



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

EE Podcast #79 Add-on Development with Mark Croxtan

[Music]

Lea Alcantara: You are listening to the ExpressionEngine Podcast Episode #79. Today we are talking about EE add-on development with special guest, Mark Croxtan. I'm your host, Lea Alcantara, and I'm joined by my fab co-host,

Emily Lewis: Emily Lewis.

Lea Alcantara: This episode is sponsored by [EE Garage](#). EE Garage provides must have add-ons built by dedicated ExpressionEngine developers with over five years of experience. NSM Better Meta, NSM Reports and NSM Override are just some of the popular add-ons backed by an excellent support team. EE Garage is committed to the community and as a thank you for those tuning in, they are providing you, our beloved listener, with one free add-on. Visit ee-garage.com/eepodcast to get your premium add-on today.

Emily Lewis: The ExpressionEngine Podcast would also like to thank [Pixel & Tonic](#) for being our major sponsor of the year. [Music ends] Hey Lea, what's been going on with you?

Lea Alcantara: I'm enjoying fall for the most part. It's really beautiful this time of year, and it's still warm enough to do things outside. But that's not as exciting as you.

Emily Lewis: [Laughs]

Lea Alcantara: I heard you moved to a new place. How is that going?

Emily Lewis: Oh, it's going. I'm still unpacking, but it's a really beautiful, old, 100-year-old adobe house which is something I've always dreamt of living in since I moved to the southwest. It comes with a lot of black widow spiders and some roaches.

Lea Alcantara: [Agrees]

Emily Lewis: So [laughs] I'm getting used to my roommates.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Lea Alcantara: [Laughs]

Emily Lewis: But I have an exterminator that's trying to get that under control. But other than that, it's been a really great experience, stressful, but I'm really excited about being settled in here.

Lea Alcantara: Well, it sounds really cool.

Emily Lewis: Yeah, this is our first podcast from my new adobe house. [Laughs]

Lea Alcantara: Ooh.

Emily Lewis: On our [last podcast](#), we talked with Adrienne Travis about template partials with the Stash add-on.

Lea Alcantara: [Agrees]

Emily Lewis: And it was really inspiring for me, I'm convinced I must try her approach for my next EE build, but I don't have one of those quite around the corner yet. But I remember you mentioning it was something you were thinking about trying for a new client.

Lea Alcantara: Yeah, pretty much. So I actually did dive into it just recently with the current client of mine just to see how it goes, and you know what, with Adrienne's article and I also read an article from Viget's blog as well, it's so far so good.

Emily Lewis: [Agrees]

Lea Alcantara: What I'm struggling with a bit is what is okay to repeat and not repeat in terms of readability and still wrapping my mind around this way of creating templates.

Emily Lewis: Cool.

Lea Alcantara: Yeah.

Emily Lewis: Now, since we talked a fair amount about the Stash add-on with Adrienne and how she's building her template partials, we thought it would be fitting to talk to the creator of Stash and other extremely useful add-ons, [Mark Croxtan](#).



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark is a partner at the [Hallmarkdesign](#) consultancy where he focuses on web design and development. As we mentioned he's the developer behind some of the most popular and useful EE add-ons available including Stash, Switchee and IfElse. Welcome, Mark, thanks for joining our podcast today.

Mark Croxtan: Hello there, how are you?

Emily Lewis: We are great.

Lea Alcantara: Thanks for coming.

Mark Croxtan: Oh, you're welcome.

Emily Lewis: [Laughs] Did you have an opportunity to listen to our podcast with Adrienne and how she's using your Stash add-on?

Mark Croxtan: Yes, I did, yeah. It's very close to the approach I take as well. She sort of simplified it a little bit because she's only using one wrapper template, and I tend to switch out the wrapper template using Switchee.

Emily Lewis: Oh.

Mark Croxtan: Yeah, and it's kind of useful if you're trying to, let's say, create multilingual sites or sites where you've got multiple views for the same set of content.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Emily Lewis: Will that even apply like in a mobile situation?

Mark Croxtan: Indeed, yeah, so you could have a sort of adaptive design where the content and the presentation with the mobile is a bit different to the sort of tablet view or the desktop view.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Lea Alcantara: So when you are creating your templates, do you actually... I know when I was chatting with Adrienne, she prefers to keep some of the HTML in her templates a lot more, so it's a lot more readable and easier to update for her.

Mark Croxtan: Yeah.

Lea Alcantara: How do you approach it? Do you try to separate it as much as possible? What exactly is, in your opinion, the right balance?

Mark Croxtan: Well, if you can be quite purist about it, then you should really try and separate the HTML markup from the view model, if you like, entirely and either put it into snippets or use Stash list so that you don't actually have to have markup in that template at all. Because really the idea is that you have a separation of concerns. You have a view model which deals with the getting of data and this sort of presentation of logic and then you have a view template which just deals with the markup. In that way you can swap out the markup more easily and also reformat it on the fly for the different purposes, and like I said before, like modeling it sideways, you might have, for example, a right to left template for languages like Hebrew and Arabic or Chinese, and a left to right template for Latin-based languages.

Lea Alcantara: [Agrees]

Mark Croxtan: And that's just an example. It gives you that flexibility to have a separation between your view and model logic and your presentation of the markup.

Emily Lewis: Has that approach been your standard approach since beginning your work with EE? And is it that why you created some of these add-ons to scratch your own itch?

Mark Croxtan: It hasn't been my approach. No, I started out probably like most ExpressionEngine developers with the header template and the photo template.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Mark Croxton: And I was always dissatisfied with that approach because the page title are being set in the body of the template.

Lea Alcantara: [Agrees]

Mark Croxton: But you would have no means really to pass it to the title tag in the head.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Mark Croxton: And so you end up replicating the channel entries tag. You get multiple tags on the page, on the template, just to solve the problem of passing data between different parts of your templates.

Lea Alcantara: [Agrees]

Mark Croxton: And so I was always dissatisfied with that approach and I recognized that it wasn't working. I read this article by John Wells about template partials, and it just sort of struck me that that approach, it just made so much sense to me immediately. In the article, he encourages you to build your own plugin and that's exactly what I did for Stash.

Lea Alcantara: [Agrees]

Mark Croxton: I then set out to make like a definitive template partials plugin. I wanted to make something that works for me and explore what I could do with it, and it just so happen I have this quite complicated project that I started and I thought, "Well, I might just give it a go." It sort of evolved from that project. All the various bits and pieces to it grew from the requirements that came out of that project.

Emily Lewis: So it was initially something, Stash came about to address your template development and then it just be evolved to do so much more?



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark Croxton: Yeah, it's solved particular problems that I've been having and encountering time and again, and initially I sort of have this idea that you should have this separation of markup from the view model, but I wasn't able to implement that fully because I still ended up as Adrienne found. She's still, in her example, just ends up with bits of content mixed up into the view model.

Emily Lewis: [Agrees]

Mark Croxton: It's not ideal and I sort of compromised a bit by using snippets and putting those little bits of markup. For example, a list item will go into a snippet so that it's kind of in a separate place, but it's definitely a compromise, and I wasn't happy with that and that's why I took it. I kind of have pushed it a bit further, particularly with lists ... Stash lists to avoid that problem.

Lea Alcantara: [Agrees] So let's take a step back a bit, you had mentioned that you had a few problems before you started using Stash. What, in particular, were the main problems that you wanted Stash to solve?

Mark Croxton: I'm sure you've had this as well. You ended up creating nested embeds.

Lea Alcantara: Yeah.

Mark Croxton: Partially, it's because of parse order. You would just find that in order to get a particular item or data to another part of your template, you need to create an embed and parse it as a variable through the embed.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Mark Croxton: And I just found that it's kind of useful to help that to encapsulate your templates, your useful elements of your template and to calibrate that. That's certainly a good thing, but the fact is that you end up with lots and lots of nesting and the nested templates, the nested embeds inevitably leads to a slow website.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark Croxton: And I found that it solved a real problem on some of the websites I was building with quite large, quite a little traffic, and I found that the embeds made a big difference. The embeds make a big difference to the actual speed that they responded, so it was important. It was also the same, and you must be aware of my plugin, Switchee, as well?

Timestamp: 00:10:10

Emily Lewis: [Agrees]

Lea Alcantara: Yeah.

Mark Croxton: And I kind of came out with a similar problem as I saw those conditions where all of the different non-matching conditions would still be rendered by the template parser, and, of course, that adds overhead to a page, so Switchee is the solution to that problem with advanced conditionals. Again, it came out basically hitting rules with speed and responsiveness of ExpressionEngine websites.

Emily Lewis: Since the initial development for these add-ons like Stash and Switchee, they've evolved over time, how do you decide how to grow an add-on?

Mark Croxton: Quite often it's from suggestions made by people that are actually using it. It comes from the forums or Twitter. I wrote IfElse and Jacob Russell said, "Could we have it on Switchee for IfElse conditions?"

Emily Lewis: [Agrees]

Mark Croxton: And I thought that was a good idea, and so I just sort of created something together just because it was just kind of like a great idea. Beside the suggestions by people through the forums, it's from problems I encountered with client work I'm doing. One example I did is the parsing of better template codes later on in the parse order template, that came out as a couple of hits in particular with one of our clients where we needed to get some injected HTML, JavaScript and CSS into the head of a page and so I came up with this parse order subversion in Stash where you can



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

process the end of a template parser. Let's say, you've got a parameter process end, process in line, process start, and that just means that you can turn the parser around its head really and process your chunk of code a different time at the end of a template parser, which is incredibly handy thing. I just use it more and more now for my templates.

Emily Lewis: And I assume that that also improves performance as well.

Mark Croxtan: Yes, it can do. Certainly, if it involves, if it means that you don't need to have an embed.

Emily Lewis: [Agrees]

Mark Croxtan: If you create the Stash variable process at the end and use, let's say, the parameters as well, you can cache the rendered output of that chunk of code for a particular period of time, and that means that the next time that template is viewed, it won't actually be processed.

Lea Alcantara: So I'm curious, you mentioned, and we are still talking about embeds, and embeds seemed to be the sticking point in terms of performance, but when I'm working on my current Stash site and using Adrienne's examples, we still need to embed the wrapper template somehow.

Mark Croxtan: Yeah, yeah.

Lea Alcantara: Is there any way in the future where that can be avoided where we can pull the wrapper template without having to use the native embed?

Mark Croxtan: Well, one embed is not too bad in actual, in fact, but I wrote a new tag code in the Stash embed which functions in much the same way as an ExpressionEngine embed that gives you a lot more custom parameters and you can really kind of change the way that content of the templates parse. You can change the order in which Stash embeds are processed using priority parameter, and you can choose to include your Stash embed at the very start of its host templates before a parsing has happened. You can include it in line as a field is tagged where you can include at the end of template parsing which is always exactly the same an ExpressionEngine embed.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

In addition to that, you can demarket areas of the template with Stash low cache type heads, so those areas don't actually get cached and tremendous kind of dynamic elements so that you can have areas of the template that are rendered and areas which are dynamic and thereby we have a much more efficient template. You can use Stash embeds to replace conventional embeds in most situations. Provided that you take advantage of the caching, then they are going to be more efficient in ExpressionEngine embeds.

Lea Alcantara: And when you say caching, caching via Stash like the cache parameters in Stash, or do you mean caching in general with EE caching options?

Mark Croxton: No, with Stash, when you save the variable or read the file and saved into a variable which is when Stash encounters an embed.

Lea Alcantara: [Agrees]

Mark Croxton: What you're doing is you're saving a chunk of data into a new row in a database table.

Lea Alcantara: [Agrees]

Mark Croxton: And that data could be partially rendered content or a mix of dynamic content and HTML that's being rendered, so you can control how much of that data is actually cached over.

Lea Alcantara: [Agrees]

Mark Croxton: Subsequent views of template by Stash, that doesn't go back to the file. It reads from the database cache.

Lea Alcantara: Okay.

Mark Croxton: If there is a parameter you can use in your website's configuration file to make Stash to always check back the file and that's very useful when you're developing a website so that you can



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

make changes and see them immediately. But if you disable that when you go to production, then you'll find that your site will load very, very quickly.

Emily Lewis: You're talking about some features of Stash that maybe folks might not be as familiar with, given the types of projects they are working on. Are there any other approaches or even just individual parameters that you think are pretty powerful in Stash that maybe folks don't know about?

Mark Croxtan: Yeah. I mean, one of the useful parameters that I use is the match against filtering, and so there is a match and against parameter which allows you to specify a regular expression against which the variable is matched, and you can use that when you're either setting a variable or a list or getting a variable or a list. So we could, for example, store if you have a channel of people and you saved that as a Stash list and then you just wanted to output all of the people where there is custom field group with last name and you want to out all the people with their last name beginning with C, you could use match against to that when you're getting the list.

Emily Lewis: I really, really, really struggle with regular expressions.

Mark Croxtan: [Laughs]

Emily Lewis: Do you have any suggestions especially for maybe a more designer-minded person to get up to speed on using regular expressions?

Mark Croxtan: [Laughs] There are good resources on the internet. I think there is Regex info. It's worth checking out. There is quite a lot of introductions to regular expressions and it's sort of basic, the basics of them. They are not that hard. [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs] Well, I mean, I know Switcher will support using regular expressions which I think is one of the things that makes it quite powerful, yet I'm not quite able to leverage that power because I'm not able to really write an effective regular expression.

Mark Croxtan: They seem intimidating like any sort of programming.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Emily Lewis: [Agrees]

Mark Croxton: But I think there are some basic things you could know. For example, a caret (*) means to start a line and a dollar symbol (\$) means the end of a line. It's just acknowledges on its own is really useful. Backslash (\) really matches a digit. There are some basic stuff that you could read up on.

Emily Lewis: [Agrees]

Mark Croxton: And it will give you an awful lot more power to what you're doing with templates. It's definitely worth learning. They are not that hard.

Emily Lewis: [Laughs]

Mark Croxton: [Laughs] There are lots of online testing, regular expression testing vocations like Rubular, I think is one, where you can put your regular expression and see if it matches up a particular string you're trying to match, and that's extremely useful.

Emily Lewis: Oh, okay.

Mark Croxton: And there is also a little matter with applications, the Windows applications which do the same thing. But you very, very quickly get the hang of regular expressions. I'm no expert really on regular expressions, but... [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Mark Croxton: Yeah, right. They can be really fun, so give it a go.

Emily Lewis: [Laughs]

Mark Croxton: [Laughs]



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Lea Alcantara: [Laughs] Well, I'm really grateful that at least in Switcher, you give an example of probably one of the more common regular expressions which is pagination, like how do I detect a paginated string. Feature requests, more examples, please. [Laughs]

Emily Lewis: [Laughs]

Mark Croxtan: [Laughs] Well, it's not a bad idea. I think, yeah, you're probably right. I mean, one of the things that Stash and Switcher, both exactly, have really good documentations, and I'm not very good at that. I was really grateful to Adrienne for writing that article about Stash. It's one of the things I'm not really very good at, I'm afraid. I do my best.

Lea Alcantara: Is that why...

Mark Croxtan: I don't get much time.

Timestamp: 00:19:59

Lea Alcantara: So is that why you provide your add-ons for free?

Mark Croxtan: Yes, that's partly the reason.

Lea Alcantara: Well, what are the other reasons?

Mark Croxtan: I've got a lot of active open source software generally with CodeIgniter in particular.

Lea Alcantara: [Agrees]

Mark Croxtan: And I just feel like I have sort of debt to repay in that respect, to give back because I've just benefitted so much. So I think it's good to support open source software as far as possible to, at least, share if the software as open source. You mentioned that the support issue is another. It's quite hard to support add-ons on a commercial basis.

Lea Alcantara: [Agrees]

Mark Croxtan: With a free add-on in some respects, people have a pure expectations of you. I mean, it's a bit possibly a bit later. I don't have much time to answer questions.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croton>

Lea Alcantara: Sure.

Mark Croxton: I try as best I can on the support forums, and I think with something like Stash in particular, that it's really complex module.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Mark Croxton: And it takes a lot of investment in time from ExpressionEngine developers to get to know how it works. I think if you had to buy it, I think you could buy it. I think it would be a barrier to the adoption of this paradigm.

Lea Alcantara: [Agrees]

Emily Lewis: Well, I remember Adrienne mentioning on our last episode that one of the things she got or she found most useful with regard to Stash was just in the forums about how people are using it and their own questions, and so in a way the support that is there is still helpful.

Mark Croxton: Yeah. It's getting quite a lot actually with that forum. There is quite a lot of useful stuff in there.

Emily Lewis: [Agrees]

Mark Croxton: If you search through it, there is a very good chance that your questions had already been asked, because the same question is really normally about partials that come out again and again as well. One of the things that comes up again and again is this idea of trying to set a variable within another tag and then trying to get it outside of that tag, and people don't quite appreciate that module tags are parsed outwards inwards.

Emily Lewis: [Agrees]

Mark Croxton: Putting tags apart and not the other way around. But I think it's like an onion where you peel the layers off, so the template parser makes multiple partials of template. It peels off the



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

outer layer of tag to reveal the inner layer and then back and the layer would have been transformed by the outer layers with the parsing of the outer layer, and so it kind of progresses like that. It's peeling away layers one at a time until there are no more tags left. That's how it works. There is a guy, Low, has written some really good resources on parse order that I'd certainly like to encourage developers to read.

Lea Alcantara: [Agrees]

Mark Croxtan: There is a PDF of this and a slideshow.

Lea Alcantara: [Agrees]

Mark Croxtan: And it sort of goes into detail about how the ordering which things are parsed, and I think an understanding of that is extremely important if you're going to make the best use of my Stash so that you can appreciate why things happen the way they do.

Emily Lewis: And then perhaps in turn more effectively use Stash to sort of subvert that parse order.

Mark Croxtan: That's right. You sort of need to understand why. I think I do worry that some people are doing this blind and making use of it, possibly because it's a cool thing to do.

Lea Alcantara: [Laughs]

Mark Croxtan: But not quite appreciating why they are doing it, and I think that's a shame. I think if more developers learn about the parse order and why it can be useful to subvert it in certain times, then that would be good and it would mean I wouldn't get so many questions. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: So that's not really a pet peeve, it's more of something you'd love to see more developers do, but do you have a pet peeve that you see a lot of add-on developers are doing that they should maybe change or look out for, or something that there is a better approach to it?



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxton>

Mark Croxton: Well, I do, but I'm not sure many add-on developers will really agree with me.

Lea Alcantara: Sure.

Mark Croxton: [Laughs] But I quite often extend existing classes and I wrote Multiplaya, for example, which extended Playa, and I find that add-on developers have quite often make some of their internal classes private rather than protect it, the visibility of it is. It's quite hard to extend a class where some of the matters are private because you can't override them in the child class. It would be great if other add-on developers could use private visibility for classes which are absolutely internal, and if there is something which in particular is core the way the add-on works like in the case of Playa, it was the relationship method. It's the way that relationships were established.

Emily Lewis: [Agrees]

Mark Croxton: And making it protected would be brilliant to people like me who want to extend it.

Lea Alcantara: [Agrees]

Emily Lewis: So why...

Mark Croxton: But I'm not sure they'd agree with me. I'm not sure if add-on developers will agree with me.

Emily Lewis: Why wouldn't add-on developer agree with that? I don't do add-on development, so I'm not really sure what the pro or con of that approach is.

Mark Croxton: Well, they probably would agree because they don't want people messing about and creating sort of subsidiary add-ons which create many bits of functionality, which they then can't control and possibly have to get support the quest for why does this work and so on.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark Croxtan: And so I understand why they would object to it, but at the same time, I think that the benefits are so important that I much prefer that they use to protect it in private. And certainly not many developers will agree with me or not on that, I think. [Laughs]

Lea Alcantara: Do you think there is such a thing as a best practices for developing add-ons for ExpressionEngine? I saw on Twitter that DevDemon was talking about a PSR-2 coding standard, but then it conflicts with the syntax guide or whatever. Do you have a sort of process thinking this is the way you need to do it?

Mark Croxtan: Well, you should just follow a coding style, a style guide, either the general style guide and this comes with the user manual in CodeIgniter, or you could adopt the PSR-1 to coding style guides. I kind of prefer the syntax of the CodeIgniter and style guide to PSR-2, but I think the most important thing is actually consistency.

Lea Alcantara: [Agrees]

Mark Croxtan: So adopting one or the other is a good thing. There are other things you can do. Quite often I see in other people's add-ons is markup and channel scripts mixed up inside a model file or inside what you got itself, the parse itself.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Mark Croxtan: It should be separated out into view files. It's the same with any kind of database interactions, they should go into model files rather than being all mixed up together, so you've got proper separation of those areas. Also, making use of the actual built-in help functions and APIs and so we're reinventing which happens a few times.

Lea Alcantara: Why do you think that is? Is it because a lot of people aren't even aware that these APIs or hooks exist? Does that why? Do you think that the developer community needs to have some more communication or at least like, "Here is the standard hooks for everything here."



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark Croxtan: Well, I think probably it's just going to be awareness if that exist with these functions.

Lea Alcantara: [Agrees]

Mark Croxtan: And some new things have been added. So if you use to gather thing in the old way, then let's say, moving from ExpressionEngine 1 to 2, you're not necessarily going to be aware of the helper functions in Codelgniter, for example. So yeah, I mean, it's just awareness really. I completely understand how people can create add-ons. They've got a deadline and they just need to use some existing snippet code they've got already. If it works, then just use that, but obviously, I mean, if we are talking about best practice, then try and make use of a framework, the ExpressionEngine framework or Codelgniter framework to do the heavy duty stuff for you rather than reinventing.

Also, make your efforts for making methods your class methods if it's general and if it's possible so that individual classes extended, for example, by me, I can add new functionality to it without bidding to completely really like parts of it. Writing lots of comments as well so that if someone else reading your code can understand why you're trying to do something.

Emily Lewis: So do you have a process yourself when you create an add-on? Is there a preplanning step involved?

Mark Croxtan: Well, not always, but quite often. I am one of these. I'm a guy under pressure that needs to get something done.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Mark Croxtan: And I just get out my text editor and start coding, but I have built a few add-ons where I sit down and sketch out what I want, and particularly if it's got a database component, I try and model the database, the data model, if you like, and I've got an app call SQL Editor I used to do that. And so I'll put the tables together and see how they relate. And if it's got kind of control panel

component, the user interface, then I'll knock that out in Illustrator and flush it out in Photoshop and work out the sort of the markup and JavaScript for it before actually building anything.

Timestamp: 00:30:27

Now, programming the add-ons, it's a good idea to sort of just putting your right class method stubs in your class and just write out under each method what people are going to be doing in that method.

Lea Alcantara: [Agrees]

Mark Croxtan: And so that you'll sort of get an outline of all of your functionality that you need to build and that's incredibly useful about that approach and just writing comments first and then building the functionality. I mean, I tend to sort of write a manual and then it's like of sort of making addition on the method outright, the different types of conditions if it happens, I'd do that. So I sort of write out a kind of almost like a big to-do list in this class and then I'm going to build up from there.

Lea Alcantara: [Agrees] I think we are almost out of time.

Mark Croxtan: Oh.

Lea Alcantara: Before we wrap up, I have one final question. As an add-on developer, what's your favorite add-on that isn't yours?

Mark Croxtan: It's called Low Search.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Lea Alcantara: Interesting. Why?

Mark Croxtan: It's really, really useful for filtering entries in various ways, and it persists your choices.

Lea Alcantara: Oh.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxtan>

Mark Croxton: Let's say you've created a form where you've made a number of different choices, perhaps the category, the date range and so on. Low has coded it, so those choices are an remembered, persistent in the data, in the string that's actually sent to the results page. So all of that data is there for you to use. He's also got for in conditionals, for example, so you can repopulate a select menu or a multi-select menu. It's great. I've been using it a lot. I think it's a really great example of an add-on. It actually utilizes the channel entries module so it runs internally.

Lea Alcantara: Oh, interesting.

Mark Croxton: Yeah, so you've got all of the functionality of channel entries exactly the same thing that Playa does as well, and so it's building on top of that in terms of providing extra functionality, and the other great thing about it, of course, is that it has an index of all of your searchable entries and that searching that index is very fast compared to actually searching through the channel data table or the channel titles table and much, much faster than the built-in search module. I'm sold on Low Search. It's great.

Lea Alcantara: [Laughs] Emily, do you have any final questions?

Emily Lewis: Well, I was curious about one thing, has there been a user, a scenario that has surprised you the way someone used one of your add-ons?

Mark Croxton: Well, quite a lot of interesting approaches that come out of the forums and people have said, "Oh, I'm trying to do this," and I thought, "Oh, I find it actually interesting on the thought of actually doing it." One example is the sort of organizing entries into categories. I hadn't thought that you could do that through using Stash, but someone said, "Well, can you do this?" I had to think about it and I realized that actually you could. There's quite a lot of things from out of just suggestions on the forums. One was like the crumb trails, remembering the pages that you visited. Someone had built an entire sort of crumb trail based on Stash, which is great.

Emily Lewis: Oh...



<http://ee-podcast.com/episodes/add-on-development-with-mark-croxton>

Mark Croxton: But they needed a method to unset that crumb trail, but of course, Stash didn't have an unset, so I read on.

Lea Alcantara: [Laughs]

Mark Croxton: So it's great. I really enjoyed that feedback that I get for the add-ons and it helps to improve the add-ons quite a lot.

Lea Alcantara: Great. Well, thank you so much for joining us, Mark.

Mark Croxton: You're welcome. Great to talk to you.

Emily Lewis: It was great talking with you as well. In case our listeners want to follow up with you, where can they find you online?

Mark Croxton: You can find me on Twitter. I'm [@croxton](https://twitter.com/croxton) or you can find me on my website at hallmark-design.co.uk.

Emily Lewis: Great.

Lea Alcantara: [Music] Thanks, Mark. Now, we'd like to thank our sponsors for this podcast, [EE Garage](#) and [Pixel & Tonic](#).

Emily Lewis: We would also like to thank our partners, [EngineHosting](#) and [Devot:ee](#).

Lea Alcantara: Also, thanks to our listeners for tuning in. If you want to know more about the podcast, make sure you follow us on Twitter [@ee-podcast](https://twitter.com/ee-podcast) or visit our website, ee-podcast.com.

Emily Lewis: And don't forget to tune in to our next episode when we will be recording live at [EECI](#) in Austin, Texas.

Lea Alcantara: Ooooooh.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs] This is Lea Alcantara.



<http://ee-podcast.com/episodes/add-on-development-with-mark-croton>

Emily Lewis: And Emily Lewis.

Lea Alcantara: Signing off for the ExpressionEngine Podcast. See you next time.

Emily Lewis: Cheers.

Timestamp: *00:35:38*

[Music stops]