Ralph Steyer

Building web applications with Vue.js

MVVM patterns for conventional and single-page websites

Springer

Building web applications with Vue.js

Ralph Steyer

Building web applications with Vue.js

MVVM patterns for conventional and single-page websites

Springer

Ralph Steyer
Bodenheim, Germany

ISBN 978-3-658-37595-9 ISBN 978-3-658-37596-6 (eBook)
https://doi.org/10.1007/978-3-658-37596-6

This Springer imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Preface

Of course, the Internet is still on everyone's lips. Digitization in general is one of the most used "buzzwords" – especially by politicians, media people, and decision makers. But the times of static websites are largely over. While it was common a few years ago that at least simple websites of private persons or smaller associations still used pure HTML code (Hypertext Markup Language), such antiquated websites can be found less and less nowadays. Usually at least content management systems (CMS) like WordPress, Joomla!, Typo, or Drupal are used or such websites are at least spiced up with stylesheets and/or JavaScript.

But if you are not satisfied with these 08/15 solutions of the common CMS or if they do not (cannot) meet the required requirements, the only option is the real programming of websites or even Web applications. On the one hand, this path usually begins with programming on the Web server side, including downstream database systems, but on the other hand, modern websites or Web applications must also be programmed in the client (the browser).

And in the browser, only JavaScript has been available for years as a universally available technology for programming. Together with HTML and CSS (Cascading Style Sheets), JavaScript forms the triad of modern websites and especially of client-side Web programming.

Now JavaScript has been around for a very long time on the Web, but for most years it was completely underestimated and dismissed as a primitive beginner's language. Only in the last few years have we realized what a treasure JavaScript is for efficient and powerful programming and that this language is anything but a beginner's language – even if you can learn it quickly as a beginner. Quite the contrary. But you have to be able to use it professionally, because in contrast to all-round carefree programming worlds from the .NET and Java environment, which have "softened" programmers over the years by hardly allowing them to make mistakes, you can't expect such protective mechanisms in JavaScript. Efficient and secure programming with JavaScript requires skills from the programmer instead of transferring them to IDEs (Integrated Development Environment) and runtime environments. But this also makes JavaScript much leaner and more efficient than its now hopelessly overloaded competitors. Professional, efficient, and secure programming with JavaScript therefore requires immense programming experience. Already in the

client, but even more so on the server side, where JavaScript has meanwhile also begun its triumphant march. JavaScript is like a scalpel. In the hands of an experienced surgeon, you can perform miracles with it. In the hands of a layman or someone not working carefully, it can do immense damage. Where JavaScript manages the balancing act of still being easy to use by beginners for simple tasks. In my JavaScript trainings, I hear again and again that many participants have already used JavaScript. Mostly they copied existing scripts and adapted them if necessary or wrote very simple scripts themselves and built them into HTML pages. And that usually works, although almost always the addition of the participants was that they did not really know why it works.

The many frameworks that have established themselves in the Web environment in recent years use some fundamentally different approaches. But most frameworks often try to extend JavaScript by things that are not possible with the core version. This makes it easier to work with JavaScript and also provides possibilities that are not or not easily available with pure JavaScript.

The various frameworks take very different approaches to ultimately end up with the same result in the code that reaches the user, namely a conglomerate of HTML, CSS, and JavaScript. Of course, always connected with resources such as images, videos, audios, and the like.

Now, Vue.js is an increasingly popular framework on the Web that takes a very specific approach. It is a reactive, client-side JavaScript Web framework that is essentially used to create so-called single-screen Web applications or single-page Web applications (only one Web page in the browser that updates parts as needed and does not reload a new Web page) according to a Model-View-Controller pattern (MVC). Strictly speaking, a Model-View-Controller pattern (MVVC) is used. But you can definitely create Web applications and Web pages with it.

Vue.js is both easy to learn and very extensible and customizable. To be able to learn Vue.js successfully, a good knowledge of HTML and JavaScript is sufficient, as well as CSS if possible, which I would also like to assume in the book. The developers of Vue.js call the framework “progressive.” This essentially means that it can be used as much for small improvements to individual details of the website as for larger projects. It also supports the creation of reusable components. Another advantage of Vue.js is that it does not require a complex installation and can even be used entirely without installation “from the cloud” (from a CDN content delivery network) if required.

So, follow me into the fascinating world of Vue.js!

Bodenheim, Deutschland
Spring/Summer 2019

Ralph Steyer
http://www.rjs.de

Contents