses a combination of:

- XHTML and CSS for marking up and styling information.

- The Document Object Model (DOM) for dynamically displaying and interacting with the information presented.
- A method for exchanging data asynchronously between browser and server, thereby avoiding page reloads. The XMLHttpRequest (XHR) object is usually used, but sometimes an IFrame object or a dynamically added <script> tag is used instead.
- A format for the data sent to the browser. Common formats include XML, pre-formatted HTML, plain text, and JSON. This data could be created dynamically by some form of server-side scripting.
- A client-side scripting language such as JavaScript binding everything together, allowing retrieving dynamic information, displaying and modifying the page elements. Other languages such as VBScript are also capable of the required functionality.

In the classic web application, communication between the client (the browser) and the web server is performed directly, using HTTP requests. Most user actions in the interface trigger an HTTP request back to a web server. The server does some processing and then returns an HTML page to the client. It's a model adapted from the Web's original use as a hypertext medium, but what makes the Web good for hypertext doesn't necessarily make it good for software applications. This approach doesn't make for a great user experience. While the server is doing its thing, the user is waiting. And at every step in a task, the user waits some more.

When an application uses AJAX, a new layer is added to the communication model (Fig. 1). The page is loaded entirely only once, the first time it is requested. Besides the HTML and CSS code that make up the page, some JavaScript files are also downloaded: the AJAX engine. This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf. Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the AJAX engine instead. Any response to a user action that doesn't require a trip back to the server – such as simple data validation, editing data in memory, and even some navigation – the engine handles on its own. If the engine needs something from the server in order to respond – if it's submitting data for processing, loading

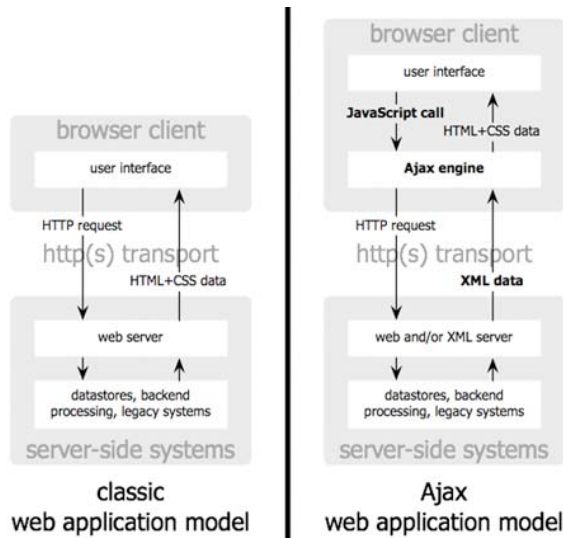


Figure 1. The traditional model for web applications (left) compared to the AJAX model (right)

additional interface code, or retrieving new data – the engine makes those requests asynchronously, usually using XML, without stalling a user’s interaction with the application [2].

An AJAX application eliminates the start-stop-start-stop nature of interaction on the Web by introducing an intermediary – an AJAX engine – between the user and the server. The AJAX engine requests information from the web server asynchronously. Thus, only small page bits are requested and sent to the browser, as they are needed by the user. The engine then displays the information without reloading the entire page. This leads to a much more responsive interface, because only the necessary information is passed between the client and server, not the whole page. This produces the feeling that information is displayed immediately, which brings web applications closer to their desktop relatives. So the user is never staring at a blank browser window and an hourglass icon, waiting around for the server to do something.

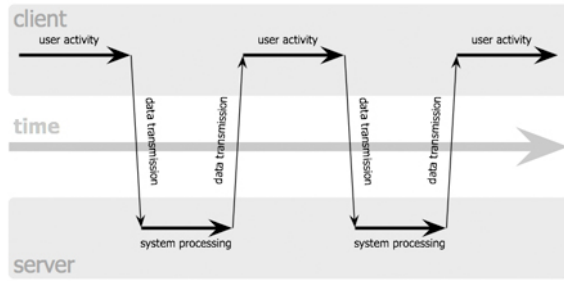
AJAX engine is nothing more than some JavaScript code that instantiates and uses the XMLHttpRequest object. This is a JavaScript object that allows sending, receiving and processing HTTP requests to and from the server without refreshing the entire page.

Although adding an extra layer to any kind of model should add to the response time, this is an exception. Through the use of this new layer – the AJAX engine – response time shortens and the user interface seems much more connected to the application logic (Fig. 2).

III. WEB APPLICATION

System for collecting statistical data in basketball and netcasting consists of three applications: main application used to collect statistical data during basketball game in relational database, Uploader application used to send data to central web server over FTP and web application used to show statistical data from following game.

classic web application model (synchronous)



Ajax web application model (asynchronous)

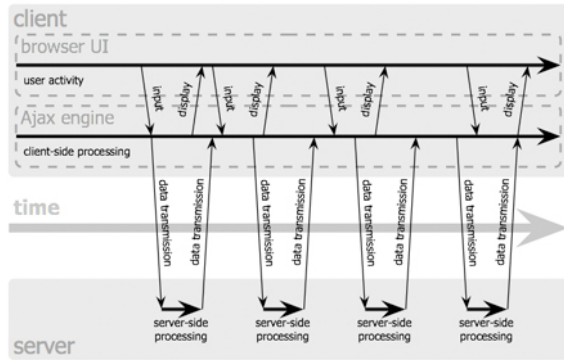


Figure 2. The synchronous interaction pattern of a traditional web application (top) compared with the asynchronous pattern of an AJAX application (bottom)

Htm file sent by Uploader application contains game statistic data in form of semi-colon delimited plain text. It’s very small in size, about 3 KB at the start of the game and only about 7 KB at the end of the game. Web application’s purpose is to show this statistical data to viewers over the internet. It is programmed using PHP, HTML and JavaScript.

PHP Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document [3]. PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not processed by PHP (although non-PHP text is still subject to control structures described within PHP code). The purpose of delimiters is to separate PHP code from non-PHP code, including HTML [4].

When user wants to follow netcasting of a certain game, he/she clicks on a link for that specific game on an internet site designed for netcasting. That starts Web application as a request for a single PHP source document, passing the game ID to it. Web server interpretes PHP code and generates HTML source document, which is sent to user’s web browser.

HTML, which stands for HyperText Markup Language, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. HTML documents are composed entirely of HTML elements that, in their most general form have three components: a pair of element tags with a “start tag” and “end tag”; some element attributes given to the element within the tags; and finally, all the actual, textual and graphical, information content that will be rendered on the display. An HTML element is everything between and including the tags. The ID attribute provides a document-wide unique identifier for an element and can be used by scripts to alter the contents or presentation of that element [5].

HTML page sent to user’s web browser consists mostly of INPUT elements as part of several FORM elements. For every single statistical data there’s one INPUT element that stores and shows that data on the page. Every INPUT element has a unique ID attribute, used by JavaScript to update statistical data stored in that element.

JavaScript is an object-oriented scripting language used to enable programmatic access to objects within both the client application and other applications [6]. The primary use of JavaScript is to write functions that are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page. The DOM is a cross-platform and language-independent convention for representing and interacting with objects in HTML documents. Aspects of the DOM (such as its "Elements") may be addressed and manipulated within the JavaScript [7].

After the page has loaded in user’s web browser, a call to a JavaScript function for fetching current statistical data from web server is made. It’s done by loading game’s htm file sent by Uploader application in an invisible IFRAME element of HTML page. When the file is loaded it is parsed by other JavaScript function to extract all statistical data from it, and using DOM all INPUT elements of the page are updated with new data. The refresh interval for getting newest htm file is set to 10 seconds. This way, game’s statistical data shown to the user is refreshed at every interval, without need for a whole page reload.

The web application is shown in Fig. 3.

IV. CONCLUSION

The AJAX approach to building web applications sheds a new light on the Web and helps close the gap

SUPER LIGA SRBIJE, Round: 5
Beograd, 23.05.2009, 20:15
Pionir, Spectators: 5825
Referee: Belaszevic L, Jovicic M, Vujinovic M.
Commissioner: Lozanov B.

[H] PARTIZAN73

[A] CRVENA ZVEZDA61

1/42/43/44/4E.T. Final

[H]1717241573

[A]1512132161

[H] PARTIZAN

Coach: Vojosevic D. - TFI: 0

No	Player	Pts	Free throws	2 points	3 points	Field goals	Rebounds	St TO	As	Blocks	Fouls	TF	Ind
5	Topic M.	5	2 4 50	0 0 0	1 2 50	1 2 50	3 0 3	0 3	4 0	0 5	5 0	6	
8	Milosevic S.	0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0 0	
10	Rasic A.	17	8 10 80	0 1 0	3 7 43	3 8 38	2 1 3	3 2	2 2	0 0	2 7	0 21	
11	Tripkovic U.	5	2 2 100	0 2 0	1 3 33	1 5 20	1 0 1	0 2	1 0	0 0	4 3	0 0	
12	Velickovic N.	9	5 6 83	2 2 100	0 4 0	2 6 33	5 0 5	2 0	2 0	0 3	4 0	14	
13	Lazma S.	12	2 2 100	5 6 83	0 2 0	5 8 62	4 0 4	0 2	3 1	0 1	1 0	15	
14	Aleksic V.	0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0 0	
15	Vitkovac C.	2	2 4 50	0 0 0	0 0 0	0 0 0	0 1 0	1 0	1 0	0 0	1 3	0 3	
20	Basic P.	6	0 0 0	0 0 0	0 2 4	50 2 4	50 0 0	2 0	2 0	0 5	1 0	4	
22	Rakocvic Z.	6	0 0 0	3 3 100	0 0 0	3 3 100	3 2 5	0 0	1 1	0 0	3 0	8	
24	Vesely J.	11	3 4 75	4 5 80	0 2 0	4 7 57	3 2 5	0 0	1 1	0 2	3 0	15	
33	Vranes S.	0	0 0 0	0 1 0	0 0 0	0 1 0	0 0 0	0 0	0 0	0 1	0 0	-2	
73 24 32 75 14 20 70 7 24 29 21 44 48 22 5 27 7 10 16 2 0 2 27 0 84													

[A] CRVENA ZVEZDA

Coach: Pesic S. - TFI: 0

No	Player	Pts	Free throws	2 points	3 points	Field goals	Rebounds	St TO	As	Blocks	Fouls	TF	Ind
4	Kezelj M.	4	2 3 67	1 2 50	0 2 0	1 4 25	3 1 4	1 1	1 0	0 3	3 0	5	
5	Roberts L.	6	0 2 0	3 6 50	0 1 0	3 7 43	1 3 4	1 1	1 0	0 4	1 0	2	
9	Kikanovic E.	3	1 1 100	1 4 25	0 1 0	1 5 20	1 1 2	0 0	0 0	0 2	2 0	-2	
10	Markovic S.	9	2 2 100	2 5 40	1 2 50	3 7 43	2 1 3	0 0	2 1	0 0	2 4	0 9	
14	Marinovic M.	15	4 4 100	4 4 100	1 6 17	5 10 50	1 0 1	1 3	1 0	0 4	5 0	11	
17	Stevic O.	0	0 0 0	0 1 0	0 0 0	0 1 0	1 0 1	0 0	0 0	0 1	1 0	-2	
20	Dragicevic T.	11	3 4 75	4 6 67	0 1 0	4 7 57	2 2 0	0 0	0 0	0 0	4 3	0 7	
21	Milosevic M.	0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0	0 0	0 0	0 0	0 0	
23	Owens A.	5	2 4 50	0 0 0	1 1 100	1 1 100	0 0 0	0 0	0 0	0 0	4 0	8	
30	Stimac V.	1	1 2 50	0 0 0	0 0 0	0 0 0	0 1 0	1 0	1 0	0 0	1 1	0 0	
33	Balic B.	0	0 0 0	0 1 0	0 1 0	0 2 0	0 0 0	0 0	1 0	0 0	1 1	0 -3	
44	Bjelica N.	4	4 5 80	0 0 0	1 1 100	1 1 100	5 1 6	0 2	1 0	0 5	3 0	9	
61 19 27 70 15 29 52 4 16 25 19 45 42 17 7 24 3 14 6 0 2 27 0 44													

Figure 3. Web application

between web applications and desktop applications in terms of usability and responsiveness. The biggest challenges in creating AJAX applications are not technical. The core AJAX technologies are mature, stable, and well understood. The most important thing to remember is to create the best possible user experience.

This paper described AJAX web application that shows live basketball statistics of a certain game to viewers over the internet (netcasting). Presented solution is, with small changes, applicable to other sports.

REFERENCES

- [1] C. Ullman, L. Dykes, “Beginning Ajax”, 2007.
- [2] J.J. Garrett, “Ajax: A New Approach to Web Applications”, AdaptivePath.com, 2005.
- [3] D.R. Brooks, “An Introduction to PHP for Scientists and Engineers”, 2008.
- [4] D. Powers, “PHP Solutions - Dynamic Web Design Made Easy”, 1st Edition, 2006.
- [5] D.R. Brooks, “An Introduction to HTML and JavaScript for Scientists and Engineers”, 2007.
- [6] T. McNavage, “Getting Started with JavaScript”, 1st Edition, 2010.
- [7] J. Keith, “DOM Scripting - Web Design with JavaScript and the Document Object Model”, 2005.