EE Podcast #78 Stash & Template Partials

[Music]

Lea Alcantara: You are listening to the ExpressionEngine Podcast Episode #78 with special guest, Adrienne Travis, who is going to share her template development process using the Stash add-on. I'm your host, Lea Alcantara, and I'm joined by my fab co-host, Emily Lewis.

 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 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] Hi Lea, how have you been?

Lea Alcantara: Good. Lots of things on the pipeline.

Emily Lewis: Oh, I know. We've both been pretty busy the past few weeks planning for our live podcast at the upcoming EECI.

Lea Alcantara: Yeah.

Emily Lewis: Now, that we know some of the details I was thinking maybe we could give our listeners a little heads up on what they could expect.

Lea Alcantara: Well, we want to end the conference with a bang, so we are going to have a micro tournament for our attendees. We'll pick three lucky audience members so you have to be there to be a part of it, and then you'll be part of our EE flavored quiz show, and yes, there will be prizes. Beyond that, we'll also have a few Q&A sessions with panelists and then we'll cap it off with a masters tournament where we'll get a few of our favorite EE personalities to find out who is the real EE Master.

Emily Lewis: [Laughs] I'm really looking forward to it. It should be a lot of fun.

Lea Alcantara: [Agrees]

Emily Lewis: In fact, I'm really looking forward to everything about EECI, seeing everyone and picking up useful stuff. I remember the last EECI I was at in 2010, I still use some of the workflows and processes I learned at the conference.

Lea Alcantara: That's really the biggest benefit of the conference, all the insights from the panels and the attendees.

Emily Lewis: [Agrees]

Lea Alcantara: It's truly invaluable. It's pretty much where I'm more or less rejigged how I approach creating EE sites, and I think EECI will do that again this year. Today's topic I think is pretty much on the same level.

Emily Lewis: Yeah, finding better ways to do our development. So let's get to it. Our guest today is Adrienne Travis who is an expert ExpressionEngine and PHP developer. She recently joined the Devot:ee team and previously worked for The Nerdery. Welcome to the podcast, Adrienne, and thank you for joining us.

Adrienne Travis: Thanks for having me.

Lea Alcantara: So before we talk about template partials, how is the new gig at Devot:ee, Adrienne?

Adrienne Travis: Oh, I love it. I love it. We are having a blast. I've got lots of plans for improving the site. I think that there is a lot of really cool stuff in the pipeline.

Lea Alcantara: Can you chat a little bit more about what's on the pipeline with Devot:ee?

Adrienne Travis: Well, one thing we are looking at, you know that I work with revamping the search right after I got there to speed it up and fix some issues we were having.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: We are also looking at improving the relevance of searches. It's kind of one the things that's on my to-do list.

Lea Alcantara: [Agrees]

Adrienne Travis: Google, they're kind of improving their search ranking algorithms. We are working on some ways to kind of do our own internal Devot:ee page rank kind of thing so that we can make sure that people are getting results that are returned, that are useful to them, because search is kind of a black art. It always has been. It always will be.

Lea Alcantara: [Agrees]

Adrienne Travis: There are some plans that we have to try and make that more relevant for people to make sure that when you look for an add-on, you get the add-on that's the most helpful to you, things like that.

Lea Alcantara: One thing that I noticed is whenever I do research for the podcast and looking for add-ons that are relevant, would you guys be thinking about ranking add-ons that have been recently updated because sometimes there are add-ons that say, "Oh, they are from EE 2, but they haven't been updated since 2010."

Adrienne Travis: Yeah, that's definitely one of the factors that I think is going to go into the sort of page rank, we'll call it, our little micro emulation of Google Page Rank. Yeah, I think that how recently it's been updated is definitely going to be a factor because, obviously, if it hasn't been updated in two years, it's probably less useful than an add-on that's been updated more recently, and the thing that I've got kind of in the back of my head is that you've always got people who criticize Google for playing with their ranking algorithm. I mean, a lot of those people are idiots, but there are people with legitimate concerns about Google has a lot of control over what you see and in what order you see it, and there are certainly ways to use that power for evil.

I don't want to be in a position where people are like, "Well, Devot:ee is giving a boost to certain people or blah, blah, blah." I mean, the goal is definitely to make sure that you can find what you're looking for, not to benefit certain developers or certain add-ons at the expense of others, you know?

Lea Alcantara: Sure.

Adrienne Travis: That's never the goal, but if you keep finding stuff that's not what you're looking for, that reflects badly on Devot:ee. It also reflects badly on the developers of the add-ons that you're finding because you're like, "Aargh, why can't I find what I'm looking for?"

Lea Alcantara: Sure.

Adrienne Travis: Sometimes that irritation transfer is so completely unrelated stuff. You know how that goes. Sometimes you're just irritated and you're looking at a page and the page is suddenly what you're irritated at. We want people to be able to find what they're looking for. I'm excited about some of the ways that we're looking at doing that.

The other thing that is in the pipeline that I'm working on some – EECI is coming up and so this is relevant – we are looking at doing some kind of internal re-architecting and making things more findable. We're working on better information architecture and things like that.

Lea Alcantara: [Agrees]

Adrienne Travis: And we are going to be actually drafting – well, not drafting – but we are going to be looking for volunteers at EECI to help us with some focus group tasks, some card sorting kinds of user experience kinds of things. We are not talking about a two-hour focus group, but we are talking about 15 minutes, and we are going to have swag to give away, not just for that, but we will be giving away swag for people who help us out with that as well as just probably giving away swag for lots of other reasons.

Lea Alcantara: Cool.

Emily Lewis: Are you guys going to be putting information out soon regarding how someone can sign up to participate in that?

Adrienne Travis: Yes, we will. As soon as we have that figured out ourselves, we will be releasing information about that. [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Adrienne Travis: And we really are interested in community feedback about that because we want to know, always want to know how we can be better, how we can better serve the community, what people would like to see, what people's pain points are, and my background is kind of odd. I'm a developer, but my degree is actually in social science.

Lea Alcantara: Oh, interesting.

Adrienne Travis: So the whole user experience thing is something that's definitely near and dear to my heart specifically.

Lea Alcantara: Before we dive directly into template partials, how about you tell us how you got started with EE and why you stuck with it?

Adrienne Travis: Well, I have been a customer of EllisLab since before there was EllisLab. I, in fact, have been a customer since it was just Rick Ellis working out of his home office making pMachine Pro.

Lea Alcantara: Wow.

Adrienne Travis: I was a pMachine Pro customer. When I log into the ExpressionEngine site and go and look at my purchases, you can still see that I purchased copies of pMachine Pro. [Laughs]

Lea Alcantara: [Laughs] Wow.

Adrienne Travis: So I've been a community member for a long time. I was using ExpressionEngine 1 actually before the public beta even. I was one of the early beta people, and I've stuck with it because I love it, because it's a great system. It's really versatile. I know it like the back of my hand at this point.

Lea Alcantara: [Agrees]

Adrienne Travis: So I mean, I'm still always finding new ways and better ways to do things, but I'm comfortable in it and I do think that it's a great product.

Emily Lewis: I was just going to say I think that's why a lot of people, once you get familiar with EE, it becomes just this flexible tool that lets you do just about anything you can imagine, and I think once you get into it, it's hard to walk away from it and go and buy something else that doesn't offer the same kind of flexibility.

Adrienne Travis: Oh yes.

Lea Alcantara: Yeah, and in particular, the flexibility, and I'm sure everyone can agree to this is the templating engine of ExpressionEngine, which is sort of why we are here because I think for several years everyone has been doing a certain approach with using templates, and it's been working and it's been great in terms of flexibility and things like that. But now that we've matured as developers and as a community, now we're really thinking beyond just the regular flexibility of design into figuring out performance and maintenance and those types of things. So Adrienne, how did you start looking into template partials or Stash?

Adrienne Travis: Well, John Wells wrote an article a couple of years back about doing template partials with the homegrown plugin, and I had read that years ago. There were a couple of projects that I was working on, both at The Nerdery and freelancing side projects, where the templates got hairy, you know? You know how it goes.

Lea Alcantara: Okay.

Adrienne Travis: As a site grows organically, especially if it's a big site, you kind of end up with templates all over the place. You end up with a lot of duplication of stuff because you can't deal with the idea of having one more conditional in your templates so you pull something else out into it. You pull something out into a different template, but it's 90% the same.

00:10:09

Lea Alcantara: Yeah, yeah.

Adrienne Travis: So you're violating the principle of not repeating yourself there, and you're also making it so that it's just a headache.

Lea Alcantara: Yeah.

Adrienne Travis: So I was just very frustrated kind of one day and looking for a new approach, and I have been using Stash some anyway for some of the other things that Stash is good for. So you can use it to subvert the parse order. You can use it to pull stuff out into other places, and I was reading one of Croxton's support threads on the Devot:ee forums where he was talking about his approach using Stash for template partials. Obviously, he's the author of Stash, so he knows it like the back of his hand.

Lea Alcantara: [Agrees]

Adrienne Travis: And I was thinking back to John Wells' article about template partials, and fortunately at The Nerdery right at that time when I was getting really frustrated, we were just starting up a new project that I was the lead developer on and it was a fairly small project, and so my thinking went, "You know, if it winds up that something is horribly broken with this approach, the site is small enough that I can spend 20 hours over the weekend just redoing all the templates.

So it was kind of a good pilot project for it because it was a real world project, but it was small and it was something that I didn't feel like there was a lot of risk in trying a new approach. So we tried the new approach, and I reread John Wells' article and I read a bunch of stuff that Croxton, like I said, has written in the forum where he supports Stash and Devot:ee. That particular forum has a lot of great information. Stash is so complicated, and so I go to that forum a lot and search for questions and see whether Croxton has answered them.

I do that with other add-ons too. I think that the Devot:ee support forums are a really good resource in general. I'm not just saying that because I work there either.

Lea Alcantara: Yeah. [Laughs]

Emily Lewis: [Laughs]

Adrienne Travis: And Croxton, in particular, tends to go into a lot of detail about, "Here is why this isn't working. Here is what you can do to make it work, or here is a totally different approach that you haven't thought of." I just kind of started experimenting and very quickly realized, "Holy crap, this is the best thing since sliced bread for EE templating."

Emily Lewis: Can you give me like a high level definition of how you would describe template partials to someone who is looking to change their way of building templates?

Adrienne Travis: Well, the idea of template partials is that instead of doing your templates, instead of thinking about your templates as a linear document where you sort of have one chunk of code and another chunk of code and another chunk of code, you think of your templates as a kind of a set of nesting baskets into which things go.

Lea Alcantara: [Agrees]

Adrienne Travis: So it's kind of like those neat stacking containers at the store where you kind of fit several different containers inside a bigger container inside a bigger container in different configurations. It's kind of…

Emily Lewis: Oh, right, like those Russian nesting dolls.

Adrienne Travis: Yeah, like a cross between Russian nesting dolls and desk organizer, yeah.

Lea Alcantara: [Laughs]

Adrienne Travis: And I talk about it as feeling kind of inside out from the more traditional way of building templates. Normally, what people do is they kind of have an outside template and then they use embeds and snippets to do pieces.

Lea Alcantara: Right.

Adrienne Travis: Whereas with template partials, what you do is you actually use the embed as your outer container and then your actual templates are the pieces. So it's kind of you have to flip your brain around and start thinking inside out a little.

Emily Lewis: You said you started this approach with that small project because you felt it would be a good fit. If it didn't work, you could undo it.

Adrienne Travis: Right.

Emily Lewis: What was the learning curve? Like would you say it added hours to your development for that first go around?

Adrienne Travis: I don't think it actually did, and some of that was due to the fact that I had some really sharp people and some of it actually was due to the fact that one of the people I had on the project had never actually developed using EE before, so he didn't have anything to unlearn.

Lea Alcantara: [Agrees]

Emily Lewis: Oh, nice.

Adrienne Travis: So that was kind of handy. The folks that I had on the project that did have some EE knowledge, there was I think a little bit of a learning curve for all of us, and I'm sure that there are things that if I went back and looked at that project, I would do differently even. It's only three months later, but I'm sure I'd do things differently, but I don't think it was that bad. I really don't, and I feel like it cut down on some of the fiddly conditional logic and I'm sure you guys know that conditional logic ends up sometimes being a huge pain to debug.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: Because, of course, you've got 12 different things trying to all happen in the same document and you've got nested If statements and you're like, "Oh God, when is this If running and when is this other thing running?"

Lea Alcantara: Yeah, which goes first, yeah.

Adrienne Travis: Right, and some of Croxton's other add-ons, we use Switchee and we use If:Else extensively.

Lea Alcantara: [Agrees]

Adrienne Travis: And they are both hugely helpful. Even then, you end up with really complicated things where you've got totally different blocks happening, and so I think that the template partials design pattern actually cut out enough of that fiddly debugging and maintenance time that even with there being a learning curve, it didn't take longer than it would have otherwise.

Emily Lewis: Does it require more planning in advance doing the template partials approach?

Adrienne Travis: That's a good question. I don't think that it really does because the things that most people are doing, I had to run a footer kind of thing anyway like snippets or an embed, and you can kind of leverage that idea toward to figuring out what you're doing. I also think that there are ways in which it's actually easier to rethink things if you're using partials because there are fewer places that you have to change things most likely.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: You don't have a bunch of container templates that are then having different pieces. You've got one or two container templates. So if suddenly all of the markup for your site has changed, you probably don't have to change 30 templates anymore, you have to change maybe two.

Lea Alcantara: So how do you plan for this?

Adrienne Travis: For template partials in general?

Lea Alcantara: Yes.

Adrienne Travis: What I do is, the very first is I take the wireframes or the comps or whatever I've got and this is an actual visual process for me.

Lea Alcantara: Cool.

Adrienne Travis: So I want to actually say that when I say to look at the wireframes or comps, your scope document is going to be important in the process or your features requirements document or whatever. But for this step of actually doing container templates, I would argue that really, really you don't want the scope document because you don't care about functionality. What you want is the comps or the prototype, the HTML prototype, or whatever you've got that's the visual piece.

Because what I do is I kind of squint my eyes and look at all the pages side by side, because what you want to do is to figure out what you want your container templates to be, you want to figure out what your different kinds of visual and markup-wise your pages are. So if you've got a page with a left side bar and then a page with a right side bar and then a home page that's got completely different markup, then you've probably got three container templates, depending on how different the markup is between the left side bar page and the right side bar page. My rule or my guideline is that if the markup is really different, you want a different container.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: If the markup is pretty close to the same such that maybe you would need one If statement to differentiate between them, then they are the same container, but so that's why I say to take an actual look at the visual markups of whatever kinds you've got and squint your eyes and figure out how many different containers you need that way.

Emily Lewis: You really need to know your content. I always am emphasizing to clients how important it is to know what the content is on the different pages of a site.

Adrienne Travis: Yeah.

Emily Lewis: And this just reiterates it because like you've said if you have a right side bar and a left side bar and they are almost the same except for one thing, that's a whole different scenario than two completely different side bars.

Adrienne Travis: Right, yes, very much so. Now, obviously, at a certain point it, you only have to go so many levels deep into the markup in this process.

Emily Lewis: [Agrees]

Adrienne Travis: If you've got a left side bar and a right side bar and the only difference is that they've got class left and class right, then obviously that's fine.

Emily Lewis: [Agrees]

Adrienne Travis: Even if they've got completely different content and markup inside of those boxes, that probably doesn't matter because you can handle that in your actual partials.

Emily Lewis: Right.

Lea Alcantara: What's interesting about what we were discussing here in terms of looking at the HTML markup and seeing how it's different visually, and I'm looking at your EE Insider article, it's not as strict DRY approach.

Adrienne Travis: No.

Lea Alcantara: Yeah, because when we had John Rogerson and he was trying to explain his DRY approach with Stash, he actually separated the HTML completely from the partial like actual ExpressionEngine loops.

Adrienne Travis: Which you can certainly do.

Lea Alcantara: Sure.

Adrienne Travis: And there are advantages and disadvantages to doing that.

Lea Alcantara: Can you explain what those are?

00:19:57

Adrienne Travis: Well, one of the disadvantages is that it can be less readable. Especially if you've got people who are more front end than back end people, they are not necessarily as clear on the what is going on here.

Lea Alcantara: [Agrees]

Adrienne Travis: I'm a big believer in don't repeat yourself, but I'm also a big believer in making sure that things are readable and maintainable.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: And so I'm always kind of walking a middle way, I think. You can also see on my EE Insider article that I have the head and body markup. It's the first thing I stick in my wrapper templates, and the reason for that is, I'm sure you guys have run into sites where like you've got an embed for part of the head and that embed has the closing head tag in it.

Emily Lewis: [Agrees]

Adrienne Travis: And then maybe the opening body tag.

Lea Alcantara: [Agrees]

Adrienne Travis: And so then you embed that and then you've got a different snippet somewhere that's got your closing body tag, and so you never actually know where those are. So what if somebody creates another embed and accidentally puts another closing body tag in it? Suddenly, you've got invalid HTML.

Lea Alcantara: [Agrees]

Adrienne Travis: So I kind of like to have the high level markup where I can see it.

Lea Alcantara: [Agrees]

Emily Lewis: At the very core structure of your markup.

Adrienne Travis: Right, and yes, it's not strictly speaking DRY, because you're probably at the very least are duplicating it two or three times for your containers.

Lea Alcantara: [Agrees]

Adrienne Travis: But you are only duplicating it two or three times, and again, I'm trying to strike a balance between I don't want to have to do a global find and replace versus I want people to be able to actually look at a template and see instantly what is going on.

Emily Lewis: I like that scenario that you're describing because I think it demonstrates that this approach isn't restrictive. Really, like if you are working with many people on a team and readability is a high priority, I think you make an excellent point in the fact that you can go that route or you could go even more DRY, don't repeat yourself. It just demonstrates this is a pretty flexible approach in and of itself.

Adrienne Travis: Yes, it absolutely is. Stash is such a game changer. I mean, and I think people say that lots of things are game changers, and I try to not use that idea lightly.

Lea Alcantara: [Agrees]

Adrienne Travis: But I think Stash really is and he keeps improving it and keeps making more useful things that you can do with it.

Emily Lewis: Well, let's talk about, so you've sort of described your process for getting the template partials set up and your process for sort of planning it, but where does Stash come in to the picture?

Adrienne Travis: Well, what Stash lets you do is it basically lets you pile up a chunk of stuff, whether that's from a channel entries loop or plugin tags or whatever, and then pull that out somewhere else. So it's kind of like on the fly global variables, except it's way more versatile than any ExpressionEngine's default ways of creating those on the fly blocks.

The thing about Stash is it lets you do things like I have three different things that need to happen on this page and because of the way they are happening, I would normally need to run three different channel entries loop, even though it's all on the same channel, but because of the order things need to go in, because I've got totally different blocks. One loop is in the content and one loop is in the side bar, for example, whatever. With Stash, you can run one loop and pull stuff out into separate Stash variables and then put those wherever you need them. The main thing that Stash does is it kind of frees you from having to think about where your loops are.

Lea Alcantara: [Agrees]

Adrienne Travis: You can kind of stick your loops wherever, I mean, not quite wherever, but you stick your loops wherever it makes the most sense to put them and then you can have the content go wherever it makes the most sense to put the content, and those two don't have to be connected anymore.

Lea Alcantara: It sounds like if there is less loops, does that necessarily equate to less queries and a faster page load?

Adrienne Travis: Absolutely, absolutely. Every time you are running a channel entries loop, you are running a whole lot of queries, and the same is true of any of the many, many add-ons that extend the channel entries loop with things like Playa, with things like…

Emily Lewis: Matrix, Category.

Adrienne Travis: Well, Matrix doesn't extend the channel entries loop because Matrix is a field-type, but for the Playa tags it extends the channel entries loop. Low Search extends the channel entries loop. All the search add-ons do. Any ObjectiveHTML, wonderful Google Maps add-on, that extends the channel entries loop, and you're running a lot of queries anytime you are running one of those loops. So absolutely, running fewer loops and just pulling the data out because Stash basically only has to do one query or two queries anytime it pulls out a chunk of data.

Lea Alcantara: Why don't we talk about whether or not Stash is even needed? Is Stash the only approach to templating in this fashion? Have you ever tried doing something like this before Stash came out?

Adrienne Travis: Well, as I mentioned way earlier, John Wells wrote an article years ago about doing this with the homegrown plugin.

Lea Alcantara: [Agrees]

Adrienne Travis: So yeah, you absolutely do it without Stash. It takes advantage of the ExpressionEngine parse order ultimately. The critical thing is that you can store some stuff and pull it out later because embeds run last basically in the parse order. The fact that embeds get processed last is ultimately the trick that makes template partials work. You can do it without Stash. I frankly don't why you'd want to because Stash is such a wonderful Swiss Army knife. It's definitely one of the plugins that just gets installed everywhere now, or modules rather, they get installed everywhere now. I just do a site without it installed.

Emily Lewis: Is there ever a situation where like Stash is overkill or maybe it's limited with another add-on? Have you had any gotchas with using it?

Adrienne Travis: I have not. Actually, there was a bug. I actually mentioned it in my EE Insider article. The one thing we found was that there was a bug with the count tags. So if you were relying on count, then in earlier versions of Stash, there was a problem, but he has absolutely fixed that. I don't think that there would be any situations where Stash was overkill because it's pretty much as lightweight as you want it to be.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Adrienne Travis: If you have to wrap stuff up in a Stash tag, unless you're using a homegrown plugin to pass it to the embed, but if all you were doing, the least invasive case of the template partial scenario is you do a template mostly the way you would have done it before and then you wrap the whole thing in a Stash tag and then pass it to your wrapper embed. So that's one tag, and if that's how you want to do it, that will work just fine. So I don't think there is any situations really where Stash is overkill because you can use as much or as little of it as you want, and it's pretty low overhead.

Emily Lewis: [Agrees]

Lea Alcantara: Speaking of that though, I'm curious. I know you just said that you don't think it's overkill because you could do whatever. But I was reading Twitter and you mentioned ObjectiveHTML, Justin's Google Maps add-on, he's been experimenting with Stash a lot too and he's trying to actually completely eliminate the embed tag, which you said is pretty much one of the main things that you definitely need for your container templates, and he's trying to eliminate that, because we all know that the embed tag does carry a bit of an overhead as well. What do you think about that approach? Do you think it's necessary? Have you tried using Stash embeds in place of ExpressionEngine embeds?

Adrienne Travis: I have played with Stash embeds some, yes, and I do think that you can probably. Stash embeds, mind you, are a new feature. They were not available at the time I wrote the EE Insider article.

Lea Alcantara: Sure.

Adrienne Travis: Because that's the newest beta of Stash have that. The thing that I don't know and I have to admit that I don't know is you can absolutely change most of your embeds to Stash embeds if you even needed embeds after switching to the template partials approach. The thing that I don't know is whether you can eliminate that one wrapper embed basically that actually folds that whole thing up into a template partial plus container.

Lea Alcantara: [Agrees]

Adrienne Travis: I don't know whether you have to leave that one as an actual ExpressionEngine embed. I would think that you might because of the way that the whole parse order thing is happening, but it maybe that Croxton has got a way around that. He's really sharp certainly, and I would not put it past him. I have played with replacing some of my internal embeds with Stash embeds because while template partials definitely cuts down on the number of embeds you need, every once in a while you still run into something where you've just got a huge parse order headache and you decide to use an embed. So those were the sorts of things where, yeah, I have been playing with Stash embeds for that kind of thing. I do plan to write a follow-up tutorial on Stash. I just have not had a chance. It's been kind of a busy summer.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Adrienne Travis: And so with the new job, you know?

Lea Alcantara: Sure.

Emily Lewis: You had also mentioned in your article just a little gotcha with Stash for particularly new folks about making sure that it's not nested within a channel entries tag, but it's actually the outermost tag.

00:29:56

Adrienne Travis: Well, you can actually use it within the channel entries tag. It kind of depends on what you're doing, and occasionally, you'll run into issues with parse order, although in the newer betas, Croxton has got stuff where you can basically tell it how far inward to parse tags, and that kind of eliminates the problem I was having with putting it inside the channel entries loop in certain circumstances because it was not parsing all the tags properly, like the inside variables inside the loop, and he's had the parse variables feature in there for some time, but for a while I was having real trouble getting it to work consistently and some of that was probably user error, but it seems a lot friendlier in the more recent version. So yeah, you can use it inside or outside the loop I would say at this point.

Lea Alcantara: Before we wrap up this episode, is there anything else you'd like to tell our listeners about Stash or the template partials approach?

Adrienne Travis: I think that the thing that I would like to say is that it really, really is worth playing with, that the benefits in maintainability and in performance are really hard to overstate because they are really, really drastic. Once you get the hang of thinking in this way, you find that you can do things that before you were like, "I'm going to have to write a custom add-on to do this because I can't get this data from this one plugin into this other module in the right order.

Lea Alcantara: [Agrees]

Adrienne Travis: I have done some really unholy things with like Playa tags and Stash, and they just worked on the first or second try. I mean, it's such a huge eye opener once you actually get the hang of it, so I would really encourage people to find at least a small project and play with it if they haven't already because I think that you'll be pretty instantly converted.

Emily Lewis: Is this even an approach that you would recommend, if the budget and time was available, to retrofit an existing project with this new approach?

Adrienne Travis: I would actually because, again, I think that you're going to reap a lot of benefits in maintainability and potentially if you've got a really embed-heavy site, you're going to reap a lot of benefits in performance as well.

Emily Lewis: [Agrees]

Adrienne Travis: I mean, an embed calls an entire another instance of the ExpressionEngine parser. It's a huge performance hit. If you've got an embed inside a loop, you basically just doomed your site to run hundreds of queries on every page load potentially.

Lea Alcantara: [Laughs]

Adrienne Travis: So yeah, especially for bigger sites and data-heavy sites and sites that are using a lot of embeds, I would say that it is absolutely worth looking at trying to get the time and budget to retrofit, and you can start small. Again, it's not like you have to make a giant total commitment. Make one container, convert a few templates.

Emily Lewis: [Agrees]

Adrienne Travis: See and do some benchmarking. See if people like it better, wrap your head around it in small ways before you go to the complex stuff. I don't think it's an all or nothing kind of thing.

Emily Lewis: Excellent. I have one quick question. This is totally unrelated to Stash or template partials, but I'm curious when I'm looking at your examples in the EE Insider article, is your naming convention when you do use snippets or global variables, Snippet:Name?

Adrienne Travis: Yes, that's actually if you are using the Snippet Sync plugin by Brian Litzinger and Isaac Raway. That's kind of the default convention for snippets and global variables in that add-on.

Emily Lewis: Okay, and you just carried that over if you make your own?

Adrienne Travis: I try, yes, because it's the really easy way to figure out what is what.

Emily Lewis: Yeah.

Adrienne Travis: And underscores are so overused, I feel like that just doing snippet_underscore or whatever, you just got yet another underscore there.

Lea Alcantara: [Agrees]

Emily Lewis: I agree. Just scanning your examples, I immediately was like, "Oh, I'm going to start doing that because it's much more, like you said, readable. It's something that's really important for some people.

Adrienne Travis: Yeah, readability is really vital, especially on bigger projects.

Emily Lewis: [Agrees]

Adrienne Travis: I mean, because your eyes can really easily glaze over and miss stuff when you're into knee deep into your 30th template of the day.

Lea Alcantara: [Laughs]

Adrienne Travis: So I place a lot of importance on readability. I place a lot of importance on maintainability too, obviously, which is why I say there is always kind of a fuzzy place in the middle where you've got to try and find what's right for your project, but readability is really vital, especially on big projects.

Lea Alcantara: Well, thank you so much for joining us today, Adrienne. That's all the time we have.

Emily Lewis: In case our listeners want to follow up with you, where can they find you online?

Adrienne Travis: The easiest place to find me online is on Twitter. I am @adrienneleigh on Twitter, and I spend a lot of time, obviously, in the #eecms. I don't currently have a blog because, well, the cobbler's children have no shoes. That's why I got no blog.

Lea Alcantara: Yeah, yeah.

Adrienne Travis: Because there is always so many side projects and so much work to be done and I'm also the world's worst clients.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Adrienne Travis: So I would like to get my website back up and around, but currently it is defunct, I'm afraid. So Twitter is really the best place to find me, and I'm always happy to give people my IM information if they message me on Twitter, and you can ask me whatever. I'm pretty friendly.

Lea Alcantara: Cool.

Emily Lewis: Great.

Lea Alcantara: [Music] 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 @eepodcast or visit our website, EE-Podcast.com.

Emily Lewis: And don't forget to tune in to our next episode. We will have Mark Croxton on to talk about add-on development, and I'm sure we'll have lots of questions for him about Stash.

Lea Alcantara: This is Lea Alcantara.

Emily Lewis: And Emily Lewis.

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

Emily Lewis: Cheers.

00:36:10 [Music stops]