



EE Podcast #95 EE Automation w/ Matt Fordham

[Music]

Lea Alcantara: You are listening to the unofficial ExpressionEngine Podcast Episode 95. Today we're talking to Matt Fordham about automating your EE development. 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 Engine Summit. The fourth annual Engine Summit returns online and live on June 12th. Engine Summit brings the experts to your desktop for a full day of deep diving into EE, EE add-ons and much more. Sign up now with discount code EEPODCAST for 20% off. Visit EngineSummit.com. See you at the Summit.

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

Lea Alcantara: Well, for some people on Twitter, people should know that I was selling my house.

Emily Lewis: [Laughs]

Lea Alcantara: So the new thing is my house is sold.

Emily Lewis: Congratulations!

Lea Alcantara: Yeah, it's a huge weight off my shoulders, so I'm super, super excited.

Emily Lewis: That's a pretty fast turnaround. I guess the market in your area up in Edmonton is doing pretty well.

Lea Alcantara: Yeah, I was a little bit worried because there was a lot of traffic. This is a thing about selling your house, sometimes you get traffic, but that doesn't mean any offers are going to happen, and sometimes there are just people who just want to snoop around. [Laughs]

Emily Lewis: Yeah, exactly. Well, that's great, and now, you don't have to keep it up nice and clean all the time now. [Laughs]

Lea Alcantara: Oh yeah. Yeah, once the offer was set and all the conditions were removed, I stopped washing dishes. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: Just out of principle because now I don't have to do it right away.

Emily Lewis: Exactly.

Lea Alcantara: Yeah.

Emily Lewis: Well, congrats. It's great news.

Lea Alcantara: Yeah, so why don't we jump into some EE news?

Emily Lewis: Right. So EEUK wrapped up I guess about a week and a half ago, and I think some of the biggest news that I heard at least on Twitter and online from that event were there were two really interesting add-ons introduced.

Lea Alcantara: [Agrees] One of them is Chris Imrie and Eric Lamb's Entry Analytics and Ben Croker's Open API. So with the first one, actually Chris and Eric gave us a little bit of review copy for the podcast, and I just installed it last week and so far, I think it's pretty neat. Just being able to see the statistics right in your control panel right away, I think it's pretty cool.



Emily Lewis: Cool. Do you think it will run some like reports off of it to maybe guide us in like what popular episodes were? Do you think it's got that sort of reporting level?

Lea Alcantara: Yeah, just right now it does tell you like the top ten entries in the past 30 days, so that gives you a little bit of an idea over like, well, what is resonating with our listeners, at least the ones that visit our website to view the episodes or whatever. So yeah, I think it gives us an idea over what is catching people's attention these days.

Emily Lewis: Nice. So the other one was from Ben Croker.

Lea Alcantara: Yes. The Open API. I have to admit I didn't really looked into it as much, but I believe that allows other people to hook into it so they can provide like authentication and other types of things like that to hook more into ExpressionEngine, which I think is always a good thing.

Emily Lewis: Yeah, it definitely looks like a developer-type add-on.

Lea Alcantara: Yeah, for sure.

Emily Lewis: I like the description that I'm reading. It's a lot of – I don't what it means, but...

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs] It's an API for Expression that provides authentication like you said, and I don't know if he pronounced it CRUD, Crud, but either way it's allowed to bring that functionality to data in the ExpressionEngine's CMS.

Lea Alcantara: Yeah, I mean, at the end of the day like he did put up his presentation.

Emily Lewis: [Agrees]

Lea Alcantara: And hopefully that will give a little bit of an insight as to what was going on. [Laughs] And he does have a video too, which I confess I haven't seen yet, but hopefully that will give us more info.

Emily Lewis: Yeah. I think it's great that at these events, developers are kind of announcing or releasing their add-ons. It builds a little buzz.

Lea Alcantara: [Agrees]

Emily Lewis: Also, when you're attending an event like that, it's always great to network and attend sessions and stuff, but if you also can get a peek at something new, that's pretty exciting, and I think it says that people are still really excited about ExpressionEngine. They are still developing new things for it.

Lea Alcantara: I agree. On the opposite end of the spectrum though in terms of developing add-on businesses and releasing new things, Erik Reagan, our previous guest, he actually write a huge blog post called, "The sidecar that never was," and he explains the thought process over why he decided that at least at this point in time that Focus Lab won't be releasing custom add-on business.

Emily Lewis: Yeah, I think it's interesting, and it somewhat echoes I think what Mitchell was talking about when we spoke with him about the competition in the marketplace and making it financially viable to have a really thriving add-on business. Hey, I think creating an add-on is one thing, but being profitable from it a completely different thing, and it's a business decision.

Lea Alcantara: Yeah, absolutely. I mean, it's not even just that profitability, how about the sustainability of it all?

Emily Lewis: [Agrees]

Lea Alcantara: You could have a slew of initial sales but to continue having those sales, and then at the same time, support, right?

Emily Lewis: [Agrees]

Lea Alcantara: I think that's the biggest hurdle that people need to figure out. How do you remain profitable while providing high quality support and still doing the other stuff that your business wants to do, right?

Emily Lewis: [Agrees]

Lea Alcantara: Some people have tackled that by simply just focusing on add-ons, period. Just like Brandon Kelly with Pixel & Tonic, all he does is add-on development, so it's not client work split with add-on development, which some people do, and that's totally fine. But people need to understand that if they do want to have a viable business with add-on development, it's a business. You have to take care of it as much as your original business if that is development and client services.

Emily Lewis: And it can lead to making some sacrifices or major changes in how your business strategy is. It's just not simple. I love the time that Erik took to kind of detail their thought process at Focus Lab.

Lea Alcantara: Yeah, agreed, agreed.

Emily Lewis: All right, so let's get to today's episode. Today we're talking about ways to automate ExpressionEngine development with Matt Fordham. Matt is a web developer and technical director at [WINTR](#), a digital creative company in Seattle, Washington. He also organizes the [ExpressionEngine Seattle Meetup](#). Welcome, Matt. Thanks for joining us.

Matt Fordham: Thanks for having me guys.

Lea Alcantara: So Matt, can you tell our listeners a little bit more about yourself?

Matt Fordham: Yeah. So I've been in Seattle for about eight years or so now, and I started out professionally in Nashville actually. I was a music producer and engineer for four years.

Lea Alcantara: Oh wow!

Matt Fordham: I went to a music school in Boston at Berklee College of Music, and as I was working with my clients there, I wanted to make myself a website and so I did and learned it. At that time, I learned flash and based off of what I was doing, my music clients, they started asking me to make websites for them. So it kind of snowballed.

When I moved to Seattle with my wife, which was where she's from, it just kind of snowballed with the web thing. I actually ended up teaching music productions in Seattle for six years at a college, but I always kept a kind of a freelance web business on the side, and then a few years ago, I just decided I needed to change pace and I switched to doing web kind of exclusively. So yeah, that's me in a nutshell.

At WINTR, I'm responsible for sort of leading all of our development efforts. We sort of provide clients with complete digital strategy from planning and design through development and execution, and we also have a lot of motion and RD skills in house, so we do a lot of different kind of stuff.

Emily Lewis: Cool. You're the second person we've spoken with recently that has a background in music, but now does web design and development.

Lea Alcantara: [Agrees]

Matt Fordham: Yeah, yeah. Specifically with music production and engineering, I think there's definitely some common lines between them in terms of trying to make something creative fit into a technical box. It's pretty much the same thing.

Emily Lewis: Do you have a lot of music clients with WINTR, or was that something that you sort of didn't have a focus on once you joined WINTR?

Matt Fordham: Yeah, that's definitely not a focus. We look to have clients in pretty much every market. In terms of music stuff, we just did the website for the Sasquatch Festival in the Pacific Northwest.

Lea Alcantara: Oh, very cool.

Matt Fordham: Which is coincidentally taking place right this very minute. We do some stuff like that, but no, music is not a focus, although we are all interested in sort of all areas of creative expression, so music and fashion and design are a big part of what we do in general.

Emily Lewis: Nice. So in terms of preparing for this episode, we all had some emails back and forth, and one of the things you shared with us was sort of an overview of a typical project life cycle for you. In that life cycle, you sort of broke things down into three different, I guess, phases maybe. Can you quickly share what those three phases that you commonly see in projects, and kind of talk about the types of tasks that fall into each.

Matt Fordham: Yeah, sure. In terms of the phases that a developer would be specifically involved with I suppose with things like planning and stuff like that. The setup of the project, the development of the project and then the deployment of the project are sort of the three main phases, and I look to sort of automate as much of that as I possibly can and it's partly my personality type. I think it's been said many times before kind of jokingly that a good developer is a lazy developer.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Matt Fordham: And I don't think of myself as a lazy person, but I really hate doing stuff over and over and over again. It drives me crazy, so I'm constantly looking for ways to avoid those repetitive tasks and build or find tools that speed things up, and also managing a team, it's especially important.

Lea Alcantara: [Agrees]

Matt Fordham: I don't want for the team members to be doing the same things over and over again, so I'm looking for ways to make their lives easier as well constantly. I think one of the challenges with automating is not so much the actual automation, but the sort of realizing and noticing that you have a problem and that you're doing something over again.

Lea Alcantara: [Agrees]

Matt Fordham: I think that we so easily get numb to doing something repetitively.

Emily Lewis: [Agrees]

Matt Fordham: We almost don't notice that we're doing it.

Emily Lewis: Yeah.

Matt Fordham: So that's the big challenge. It's sort of noticing where the pain is and then solving it is I think a lot easier actually.

Timestamp: 00:10:01

Emily Lewis: So let's talk about that for a second. So how did you and your team identify those areas where you could use efficiencies, that you weren't being repetitive in your tasks?

Matt Fordham: Yeah. Well, it's hard to describe. It's almost the opposite. It's almost like where do we really like spending our time.

Lea Alcantara: [Agrees]

Emily Lewis: [Agrees]

Matt Fordham: And it's not setting up projects. It's not deploying projects, so it's like the things that we really like doing often overshadow the things we don't like doing, and when we realize that there are things fighting for our attention when we are trying to do the cool stuff, that's when we sort of say, "Hey, it will be cool if we didn't have to do that because we'd really like to focus on this cool stuff." So it's those moments where we are in the flow as it were and something kind of jumps in the way like a roadblock almost that I sort of take note and figure out a better solution for next time around so that we don't have to sort of spin our wheels and lose that flow. We're constantly in this creative back and forth within the company between designers, developers, the client, and the stuff that needs to be automated just gets in the way.

Lea Alcantara: [Agrees]

Matt Fordham: So that's how we notice it, it's when we are sort of taken away from what we really want to be doing.

Emily Lewis: So Matt, it sounds like what you're describing is something that's ongoing. You guys are constantly looking for efficiencies in your projects?

Matt Fordham: Yeah, absolutely, and one of the advantages of doing so many projects and starting projects so frequently is that there is always an opportunity to try out a new idea or to sort of hone things the next time around. So whether it's trying out a new tool or sort of tweaking the process a little bit, and also documenting the process is a big part of my job so that it becomes official. I found over time that you can have the best intentions in terms of managing a team and developing a process for a team, but unless things are documented, unless things are sort of made official in some capacity, all those good intentions kind of get pushed aside. So yeah, absolutely, it's a constantly ongoing sort of honing of the system for sure.

Emily Lewis: So let's talk about some of the efficiencies that you've come up with for that setup phase of your projects.

Matt Fordham: Yeah, so the first part of it is a project template, and I know that it's been discussed on the show a lot. So our project template, as other people have mentioned, is an ExpressionEngine install that is in Version Control so we have a Git Repo that contains the template project, and that's an installed copy of ExpressionEngine, so there's a database dump within it. When we create a new project, well, the way we used to do it, I don't talk about the way we do it now, but the way we used to do it is copy that directory and import the database and you're basically good to go. It's sort of a base install.

Lea Alcantara: [Agrees]

Matt Fordham: An important piece of the base install for us is the config piece of it, which Erik was talking about recently, making sure that it works in multiple environments without tweaking stuff in the database so it works locally as well as working on a staging or production server. So we're trying to get all those configuration settings into sort of a configuration bootstrap file.

Lea Alcantara: Is that something you've customized for WINTR, or are you using somebody's config?

Matt Fordham: Yeah, I'm using a customized version of the Focus Lab one. It works great. It works super great.

Lea Alcantara: Okay.

Matt Fordham: Having the template, well, we've been doing that for a long time, it's great. It certainly speeds up the process, but I realized a few months ago, well, maybe six months ago that the process of creating and copying that template itself was annoying. So I thought a little bit about how I can automate the actual sort of setting up of a new instance of a site based upon the template. So I thought about it for a long time and I was inspired a little bit by Carl Crawley's talk at EEI on the fall. He had written a bash script which basically did that. It would check out the template from Repo and it set up a new instance of it, and that's what I wanted. I always thought about doing it, but I wanted something a little bit less ExpressionEngine specific so I could use it on other types of projects.

So I made this tool called Strap which is a RubyGem, and basically you would create these templates within it and I have an ExpressionEngine template that I created for it, and what it does is it's a single command that I run in the command line and it will create my new project for me. So it will check out my template into a new directory. It will create the database. It will import the base database install. It will set permissions on all the files in each piece and have their permissions tweaked. It will then initialize and push to a new Git Repository for this new project.

Emily Lewis: Wow!

Matt Fordham: So going from my project template to a brand new project is just like a couple of commands on the command line.

Emily Lewis: Wow!

Matt Fordham: I had to tweak my little ExpressionEngine template for the tool to sort of make it quick each time, but that's awesome, like the command takes like five minutes to run because of the downloading and uploading piece of it. Once it's done, the database is ready to go and so I can log in and just get down to business.

Lea Alcantara: Wow, five minutes, definitely that's worth the investment of time.

Matt Fordham: Yeah, and also if somebody else in the team is tasked instead of setting up a new project, it's a way of ensuring that they do it exactly the same way that somebody else would do it.

Lea Alcantara: Yeah.

Emily Lewis: [Agrees]

Matt Fordham: So it's a time saver, but it's also sort of a consistency check.

Lea Alcantara: Yeah, absolutely.

Matt Fordham: So that's Strap. I don't know if many people will use it as a name, but it works great for me, and as I said, you can use it with any kind of project, anything that has a Git Repo basically. You can have it automatically check it out and do some stuff and then push it to a new Git Repo.

Emily Lewis: That process, the five minutes, it sounds fantastic. I'm curious though that template that you are bootstrapping basically each time, how do you guys maintain them?

Matt Fordham: Yeah.

Emily Lewis: Like I imagine when you're going through a project, you have something came up, and you learn something from it and you want to make sure that the template reflects that. So are you just constantly documenting or are you literally making changes to that template that's been copied basically during your bootstrapping process?

Matt Fordham: Yeah, that's a great question, and the big challenge is making sure that you don't forget that stuff and then start it with a slightly deficient big template next time.

Lea Alcantara: [Agrees]

Matt Fordham: So it depends on how busy things are. I'll do one of two things. I'll either make notes so I just kind of keep text file around. If I like installing a new add-on that I can't live without or if I decided to change some kind of preference, I'll just put a note in that text file and then before the next project, I would start off by opening up the template, tweaking it, usually upgrading ExpressionEngine or some add-on, pushing it so that my template is now up to date, and then I start the whole bootstrapping process.

If it's not that busy, I'll just go ahead and make the adjustments to the template at Repo when I'm thinking of those adjustments and pushing them. It just takes a minute probably and then the next time we pull from that template repo, those updates are in there.

Lea Alcantara: So when do you decide to update or upgrade your repo based on when ExpressionEngine itself upgrades to say 2.6 or whatever, or the next 2.7? Do you wait a little bit, or do you use your older version, or until it's battle tested by the community, or do you just upgrade like everything the moment it's been updated?

Matt Fordham: No, with ExpressionEngine, I tend to wait a little while in I would say a few months of a release being out, and then we'll update it. It's usually just because there's no time to update it, and if things aren't broken, I don't worry too much about fixing it at that point. But I like to keep it updated within a few months.

Lea Alcantara: Sure.

Matt Fordham: Yeah, but we don't have a system for that. We don't have specific release plans in place for that. Usually if it's working, we don't sweat it too much.

Lea Alcantara: All right. So is there anything else that you like to do during the project setup process? You've created the template. You bootstrapped all of your installs and your add-ons and you've done Strap, is there anything else that you do? Or is this basically once it's set up, you're ready to go straight into development?

Matt Fordham: I usually actually jump to the last step and deploy the project immediately.

Lea Alcantara: Oh, okay.

Matt Fordham: Yeah. Just so that while we're developing, we can be pushing constantly and people can always sort of see the latest up online. So that's usually something I like to have and be ready so that it's easy to deploy and that we don't waste a day when it's time to sort of set it up, that it's nice to have it set up already. So that's usually something I'll skip ahead to, and usually it only takes a few minutes.

Lea Alcantara: [Agrees]

Matt Fordham: But then we're pretty much all ready to start development.

Lea Alcantara: Once you're ready to start developing after everything has been deployed at least the first step of deployment, what is the first steps of automating the development process?

Matt Fordham: Well, a lot of the automation that takes place during development happens with front end stuff, which isn't really ExpressionEngine specific, but I think it's worth mentioning at the very least.

Lea Alcantara: [Agrees]

Matt Fordham: And that stuff is already set up. It's in the project template just sort of be ready to go, so really once the project template is up and running, we're jumping straight into development because the tools that we use are already there. So we use Coffeescript and Sass with Compass, and a lot of folks use tools like CodeKit and other sort of desktop applications.

Lea Alcantara: Yeah.

Matt Fordham: We don't partly because of this automation issue. I want to have the configuration of all that stuff be something that's checked into the repo, and with a tool like CodeKit, it's difficult to do that. You kind of need to trust that each developer is going to be setting it up the same on their own.

Lea Alcantara: Yeah.

Matt Fordham: So we use a tool called Guard, which it does a lot of different things. At its heart it's a tool that will watch for file changes and will run some tasks whenever a file gets saved.

Lea Alcantara: Okay.

Matt Fordham: So we use it for a few different things with ExpressionEngine projects, although we use it with all our projects regardless of the technology. With ExpressionEngine projects, it's looking for Coffeescript files to be saved. At which point, it will compile into JavaScript. It's looking for Sass files to be saved, which at which point, it will compile into CSS. It also watches sort of the public directory and will refresh the browser whenever anything gets updated. So we're using the LiveReload extension for that, which they make an extension for all the major browsers. One other thing that this pipeline does for us is that whenever we saved any of the Coffeescript stuff or the JavaScript stuff, it will concatenate and minify all of the JavaScript files together.

Lea Alcantara: Oh wow!

Matt Fordham: Right then when you hit save.

Lea Alcantara: Oh.

Timestamp: 00:19:52

Matt Fordham: So basically the process is you save your Coffeescript file, it concatenates all the JavaScripts together. It compiles from Coffeescript to JavaScript, concatenates everything, minifies everything and then that triggers a browser reload so you can see your changes update immediately. What's cool about that is that when it comes time to deploy, we don't have a build script to worry about in terms of minifying JavaScript. It's like it's already done, you know?

Lea Alcantara: [Agrees]

Matt Fordham: So there's no danger of somebody pushing to the server unminified stuff. It's just done in real time basically.

Lea Alcantara: Oh, that's pretty convenient, yeah, because right now if you don't have that process, there is an add-on by John Wells. It's called Minimee, and that's what we use right now at the EE Podcast, but you have to remember to turn it on because we debug, we have to turn it off, take a look at the stuff, turn it back on and sometimes we forget to reminify after.

Matt Fordham: Yeah, exactly. Yeah, so that's what our front end sort of process looks like. Again, it's nice because all of that configuration is checked into the repo, and all those tools are RubyGems, and Ruby has a nice tool called Bundler which is a single command that you run, it's bundle install, and that will install all of the dependencies that your project has so you can ensure that a developer has everything they need just by running that one command.

Emily Lewis: [Agrees]

Matt Fordham: That's assuming they have Ruby installed, which is a whole another conversation.

Lea Alcantara: [Laughs]

Emily Lewis: So when it comes to your Coffeescript and Sass files, do you start with like a core file, like a template, even for those?

Matt Fordham: Yeah, absolutely. More so for the Sass stuff. The Coffeescript tends to be a little bit different for each project. We have some basic ways in which we organize the code, but for Sass, we have pretty much like a boilerplate set of files that and a pattern that we use for all of our projects. So we know that if you're applying styles to a base HTML element, it goes in the underscore base Sass file, for example.

Emily Lewis: [Agrees]

Matt Fordham: And so there's basically a Sass partial as it were for each of the kind of things that you would do with CSS, and we have a sort of mix-ins that we use in Sass. So we have a mix-ins that we use for responsive stuff, for retinographics, for creating sprites automatically. So we're all using the same sort of base configuration of Sass files so

that we know what we are doing, and we also have all agreed upon a way of naming classes and things like that so that there's no doubt as to what something should be named and where it should go.

Lea Alcantara: Is that documented somewhere like the class names?

Matt Fordham: Yeah, it is, absolutely. So basically, I maintain a separate repo which is that Sass sort of boilerplate stuff, and within that I have lots of comments, code comments about, "Here's how you create this thing, and here's how it should be documented, and et cetera, et cetera."

So yeah, I tried to get that stuff all written down. We also have a sort of a series of Google Docs that we use internally that we would have like a new developer read through and kind of study a little bit when he starts just to sort of get up on things like code style and how do you format your comments in the code and stuff like that. It's taken a while for us to develop that library of documents and practices, but we're definitely finding that it's worth the effort. As soon as there are more than two developers, it's sort of helps a lot. Just kind of a tangent, but I like very much to not know who wrote certain codes.

Emily Lewis: Yeah.

Matt Fordham: Do you know what I mean?

Lea Alcantara: Yes.

Matt Fordham: If you can look through the code and say, "Oh, so and so did that," that's a bad sign I think. You want it to sort of look like there's one unified developer working on the project. [Laughs]

Lea Alcantara: Yeah, for sure. So why don't we bring it back to ExpressionEngine development. Is there anything that you do during development that helps speed things up? Are there any automation add-ons that you use?

Matt Fordham: Yeah, for sure. So one that's really useful is Seed. The Seed add-on will basically create dummy entries for you so that if you create a channel, like let's say it's a news channel, you can have it create 30 news items with some just random text to populate things immediately. So for example, if you are testing pagination and you didn't want to sit there and create new entries for half an hour.

Lea Alcantara: Sure, yeah.

Matt Fordham: That's what that is for, and that's an obvious timesaver.

Emily Lewis: I'm curious, I've never used it. I've heard it about it. It sounds really interesting. Does it provide like an undo, so when you create all those test entries, that maybe you want to remove them at some point in the future?

Matt Fordham: It might, but I don't think so. I guess you'd just have to just select the whole bunch of them at the same time when you're out there.

Emily Lewis: [Agrees]

Matt Fordham: I guess you could add associated category to them to test out a category and then delete everything within that category at once.

Emily Lewis: Right.

Matt Fordham: I'm just thinking off the top of my head. Another really useful add-on is DataGrab, which I think you guys have talked about before.

Lea Alcantara: Yeah.

Matt Fordham: Clients give us stuff obviously to put into the CMS and if we can get it in like CSV format from a spreadsheet and importing it with DataGrab is so much better than importing stuff manually.

Lea Alcantara: Yeah.

Matt Fordham: So yeah, that's another one, and then a third one that I really like. It's not really automation, but it's a major timesaver is Field Editor.

Lea Alcantara: Oh, agreed.

Matt Fordham: So Field Editor is another useful add-on. It basically allows you to view, edit and create all of the fields for a channel field group at the same time on one single screen instead of clicking into the edit form for each field separately, and you can duplicate fields between channels and you can also import and export channels in too. I think it uses a JSON format. That's really useful for quickly creating channel groups to field groups, which obviously can be a time consuming task.

Lea Alcantara: So I find that a little bit interesting. In your default EE install back in the project setup, you don't have anything precreated for a group there already?

Matt Fordham: Yeah, that's right. The channels and the field groups are empty.

Lea Alcantara: Okay.

Matt Fordham: Yeah. So typically, our projects don't usually follow a pattern.

Lea Alcantara: Okay.

Matt Fordham: So there's often some kind of blog or something like that, but usually those channels are totally bizarre, which is why we love ExpressionEngine, you know?

Lea Alcantara: Sure.

Matt Fordham: Because it allows us to kind of model the data in whatever way it makes sense for the client. So usually we start from scratch with that stuff.

Lea Alcantara: Awesome. So when we first started emailing back and forth in your document, you had this really interesting thing about generating a private preview EE templates to automatically generate. Can you talk a little bit about that?

Matt Fordham: Yeah. So using something like the built-in Live Look functionality or using the EE Garage Live Look. Have you guys used that before? The trick with that is that you have to give it a template, right?

Lea Alcantara: Yeah.

Matt Fordham: So it can show you what's going on, and that's great just out of the box if you are wanting a client to be able to preview stuff within the context of the control panel and just save time without having to go around to the front end to look at it. But if you're wanting this preview functionality to sort of be a private preview thing, then it gets a little trickier because you need to have a template that will only show private things. Otherwise, you're exposing that sort of "private stuff" to the world in that template.

Lea Alcantara: Sure.

Matt Fordham: So I think there was a few ways you can handle this, but the way we handle it is by duplicating the template and adding to the channel entries tag the status of preview. So when somebody wants to preview something, they save it with the status preview and then only their user group can view that status, so then they can view that template, and that works great. The problem there, obviously, is the duplication of the template, which is a huge drag.

Lea Alcantara: Yeah.

Matt Fordham: So on one of the last projects we did, it was actually for that Sasquatch Festival thing, the lineup announcement is a big deal and it's very secret as to who is in the lineup, and the last thing we want is for the lineup to get leaked before that festival is announced.

Lea Alcantara: Sure.

Matt Fordham: So it was really important that our preview templates were kind of locked down. We had to do this, but I was looking for a way to sort of not do all this copying and pasting. So I wrote a script, a Ruby script, and I haven't figured out a way to make it into sort of a really reasonable tool yet, but I wrote a script that basically when I run it, it duplicates the templates, it will search through the text of the template and automatically append the status of preview to all of those status parameters in that thing.

Lea Alcantara: Yeah.

Matt Fordham: And then copy it over to a preview template group, so that way I don't have to do it manually, so it's just a matter of running the command and all the templates automatically show up in that preview directory with the preview status all set up.

Lea Alcantara: All set up, yeah.

Matt Fordham: Yeah, exactly. So it's a little tweaky. It took me a moment to write it. I've been trying to think about a way to sort of make it a little bit more reusable and something that potentially other people can use easily. But to me, the key there is just sort of thinking about doing that kind of thing. It sort of hadn't occurred to me before, I had to do a whole bunch of them, so it took a minute to write the script to do that and I think you can do that with bash, you could do it with Ruby, you could do it with PHP probably. So I'm going to try and get something out there that can potentially work with the control panel. I think that would be nice.

Lea Alcantara: That would be cool.

Matt Fordham: Yeah.

Lea Alcantara: I would appreciate that.

Matt Fordham: [Laughs]

Lea Alcantara: [Laughs]

Matt Fordham: Yeah.

Lea Alcantara: So once we're all finished with the development, it's time to really do the final deployment. How do you automate that?

Matt Fordham: Yeah. So our projects are deployed in one of two ways now. They're either deployed to kind of a more traditional VPS or they're deployed to one of the sort of newer "Cloud" hosting options.

Lea Alcantara: [Agrees]

Matt Fordham: And in both cases, we use Version Control to take care of the deploy. If we're deploying to a VPS, we use a Ruby tool called Capistrano to take care of that.

Lea Alcantara: Yeah.

Matt Fordham: And what Capistrano does is it doesn't have to be running on your server. Your server doesn't have to run Ruby or anything like that. It basically just automates your machine. Well, when you run the Cap-deploy task, it will log into the server. It will do a Git pull to the server. I guess it does a checkout, and then it does this clever thing where it will create a symlink, which is like an alias to a folder called Current, so that when you point your server at the Current folder, and then that in turn points to a specific release, so that if you needed to roll back a release, all you have to do is run another task which is Cap-deploy rollback and it updates that symlink to go to the previous release.

Lea Alcantara: [Agrees]

Matt Fordham: So instead of like updating the same directory of files on the server, it actually creates a new directory of files and points the server to that directory so you can very easily sort of go back in time between your releases if you mess something up.

Lea Alcantara: Interesting.

Timestamp: 00:30:01

Matt Fordham: So that's cool. Another nice thing about Capistrano is that you can write any tasks that you would want to perform in the server. So you can write a task that a really useful one for us is taking any images that we've uploaded locally on a machine and pushing them up to the server. So like oftentimes we will be developing and we will be entering content for clients, and we don't have the images checked into the repo, but we need to get them on the server, and the only other option is FTP, right?

Lea Alcantara: Yeah.

Matt Fordham: So you're like just drag them up so that when you import the database to the server, the images aren't missing.

Lea Alcantara: Yeah.

Matt Fordham: So we wrote a Capistrano task that simply uploads all the images from local to production. We have other ones that will take the database from production and dump it locally and import it to our local database, things like that. You can also write a task to install some software in the server. If you always use something like APC to speed up PHP, you could write a task to log in, install APC and log out again. So you could really automate pretty much anything about your server, and the goal for us is to never open FTP.

Lea Alcantara: Interesting.

Emily Lewis: Can I ask why?

Matt Fordham: Yeah. Well, there are a couple of reasons why. Number one, it's time consuming. Number two, if FTP is involved, there's a higher chance of breaking something and having something not be in Version Control. So all of the code itself for the site is in Version Control. It's deployed from Version Control, so doing like logging into the server and making changes to a template is very much not allowed in our environment because the next time you deploy, those changes will get wiped out. But the stuff that's not in the repo, really it's just user uploads through the control panel, that stuff needs to be managed a little bit differently and backed up separately as well because it's not part of the repo.

Lea Alcantara: Interesting.

Matt Fordham: Yeah. The other kind of project is this Cloud thing, and we do a lot of Ruby work and Heroku is a great platform for that, and basically all you do is a Git pushed to Heroku and it will spin up the server for you and deploy the project. It sort of senses what kind of project it is. Heroku doesn't work at this point really well with PHP so we've been using a company called PagodaBox for that. As great as Capistrano is, this is what we want to do in the future. It's create an app on PagodaBox. They basically give you a new repo and you push it to this repo and pushing to that repo will spin up the server right then and there and deploy it. So there literally is no logging into the server to configure something. It just happens automatically, and the cool thing about this is that you can spin up multiple instances of the server.

Lea Alcantara: [Agrees]

Matt Fordham: So going back to this site, we just launched for Sasquatch. When they announced the lineup in like a few hours, there were hundreds of thousands of visitors, we needed something that would scale up super high and then 24 hours later, scale way back down again. With PagodaBox, we were able to do that. We could have a dozen instances of each of 1 Gigabyte of memory or something, which for a month would have cost like a \$1,000 or something, and then

when that rush is over, scale it way back down again and you end up paying \$30 for the day. It would have been catastrophic without a really, really big server.

So that's such a great way of managing scaling, and it's so easy to just do a Git push to deploy a website. It means that we don't have to be server administrators and there are so many tasks that we don't even have to worry about automating because they're doing it for us.

The downside is obviously that you have a little bit less specific control over the environment and you can't get in there and do whatever, but they've accounted for a lot of things that you're likely wanting to do.

Emily Lewis: So let me take you a step back and make sure I understand. Is PagodaBox an app that resides on your Cloud host account, or is it actually the host provider?

Matt Fordham: Yeah. Sorry for not being clear about that. It's actually the hosting provider.

Emily Lewis: Okay.

Matt Fordham: So when you create an app, they give you a subdomain, which is like myapp.pagodabox.com or something like that, and Heroku does the same thing, and there's another one called AppFog because that's what you get and that's where your app shows up once you do this deploy, so you do a Git push and then after it's done, once you see it, when you see it deploying, you'll see it run for all these scripts that they have like setting up your server and provisioning the database, all that stuff, and then when it's done, you go to that subdomain they give you there's your app.

I'm simplifying a little bit obviously. You need to import your database and things like that. But then with your domain, you would do like a C-name record for your domain to point it to that subdomain instead of pointing it an IP address, and that way, they can dynamically create extra servers for you and sort of take care of all the IP addresses automatically and your subdomain never changes. It always points to your app. It's interesting to think about it. It's like literally a different server every time you deploy, but it's the same code.

Lea Alcantara: Interesting.

Matt Fordham: Yeah. And they allow you to share things between releases, so you can use your uploads. It gets shared between each of your releases on the sort of shared drive so that it's okay for there to be some stuff that doesn't change. It doesn't have to be into Version Control, for example.

Emily Lewis: So was the Sasquatch Festival projects one of the first ones you've done with PagodaBox, or maybe I should be more generic, was it the first one you've done with like a Cloud hosting solution?

Matt Fordham: We've done many sites with Heroku, with RailsApps, and we really liked how flexible it is. It wasn't how easy it was for us to deploy. We can deploy to a brand new site within minutes without having to really log into the server at all, but Sasquatch was the first ExpressionEngine site that we did like that with PagodaBox. So I kind of went out on a limb a little bit with PagodaBox because it was such a high traffic site, but no regrets. It worked really, really well.

Emily Lewis: Oh yeah, the scaling you described, it sounds like a Cloud solution is only one that could cost effectively allow you to do that.

Matt Fordham: Yeah. The only alternative is to have a VPS or to have like a bunch of different VPS servers that are buying some kind of load balancing, and you have to manage that a little bit more on your own. You're going to be paying for them, and so to scale them up and down by the minute, and then PagodaBox will charge you literally by the minute. So yeah, it's really hard to recreate that situation without kind of building their service from the ground up, which obviously you'd be crazy to want to do. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: So I'm curious, do you measure any of the benefits that you've gained by automating all these things? Because the entire point is to save some time, do you know how much you've saved?

Matt Fordham: It's such a cumulative thing, and it's something I've been doing my whole life as a developer that I sort of have nothing to compare it to, but I can tell you that measuring it in my emotional well being, it definitely measures very high. [Laughs]

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Matt Fordham: I'm much happier.

Lea Alcantara: Very cool.

Matt Fordham: Yeah, I'm much happier having these things taken care of and knowing that other developers can step in and help out on a project without too much set up and that things like server updates can happen without risks.

Lea Alcantara: Sure.

Emily Lewis: And I would imagine your team members, anyone could come in and look at the project and understand where it is. I must be nice to be a developer on that kind of team where you do know there is a reference point for you at any given point in time.

Matt Fordham: Yeah. I hope so.

Lea Alcantara: So before we wrap up, Matt, do you have any final thoughts about anyone thinking of automating their development and project processes with ExpressionEngine?

Matt Fordham: Yeah, I think the key to all of this stuff is just being aware that it is possible, and that usually somebody else if you realize, as I said earlier, that something is taking you extra time or causing you pain, somebody else probably thought about it already, and if you search around a little bit, you're likely to find the solution. So just being conscious and aware of how you can automate I think is the key. I'm just kind of taking on and subscribing to that philosophy.

Lea Alcantara: Awesome. So that's all the time we have for today. Thanks for joining us, Matt.

Matt Fordham: Thank you for having me.

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

Matt Fordham: I'm [@mattfordham](#) on Twitter, and people can email me at matt@wintr.us.

Emily Lewis: Great. Thanks again, Matt.

[Music]

Matt Fordham: Thank you.

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

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

Lea Alcantara: And 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: Don't forget to tune in to our next episode when Ryan Battles is joining us to talk about EE error pages. Be sure to check out our schedule on our site, 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 unofficial ExpressionEngine Podcast. See you next time.

Emily Lewis: Cheers. [Music stops]

Timestamp: 00:38:12