

Optimizing JavaScript web applications

Directeur: [David Bromberg](#)

Developing web applications has always been cumbersome, and is usually divided in two parts: front-end and back-end. The main purpose of the front-end, usually developed with a mix of JavaScript, html and css, is to interact with users, and present data in a well-defined style.

On the other side, the back-end is the part of the application that is never visible to the end users: it embeds the logic of the application, and is usually developed with other programming languages. Front-end and back-end are completely separate domain, owned by developers with a different set of skills, increasing drastically its complexity. In particular, front-end and back-end being developed separately, a lack of performances inherently occurs, and definitely requires stronger development coordination, and a uniform view of the targeted web application.

Through the years, several methods have emerged to deal with this complexity, such as for instance GWT, Vaadin that promote to use Java as a single programming language to enables developers to produce high-performance web applications without being an expert with familiar web technologies such as JavaScript, html and css. The java code targeting the front-end is compiled in JavaScript. Such solutions encounter different degrees of success, and still require a hard learning curve to use a specific client-server design patterns.

Further, it fails to hide complex abstraction differences between JavaScript and Java. In many ways, JavaScript is more powerful and expressive than Java, and using Java, as pivot language, does not appear as the most adequate solution. With the rise of Node.js and JavaScript on the back-end side, the use of JavaScript as a pivot language becomes a reality. A new door is open to many front-end developers, enabling them to move to the back-end side, and to develop at once both the backend and the front-end.

This paradigm shift opens the way to new tools and interaction paradigms to solve UI-specific concerns and to rethink web application architecture to overcome significant performances bottlenecks in Web applications.

In this work, we propose to explore and evaluate the impact of the use of JavaScript as a pivot language to develop web applications, and to propose novel kinds of encapsulation of web application states shared between the front and the back end sides. As both sides are developed in JavaScript, It enables many new ways of improvements.

This work encompasses a state of the art of existing JavaScript framework usually used at the front-end side. Further, we will provide methodologies, code patches and algorithms to optimize their smooth integration with javascript back-end sides.

Références :

Saba Alimadadi, Sheldon Sequeira, Ali Mesbah, and Karthik Pattabiraman. 2014. Understanding JavaScript event-based interactions. In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014). ACM, New York