



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

EE Podcast #68 DRY Techniques with EE

[Music]

Lea Alcantara: You are listening to the ExpressionEngine Podcast Episode #68 with special guest, [John Rogerson](#), joining us to talk about DRY techniques in EE. I'm your host, Lea Alcantara, and I'm joined by my fab co-host, Emily Lewis.

This episode is sponsored by [Engine Summit](#). Time is running out! The online live Engine Summit **coming up at the end of the month** covers the ins and outs of ExpressionEngine without the travel hassle of a traditional conference. It's like bringing the experts to your desktop. At the end of the Engine Summit, it's the popular EllisLab roundtable discussion hosted by CEO of EllisLab, Leslie Camacho, talking about the state of EE and answering your questions. Sign up today and save 20% when using the discount code "EEPODCAST" at [enginesummit.com](#).

Emily Lewis: The ExpressionEngine Podcast would also like to thank [Pixel & Tonic](#) for being our major sponsor of the year. [Music ends] Hey Lea, I cannot believe it's been two weeks since our last podcast. Time has flown by for me. How have you been?

Lea Alcantara: Insanely busy. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: I just finished the client's site launch so it's been go, go, go the past week. But I've been pretty pleased with the EE implementation almost more than the design.

Emily Lewis: Nice.

Lea Alcantara: Yeah, so that's why I'm excited about our guest today because I'm always looking to see how to improve the next EE site that I build.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: Yeah, me too. So let's get into it and introduce our guest then! We have [John Rogerson](#). He's an EE developer who specializes in higher education. He currently works for [Sewanee University](#). He has joined us on the podcast before talking about his experiences with multi-site manager for the Sewanee sites, but today he's going to talk to us about the use of DRY principle in relation to EE and web development. Welcome John!

John Rogerson: Hi Emily. Thank you for that.

Emily Lewis: You're welcome. It's a pleasure to have you on the podcast today.

John Rogerson: Thanks, it's great to be on again. I'm really honored that you let me back on.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: So before we get into the nitty gritty of the DRY principle, why don't we define what DRY is? What is the DRY principle?

John Rogerson: Well, DRY is an acronym for Don't Repeat Yourself.

Lea Alcantara: Okay.

John Rogerson: And so obviously with ExpressionEngine, we would spell repeat with two Es.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

John Rogerson: [Laughs] And really the idea is that, and I guess we've all been in this situation where you are developing a site and you start doing the same thing again and again and again in terms of typing, and obviously, the opposite of DRY is WET.

Lea Alcantara: Yeah.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Which is We Enjoy Typing.

Emily Lewis: [Laughs]

Lea Alcantara: Oh nice, nice.

John Rogerson: So I guess if you enjoy typing, DRY isn't for you.

Emily Lewis: [Laughs]

John Rogerson: But on that, it's obviously a very basic level. The idea is to optimize your work flow when you are developing a site.

Lea Alcantara: [Agrees]

John Rogerson: And I think with ExpressionEngine, there are opportunities to really go down the DRY road if you like, but they are not obvious and they are not kind of native.

Lea Alcantara: Yeah.

John Rogerson: Although you can still do a lot of the principles natively, but I think there has been a couple of really good articles and tutorials and a couple of add-ons that are available now that really help you implement a DRY kind of process in your development.

Emily Lewis: So on a high level, it's sounds like it's just a way to optimize and be efficient in how you work.

John Rogerson: Yeah, I think efficiency is the best way to really look at this.

Emily Lewis: In some ways when I was reading some of the information you provided us before today, some of the things seem similar to me to object-oriented programming. Are they related? Are they in the same family? Or is it just that they are both sort of embrace a more efficient way of writing code?

John Rogerson: Well, I think that with these things, I mean, I think they are so much kind of political stuff behind everything and so I wouldn't like to say that DRY is object-oriented programming or vice versa. I mean,



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

to me they are just different approaches that might have similarities between them, but really it's what works for you.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: It's on a case-by-case basis. I think that's the best way to approach any kind of like strategy for developing a website. To me, I mean, the main reason why I would want to use a DRY process is really just to make things easier, especially not just in development but obviously on an ongoing maintenance basis as well.

Lea Alcantara: [Agrees] And I think one of the unique challenges with developing for ExpressionEngine is the ability that you can develop the site any way you want, right?

John Rogerson: To start it, yeah, and that I think is a great point. I mean, I guess with some other CMSs, they probably wouldn't give you the flexibility to do something like this. They would be a little more rigid, so that's a really good point there.

Lea Alcantara: Yeah. I think there was a recent article somewhere talking about it. I think it was on A List Apart about making your content, your CMS and your coding like content agnostic so you need to create different silos for different things, and I think ExpressionEngine gives you that flexibility to arrange all your little code snippets and content and information into different things. I think at the very beginning of like the ExpressionEngine life cycle when it first got released because you can do anything and everything with EE. Everyone kind of just did their own thing, but I think as the community has matured, more and more people are adopting this type of DRY technique. Have you noticed that yourself?

John Rogerson: I have, yeah. I mean, I think even in our pre-discussions via email, we discovered that there isn't one way to do this.

Lea Alcantara: Yeah.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: That there are different ways to achieve the similar kind of goal.

Lea Alcantara: Yeah.

John Rogerson: Which is not to be doing the same thing again and again and again in terms of your template development.

Lea Alcantara: Sure.

John Rogerson: I mean, I think one of the differences maybe between EE and something like Django is that they kind of have different parts of their system that do these different things whereas with ExpressionEngine you have the templates, and really that's where your action is and there isn't like another area where you deal with the logic of your channel entries and whatnot, but you can kind of get there by using different templates for different purposes. I think that's where the template partials concept comes in.

Lea Alcantara: [Agrees]

John Rogerson: The first I knew of that was via an article by a really, really great plugin author and blogger called John D Wells.

Lea Alcantara: [Agrees]

John Rogerson: And actually, when he wrote his article on the template partials concept, it was actually under the guise of you develop your own plugin to do this. Part of his article is actually developing your own plugin. Shall we talk about the partials concept briefly?

Lea Alcantara: Sure. Sure, why don't you tell us what template partials actually mean. What is that?

John Rogerson: Sure, I mean, the partials concept is kind of addressing the sort of DRY idea, but also to avoid using multiple embeds.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: I mean, I know when I first started developing ExpressionEngine, embeds were really very popular way to reuse. When you have the parts of each template that are being used like your head and your footer and all that kind of stuff which appears in every page, we would use an embed to bring that in, and then I guess embeds have kind of gone of out popular just a little bit because it's now acknowledged that each time you call an embed to your template you have to begin the whole kind of life cycle of the engine in Expression.

Lea Alcantara: [Agrees]

John Rogerson: With this partials concept, what Mr. Wells was actually talking about was having one embed per template and that embedded template would be your actual kind of HTML area and that would potentially you might only need one kind of view if you like, which would be getting into another area, but let's just call it the View template.

Lea Alcantara: [Agrees]

John Rogerson: And that would be where you have all your HTML.

Lea Alcantara: Okay.

John Rogerson: And then on the template that calls that embedded template, you would actually list out your actual logic which is basically your channel entries and any other kind of ExpressionEngine tags.

Lea Alcantara: [Agrees]

Emily Lewis: But would that also include like snippet tags or global variable tags?

John Rogerson: Absolutely, yeah. I mean, this isn't at the expense of using any of those types of things, and what you find in those is that if you want to go really kind of like to the pure dry approach, you would actually want to make sure that in your template where you are doing all the logic, you have absolutely no markup.

Lea Alcantara: Interesting.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: It's purely just ExpressionEngine tags. Now, natively, you can't really do that because in the embed you have to actually kind of import data from one template to another.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: And natively, that's not really possible. Now, somebody maybe a lot smarter than me and they actually know how to do that natively, but I don't think they really is, and one of the points of the article was to show you have plugins in which basically we transfer data that you are generating in your logic and then there would be like a call to that storage of that data in your View template.

Lea Alcantara: [Agrees]

John Rogerson: And I guess, to the point of all these is actually where I see a lot of similarities between just this basic HTML and CSS differentiation that I think we all now acknowledge that it's not really a good idea to mix your presentation with your HTML.

Emily Lewis: [Agrees]

John Rogerson: Like you don't want to use font tags on all the rest of it like you did back in the day and know you can have one CSS file that controls all the presentation on your site and really it's the same principle. To me, I mean, that's my interpretation over there, and I think that's a good way to go because if you want to change your data, you don't have to change multiple templates, you know?

Lea Alcantara: [Agrees]

John Rogerson: And again, if you want to change your design and really what we are talking about is, it's more of your markup rather than your design because that would be probably in your CSS, but you only have to edit the templates which are delivering your different views. So again, on an average site, usually the home page is different from your interior pages.

Lea Alcantara: Sure.

John Rogerson: And really might mean you only need those two view templates and I usually myself, I'd usually make those hidden because they need to be seen on the site. I mean, those embedded templates are hidden.

Lea Alcantara: [Agrees]

Emily Lewis: So you typically, if you are following this approach, you may only have two templates?

John Rogerson: Yeah.

Emily Lewis: One for home and one for the main that other page views?

John Rogerson: In terms of your views, yeah, and then your other templates would be how many different ways you want to call or query your data, or display and generate your content would be the better way to put it, I guess. So you would have a template that would be your home page template where I guess you might show your three most recent entries, right?

Lea Alcantara: Yeah.

John Rogerson: So that would be your content template and that was called the home page template, and then I guess your interior template would probably just be calling whatever piece of content you are wanting to see individually and you would be calling your different view because that probably wouldn't be the same view that you would want to have on your home page.

Lea Alcantara: [Agrees]

John Rogerson: I think that would be the answer to that here.

Emily Lewis: This, it sounds simple, but it also sounds a little, not confusing, but that it would definitely take a little bit of time to get used to this way of thinking about your templates. So my question is, it's a little bit two part. What made you start striving for this? Was it this article by John Wells, or was it that you do the situation with work or a project and you've needed to create something that was more efficient, and then how long did it sort of take you to get the mind shift here?



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Well, that's a really a great question. I mean, I think really because I'm really into the idea of separating out the different parts of your site in terms of development, this just seemed really natural to me.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: And I only really develop in ExpressionEngine, but I appreciate the concepts which are discussed in other CMSs and frameworks, and a lot of those like I think especially Rails and Django are very much about this is like modern web development.

Lea Alcantara: [Agrees]

John Rogerson: And I'm not interested in the whole kind of war between PHP and other program and languages.

Lea Alcantara: Languages.

John Rogerson: But their criticism is usually about PHP is that you litter your templates with the actual programming codes. Your views, for example, they just think that's inefficient.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: But I think, I guess, there is an argument to that, yeah, is it inefficient? In terms of the learning curve, I don't find the learning curve difficult in terms of conceptually understanding the benefit. I don't think it's going to be any surprise that the real problems that come out of this is the parse order.

Lea Alcantara: Yeah, yeah. I was about to say that in order to do this type of technique properly and know which things need to be in templates, which things need to be in snippets, which things need to be in wraparound, let's say, Stash or something like that, you have to understand what order do you want things to load and what variables need to load first before something else.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Yeah, I mean, that can be tearing your hair out kind of stuff.

Lea Alcantara: [Laughs]

John Rogerson: It's funny in the parse order, you actually use it to your advantage in some ways with the concept of calling one embed.

Lea Alcantara: [Agrees]

John Rogerson: Because you can put that anywhere in that logic template that call to the view template because that embed will always be called last in the parsing order so it doesn't really matter where you put it.

Lea Alcantara: [Agrees]

John Rogerson: And I think that can just be confusing for some.

Lea Alcantara: [Agrees]

John Rogerson: Because they think the embed should be in a certain place in your template while it actually doesn't need to be.

Lea Alcantara: [Agrees]

John Rogerson: And you've mentioned Stash there.

Lea Alcantara: Yeah.

John Rogerson: Now, Stash is a plugin which is actually directly inspired by the template partials article.

Lea Alcantara: [Agrees]

John Rogerson: And Stash is a very, very powerful plugin that you can do a whole lot of stuff with that it doesn't need to be even template partials.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: [Agrees]

John Rogerson: But the concept of the Stash plugin is one of its purposes is to actually enable that kind of storage of content from template to template. So a real quick practical application of this is you would wrap a simple channel entries query. You would wrap it and one of the Stash tags which would essentially save whatever content is output from that and then you could, from your view template, you would use a different Stash tag to basically call that particular Stash of content.

Emily Lewis: So...

John Rogerson: And so it actually stores it in the database and you can cache it and do a point of stuff with it, but that's the basic reason or basic concept behind that.

Emily Lewis: And is the concept, I mean, this maybe a little fundamental, but is it essentially the same as when you do, for example, use an embed tag, you can include other data that's passed along.

John Rogerson: That's right, yeah.

Emily Lewis: This is just doing it using Stash and from what I understand because it's saved in the database, it's more efficient in terms of how that data is saved and then rendered when it's called into another, the view template?

John Rogerson: Yeah, I mean, I think also with, and obviously we've all used that methodology with the embed by passing variables, but I think that all you can really pass is the variable.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: You can't pass like an actual entire channel entries query.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Which is what Stash does, and I think this is where maybe it differentiates from using like Low Variables, for example.

Lea Alcantara: [Agrees]

John Rogerson: Where I maybe wrong because Low Variables just seems to be getting more and more powerful each time that it's got a new release.

Lea Alcantara: [Laughs] Yeah.

John Rogerson: But I don't think you can actually use Low Variables for generating content on the fly.

Lea Alcantara: [Agrees]

John Rogerson: I maybe wrong there, but I think that would be the difference between it and Stash, and obviously, you can. I mean, I use Low Variables all the time.

Lea Alcantara: [Agrees]

John Rogerson: Stash is not a replacement for it at all.

Lea Alcantara: [Agrees]

John Rogerson: I mean, all of these stuff is very complementary to each other.

Lea Alcantara: [Agrees]

John Rogerson: I would guess the one thing you don't want to do is use more than one embed. If it is a fundamental principle here in terms of just building your templates, you do not want to use more than one template. And actually, I think the other thing that's interesting in terms of this being a DRY approach is when you are in your logic template and you are doing your channel entries, you don't have markup within inside that channel entry, you use a snippet.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: And the idea would be that you could reuse that snippet so maybe you would have a title in your body content and that maybe your custom fields would be your body content and you would use your title and so you could have like an H1 or own your title and maybe just leave the body content if you are using a WYSIWYG or something like that.

Lea Alcantara: Yes.

John Rogerson: And you don't need to generate or you only need one snippet to take care of that. It doesn't matter which channel you are using because that might be all you need to have a display.

Lea Alcantara: I think...

Emily Lewis: Right, because like every page would have a title, every page would have blank content.

John Rogerson: Blank content.

Emily Lewis: Okay.

Lea Alcantara: Yeah.

John Rogerson: Yeah, so that's where you would bring snippets into the equation because I guess you don't need to do that, but if you want to be pure about your DRY approach, then you don't. That would be a way to avoid having markup within your logic template. I actually feel it's a little bit cheating really to do that, but it makes sense if you want to do that, especially when you don't want to be typing out the same H1 and wrapping your content in a P tag or whatever, you see?

Emily Lewis: [Agrees]

Lea Alcantara: Yeah, yeah. I think though with this entire concept, the thing that I'm struggling with a little bit is completely removing the HTML from the logic.

John Rogerson: Yes.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Lea Alcantara: Because I'm always use like right now, for example, with the site I mentioned that I just launched. So I'd have my like header variable with all the meta tags and things like that.

John Rogerson: [Agrees]

Lea Alcantara: But I might need to use a plugin or a channel entries thing or whatever for the title tag.

John Rogerson: Yes.

Lea Alcantara: And then so I just shove that in there. I feel like it's redundant to take that out, make another variable or make another thing, and then just have... do you know what I mean?

John Rogerson: Oh, absolutely.

Lea Alcantara: Because it's only going to be in the header.

John Rogerson: Oh yeah.

Lea Alcantara: Do you know what I mean? It's not going to be in like ten different templates that needed to be reused.

John Rogerson: Absolutely, Lea.

Lea Alcantara: So at that point then, I guess my question is, where do you draw the line over how much you take out?

John Rogerson: [Laughs] Yeah.

Lea Alcantara: And you mentioned the body copy with the H1 and the thing. However, if you only have the one-page inside page template, do you really need to take out... ?

John Rogerson: Oh, no, you don't.

Lea Alcantara: Do you know what I'm saying, right?



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Oh, absolutely.

Lea Alcantara: Yeah, because like if we only had like that one template group and we have two templates, is it really much more efficient if there is only really going to be one title tag and one body copy?

John Rogerson: Well, I think we're looking into...

Lea Alcantara: I guess the question is... yeah.

John Rogerson: It's like philosophical territory here, you know?

Lea Alcantara: Yeah. [Laughs]

John Rogerson: Which is fine, I mean, that's I guess one of the reasons why this really appeals to me is because I like the idea of actually thinking about how you do this rather than...

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

John Rogerson: And I think what you are saying is true, there probably isn't any real advantage to doing this, but at the same time, you could apply a similar kind of philosophical approach to how you actually do your plain old HTML and CSS.

Lea Alcantara: [Agrees]

John Rogerson: Because there might be occasions where it might be just easier to take the shortcut, but there is nothing wrong with that.

Lea Alcantara: Sure.

John Rogerson: I mean, it's just horses for cars as I like to...



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: Yeah, I think all of us have been in situations where there are going to be those really simple straightforward sites that don't have a lot of edged cases that require to think about different ways of doing things because everything is kind of the same and then you get those random projects where everything is the same except for these three sections and they all have these unique needs and you have to sort of evaluate what's best in those situations.

John Rogerson: Yeah. I think that there is a kind of cutoff point where the whole DRY ends. We haven't really talked about MVC which is not really DRY, but it's another kind of concept. It's really been talking about it too as I was mentioning it, but really there is a cutoff point where on a simple site, it's kind of like unnecessary. It probably wouldn't save you much time and it probably wouldn't make much difference in terms of your optimization of your actual site speed.

Lea Alcantara: Yes.

John Rogerson: I think though once you get beyond a certain point where you might have more than two or three channels.

Lea Alcantara: [Agrees]

John Rogerson: This is definitely going to make it quicker to develop your site. It's going to make your site quicker, period, and it's just going to be easier to change things, maintenance-wise.

Lea Alcantara: [Agrees]

Emily Lewis: I mean, I can see that as a single developer making sense, but I'm also curious, do you work with other developers and has this been something that's helped in that sense as well?

John Rogerson: I'm a freelancer, I would say yes. But yeah, I kind of have the double-edged sword of working on my own, so I can kind of work around and do things like this and not have to like worry a bit what someone else thinks.

Emily Lewis: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: But yeah, I mean I think, obviously, if you are in a collaborative relationship and when you are building a site, you obviously need to understand what and how the other developer would handle this and work with this and obviously be on board with it because then we are going to start from a very salient point definitely, yeah.

Emily Lewis: You sort of mentioned MVC and you said we've sort of addressed it without really addressing it. So do you want to take a step back and sort of define what that is and how we've sort of thinking in that approach with what you've described?

John Rogerson: Yeah, I mean, all of these, I know I am not a programmer, which actually I think is a good reason why this is good because if I can understand this and actually do it, then really anyone can. But I think the way, I mean, this is all just my interpretation, but to me the idea of MVC which I believe stands for Model View Controller.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: With that, you separate out your logic, i.e., your querying the database. You separate that out from your kind of view, which would be how your content is displayed.

Lea Alcantara: [Agrees]

John Rogerson: So the logic is what is the content? What channel does it live in if we are going to talk about ExpressionEngine, and how many pieces of that content do you want to show? Are you showing just one piece of the content, or are you showing three items, et cetera? I think the idea of the MVC is really tied in again with the optimization idea that's part of the DRY approach. I seemed to be mentioning Django a lot which is really dangerous, I guess. I mean, do you think so.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Especially when I don't have really any real experience in using it, but I think there is a lot of good ideas in that environment and also I think Rails is like this too which is coming from that angle.

Lea Alcantara: [Agrees]

Emily Lewis: And it's really just keeping everything separate, everything in its own little things. Just like we start it, like you said, we started embracing in the early days of the web of separating our behavior, our JavaScript from our CSS or presentation from our HTML, which is our structure, and now you are saying we are going to, in this way of thinking, we are going to separate our data, our control view, which is like the channel entries tag from the template view tag which is in this case, the presentation is sort of like HTML, right?

Lea Alcantara: [Agrees]

John Rogerson: That, yeah, absolutely, and I think when I'm doing this in ExpressionEngine, I really feel like I'm building an application rather than a website.

Lea Alcantara: [Agrees]

John Rogerson: Even if it is just a straightforward website, you are embracing real programming sort of techniques.

Emily Lewis: [Agrees]

John Rogerson: Which I think that you could make a point, "Well, what is the point of doing that?" And I guess, I think we've mentioned this earlier is if we think about content modelling, which was an article by Rachel Lovinger, I think her name might be in A List Apart recently.

Emily Lewis: Yes.

Lea Alcantara: Yes.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: And that really ties in with how I use ExpressionEngine anyway. It is because you can really separate out your individual chunks of content because of the custom field and its functionality.

Lea Alcantara: [Agrees]

John Rogerson: I actually see this whole MVC or DRY or whatever you want to call that approach and it's actually really tying in with that as well.

Lea Alcantara: [Agrees]

John Rogerson: Which is a really powerful part of using ExpressionEngine, and that this is something which I think a lot of developers have been doing and not even really thinking about, and I see it on EE Insider I think maybe today or yesterday, they even kind of said that in relation to that, and I kind of responded to the article that this is something that we just do and it's not news to us, this idea of content modelling. We haven't really called it that.

Emily Lewis: Yeah, I like the idea. I mean, like you said, it sort of discussing it philosophically. I do think that when it comes time for me to attempt to apply this, I think because of my more designery mentality, I think it will be a bit of a struggle because I've gotten so used to kind of keeping my HTML as part of my logic all in one sort of thing.

Lea Alcantara: [Agrees]

Emily Lewis: It will be a completely different way of approaching it.

John Rogerson: Yeah, I mean, we haven't really talked about it. I've touched on a couple of things on this, which is really I think definitely the original motivation for the template partials article is to cut down on your embeds.

Lea Alcantara: Yeah.

Emily Lewis: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: And I think that's an important thing that we need to kind of, and I know that there are lots of different methodologies to deal with this, I mean, especially with the snippets that have become part of the EE2 and also using Low Variables and using the field replace stuff that there have been a couple of really good articles on that.

Lea Alcantara: [Agrees]

John Rogerson: And I think that people now really understand that peppering your template with embeds is not a good thing if all it's doing is just calling parts of the template that you don't want to type again and again.

Emily Lewis: [Agrees]

John Rogerson: So that is a DRY approach in a lot of respects. It just comes at great expense to your performance, and I've really, really...

Emily Lewis: It sort of...

John Rogerson: Sorry, go on.

Emily Lewis: No, please finish.

John Rogerson: No, I was just going to say really there is a performance advantage to that.

Emily Lewis: [Agrees]

John Rogerson: It's 100%. I mean, even if you cut down your embeds, you will really notice.

Lea Alcantara: [Agrees]

Emily Lewis: And it seems to me that DRY, it can sort of be a little or a lot. You don't necessarily have to dive in and completely I guess be a purist with it.

John Rogerson: Yeah.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: You can sort of attempt to write things in a more efficient manner where you are not repeating yourself, but you don't necessarily have to go to the situation like Lea was describing where she takes a single scenario like in the header where she has to create an extra snippet rather than be getting it all in the same resource.

John Rogerson: Yeah, I mean, I have to say that using structure actually really helps with this whole approach because usually if you are using structure, one of the kind of outcomes is you really cut back on your channels because you might only need one channel for your entire site.

Emily Lewis: [Agrees]

John Rogerson: And you would be organizing your different sections for your structure. So if you are using structure and there is an absolutely no connection between this and structure, this is just a kind of benefit, I guess, of using structure is that really you can have an entire site with just two templates.

Emily Lewis: [Agrees]

John Rogerson: One being your view template and one being your logic template, and actually with structure, you don't need to worry about the title stuff because it comes with a title variable built in and bread crumbs and all that kind of stuff. But I certainly would never say, "Well, you need to use structure to really do this." But that's a really nice efficiency gain which is just a byproduct of taking that structure approach.

Emily Lewis: [Agrees]

John Rogerson: Yeah. I mean, I have to say, I mean, I was a real skeptic on structure at first because I kind of hold on to the native template group or template model.

Emily Lewis: [Laughs]

Lea Alcantara: Sure.

John Rogerson: But now, I'm just like, "This is easier." I mean, actually the goal is to tell the message. It's just easier.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: Easier.

John Rogerson: Yeah.

Lea Alcantara: Yeah. So as we are building a site using DRY methodology, of course, we are trying to reduce redundancies and duplicate a code, but what if you are approaching a site that you had already created, are there any tools or ways for you to detect what's duplicated? How do you know something is duplicated?

John Rogerson: So that's a good question. I don't have an answer to that one. But actually I have retroactively fitted this onto the sites, and I guess you could use some find and replace tools in your text editor if you are saving your templates as files. But that's a good question, I really don't know. I mean, you just need to know your templates inside out on your site and on the span of...

Lea Alcantara: So manually looking through.

John Rogerson: Yes.

Emily Lewis: Yeah, I was thinking about that question myself yesterday and at first I was like, "Oh, you know those tools that compared two documents and see what's changed in between them. It would be nice if there was something that along the same lines that could compare to files and see what wasn't different, but what was identical."

John Rogerson: Well, I think someday I should write the add-on.

Emily Lewis: [Laughs]

Lea Alcantara: Or I mean, isn't that something that's done through Version Control?

Emily Lewis: But I think that only checks for things that are different.

Lea Alcantara: Oh yeah, yeah.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Emily Lewis: Yeah.

Lea Alcantara: Yeah, yeah.

Emily Lewis: Like...

Lea Alcantara: Yeah, yeah. Not the same. Yeah, you're right.

Emily Lewis: I'm still a newbie to all that, but I think I'm right with that. I don't think it looks for things that are same.

John Rogerson: Yeah, I think you are right.

Lea Alcantara: Yeah.

John Rogerson: I'm a newbie too and that's my understanding of it, but I think that a really good add-on would be an add-on that actually at a basic level, it might check for how many embeds you are using.

Emily Lewis: [Agrees]

John Rogerson: And it might flag a template where you are using more than one embed. I mean, I think that would be really useful for a new comer to ExpressionEngine.

Emily Lewis: [Agrees]

John Rogerson: Because if you still go, if you look a lot of the tutorials, they are kind of basic stuff. I bet most of them probably still use embeds.

Emily Lewis: Yeah, well, and then I also think about if you bought a book last year or 18 months ago, it's still going to be referencing embeds because it does feel like they will push against embeds has been relatively new I think.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Yeah, absolutely. Let's not be embed haters because they have a place.

Emily Lewis: But...

John Rogerson: We are still using them. It just we are using them in a hopefully...

Lea Alcantara: Limited manner. Yeah.

John Rogerson: Yeah, hopefully just more efficient manner. Yeah, more efficient way, yeah.

Lea Alcantara: And I just wanted to add to it. I mean, obviously there is no add-on right now that just checks how many specific embeds there are in all these templates, but there is the output profiler and the debugging tools that ExpressionEngine has.

John Rogerson: Yeah.

Lea Alcantara: And it will tell you how many queries the page needs in order to load, and if it's massive, [laughs].

John Rogerson: Yeah [laughs]

Lea Alcantara: Then you need to do something about that.

Emily Lewis: That...

John Rogerson: Yeah.

Emily Lewis: That would be a sign.

Lea Alcantara: Yeah.

John Rogerson: That's excellent. I forgot all about that. I mean, that is one way to kind of I guess reassure yourself that you are doing the right thing.

Lea Alcantara: [Agrees]



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: It's just check how many queries per page, and if you are cutting down on your queries, then you are doing a good job, you know?

Emily Lewis: [Agrees]

Lea Alcantara: Yeah, for sure.

Emily Lewis: So I'm curious, John, since you've started doing this, what's been the bottom line for you, you as the developer, but also I guess your employer and the projects themselves?

John Rogerson: That's kind of all to me. [Laughs]

Emily Lewis: Because you are in charge of everything, so you get to this.

John Rogerson: Yeah, basically, yeah.

Emily Lewis: Yeah.

John Rogerson: Yeah, but that's a good question. When the bottom line, if you are working for an agency or if you are freelance, et cetera, the bottom line is how is this making your development cycle quicker?

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

John Rogerson: Is this enabling you to make changes quicker when after you've launched the site when the client would like a different number of articles being output, things like that, I mean, that would be how I would measure the bottom line. It would certainly be it, and also I think the basic performance of your site been rendered. Those will be the kind of straightforward measurements that I would always make.

Emily Lewis: And is this what you are sort of doing with everything, or do you still have maybe some sites that you haven't shifted to this methodology and so you kind of have to go back and forth? And if so, is it better working with the DRY?



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

John Rogerson: Yeah, I'm a purist. [Laughs]

Emily Lewis: [Laughs]

John Rogerson: I'm never going back.

Emily Lewis: [Laughs]

Lea Alcantara: It sounds good. Before we wrap up, do you have any final thoughts or recommended resources for anyone who wants to get started on a more pure DRY technique?

John Rogerson: Yeah, I mean, I would just say you don't have to be pure, I mean, at first.

Lea Alcantara: Okay.

Emily Lewis: [Agrees]

John Rogerson: I would say that I think the most basic level, I think the idea of having one embed per template should be your goal initially and that to really understand the parse order.

Lea Alcantara: Yeah.

John Rogerson: Remember, we haven't really talked about that much, but you really need to understand that and even when you feel like you understand it, it will still come back and bite you.

Emily Lewis: What do you use to help you with parse order? Is it just going through what you've developed?

John Rogerson: Yeah, I mean, there is always the very useful PDF document that wrote this.

Emily Lewis: Right.

Lea Alcantara: [Agrees]

John Rogerson: That I think most people they have them all with this parse order will have reference and I think also I know in my personal experience with using the Stash program that there are lots of, especially on



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

the Devot:ee forum for starter, they are lots of people who have come up against this problem and our parameters for the Stash program that you can use to kind of get your own way with the parse order, although there are simple principles that you can use in terms of how you use the Stash program in a way you fit it in your template that actually help you get around it. But I think the simplest way or ideal way with the parse order is to always remember that it begins from the inside out.

Emily Lewis: [Agrees]

John Rogerson: So it will process whatever is at the kind of core first and then it will pass out from that. So that's what is sometimes confusing because you might have something in to kind of deep in your cycle of the queries that might use a variable that hasn't been generated yet because the parse order hasn't got to where that variable is generated.

Emily Lewis: [Agrees]

John Rogerson: And that was always where I found it difficult at first.

Lea Alcantara: All right. So that's all the time we have for today. Thank you, John, for joining us.

John Rogerson: Thank you very much. I hope it's been informative and not confusing too much.

Emily Lewis: [Laughs]

Lea Alcantara: [Agrees]

Emily Lewis: It definitely has. I'm looking forward to trying some of the stuff myself.

John Rogerson: Cool.

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

John Rogerson: Sure. Well, I'm [@iboxifoo](#) as it sounds on Twitter and that's just iboxifoo.com is my personal website where you can contact me further.



<http://ee-podcast.com/episodes/dry-techniques-with-ee>

Lea Alcantara: Cool.

Emily Lewis: Great.

Lea Alcantara: [Music] Now, we would like to thank our sponsors for this podcast, [Engine Summit](#) and [Pixel & Tonic](#).

Emily Lewis: We would also like to thank our partners, [EllisLab](#), [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 when we are going to tackle .htaccess ([dun, dun, dun](#)). Be sure to check out our schedule on our site at [ee-podcast.com/schedule](#) for more upcoming topics.

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.

[Music stops]