

Novel Technology: Single Page Web Application

Bipradip Roy¹, Sk Rubel²

1 Dept. of Computer Applications, Dayananda Sagar College of Arts, Science and Commerce

2 Dept. of MCA, Acharya Institute of Technology, India.

ABSTRACT

Web development technique for developing interactive web applications, termed as AJAX has emerged. The single-page web interface is composed of individual components which can be updated independently. Until a year ago, the difficult task was migrating legacy systems to web-based settings, today a new challenge of migrating web applications to single-page AJAX applications exist.

Gaining an understanding of the navigational model and user interface structure of the source application is the first step in the migration process. In this paper, we explore how reverse engineering techniques can help analyze classic web applications for this purpose. Our approach, using a schema-based clustering technique, extracts a navigational model of web applications, and identifies candidate user interface components to be migrated to a single-page AJAX interface.

Key Words: AJAX, Bootstrap.

1. INTRODUCTION

A single-page application (SPA) is a web application or web site that fits on a single web page with the goal of providing a more fluent user experience similar to a desktop application. In a SPA, either all necessary code – HTML, JavaScript, and CSS – is retrieved with a single page load. The page does not reload at any point in the process, nor does control transfer to another page, although the location hash can be used to provide the perception and navigability of separate logical pages in the application.

1.1 Advantages

All data has to be available via some sort of API - this is a big advantage for my use case as I want to have an API to my application anyway. Right now about 60-70% of my calls to get/update data are done through a REST API. Doing a single page application will allow me to better test my REST API since the application itself will use it. It also means that as the application grows, the API itself will grow since that is what the

application uses; no need to maintain the API as an add-on to the application.

More responsive application - since all data loaded after the initial page is kept to a minimum and transmitted in a compact format, data requests should generally be faster, and the server will do slightly less processing.

SPA is almost like client server technology, where HTML page is static and all dynamic changes occur in browser. In earlier PHP, JSP, ASP, HTML was mixed with server side logic and it was generated on the server. Server had to process more load.

SPA separates UI and data, SPA communicates with server only with JSON REST API (Send/Receive JSON using AJAX), this also allows both parts to be independently developed and tested.

You can easily fake JSON data communication to test SPA, and you can also easily fake JSON requests to server to write unit tests. SPA is fast, as most resources HTML, CSS, Scripts are only loaded once, throughout life span of application, only data is transmitted back and forth. SPA can use caching and local storage effectively. It is easy to scale, and it is easy to cache resources. SPA works and feels more like an application then a web page.

Back button issue, mostly all communication is performed using AJAX promises, loss of back page is not an issue. In HTTP Post operation, failure on server side would mostly require to redo entire data entry. One would lose back button and accidental navigation on other pages would result in redo the data entry. These issues can be fixed very well. SPA is easy to debug with chrome, as you can monitor network operations, investigate page elements and data associated with it. Rendering html obviously takes some resources now every user visiting your site is doing this. Also not only rendering major logics are now done client side instead of server side.

1.2 Disadvantages

Client must enable JavaScript, Single Page Application build with JavaScript, So JavaScript should be enabled in client browser. JavaScript enabled in all modern browsers by default.

Security: Compare to traditional page Single Page Application is less secure due to Cross-site scripting (XSS). Memory Leak: Memory leak in JavaScript can even cause powerful system to slow down. Duplication of code - for example, model code. I am going to have to create models both on the server side (PHP in this case) and the client side in JavaScript. Business logic in JavaScript - I can't give any concrete examples on why this would be bad but it just doesn't feel right to me having business logic in JavaScript that anyone can read.

JavaScript memory leaks - since the page never reloads, JavaScript memory leaks can happen, and I would not even know where to begin to debug them. The biggest disadvantage is that the client must have JavaScript enabled and be powerful enough to run a fair amount of it. It's also harder to satisfy accessibility concerns or anything else that relies on parsing static HTML. Inability to link to a specific part of the site, there's often only 1 entry point. Dysfunctional back and forward buttons. The use of tabs is limited or non-existent. Take very long to load. Can't reload a page, a sudden loss of network takes you back to the start of the site.

2. TECHNOLOGY

2.1 HTML5

HTML5 is the fifth revision and newest version of the HTML standard. It offers new features that provide not only rich media support, but also enhance support for creating web applications that can interact with the user and servers, more easily and effectively than was possible previously.

Technically, HTML is not a programming language, but rather a markup language. It was finalized, and published, on 28 October 2014 by the World Wide Web Consortium. Its core aims are to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices. HTML5 is a response to the fact that the HTML and XHTML in common use on the World Wide Web has a mixture of features introduced by various specifications, along with those introduced by software products such as web browsers and those established by common practice.[5] It is also an attempt to define a single markup that can be written in either HTML or XHTML.

HTML5 introduces elements and attributes that reflect typical usage on modern websites. In particular, HTML5 adds many new syntactic

features. These include the new <video>, <audio> and <canvas> elements, as well as the integration of scalable vector graphics (SVG) content (replacing generic <object> tags) and MathML for mathematical formulas.

2.2 CSS3

CSS stands for Cascading Style Sheets. CSS is a style sheet language used for describing the presentation of a document written in a markup language. CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

CSS3 has been split into "modules". It contains the "old CSS specification". In addition, new modules are added. Some of the most important CSS3 modules are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

2.3 Bootstrap

Originally released on Friday, August 19, 2011, bootstrap is a free and open-source collection of tools for creating websites and web applications. Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Bootstrap includes css files, javascript files, and images.

3. CONCLUSION

In this paper, we emphasized the rise of single-page AJAX applications and the need for support in migrating classical multi-page web applications to this new paradigm.

REFERENCES

- [1] M. Abrams, C. Phanouriou, A. L. Batongbacal, S. M. Williams, and J. E. Shuster. UIML: An appliance-independent XML user interface language. In WWW '08: 8th International Conference on World Wide Web, pages 1695–1708, 1999.
- [2] G. Antoniol, M. Di Penta, and M. Zazzara. Understanding web applications through dynamic analysis. In IWPC '04: 12th IEEE International Workshop on Program Comprehension, page 120. IEEE Computer Society, 2004.
- [3] P. Barone, P. Bonizzoni, G. D. Vedova, and G. Mauri. An approximation algorithm for the shortest common supersequence problem: an experimental analysis. In SAC '01: ACM symposium on Applied computing, pages 56–60. ACM Press, 2001.
- [4] J. Conallen. Building Web Applications with UML (2nd Edition). Addison-Wesley, 2003.
- [5] J. R. Cordy, T. R. Dean, and N. Synytsky. Practical languageindependent detection of near-miss clones. In CASCON '04: Conference of the Centre for Advanced Studies on Collaborative research, pages 1–12. IBM Press, 2004.
- [6] A. De Lucia, R. Francese, G. Scanniello, and G. Tortora. Reengineering web applications based on cloned pattern analysis. In IWPC '04: 12th IEEE International Workshop on Program Comprehension, page 132. IEEE Computer Society, 2004.
- [7] A. De Lucia, R. Francese, G. Scanniello, and G. Tortora. Understanding cloned patterns in web applications. In IWPC '05: 13th International Workshop on Program Comprehension, pages 333–336. IEEE Computer Society, 2005.
- [8] A. De Lucia, G. Scanniello, and G. Tortora. Identifying clones in dynamic web sites using similarity thresholds. In International Conference on Enterprise Information Systems, pages 391–396, 2004.
- [9] G. A. Di Lucca, M. Di Penta, and A. R. Fasolino. An approach to identify duplicated web pages. In COMPSAC '02: 26th International Computer Software and Applications Conference, pages 481–486. IEEE Computer Society, 2002.
- [10] G. A. Di Lucca, A. R. Fasolino, F. Pace, P. Tramontana, and U. De Carlini. Comprehending web applications by a clustering based approach. In IWPC '02: 10th International Workshop on Program Comprehension, page 261. IEEE Computer Society, 2002.
- [11] G. A. Di Lucca, A. R. Fasolino, F. Pace, P. Tramontana, and U. de Carlini. WARE: A tool for the reverse engineering of web applications. In CSMR '02: 6th European Conference on Software Maintenance and Reengineering, pages 241–250. IEEE Computer Society, 2002.
- [12] D. Draheim, C. Lutteroth, and G. Weber. A source code independent reverse engineering tool for dynamic web sites. In CSMR '05: 9th European Conference on Software Maintenance and Reengineering, pages 168–177. IEEE Computer Society, 2005.