

EE Podcast #93 Dev Environments with Erik Reagan

[Music]

Lea Alcantara: You're listening to the unofficial ExpressionEngine Podcast, episode 93! Today we are talking about development environments with Erick Reagan. I am your host Lea Alcantara and I am joined by my fab co-host...

Emily Lewis: Emily Lewis.

Lea Alcantara: This episode is sponsored by [CSS Dev Conference](#). This year CSS Dev Conference is situated in the Rocky Mountains of Colorado at the beautiful Stanley Hotel. Join us for two days devoted to CSS and its super-friends JavaScript, Sass, Compass and more. Speakers include Eric Meyer, Nicole Sullivan, Jonathan Snook and maybe you? The call for more speakers is open right now, if you live and breathe CSS visit [CSSDevConf.com](#) to submit a talk or register for Early Bird to get in as an attendee. Follow [@cssdevconf](#) on Twitter for more info.

Emily Lewis: The ExpressionEngine Podcast would also like to thank [Pixel & Tonic](#) for being our major sponsor of the year.

[End Music]

Emily Lewis: Hi Lea, how are you doing?

Lea Alcantara: You know, I think I'm getting boring because I have the same complaint each episode for the past three episodes, [laughs] but its blizzarding right outside my window right now, [laughs] so I think I am a little bit justified about being annoyed by the weather.

Emily Lewis: Well at least you got a couple days of some nice warmer weather in Columbia, South Carolina for the Converge Conference.

Lea Alcantara: Yeah, I am so pumped at having attended ConvergeSE this year and speaking, Emily you spoke there, too, how did you like the conference?

Emily Lewis: I really enjoyed it, it's always great to see some faces I haven't seen in a long time. But there is something about this conference that, at least for me, was new in the sense that the way they had it structured. Friday was sort of like a technical day of workshops and then Saturday were talks that were a little more human, in talking about personal experiences and inspiration. It was a really nice balance, and I felt it was just like a diverse mix of topics, a diverse attendee group, a diverse speaker line-up. I am feeling very energized and excited now.

Lea Alcantara: I totally agree, I saw a tweet from Nevin from EngineHosting, and he said that it kind of reminded him of how he felt the first time he went to SXSW. You know that feeling where you're just surrounded by your peers and they are super-inspired and energized. I really loved the fact that the second track on Saturday was full of those personal stories and the inspiration. I feel like it humanizes the tech community a lot more and I think that needs to be done a lot more, just overall. And I think by doing so would address some of the issues our community is dealing with right now.

Emily Lewis: Yes I agree, it was a really nice mix of people. There were a lot of EE folks there, Ryan Irelan, Nevin Lyne as we mentioned. Kelly Cook, who if you remember, she was one of our contestants on our live podcast last year at EECl, and oh, the guys from FortySeven Media, Kicktastic were there. It was really nice, it was a nice mix of attendees, developers, designers, some project managers, business owners – that was also really cool.

Lea Alcantara: Yes, I think it was a great mix of really our industry, it wasn't just tech and it wasn't just design, everyone was kind of represented, that's what I really liked about Converge.

Emily Lewis: Yes, I definitely recommend it and will definitely go back. Let's talk about the big news in the EE community, which is 2.6 is now out.

Lea Alcantara: Yes, so with 2.6 that means there are new features, which is awesome. They have a new improved relationship field that supports multiple relationships. They also have a simplified date and time system that no longer asks [woo-hoo!] if you're observing Daylight Savings Time. They have a better forgotten password flow and they are also changing the Strict URLs to default to follow their recommendations.

Emily Lewis: Right, I am actually really excited about the forgotten password thing, I haven't installed 2.6 on anything yet but apparently the remember me functionality is working [laughs].

Lea Alcantara: I know, that's great, especially because not all members of a site are administrators or people like us, the ones that access particular pieces of membership information, for example. And I know for my less tech-savvy clients or members, they get confused by the forgotten password flow, a lot, for whatever reason. I'm not sure why, but this better forgotten password flow will probably help things out.

Emily Lewis: Yes, I've got a client I'm working on now, we're just beginning the process of the EE phase, so I'll be installing 2.6 and it'll be interesting to see how it goes.

Lea Alcantara: Cool.

Emily Lewis: Alright, let's get to today's topic. Our special guest, Erik Reagan is returning to the podcast, this time to talk about development environments. Erik is a partner and Technical Director at [Focus Lab](#), a web design and development studio in Savannah, Georgia. At Focus Lab, Erik leads the development team and specializes in web architecture and project planning. He is also an active member in the ExpressionEngine community, speaking at EE events, blogging about EE and helping out folks on the #eecms hashtag. Welcome Erik, thanks for joining us!

Erik Reagan: Thank you, thank you both for having me again.

Lea Alcantara: For those that aren't familiar with you, you can tell our listeners a little bit more about yourself?

Erik Reagan: Sure, I am a husband, I'm a dad, I've got a little girl and another on the way, and have been in Savannah my whole life, born and bred here. And that's where our little Focus Lab is founded. We've got a small team of a few folks locally and a few who are remote, and we are just having fun building stuff.

I like guitar, I like music, I was a music major in college but ended up doing web development instead. Let's see, what else is fun ... I really like to play pool, so any kind of ExpressionEngine or tech-related event I get to, I always try to find a local pool hall to see what it's like. I think that is a well-rounded little synopsis.

Lea Alcantara: Pretty cool.

Emily Lewis: It's interesting you mentioned you were a music major. Do you think that gave you a really good foundation for coding, because music in its own way is as technical as it is experiential?

Erik Reagan: Absolutely. There is a lot of correlation between what it takes to understand and to do music from a technical level, and programming things. There are a lot of folks who do both. In the friends that I have made over the years, initially I was surprised — now I just kind of expect it — but I was surprised at how many people who were in the programming world were musicians. And vice versa, how many musicians like to dabble in code. There is definitely a lot of correlation in kind of how the brain works and which people are going to like those tasks.

Emily Lewis: For today's topic we are going to be talking about development environments. Before we get into some of those details, I was hoping you could define what a development environment means?

Erik Reagan: That's a great way to start. The simplest definition I can offer is that a development environment or a "dev" environment — I will probably shorten it for the bulk of our chat. A dev environment is the server on which you can break anything and it won't matter. This means that the client's live site doesn't come down or sales aren't lost, or people aren't rightfully mad. [Laughs] I guess some people might get mad, but not really in a sense that a business isn't running. It's any version of the website that can be messed with without any detrimental consequences after the fact.

Usually it's a different computer, a different server from the live one. But sometimes it's actually still on that same server, just maybe a different domain, like a sub-domain or something. A development environment is one where you can break it and it's okay.

Emily Lewis: If the development environment happens to be on a different server, is that server configuration identical to what's on live or production?

Erik Reagan: Traditionally speaking, at least one version will be. I will talk more about that, but I have multiple development environments in our work flow, and in a traditional sense — I hate the phrase — but a lot of people might consider a "best practice" is to have at least one version that is identical or near identical to that live server so that if something is going to break in the process of making a change, you're going to discover it before it actually gets into that live environment.

Emily Lewis: Okay, now I have often heard, in fact it was back in my days of working at Pitney Bowes, we had development, staging and then production. What is your take on a staging environment and how would you define that?

Erik Reagan: That's actually pretty much what I just mentioned, that would be the staging, where staging should be pretty much be identical to production, as much as possible. On the development environment side, for us, that's usually a Focus Lab server that we just use for client projects for when we're in development mode or creating something that hasn't been launched yet. The staging side though, when possible, we try to use the same server setup or sometimes the same exact

server, not always recommended. It kind of depends on the load of that server and what you're able to do with it. We treat a lot of this on a case by case basis, but that is a flow that we follow where we do have a development environment and staging, which is still not live, but at least acts the same way that server will act.

Emily Lewis: And staging, you don't actually do anything except push from dev to staging or do you do work on staging – directly on it?

Erik Reagan: The former, we will push from a dev to staging and at least click through the site on the front-end to make sure stuff works. Sometimes we'll do some control panel testing before it makes its way to a production environment just to make sure that maybe like if we build some custom add-on or something like that, that it works in that new environment. But for the most part it's just a place to test, but we don't actually build there.

Lea Alcantara: What I am curious about, especially because I'm by myself, just having local and then staging and then the live environment, that's a pretty simple straightforward workflow. But for your company and other people's companies, you've got several developers working on the same thing. I am looking at your EE Master Config and that adds a couple more config files and stuff. How do you deal with sharing the development and local development, and how do you coordinate that?

Timestamp: 0:10:03

Erik Reagan: Sure, one important piece to note is we use Git for our version control, and so all of our code is tracked there and pushed to a repository that just allows us to pull and push those changes as individual team members. On the code side we're all working on our local computers apart from anything else, which means I can break something without messing up one of the other developers on the team. And then once I get it done and fixed, I can send that code over to him; make sure he's got the latest.

The only challenging part to having multiple people working on stuff in tandem is the database, which is actually kind of tricky ... it can be. For us, what we found to be the most efficient is to basically point our local computer or our local ExpressionEngine settings to use a remote database, which is the database of our development environment, which sits out on a host somewhere. That means that when we load the control panel or the front-end, that everything is fast on the front-end where it's requesting template files or other ExpressionEngine documents. But whenever it has to talk to the database it's a little bit slower.

It causes some latency in terms of how you're testing things, but it also makes it a lot easier; less headaches for a team to work on something. It kind of depends on the certain feature we're building. For example, if I'm working on a custom add-on I'll do the vast majority of that with my local database, because I don't want to do things that could break a development in the database. But once I'm done, I'll push that out to the development environment and we'll all start using that development database again, or at least I will. That helps.

When we are testing content and everything, nobody wants to have to enter test content on my local computer and then tell Person-B to go enter the same test content on theirs or create their own test content. For the most part we share remote database to kind of streamline the in-progress build process.

Emily Lewis: When you put together the EE Master Config repository, why did you decide to put that out there for the public?

Erik Reagan: I started the config setup when ExpressionEngine was in 1.6.9, and I had a project that had a specific approval process requirement, so I would work on the site on my local computer and then push it to another server through our Project Manager, and a QA or Quality Assurance person to approve. Once that was done, it would go to the live or production site. And that was the first project I worked on where I ever had additional environments, so I had to figure it out. The config setup was

born because I was tired of constantly changing server path settings when moving things from one server to another. And ExpressionEngine, by design, stores a lot of that in the database and that was a headache. I would also change things like the webmaster email address for email testing. I would always get non-production tests.

That led me to research what config settings could be overwritten in the config.php file. My biggest goal was using the same file for each server to more easily track and deploy these changes in Git and our deployment tools. It was a little messier in the EE-1 days. ExpressionEngine would actually overwrite the entire config file without any care to what was in it whenever a control panel setting was changed, even if it was one setting. That was really frustrating, so we had to find a way to lock-down that file on the server side and then we never really had clients who were changing settings. But the occasional one who did, it would be a headache because they would change it and it would say that it was changed in the “success” message, but it was actually not changed because the file couldn’t be written. It was an interesting conundrum.

There was also a path.php file to consider and use in EE-1. Thankfully EE-2 made it a little simpler. So it helps in the development because I can easily define environments based on parameters and most people it’s just the domain name, the parameter that you would base that environment on. That’s what we usually recommend. It makes it a lot easier.

The reason we made it public is because mainly of Aaron Waldon, Causing Effect fame. He told me to. [Laughs] I always thought it would be worth sharing. I just never got around to it, and I guess it was about two years ago or so, Aaron and I were working on a project together and he and I were discussing the config setup a bit so he could have his local environment. He nudged me to share it with the community, so we decided to document it a bit more and throw it up on Github. Honestly it hasn’t really changed much since, we have added a couple new lines here and there as ExpressionEngine has added to what is configurable from this file, but we haven’t really changed a lot of it.

Emily Lewis: If I am understanding correctly, because I am just sort of looking at the documentation on Github. It detects what environment you're working from based on the domain and then from that it pulls the relevant variables for if you're working locally for the local variables.

Erik Reagan: Correct.

Emily Lewis: That is really straightforward for someone who would just be using and not coming up with the concept. [Laughs]

Erik Reagan: Sure, when we made the config setup for EE-2 we had an opportunity to kind of revisit how we did it. Sometimes you just build something and you just kind of keep using it for a while and you don't really change it, because there is not a need to. When EE-2 came out, we had to change it because there was enough under EE-2's core that meant we needed to make some shifts and some changes. And Matt Weinberg had actually already written an article on EE Insider about how Vector Media Group does multiple environments with their config setup. Leevi Graham at Newism had also put together their bootstrap before I had even mentioned what we were working on, or what we were using I should say.

Some of the things that we did before we made it public on Github was consider some of the things that they had done and basically make what we had already done even better. Because you know, when you work with other people, when you learn from their ideas you can make something that much better. What we had wasn't even what it is today, but those guys' ideas had some influence on it, and I think it does make it, so far, fairly easy to understand.

There is a little difference on Leevi's versus ours, his is all in one file which some people really like, having it all in one file. It's just a personal taste thing, I like having separate files, and technically there's a little bit extra overhead because php has to load a few files, but it's nominal, it's so small it's not hurting anything. I like to keep things very clean, very minimal and that's just kind of a company thing at Focus Lab, anyway. From our design to our development, we try to be very minimal.

Lea Alcantara: So there are a lot of cogs, I guess, when you're designing and developing with multiple environments. Is this something that's needed for all projects? What determines whether you need a development environment for a client? Like is it a Master Config file default for all projects whether they are big or small?

Erik Reagan: Honestly I think very few things are actually needed or default for all projects. So in a literal sense I'll say, no. A development environment is not always needed. I would always recommend it though. The main reason is so that you have a play area for new features or add-on's, etc. Where you can just break things and not worrying about it. Playing with these ideas and features in the context of a specific client site is pretty important. That way you know if there will be any clashing with other implementation aspects like add-on's and what not.

At Focus Lab we do use our Master Config setup and multiple environments on all projects, big and small but it's not a necessity.

Emily Lewis: You mentioned before that you rely on Git for version control. My understanding is that you can't version a database, so you did mention that when you are working locally you're pointing a development database, but is there a need for versioning or coordinating with your different developers if you're all accessing that same database, entering test content, and such?

Erik Reagan: Yes you definitely all have to be on the same page and have a coordinated workflow. And for us, how we manage that content across multiple environments is really a case by case treatment, again. The rule of thumb we try to stick with is that structural changes always flow, what we call "upward" and content changes always flow "downward."

Structural changes are things like creating new member groups, new channels, new fields, changing Publish Layouts, configuring add-on's, etc. Content changes are mainly things in the channel model but may also involve things like e-commerce product details for tools like BrilliantRetail, or a new

variable for Low Variables, which you guys discussed in a recent episode. That's kind of the structural side and the content side.

The upward and downward ... structural changes flow upward. And by that I mean we start new structural changes in a local or a dev environment, and then we enter a bunch of test content to make sure it functions the way we expect. Sometimes if the exact content is already available we'll test with it, but sometimes not. Then we deploy those changes to the next environment in the chain. And by "deploy," we are using a tool like Capistrano or Beanstalk or Deploy HQ, it kind of depends on what your comfortable with using. And that's usually like file base changes. If there are database changes as well, we use something called Navicat, which is just another database tool and it has a specific feature called "data transfer" which makes it easier than it would have otherwise been to move data from one database to another in a different environment.

Content on the other hand flows downward, and by downward flow we are talking about content from production is always going to be considered correct or accurate. After certain feature deployments we'll copy down the content from production to staging and then down to dev and however far down that chain goes.

Going back to how I started I guess, it's always a case-by-case basis. On sites that have a lot of active changes and features, it gets pretty tricky to deploy only certain changes at a time, that kind of gets more into how we use Git in tandem with database deployments. It goes possibly a little beyond the scope of the discussion today, as there are so many different rabbit-holes you could go down of "what if's" that we talk about with our clients pretty regularly ... but that would be a three-hour podcast.

Lea Alcantara: [Laughs] Just to touch on that a little bit, let's say you finish all your development and you're about to push it live, but at that time the client's live site is actually being updated and everything like that. Do you tell your clients not to update for a certain amount of time or you take the

site down at any time? And if there is a downtime, when's the best time to push changes to the actual live site?

Timestamp: 0:19:57

Erik Reagan: One of the things to consider, I don't want to beat a dead horse, but it's always going to be a case by case. And with Focus Lab, the vast majority of our clients don't have content changing every day, and that's an important detail to think about. I'm not going to have the same problems that some other people will have, or the same problems to solve that other people will have just because of the type of projects that we're talking about in terms of the context.

What we do basically, it's a rule, we just say "okay Mr. or Mrs. Client, this is the date after which you cannot add new content, here is the reason, do you agree – yes or no?" And if they say no, we get them to say yes. [laughs] Basically it's just an education thing to make sure we're all on the same page. We make sure we are not going to mess them up businesswise at all. If for any reason the inability to publish content would mess them up, then we just talk through that and figure out what a solution would be. The trickiest one we have done wasn't even tricky. The clients we work with don't have such time-dependent needs on publishing content.

For us, what that means is at a certain date they stop entering content. And if they have changes, then they are going to make them after whatever date we give them. So a site may or may not have to go down, or the updates it just depends on if it's a feature.

I think you mentioned, maybe it sounded like the context of your question was a site rebuild or redesign where there's a live site and then a whole new big redone site ready to go. With those, what we would do is temporarily use the staging environment for what we would call "completely accurate content" and right before launch that would be the environment where everyone says this is what it has to look like. And then once that deployment is done, staging all of a sudden is the mirror of production. And when production is finally done they can go back in and start changing things again.

Emily Lewis: I am curious, is this something that you include in your original contract with the client, defining the process and setting their expectations of these things? Or is it something you do once you reach the appropriate stage of a project to begin those discussions?

Erik Reagan: We talk about it in the sales process, even before the contract is out there. If we have somebody new, then they are going to learn about what that process looks like before they sign the dotted line. We try really hard to set expectations as best we can, and so we talk a lot about that before we even get involved in the contractual relationship.

Now, if it's somebody we have already had a relationship with, chances are they already know a little bit about the process. But if they don't it's still more a meeting/communication thing than a contractual detail they signed off on, at least in the contract sense. It's more about communicating effectively than anything for us.

Emily Lewis: Is this process, does it add a significant amount of time to your development?

Erik Reagan: Initially yes. As we were trying to find a good process for this, it was pretty time consuming, a lot of that in terms of cost. We were just eating that cost as we were learning what we wanted to do. But as we have done it more and more, it's really not that time consuming. Again that goes back to the type of projects we work on, though. If it was a more highly-trafficked news site for example, or I know we've got some campaign sites out there – not we Focus but the ExpressionEngine community — has some campaign sites out on ExpressionEngine. There are different sites with different needs, and for the stuff we've worked on, we haven't had very complex needs.

Lea Alcantara: What I am curious about is that this is dev servers and dev environments, so clearly developers are all within the loop. How about your designers, do you provide special training for your designers to work with this? Do they even work in this type of development environment?

Erik Reagan: They are usually at least aware that there's a workflow on our Git side, but really our design team doesn't touch code, at all. There is really not any training involved.

Emily Lewis: Has it been necessary for you to do training with developers, especially like maybe someone new to the team?

Erik Reagan: Always, always. It would be irresponsible for me to think I didn't need to train them, even if they came from a company that does awesome things, chances are very likely that our process at Focus Lab is going to be different. So there is always time, especially if we are...maybe we've got a little bit of overflow and we're working with a contractor for a little while on something. We're going to have to account for time with them to show them what our workflow is like, answer questions, allow for a little bit of learning time as the developer gets used to what we do. For someone new to the team or new to the workflow, there is always going to be time for some type of training. Usually it's pretty minimal, though.

Emily Lewis: You mentioned you work to educate your clients about this process, but is there actual training needed for clients, or special documentation you provide them?

Erik Reagan: No training, per se. I guess it's more like awareness. We explain the workflow and the approval process for changes and how environments play into that. We also have visual indicators for environment, so clients are more easily aware of which environment they are in when they are looking at like a test version or when they aren't. And that's something like on the front-end, where in the browser they see this insanely hot-pink tag that says "staging" or "dev" or "local" or something. There is some awareness that needs to go into that, but in terms of the documentation, usually we provide a PDF or sometimes video training for content management, and we'll include a little bit about how to know which environment you're in and the purpose of the environments. We at least teach them a little bit about that. But it's probably a pretty small portion of the overall training that we would provide.

Emily Lewis: Now since you've been taking this approach, especially since things have evolved from your work with Master Config with 1.69 to now, what have you seen have been the benefits for you in terms of internal development as well as the project management itself?

Erik Reagan: First off we use a little bit of what I have heard some folks consider a bootstrap, where we can just easily start a new EE site based on a repository that we already have, and part of that includes our Master Config setup. So there is a lot of time that is saved in starting up a new project and often you have to think about different server paths and different configurations. In having our Master Config setup saves quite a bit of time on the getting things started, and also on the getting things deployed, even for the first time, aspect.

I guess I should at least specify that if you just install ExpressionEngine with the built-in installer and don't use any type of customized config setup, all of your path settings are stored in the database. So if you ever have to move to a different server than you installed it on at any time in the future, chances are it's not going to work. There are a couple of tools out there that are kind of control panel centric tools that help change those paths more easily, but having a Master Config setup uses php to dynamically define what those paths are so that you can, with almost no effort, drop these files onto a new server and it works.

Now you also have to put the database on the new server, but the point is that there is a lot of time saved than having to go from Point-A to Point-B. If you ever move servers in the middle of a project and you have Master Config set up, it's probably going to be pretty easy to get it from one environment to another, on the sense of things working. Now the timing of things is another area where context matters and it's a case-by-case thing.

Emily Lewis: We have talked to you before on the podcast about documentation ... when you are dealing with multiple environments, do you maintain independent documentation for each or is there

just documentation. I am thinking most specifically about your Dev Docs type of documentation, or is there just a single document for production?

Erik Reagan: We have it in every environment, it's a single document or a single directory of files, and there is a section for each of our sites that has an overview of the environments. It talks about the server, the software on the server, php, and MySQL versions and stuff like that, the IP addresses of servers, nothing extensive though. It's really just like a table with four or five rows and four or five columns, it's very basic and once it's set up initially in the project it usually doesn't have to be changed. But that's kind of the extent of our documentation for our developers sake for our environments.

Emily Lewis: I have a question that's not really about development environments ... I know at one point you were part of the EE Reactor team. Is that group still active?

Erik Reagan: Yes it is. The idea behind EE Reactor I absolutely loved, so I jumped at the opportunity to be involved with that, and especially early on there were a lot of really good discussions between folks on that team and the EllisLab team. If you peruse the changelog for the most recent release of ExpressionEngine ,you'll see there are some changes that were made by the Reactor team. If you look through, there are changes in I think almost every release since it came around, at least one or two that were made only because of the Reactor teams existence.

I can say though that it's very quiet, I think we were all, all the members were very excited and then quickly realized we were all very busy as well. [Laughs] I can say that the majority of my participation has been in discussion and not in code, and actually doing anything. The problem I found with myself is that all of the ideas that I had were kind of big ideas and not small ideas, and big ideas take a lot longer to discuss and quite frankly should not be a Reactor team. They should be done by EllisLab. We have had a lot of good discussions with folks like Wes Baker and right now the Reactor team exists and I am not really sure – I know that the pace has been quiet but steady, and I am not too

sure of what the future holds for it, but I know that it's out, it's alive. I haven't personally contributed any code any time recently but it's still going.

Emily Lewis: Now when you did have an opportunity to contribute code, was there something like a development environment that you guys were all working from or that you did something locally and then through Git or some type of version control pushed it to a single environment that all of the team members could access? How did that work?

Erik Reagan: The beauty of Git is that the entirety of the code base is wherever you push it to, and so we would all have our own local instance of the repository and push it out to one where the EllisLab team could pull down the change and test it. Or even if it didn't even get that far, sometimes it was more of code is pushed to that repository and simply discussed.

I did probably more reading than I did discussing, just because of the good conversations that come from proposing new ideas. It's really beneficial for me to just kind of read through the discussions and see how other Reactor team members said "hey I've got this idea, here's the code" and then someone else in the lab would say "okay, that's a cool idea, have you thought about this it will impact this area over here" or stuff like that. It's cool to watch that, but that all happened in a private environment where if the changes were approved or were good and done they could easily be merged into the code that EllisLab maintains.

TimeStamp: 0:30:09

Lea Alcantara: Well cool. Thanks Erik, I think that's all the time we have for today. Thanks for joining the show.

Erik Reagan: Absolutely, thank you for having me.

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

Erik Reagan: They can find me on Twitter [@erikreagan](#) or [@FocusLabLLC](#). The way it sounds, there is no “S” on Focus Lab ... everybody and their grandmother tries to put one there, but there is really not – I’m just saying.

Emily Lewis: [Laughs] Cool, thanks a lot Erik.

Lea Alcantara: Now we would like to thank our sponsors for this podcast, [CCS Dev Conference](#) and [Pixel and 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 Mitchell Kimbrough of SolSPACE is joining us for our “Get To Know #eecms” series.

Be sure to check 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!

[End Music]