

Book reviews

Building Web Applications with Erlang, by Zachary Kessin, O'Reilly Media, Inc., 2012, ISBN 97-81-4493-20652
doi:10.1017/S0956796812000342

This book presents a very interesting and timely subject: web application construction using the functional programming language, Erlang. The subject is especially relevant in the light of the growing popularity that Erlang has been gathering in the last few years. With the success of the so-called “cloud model,” more and more services are migrating to a distributed web environment, where high-reliability, fault-tolerance, and easy maintenance have become essential ingredients for success. In this context, the characteristics of Erlang (built-in concurrency, lightweight process-oriented philosophy, transparent distribution, and hot code update capabilities) have granted this functional language a positive reputation, especially after being part of well-known products as Ejabberd, Amazon’s SimpleDB, and CouchDB among others, which have also been the subject of books published by O’Reilly Media, Inc.

The author presents his book as an introduction to large-scale web services and starts off by explaining differences between “traditional” web services (built using PHP/Apache and the like) and the construction of web services using Erlang and Erlang-based libraries and tools. Throughout the volume, different aspects of web applications are dealt with (such as appmods, REST, websockets, and content uploading and streaming), with some hands-on orientation, provided via source code examples. This must not, however, be taken to be an instructional text for either Erlang or the OTP libraries, or indeed other well-known Erlang applications such as the yaws web server. It is not a text on web services and applications, rather it is a handy guide for web service professionals to know how to do everything they already know, but using a new set of tools.

As clearly stated in its preface, this book does not address beginners, neither of the Erlang language nor the web services and applications world. The perfect fit for the book are people who are very familiar with building web applications in other contexts and want to see what are the dos and don’ts when using Erlang for that same purpose, provided they have already been in touch with Erlang. A different reader, such as an Erlang beginner, will not get the same understanding about what is the strength behind this language that makes it such a good candidate for the scenarios that are mentioned. A newcomer to web services, on the other hand, will find that familiarity with components and so forth is assumed on the reader’s side.

This book comes to fill in an important gap: To the best of this reviewer’s knowledge, there are no other titles that deal specifically with the subject of web services from the point of view of the emerging functional language Erlang. However, this short volume leaves room for wider, perhaps more ambitious, works. It would be interesting to read about web services without the constant comparison with traditional technologies. After all, Erlang beginners are told once and again that they need a shift in perspective when moving into the Erlang programming paradigm. So it may be worth to put together a volume in which web services construction is addressed for Erlang and web application beginners. Of course, such a book is likely to be very long and dense, so perhaps narrowing the scope to address REST web services would be interesting, should Erlang consolidate as a firm alternative in this business area.

The book begins with a nice introduction to Erlang, in which strengths and weaknesses are well presented, and pros and cons are discussed in a very objective and balanced way. Also, a review of non-SQL alternatives to support a web service or application data persistence is presented. This is followed by a chapter on simple, static-content web site configuration using

yaws, the Erlang-based web server. Background knowledge about web services is assumed, and a number of concepts and configuration possibilities are explained comparing and contrasting with PHP *et al.* The reader might find a bit confusing, strange, or even frustrating that some very simple things are illustrated with basic examples, and right afterwards more interesting and advanced pieces are left to the readers' own extrapolation capabilities. The overall sensation is that it is hard to tell if the writer wants us to follow the book as a hands-on tutorial or manual, since we will find ourselves lacking basic information at some points; but at the same time the amount of source code in examples is so prolific that it is hard to resist the temptation to try and do it oneself.

The book is very ambitious in that it mentions every aspect about web services that one may think of, but in many cases after a few thoughts the author leaves us with one of the most repeated sentences in the volume: "beyond the scope of this book." However, a great point in its favour is the great amount of time and effort that the author has devoted to actually research and select complementary sources of information, which he points to the reader. In Chapter 4, the reader is presented with the first full example, which puts all previously examined pieces together. From then on, we see a series of examples of good practice, again especially focused on explaining key differences with other frameworks such as PHP+Apache. Beside useful pointers to external resources for further information and material, a clear explanation on how to interact with Amazon S3 service is also presented.

Although the first part of the book is heavily oriented to REST-services, Chapter 6 is devoted to websockets, a sort of TCP-socket-like behaviour over HTTP, which represent the opposite philosophy to REST, thus complementing the content so far.

The rest of the book shows specific examples of how to do specific but very common tasks (content upload and streaming, for instance), including a discussion of HTTP client alternatives to be used for testing purposes (CURL, https, etc.). Last but not least, a nice full OTP-flavoured example is included in the last chapter, which in a way is one of the barriers for readers who are not so familiar with web services: a reader cannot see the whole picture until the end, and chapters do not seem to follow a cohesive storyline, they just cover different aspects, which again bring to mind the dichotomy between manual and book.

This book was reviewed in e-book format, which was unfortunately not a very polished edition. Some figures were too small to be read in a 6-inch e-reader, and the verbatim font used for source code sections was not only poorly formatted but also its greyscale colour was too faint to read at times.

Overall, this is a nice book which provides easy reading for professionals, experts in the web-services world that have come across Erlang and want to see that the same things they are used to can be done with Erlang and Erlang-based tools in a very simple and quite similar way to what they already know. It is definitely a volume this reviewer would recommend to read in a digital format, preferably with Wi-Fi capabilities, such is the number of online resources and references that have been meticulously researched and kindly presented. However, this is not a volume for those seeking the basic knowledge to implement a web service, because neither does it give an introduction to what Erlang is capable of nor is it a reference book.

LAURA CASTRO

University of A Coruña

Pearls of Functional Algorithm Design, by Richard Bird, Cambridge University Press, September 2010, £35.00, US \$ 60.00. ISBN: 978052151338 (hardback), 286pp
doi: 10.1017/S095679681200041X

When I first started learning Haskell, I did so by adapting a Sudoku implementation into dealing with partial Latin Squares. That implementation was from a Functional Pearl by