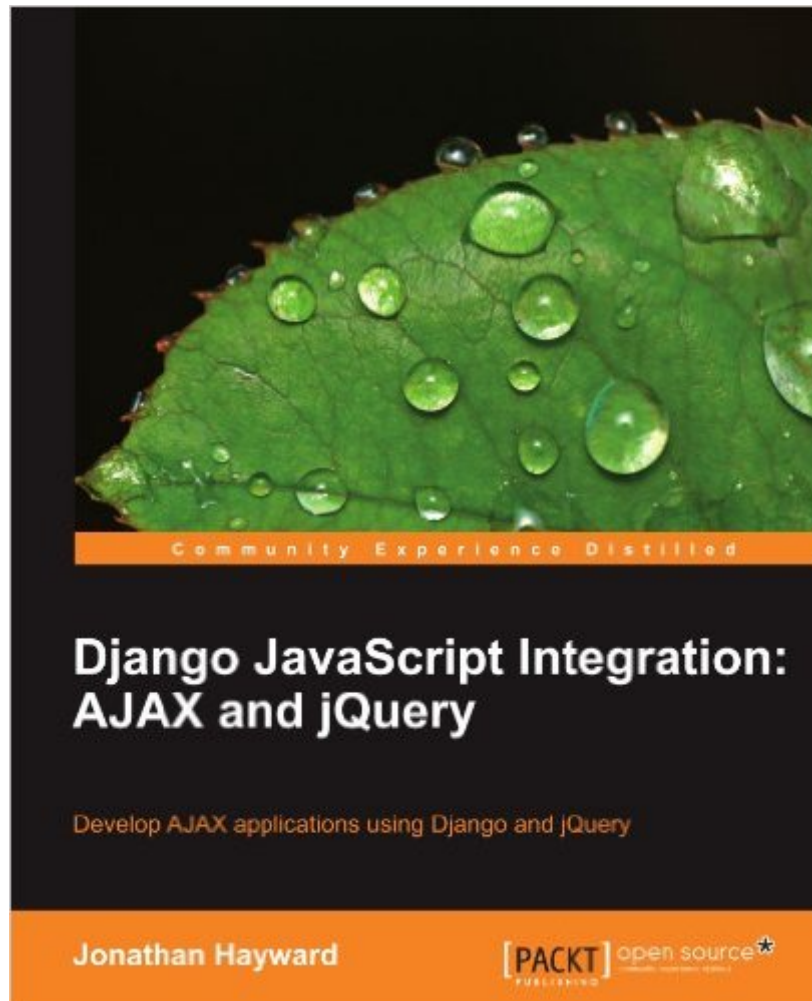# Django JavaScript Integration: AJAX And JQuery

Synopsis

This book will teach you how to enrich your AJAX applications with Django and jQuery in a practical manner with minimum fuss. It introduces AJAX basics, gets a little deeper, and then goes chapter-by-chapter through a carefully chosen case study. It demonstrates how different pieces of a Django AJAX application work in themselves and are put together and specifically leads you through building an intranet employee photo directory with social networking features. This book is for people looking to integrate AJAX / JavaScript functionality into their web applications. It is for Django users who are looking to easily integrate AJAX features into their applications. Conversely, it will also be a priceless companion for users familiar with Django and jQuery who are and looking to integrate them in their AJAX applications. A working knowledge of Django and basic familiarity with AJAX and jQuery are assumed.

Book Information

Paperback: 324 pages

Publisher: Packt Publishing (January 4, 2011)

Language: English

ISBN-10: 1849510342

ISBN-13: 978-1849510349

Product Dimensions: 7.5 x 0.7 x 9.2 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 1.6 out of 5 ststarsÂ Â See all reviewsÂ (5 customer reviews)

Best Sellers Rank: #2,596,437 in Books (See Top 100 in Books)   #83 inÂ Books > Computers & Technology > Programming > Languages & Tools > Ajax   #1076 inÂ Books > Computers & Technology > Networking & Cloud Computing > Data in the Enterprise > Client-Server Systems #4888 inÂ Books > Computers & Technology > Web Development & Design > Web Design

Customer Reviews

I was actually asked by a representative of the publisher to review this book because I have blogged on a similar topic before [...] - this very review also appears on my blog in slightly modified form. Never one to turn up my nose at an opportunity to read a full-length book on a topic I find interesting for free, I agreed. I was hoping for a detailed technical discussion of Django and jQuery integration, but by and large I was disappointed. Hayward comes across as a mediocre technical programmer at best, and the code examples are riddled with inefficiencies, eccentric departures from mainstream best practices for both libraries, and out-and-out errors which should not have

made it into the final text. I wanted a technical discussion and I was left feeling uninspired in this area. As a treatise on UI design which draws on Django and jQuery for its technical inspiration, and as a source of some ideas with possible merit for programmers who are already experienced in this area, the book has more merit. All in all, I'm glad I didn't pay good money to read the book, however.First of all, I have to get something out of the way: this is not a book I would ever recommend to beginners. Not because the treatment is at a higher level, but somewhat the opposite. To begin with, in the current edition of the book, the text introducing jQuery is riddled with basic errors which will confound beginners. I honestly believe that Hayward knows better, but this section of the book seems like it may have been rushed into print with inadequate proofreading and copy editing. Anyway, it's sloppy, and only readers who are reasonably experienced with jQuery will spot the errors without being lead down any garden paths. While Hayward is apparently much more comfortable in Python and the Django and these parts of the text are much more solid, I don't find that Hayward especially shines in this area from a technical standpoint, either. In general, the jQuery and Django sites still provide far superior documentation and basic tutorials for their libraries. Beginners would be far better served to review the tutorials provided on these sites - Hayward adds nothing in particular until you have acquired a strong grounding elsewhere.What I was hoping for was a technical discussion on how jQuery and Django interoperate, and in this area I also found Hayward fell short for me. Yes, the basics were explained, somewhat. But Hayward doesn't honestly seem to like or trust Javascript or jQuery especially... to me, he almost seems to treat jQuery as the best of a bad lot of Javascript libraries. Maybe it's just me, but I happen to love jQuery, and I felt like it deserved better. It's discouraging for a technical writer to evidently have a tin ear for beautiful code. He doesn't focus on what I would consider mainstream jQuery practices, instead preferring to recommend eccentric or heterodox practices like using the $.ajax() method as much as possible to make AJAX requests, and avoiding using jQuery's own facilities for using JSON, apparently believing them to be a security risk (a point which I don't think is adequately substantiated). One fact I find positively risible is that Hayward binds events to elements in the old style via properties on those elements and completely bypasses the jQuery way of handling events, or for that matter any semblance of non-jQuery Javascript current best practices in this area.I find there are eccentric elements on the Python side, too. This is another reason why Hayward's book is really only for experienced programmers - I'm not saying that Hayward's opinions are necessarily wrong about these matters, although in some places I don't happen to agree, but they require engagement at the level of someone already familiar with the pertinent issues and would be unwise to take as Gospel. Like for example, he splits a user query into "word" chunks using the bit of

code:split_query = re.split(ur'(?u)\W', query)while u'' in split_query: split_query.remove(u'')Ignoring the ubiquitous u's, which is probably laudable but something I would not bother with, why would you do this? Why not split on \W+ and not have to remove any empty strings in the first place (best option), or else use a list comprehension like split_query = [x for x in split_query if x != u'']? Again, Hayward's code seems rushed and inelegant, even in his area of greater strength.What's missing is important workaday tasks in Django and jQuery integration like serializing forms in jQuery and feeding them to Django, and what to do with that data in Django. One critical issue that is missing is the $.ajax() method's "traditional" parameter, which if not manually set to true will cause surprising loss of backward compatibility for Django code written to operate with jQuery 1.3 and below (and requires extra processing steps for Django code written for jQuery 1.4 and up). This is related to jQuery optimizing for the behavior of PHP and Ruby on Rails, and is something Django developers need to watch for.One thing that stood out for me: In his jQuery code, Hayward hard-codes URLs, including some rather ugly ones with query strings built in, directly into his AJAX calls, saying (I paraphrase) that there is not really a good way to follow accepted Django best practices here of not hard-coding URLs in templates. If the jQuery code is generated by templates to which you have pushed the relevant parameters into the context, you simply have to add the URL you want to your urlconf in Django and use the {% url %} tag in your Django template. If you want to feed data to the view via the request rather than via the URL parsing, that is what the data parameter of the AJAX call is for - there is no need to manually hack together query strings. This is not a hard solution to implement, and emblematic of the kind of discussion I wanted to see which I feel is missing from this book.In general, this is not a technical manual, and technical issues are not the author's primary focus and concern. If you come to this book hoping for an advanced technical discussion of how jQuery and Django, taken in their most conventional and mainstream style are best able to interoperate, you will be disappointed. This is not a programming treatise, it is a philosophical treatise from the point of view of a programmer about user interface design as it affects web application development using jQuery and Django integration as its practical example. On this level it's a stronger book, although frankly in many places I found the philosophical excursions a little tedious and trite and not particularly enlightening. Hayward makes a decent argument for his views on proper programming to interact with users. In a few places, I found Hayward's contentions about optimizing the user experience to be genuinely interesting and enlightening. Given the profusion of dreadful user interfaces out there, many programmers could certainly use to take such arguments under serious consideration, although I can't help but feel there must be better books on the market to address this kind of topic in a straight-forward way without dashing anyone's hopes of achieving a

more technical enlightenment.While Hayward's coding, at least as shown in the examples in the book, is mediocre at best and certainly cannot be said to sparkle anywhere, his philosophical system of user interface design which he uses Django and jQuery to illustrate is reasonably serious and adept, and it is on this level that the book should best be engaged. There are also some practical ideas in Hayward's central example throughout the book, the photo directory, from which a reader may glean some useful ideas.

There is a formula for books that cover a specific topic such as a web framework.The first chapter is an overview of the technology. The second chapter covers installation of the software. The third chapter jumps into an example. Subsequent chapters expand on that example. Final chapters may provide alternative ways of accomplishing something from earlier chapters, or cover a specific aspect of the technology that is more complex than people realize.The reason this formula exists is because it works. Unfortunately Jonathan Hayward chose to completely ignore the formula and put together this confusing and poorly organized book.Chapter 1 and 2 are a sort of high level overview of the topics in the book. Like the rest of the book even these chapters aren't well organized or structured. Chapter 2 _briefly_ touches on how to create a Django project- but there is no real discussion on how Django works, how it is structured, what dependencies it has, why you might choose it over Pylons, or any other even remotely useful information.Chapter 3 is nothing but the Authors personal diatribe on UI design. While he makes several good points that would be appropriate in a book on UI design- they are a complete waste of space in a book on Django and jQuery integration.Subsequent chapters are a mix of code and topics in only the barest semblance of order. Models are introduced without any real explanation of why or how they derive from Django.This book would have been infinitely more useful if the author had taken us through creation of his example application in more depth and with fewer asides and detours. He seems to enjoy telling us that there are two ways to do things- explaining one- then telling us not to do it that way for these reasons and then moving on to the correct way to do it.For example- the author says there are two ways we could perform a search- transmit the entire database to the client and do the search in javascript, or do it on the server side. Why he feels the need to explain that doing the search on the client side is bad, but does not feel the need to go into any depth on Django itself, is beyond me. The sort of person that does not understand the client side search problems probably can't program in Python in the first place.All in all I found this book confusing, disorganized and just generally not what I was expecting based on the title.

Technical review are already given by more professional reviewers, here I only describe my own reading experince.I really TRIED to read this book, I tried three times, very hard, and I could not go very far. The feeling of reading this book is very strange. It's like a person who is not very logically clear tries to describe to you some idea in a very professional tone and demands you to listen very carefully. For one reason, the author really likes long side track to his mainstream stories --- in fact, he even divide his side tracks into sections! For another reason, even when coming to paragraph-sized explanation he is at most doing an on-par job explaining stuff. When reading the book, I cannot help feeling that I am facing a very energetic professor who is so excited that he spent hours explaining things that have little relevance to the topic he started, and that I was too polite to stop him and have to listen.Just to be fair, I'm a beginner to Django and I'm looking for a systematic book that is either beginner-friendly or manual-like (I deal with both kinds of books fine), but I really do not like this book.Again, this is my personal reading experience and for more professional review please see other reviews.

Django JavaScript Integration: AJAX and jQuery Two Scoops of Django: Best Practices for Django 1.8 Developing Responsive Web Applications with AJAX and jQuery A Software Engineer Learns HTML5, JavaScript and jQuery JavaScript & jQuery: The Missing Manual (Missing Manuals) WordPress and Ajax: An in-depth guide on using Ajax with WordPress Ajax Construction Kit: Building Plug-and-Play Ajax Applications JavaScript and AJAX For Dummies Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional (Beginning: From Novice to Professional) Scriptin' with JavaScript and Ajax: A Designer's Guide (Voices That Matter) M&A Integration: How To Do It. Planning and delivering M&A integration for business success Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript (Effective Software Development Series) Learning JavaScript: JavaScript Essentials for Modern Application Development Django: The Life and Music of a Gypsy Legend Django Django: World's Greatest Jazz Guitarist Murach's jQuery, 2nd Edition jQuery in Action Mobile Web Development: Building mobile websites, SMS and MMS messaging, mobile payments, and automated voice call systems with XHTML MP, WCSS, and mobile AJAX Developing Web Widget with HTML, CSS, JSON and AJAX: A Complete Guide to Web Widget