

EE Podcast #96 EE Error Pages with Ryan Battles

[Music]

Emily Lewis: You are listening to the unofficial ExpressionEngine Podcast, episode 96! Today we're talking to Ryan Battles about EE error pages. I'm your host, Emily Lewis. Unfortunately, my fab co-host, Lea Alcantara isn't able to join us today, but we've got a special guest co-host filling in...

Anna Brown: Anna Brown.

Emily Lewis: This episode is sponsored by [Entry Analytics](#). Entry Analytics brings the performance metrics of Google Analytics directly into an ExpressionEngine site. View metrics for your individual entries in the control panel, create custom reports using template tags and more with Entry Analytics, [available at Devot:ee](#).

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

[Music ends]

Hi Anna, thank you so much for filling in for Lea today!

Anna Brown: Hi Emily, I'm really glad to be here. It's a pleasure to be invited. I really like what you guys are doing with the show, and I'm very happy to participate. Thank you.

Emily Lewis: Oh, thanks. Well, you're also an active member of our community so it's great to have you on board, especially not as a guest, you get to guide our guest today.

Anna Brown: I will do my best. [Laughs]

Emily Lewis: [Laughs] All right, well, let's kick off with a little EE news. It's been a long time since Lea and I talked about the EE Stack Exchange, but you've been an active and vocal supporter of the efforts since it began. Would you mind filling our listeners in on where things stand and what's still needed with the EE Stack Exchange?

Anna Brown: Sure. I think as most people know, we're in public beta still for the Stack Exchange site. The Stack Exchange site can be found at expressionengine.stackexchange.com. The site, it really is a public support site for ExpressionEngine, and there's not really an ETA for when this will get out of public beta. That's really up to the Stack Exchange folks, but our numbers to me look great. We just surpassed 3,000 questions on the site.

Emily Lewis: Wow!

Anna Brown: We're at 4,600 answers with 1,600 users with about 1,100 visitors per day, and I think our stats are great. It's definitely really great participation in the community, and I think going forward, we all just need to be better about visiting the site daily or as much as we're able to do and to answer questions, and ask questions for that matter.

Emily Lewis: So you said the folks at Stack Exchange will be the ones to determine when it comes out of public beta. Do you know what their criteria is?

Anna Brown: I don't know that that's very clear. They have probably an internal discussion about the longevity of the site.

Emily Lewis: [Agrees]

Anna Brown: So they're looking to see, if they launch it, they really don't take it back so when they launch it to a true real site and it's going to be there pretty much forever.

Emily Lewis: Awesome.

Anna Brown: Wherever the numbers land, they then will make that decision, so I don't know what that point is, but I think we're doing exceptionally well when you look at other sites that are in public beta. We, compared to them, are doing quite well.

Emily Lewis: So I have to admit that I haven't used Stack Exchange myself. I'm curious, and one of the reasons why is that I tell myself I don't really have time, and I know you're busy, so how do you make time to stay involved in Stack Exchange.

Anna Brown: Well, I've been slacking a little bit, but in the beginning, [laughs] I was exceptionally active. I just made it a part of my day. In the mornings while I was drinking coffee, I took a look at the site. I answered questions that I could answer. I participated when I could. I'm not always available to do it, but I promoted on Twitter just to make sure that people remember that it was there, so I was actively just reminding people daily, which generally helps to keep the momentum going. Participating doesn't mean answering questions either. I mean, you can be the person asking questions.

Emily Lewis: [Agrees]

Anna Brown: So when you have relevant questions to answer as well, so I would use it in that way if it works for you.

Emily Lewis: Then I'm correct in that you can pose a question and also answer it yourself?

Anna Brown: Yes, absolutely, and you can do that. In the flow of actually asking the question, there is an option to answer it yourself. So if you feel that whatever situation you're in is applicable to other people and it's something that should be archived, then this is a great place to do it.

What I think some of us are seeing now is that when you Google an issue, these Stack Exchange questions and answers for that matter are coming up quite high in the search engines. So as that happens more traffic will grow, and people will know that the site is a great place to get public support from the community. I'm just exceptionally proud of what we've been able to accomplish in 206 days.

Emily Lewis: Yeah, it's really impressive.

Anna Brown: For 206 days we've been in public beta and I would like to publicly thank Patrick [Poehler] for proposing this at the ExpressionEngine conference last year, for proposing this site. It has been proposed several times before in the eight years I've worked with ExpressionEngine and it just hasn't gained momentum, and I'm not sure exactly why it did this time, but I'm thrilled to see us as a community rally behind something that works for us, supports our daily jobs and is a great tool for all of other developers in this area. So I'm very proud of us.

Emily Lewis: Excellent. Now, before we get to today's episode, you've been talking a little bit about your involvement promoting Stack Exchange, but can you tell our listeners a little bit more about yourself and what you do each day?

Anna Brown: Sure. Yep, absolutely. So well, I'm here in Albuquerque right down the road from you.

Emily Lewis: I know, I'm waving at you right now. [Laughs]

Anna Brown: [Laughs] Are you? I don't see you enough [Laughs] for somebody who lives right down the road. So I'm based here in Albuquerque, New Mexico. I'm actually in Corrales, which is on the north end of town. I'm a full-time ExpressionEngine freelance developer. I work for agencies, one particularly here in Albuquerque, and then a couple in San Francisco, and then sort of just random projects across the country. I've worked as a full-time freelancer for about eight years now, and I wouldn't be doing anything else really. I will be doing my first public talk on an ExpressionEngine project of mine at the Peers conference at the end of the June.

Emily Lewis: Oh.

Anna Brown: And this is a conference in Chicago. It's June 26th through the 28th. So if anybody is looking for a conference in Chicago right around that time, please come and join us. We would love to have great participation at the conference, and I would love to have full room for my talk.

Emily Lewis: [Laughs]

Anna Brown: Everybody should just know I'm extremely nervous. [Laughs]

Emily Lewis: So you mentioned you're nervous. This is your first time speaking. What made you decide to do it because I know that I had bugged you for a couple of years about doing some local stuff, and it just wasn't something that interested you, so what's different this time?

Anna Brown: Well, like I said, I've been a developer in the ExpressionEngine community for eight years, but I worked in a little bubble out here in New Mexico and there aren't a lot of developers here

in Albuquerque. I mean, there are, but I don't know them. So I really was working in this bubble, and I was in the forums a little bit and a little bit in the community. But I didn't get to these conferences, so last year I happen to win a ticket from Devot:ee for the EECl Conference in Austin, so I had to go.

[Laughs]

Emily Lewis: [Laughs]

Anna Brown: So that was really sort of, I guess, my stepping out in the community and meeting people for the first time. And honestly I didn't understand the importance of that, and I'm just beyond thrilled that I won that ticket, and that I had the opportunity to meet my community in person.

So when I was there, I realized that we needed more women to step up in this community to present and give talks and talk about our experiences as developers and our projects. I also knew that I had something to give back ... I work on a project that I think is something of an anomaly in ExpressionEngine community, and I've learned an enormous amount about ExpressionEngine and how it will work with, or how it scales with a high-traffic sites and a busy publishing sites. So I have some valuable lessons to present. So when Jessica [D'Amico] talked about putting Peers together, I absolutely said, "Sign me up." I'm thrilled to be presenting. I'm sort of humbled to be on the same stage as some of the other speakers, and I just hope I don't flub it. [Laughs]

Emily Lewis: [Laughs]

Anna Brown: So everybody, rally and show up please. I would love a full room of very familiar faces.

Emily Lewis: Yeah, that sounds great. Well, it's very exciting. I'm really excited for you.

Anna Brown: I'm very nervous.

Emily Lewis: All right, so let's get to today's episode. We are talking about EE error pages with [Ryan Battles](#). Ryan is a web developer who's passionate about startups. He started his own [Jovia](#)

[Web Studio](#) back in 2008, specializing exclusively in ExpressionEngine development, but last year he joined [Crushpath](#) as their Lead Front-end Engineer.

Welcome, Ryan, thanks for joining us today!

Ryan Battles: Thanks for having me! It's a pleasure to be on the show!

Anna Brown: Hi Ryan, great to see you, or great to talk to you. [Laughs]

Ryan Battles: Yes, it's good to reconnect. I know that we all, three of us, connected down in Austin at EECl.

Anna Brown: Yeah, so it was great to meet you there.

Ryan Battles: Yeah, yeah. I was just listening to you describe your story for getting down there. That was a pretty interesting story, and I'm glad you won the ticket as well.

Anna Brown: Right. Ryan Masuga, thank you. It's really Ryan's fault that I got there. [Laughs] But thank you, Ryan. Thanks for joining us.

Emily Lewis: Ryan, can you tell you our listeners a bit more about yourself?

Ryan Battles: Yeah. So I started off as a schoolteacher. I actually taught high school in junior high computer classes for six years before I decided that I wanted to go into business building web pages myself. I was teaching my students how to build web pages, and some of them had a passion for it and some of them didn't care. And after a while, it just kind of grated on me that these kids who are taking these classes didn't think it was that neat because I thought it was awesome. So I knew I had a calling to kind of start my own thing. So I left teaching and started Jovia Web Studio, as you said, and have been building pages for the web for the last, I guess that would be five years I've been doing it now professionally.

Timestamp: 00:10:00

So yeah, last year I decided to close out Jovia Web Studio after a great four years of serving clients. I just had a neat opportunity to join a startup in its early stages. I had a chance to meet the co-founder behind the startup. He seem like a really great guy, a really interesting opportunity to get connected with.

So I made a tough decision, I closed down my studio and started working full time for these guys out of San Francisco, and it's been a great ride. Just the last year has been full of all kinds of learning opportunities and opportunities to work on items that I didn't really stretch myself to work with clients, but now that I'm working on a web app, there's a lot more fun in code that I'm being exposed to that I wouldn't have normally so.

But we still use ExpressionEngine. We use ExpressionEngine for our marketing site for Crushpath, and any side projects that I do, ExpressionEngine is my go-to. I still am very active in running Director-ee, the online community of ExpressionEngine developers and the official ExpressionEngine job board. So yeah, I have side projects and spend my days working for Crushpath.

Anna Brown: So your job with Crushpath is the developer?

Ryan Battles: Yes, yeah. I'm the head of the front-end development there so lots of JavaScript, HTML, CSS, bug fixes, and things like that. It's nice to be on a team full of talented people who have strong back-end skills because that stuff used to scare me a lot with EE.

Anna Brown: Yeah, right, yeah. [Laughs]

Ryan Battles: Yeah, I've never built a plugin or anything for ExpressionEngine. So all that stuff I always just bought what was out there. So it's nice to be surrounded by other talented people.

Anna Brown: I agree. I was the same way. So you work remotely, right? You're in Columbus, Ohio?

Ryan Battles: Yeah, sort of. It's pseudo-remote. We have like several hubs in Crushpath. There's actually five of us now in Columbus so we do work out of a co-working space, but we have a

dedicated office here. So I work remotely with the team in San Francisco, but a couple of people are also working remotely with me, and we're usually not on the same projects, but we just share the office space so we have an office to go to.

Anna Brown: That's neat. That's actually a nice situation.

Emily Lewis: Now that you're at Crushpath, you mentioned you have a passion for teaching and I've seen you give presentations before, do you do any internal training, or working with other team members to talk about best practices, what you're doing?

Ryan Battles: Not necessarily here with Crushpath. I'm still pretty active. Actually, with my personal blog, I just redesigned it and then republished it on a different platform just to kind of expand my skills. But yeah, my blog is where I like to write a lot of how-to articles. Anytime I come across maybe a difficult JavaScript problem that I need to solve or a neat way to do things with the PHP file, anything I have to research and figure out, I usually turn around and write out what I did in tutorial and post it to my blog. That is helpful to me in the future because I often need to know what I did in a certain situation.

Emily Lewis: Yeah.

Ryan Battles: So I go back not to the code that I wrote, but to the blog post I wrote because in the blog post I usually strip out the specific code, like anything that was not directly related to the solution and just put the bare bone solutions in the blog post, so I'm able to figure it out quicker having it written it down.

Emily Lewis: Nice. Nice. All right, well, let's have you educate our listeners about error pages. Let's just start with the basics, what is an error page?

Ryan Battles: So an error page ... anytime something goes wrong in the handshake between the client and the server, so sometimes that's going to be the server is not working correctly. Sometimes it's you requesting a file that's not there, and sometimes it's you are trying to poke around the site in a

place that you're supposed to be. So there's a variety of reasons why an error might be returned from the server, but typically it's to signify something has gone wrong.

Anna Brown: And what are the standard error pages that a user might see?

Ryan Battles: So you've got your set of redirect errors, which you actually don't see those. The server sends out like a redirect if, for example, you go to a page that has moved, you can tell the server to tell the browser, "This page has moved." And then the URL on the browser will automatically switch over to the new place so if you ever like typed in the URL and it blinks and then it goes somewhere else, that is actually an error of sort. Well, I guess it's not an error, it's just a redirect code.

As far as the actual errors that you're probably going to see, errors are split up in the client errors and server errors. So the client error is that you're most likely to see. They all have status codes next to them.

So the 403 error is a forbidden error. It means you're trying to go somewhere you're not supposed to. A 404 error which is what I gave my talk on last year at EECl is about the file not being found, and with ExpressionEngine, which we'll probably get into, you kind of have to set that up and have a strategy for how you're going to handle that these files are not found.

Then a 408 error happens when the request has timed out, so if you're trying to do something complex on your server and process something to show the page and the server times out, it takes too long, they might see that error.

In all of these, there are default error screens that every server has, so if you ever just see like a black and white with just really big letters that says like 404 error, that's just the server's default.

And then the final one is the most common server error you're going to find is a 500 error. It just says, "Internal Server Error," and the thing that kind of stinks about that is that it's not very specific at all.

Anna Brown: No, it's not. [Laughs]

Ryan Battles: Yeah. [Laughs]

Anna Brown: It's actually the most frustrating one that I find. [Laughs]

Ryan Battles: Yeah, yeah, that's just the hardest to debug.

Anna Brown: And that really is one that hits developers the most it seems, because if a user is seeing that, the developer has not done their job, right?

Ryan Battles: Correct, yeah, there's something wrong with the server, and I usually get those whenever I'm messing around with the .htaccess file.

Anna Brown: Yeah.

Ryan Battles: And I'm doing something clever and all of a sudden my server is broken or something. [Laughs]

Anna Brown: Yeah, or not so clever. [Laughs]

Ryan Battles: Right, exactly. [Laughs]

Emily Lewis: So Ryan, you mentioned, at least with the 404, there does need to be a little bit of setup. But generally speaking with these errors you mentioned, do they just come automatically on the server, or is there something that needs to be set up in order to enable the user to see these errors?

Ryan Battles: All these errors will come, like I said, by default. There is just the white and black simple error message that the server has, and it's not pretty. It's not useful. It's not branded. It's not even informative, and the fact that they highlight or make real large the status code ... I mean, as a user experience person, you know that that's not something you want to do. You don't just want to be shouting error codes at your visitors.

The best thing you can do is guide them along and explain in human terms what's going on here and how they can resolve the issue. So yes, it does come out-of-the-box with these error pages, but it's definitely a great practice to customize these.

Anna Brown: Do you customize all of those? I mean, are you customizing a 500-page?

Ryan Battles: I have not yet customized a 500-page.

Anna Brown: I haven't either.

Ryan Battles: But in kind of doing a little bit of research before this podcast, I came across some pretty interesting articles that people have written about customizing that 500-error page, some pretty cool things that you can do with that. So typically, I do customize the 404 error page. That's a pretty standard one a lot of people do. Beyond that, I haven't, but I think I might start rethinking the 500-error page. Not that my servers ever have that.

Emily Lewis: [Laughs]

Anna Brown: So what did you find in your research, that you can configure it to pull more information so you actually knew where the error was originating from?

Ryan Battles: Well, sadly no.

Anna Brown: Yeah.

Ryan Battles: But what you can do is, and probably the cool thing I saw somebody do is, first of all, they designed it so that at least instead of a black and white screen, there is the branding going on so that you know that you're on the right site, but they had a form on there that said, "Let us know if you see this page or if we can help you out in any way." And it also give some kind of offline contact methods like the phone number in case you want to call the company.

Anna Brown: Okay.

Ryan Battles: I also have found that you don't just want to put it on them to contact you like it's also a good idea to just automatically send yourself an email.

Anna Brown: Yeah.

Ryan Battles: So that you know that the 500-error page is being tripped, and one company I saw, actually they put like an email input box so that if you want to be notified when the site is back online, they'll send you an email.

Anna Brown: Interesting.

Ryan Battles: So just a few interesting ways to customize that 500-page so that they can get in touch with you on another way, I mean, because maybe they're just coming to your site to find your phone number. At least you can give them that.

Anna Brown: So if you are customizing the 500 page to automatically send an email to you if it ever loads, that form is automatically submitting when you load that page, or how are you doing that? How are you configuring that?

Ryan Battles: Well, I'm not sure how much you can do on the server-side level. I probably wouldn't trust anything on the server side level if your server is not working right. But you can, let's say, if you just serve up the basic HTML page, you can have a form on there that's submits to another site that you have like a third-party site somewhere set up.

Anna Brown: [Agrees]

Ryan Battles: And you can also because it's loading front-end code, you could load your JavaScript, and I guess you could probably...

Anna Brown: Do an auto-submit.

Ryan Battles: Yeah, you could do like an Ajax auto-submit or something like that.

Anna Brown: Got it, okay.

Timestamp: 00:19:48

Emily Lewis: So you mentioned that you frequently customize your 404 pages, so let's talk about those just a little bit. It sounds like the goal of these messages is to inform the user that something is happening, but the default messages don't really tell them what's happening. So in terms of customizing your 404 pages, what are you adding to tell your visitors, your users, when they visit that page?

Ryan Battles: Well, again, one of the things that's important to do is reinforce your branding, just so that they can be sure that they are comfortable that yes, you are on the right site, because a lot of times they'll get the 404 page because they came to a bad link from another site or maybe an outdated search result. So if they land on your page, the first thing you want to do is reassure them like, "Hey, you're on the same spot. Don't hit that back button."

Then the best practices I would say are to maybe apologize for the error, really briefly explaining what happened in layman's terms, even explain what might be responsible for this, but again, you don't want to go too copy-heavy because they're not sitting there trying to read all your 404 explanations.

Anna Brown: No, they are not. [Laughs]

Ryan Battles: [Laughs] But probably the most useful thing you can do is explain what to do next, and there are some useful things that you can do with ExpressionEngine, especially because it is a dynamically generated page that you could try to guide them in the right direction.

At the very minimum, give them a link to your home page. Even better it might be to give them a link or to list out your navigation if it's not already in the template that your server. Maybe even if it's a blog, show them some of your most popular blog post that most people are trying to get to. Anyway, you could at least maybe hit that 80% of the people that might be there in the first place.

Anna Brown: That's a good idea. As developers, what do you think we miss or forget to include when we're building out these custom 404 pages?

Ryan Battles: I would say a lot of times, it's just simplicity. As developers, it's hard to find a designer and developer that are equally skilled at both. So a lot of times if the developer is the one building the 404 page, it might be overly technical. I've even heard a lot of people say, "Don't even use the term 404 because it's like what do they care, what do they know about the error 404, you know?"

Anna Brown: Right, so what would you use instead?

Ryan Battles: Well, just say, "Page Not Found," or "Oops, we're going to fire Ronnie." [Laughs]

Emily Lewis: [Laughs]

Ryan Battles: Turn it into a story that...

Anna Brown: Firing Ryan. [Laughs]

Ryan Battles: Yeah, exactly.

Emily Lewis: Well, that brings up something that I've seen often ... blog posts about like the best 404 page messages, many of which are really funny. Do you think that that's a good direction to go in?

Ryan Battles: As long as it handles the important thing of explaining what's going on. Funny is great after they're in on the joke too. [Laughs]

Emily Lewis: [Agrees]

Ryan Battles: So like instead of like disguising it as an entry and having something funny on there, it should be like, "Hey, definitely you're on the wrong page. We're sorry." And if you have like something funny like a photo or something cute, maybe even a video I've seen, as long as you're pretty sure that they are understanding of what's going on there, and they are okay with a little laugh. But yeah, I would never put something funny in there at the expense of confusing the visitor.

Anna Brown: Yeah, that make sense. I've seen one that I loved, and it was a video. I was quite involved though, but with the 404 page, I can't remember the company, but it was sort of a whole

battle scene. There are some people, there are helicopters flying over and army in the fields and they are searching for this 404 that killed the page, or something. It's so incredibly elaborate and well done. I wish I could actually point people to it, but have you seen that one?

Ryan Battles: Yeah, yeah.

Anna Brown: Yeah.

Ryan Battles: I've seen that. I actually played the video in the talk I gave at the EECon.

Anna Brown: Oh, okay, yeah.

Ryan Battles: But I can't think of the name of the company. I will look it up afterwards and we can add it to this.

Emily Lewis: We can post it on the show notes.

Ryan Battles: Yeah.

Anna Brown: It's quite elaborate. [Laughs]

Ryan Battles: Yeah. [Laughs]

Emily Lewis: So you've talked a little bit about customizing in particular the 404 page to give your user a little more context and information. Beyond customizing the actual content the user sees, when we're talking about 404 pages in the EE, what are the other things that we need to think about?

Ryan Battles: Sorry, you're talking about like...

Anna Brown: Really, like how does ExpressionEngine handle these page errors?

Ryan Battles: Got you. Got you.

Anna Brown: Okay.

Ryan Battles: The parse order of an ExpressionEngine page, if you're ever bored and looking for some exciting reading, check out Low's PDF on the parse order of an ExpressionEngine URL.

Anna Brown: Can I just interject and say most of us can't even understand that document. [Laughs]

Ryan Battles: [Laughs]

Emily Lewis: [Laughs]

Ryan Battles: I think at one of the EECl talks, I think it was like Low who started off saying like, "You know, who here understands the parse order?" And EE Hulk tweeted like, "False question, nobody understands parse order." [Laughs]

Emily Lewis: [Laughs]

Anna Brown: [Laughs]

Ryan Battles: But yeah, it's confusing, but it's time well spent as an EE developer to learn what's going on because it also kind of unlocks more of what you can do with an EE template to know what's being parsed when. But a URL parser for ExpressionEngine, first is that there's a whole set of rules about what can happen with EE. A lot of times, somebody can add extra URL segments to a URL and it will still load a page just fine. For example, if you just take your average blog post URL and just add another slash and start adding a bunch of random characters afterwards, it will still generate that page out of the box.

Anna Brown: Is it ignoring those additional segments?

Ryan Battles: Yeah. They'll just ignore them. I mean, they'll be there for you to use as your segment variables.

Anna Brown: Okay.

Ryan Battles: But for the most part, if they're not anything related to pagination or categories, the parser will just leave them alone, which can be a negative thing because somebody can start adding, like they can create a valid URL to your site that will return a 200 status, which earlier when I talked about those status codes, 200 is the code that means everything is okay and that tells Google to go ahead and index this page and all that stuff. So somebody could like type in some more words after

your URL and then start linking to them around the internet, and that sort of URL might start getting picked up by Google.

Anna Brown: You know that can be a problem.

Ryan Battles: Yeah. You're letting other people control the URLs that Google points to.

Anna Brown: [Agrees]

Ryan Battles: So you can see where that can be a spam issue to somebody fooling around with your search results, I'm not sure.

Anna Brown: And is that because ExpressionEngine defaults to the `dynamic="yes"` parameter so it makes some assumptions from these URLs that are being called?

Ryan Battles: Well, it's actually a feature because you can use that power to pass along some variables to use in the template by using the segment variables. So all you want to do in those circumstances is just check links. If you're building out of template, it's going to be your single-entry blog post and you know that it's not going to have pagination and you know that you're not building any other URL parameters off at the end of it, you can just say a segment 3 or whatever the one after the last segment is. You can say like, "If segment 3 is not empty, then send them to the 404 page." It redirects people to the 404.

Anna Brown: Okay.

Ryan Battles: And the manual thing that you have to do within each of your templates is just to check ... basically saying, "I don't want any other URL segments after 3 or after 4. If there are any, just kick them back to the 404 page." Or what some people do is they just strip those out and do a 301 redirect. So if you were on a blog post/blahblahblah, it would just basically chop that off of the URL.

Anna Brown: Oh, interesting. So then what Google would index is really the correct, or the developer's version is what correct there?

Ryan Battles: Correct.

Anna Brown: That's a great idea. I have to interject here. I've recently had an issue with this where I didn't know this was necessarily happening. I didn't have these custom 404 in some of my major templates on a project, and I didn't find this until I actually installed New Relic on this project. And I won't get into what New Relic is, but I installed it and I started to see that some of the pages that we're having major performance problems or the page loads that we're having performance problems were these ones that weren't correct URLs. And because the segments were incorrect for various reasons, those page loads were in the 20- to 40-second time because ExpressionEngine was trying to do its work; it was trying to do the job correctly, but the segments were not valid ones. So it was really ending up loading an enormous amount of data that it didn't need to, or doing large search to try to figure out what to load, and then it eventually was loading something, but it never was valid content. And so I was quickly able to identify that and get these custom 404 redirects in there and I eliminated that issue, and that actually was a huge benefit to my database.

Emily Lewis: Did you use the approach that Ryan just described about saying, "If segment X is not blank, redirect"? Is that what you did?

Anna Brown: I did that in some cases. One place where I couldn't do that was on a category detail page, so I had to run a query that first validated whether it was a valid category, so if that segment was a valid category segment or category URL title, then I would load the page. If not, I would 404 it.

Emily Lewis: So Ryan, I'm curious. Anna describes a situation where she had a site that was triggering some 404 errors that she didn't even realize were happening. Are there some sort of ways to look up logs of 404 errors, or view them from the control panel?

Ryan Battles: You know I don't think that there is an ExpressionEngine plugin right now or add-on, I should say, that does that. I tried searching for one. I Googled it. I looked on Devot:ee. I looked and even asked out on Twitter. I know that there are some for WordPress because those were all the

search results that kept coming up and it seems like, unless I'm missing it, that would be a great add-on for somebody to have.

Anna Brown: It's a nice little opportunity. [Laughs]

Timestamp: 00:29:57

Ryan Battles: Yeah, a little module that you can have a little display in the control panel that shows the 404 errors that are being triggered. There are a few other plugins though. I know that there are some that will email you when somebody hits a 404 page.

Emily Lewis: Oh.

Ryan Battles: So that one will send you an email, not only of the URL that was hit, but the template that triggered it. So a very EE...

Anna Brown: That's an ExpressionEngine add-on?

Ryan Battles: Yeah, yeah. That's from Chad Crowell. That's called Encaf 404 Email.

Anna Brown: I have not heard of that, so that's good to know.

Ryan Battles: Yeah, that's a handy little plugin that will do that.

Anna Brown: So in my case with my templates, it wasn't throwing a 404 error. That was sort of the issue was it wasn't doing that. So how do you track that down unless you are constantly really watching the URLs that are loaded? I mean, if there isn't an error throwing, but there really should be.

Ryan Battles: Yeah. That's a great question. I know that there are server logs. I mean, you ask also like is there some in the server? You can check your server log just by going into the root of your hosting. Typically, there are some sort of log or you'll see a folder that looks like it's related to error logging, but that's just basically a text dump of all this ... everything going on on the server.

Anna Brown: Right.

Ryan Battles: And so you kind of have the find, hit control F and look through for the 404 code.

Anna Brown: Yeah, it could be a little overwhelming.

Ryan Battles: Right, yeah. It's not very intuitive. So yeah, I don't know. I would say in your case, probably something like New Relic or some of these tools. That's why they are there.

Anna Brown: Yeah.

Ryan Battles: There's a hole that needs to be filled.

Anna Brown: Well, obviously, I didn't start the project great. I mean, you're saying that's part of your build process that you make that redirect. I mean, you make some assumptions about what is a correct page load and what isn't, and then you code around that.

Ryan Battles: Yeah. Well, in truth, your problem also sounds like it's coming from the processing power of loading that template.

Anna Brown: It was.

Ryan Battles: Yeah, if you're checking the URL for certain things, every time you do any kind of checking or looking up or comparison, I mean, you're adding a little bit of load. So in those cases, I would say if you're starting to do some fancier or creative things with your programming like it sounds like you were doing is, the best practice is just along the way, test the failing. Like what you're looking to test against, so kind of like make yourself fail by always entering in the wrong data and see what happens and how long it takes to load.

Anna Brown: Yeah. Well, that really make sense. I mean, as developers, I don't know that I go to that place. I mean, that I don't know that I do that enough, so actually I'm going to start making a little concerted effort to make the page load fail. I'm assuming everybody loads it correctly, you know?
[Laughs]

Ryan Battles: Yeah.

Anna Brown: And that it's not my job. Actually, I need to be mitigating those failures, so that's actually very good advice.

Emily Lewis: Now, Ryan, does anything need to be done in the .htaccess file when it comes to 404 pages or other errors?

Ryan Battles: Well, that depends. If you are using the index.php removal tricks that most of us do within ExpressionEngine site and I mean, depending on the technique that you're using, every URL that somebody types in that doesn't directly correlate to a file or folder on your server will go through ExpressionEngine's index.php. So in that case, the majority of your 404s, you can handle right there within ExpressionEngine by using the tools that they have as far as enabling Strict URLs and all that stuff.

Emily Lewis: [Agrees]

Ryan Battles: But outside of that, you can with .htaccess file write down some commands that will say which ... like you can direct a particular HTML file to be loaded when somebody hits a 404 error or a 500 error. So that's where you can build that custom 500 error page just by writing in your .htaccess file when a 500 error has occurred, open up the 500.html.

Emily Lewis: [Agrees]

Anna Brown: So I know based on my experience that sometimes the 404 error page as coded or configured in ExpressionEngine will load, but the header error that's returned is not correct. It's a 200 header. So how can we verify as developers that the correct 404 header message is being displayed or served?

Ryan Battles: Yeah, that's a great question because that's not something you see by default anytime you look at a page.

Anna Brown: No.

Ryan Battles: So there are a couple of methods. Probably the designer-friendly method is just by going to a website that has Google check your header status, one site is sitestatus.net. It's a free service where you can just type in the page and URL and they'll tell you what the status code

returned it. But for people who are more developer-minded, you can use your web inspector. I use Chrome, so it has web inspector, but there's Firebug, and Safari also has a web inspector. But you can go to the, I believe it's the Resources tab. No, it's under Network, and it shows all the files that were loaded with that web page request and their status message.

Emily Lewis: Oh nice.

Ryan Battles: So that's not only useful. Just for the page itself, you can see like, "Hey, my 404 template is actually returning a 200 status." But you can see if there are any other files that are being called like an image or a font; something that's being called from your HTML that's not loading, and maybe you're not noticing it, but that you'll see that now have a 404 error in there. So yeah, you'll see that if everything is configured correctly, if you trigger a 404 page when you open up the web inspector and looked at that tool, it should say, "404."

Anna Brown: Got it. I use Firefox as my main browser, and I use the Web Developer add-on and under the info section header or toolbar, whatever, they have a view response header link there so that actually gives me some neat information too, or additional info.

Emily Lewis: Nice. I want to change tacks just a little bit and talk about EE error messages that aren't really tied to these sort of default server messages that we've been talking about like a 500 or 404. What I'm thinking about are like system messages that get presented to the user if, for example, they submit a form and they haven't completed a required field. EE delivers some standard error messages that go to the user. Have you, Ryan, ever customized those kind of messages. And if so, like what sort of approach did you use or add-ons did you use?

Ryan Battles: I have not customized the messages themselves. I have customized the design of that page simply because there are so many possible error messages that can come back. I just figured whatever, I have trust in the system I guess, and every add-on as well. I know, for example, Solspace's Freeform ... they will return those messages if you are not submitting a required field or if it's not formatted correctly.

Emily Lewis: [Agrees]

Ryan Battles: So just by customizing the design of those pages in ExpressionEngine goes a long way. That's under the Design tab under Message Pages and then you select User Messages. You can customize the design of that, but that, it's not based off of a template, it's just based off of like a dumping in the HTML with a few variables in the right spots there.

Emily Lewis: [Agrees]

Ryan Battles: So without an add-on, you're kind of stuck. Like if you want to reproduce your header, you'd have to manually copy and paste because you can't just include your header template.

Anna Brown: Right.

Ryan Battles: So there are add-ons out there, I believe, that will allow you to customize those error message pages, but typically I'll just keep them pretty simple and again reinforce my branding of the site on those pages, but I'd just leave the message alone.

Anna Brown: On a couple of projects, I had to customize these error messages quite significantly, and I'm a big fan of the Custom System Messages add-on. It allows you to really just completely take control of those system messages in your actual template. And in my case, I was building sort of a custom My Account area where we were pushing a lot of the actions into overlays so I really needed custom fine-tuned control of those error messages so I could point people backwards or point them forwards and get them to where they needed to go. So for anybody out there that really needs that finite control over those messages, take a look at Custom System Message add-on. It's great, and it's really relatively inexpensive.

Emily Lewis: Yeah, I'm looking it now. It looks like it's from Brian Litzinger's Bold Minded. It's \$15. Anna, when you were using that, does it let you embed other templates or use Snippets or Global Variables?

Anna Brown: I don't see why not. You're running everything out of a normal template.

Emily Lewis: Nice.

Anna Brown: And you could write conditionals, so if it's X error, you can have it deliver this copy. If it's this other one, you can have it deliver a unique copy for that error. So you really have amazing control over it. It moves everything into templates.

Emily Lewis: Nice.

Anna Brown: The biggest issue I have with the default system messages in ExpressionEngine is that you don't really have a lot of control over them.

Emily Lewis: [Agrees]

Anna Brown: Sure, we have the design control, but we don't have much more control than that, and so this tool is amazing. This add-on, I really highly recommend it if you have to get into the nitty gritty of those errors.

Emily Lewis: Excellent. Well, before we close up the talk today, Ryan, I just wanted to ask, do you have any just general tips or suggestions for our listeners about handling errors and communicating errors to users?

Ryan Battles: Yeah, just to reiterate what I said about it, I've read the book, *Don't Make Me Think*, by...

Emily Lewis: Steve Krug.

Ryan Battles: Yeah, exactly, and that's a great tool, and in fact, even just remembering the title...

Anna Brown: Yeah.

Ryan Battles: If all you get away from it is just thinking, "Whatever I'm doing, don't me the user think."

Anna Brown: That's probably always, I mean, that's enormously huge advice right there.

Ryan Battles: Yeah.

Anna Brown: We probably don't even need the book... [Laughs]

Ryan Battles: Yeah, and also we talked about EE errors and we said something about SolSpace's Freeform, how you'll get an error message if you forget to submit a required form field. I like to use jQuery whenever possible or just JavaScript to do some front-end error checking if you can before it gets sent to the server.

Timestamp: 00:40:07

Emily Lewis: Right.

Ryan Battles: Because not only do you save them time in going back, submitting a form and then coming back and risking losing data that they may have entered, but it also just gives them that immediate feedback like, "Hey, this doesn't look like an email address. Can you reformat it?"

Anna Brown: Right, and as the developer, you're in control of exactly what they're going to see with the flow and the look and feel of it all.

Ryan Battles: Right. So just always be thinking kind of defensively when it comes to programming and design.

Anna Brown: [Agrees]

Ryan Battles: Just what could go wrong? How can I prevent that?

Anna Brown: Yeah.

Ryan Battles: And building the tool or the site so that errors are minified if possible.

Emily Lewis: So when you're doing that, like jQuery form validation, for example, are you rolling your own, or is there a widget or an add-on or something that you like to turn to?

Ryan Battles: Well, there are two. One is I use the jQuery Validate Plugin, and that's kind of out-of-the-box, super simple, easy, and I think I've used it on almost every site since I've discovered it. It's

incredibly easy to use. Another one that I use that I found ties in well with a lot of EE add-ons ... for example, Solspace's Favorites...

Emily Lewis: [Agrees]

Ryan Battles: I know I'm saying a lot of Solspace ads here, but their Favorites module is cool because when you click that you want to favorite an item, it will basically take you to a template that just says, the message, "Added the favorites." And I mean, it's meant to be used in a way that harnesses Ajax a little bit more than just actually taking them to that template and showing them, "Yeah, we've confirmed this." So there is another plugin called the jQuery Form Plugin that allows you to really easily do Ajax submits.

Anna Brown: Yeah.

Ryan Battles: And then when it will read that page or that success message that you get and then you can just tell it what div to dump that message into.

Anna Brown: Right.

Ryan Battles: So it's nice because they click "Favorite" and then you can change the star to yellow or say, "Successfully added to Favorites," and then have it fade out, so it's kind of a neat way to do some form handling by submitting it via Ajax. And then if there is an error code there, I know some people have used that with kind of in conjunction with the system messages for like logging in and stuff like that. You can actually take those messages and then display them inline.

Emily Lewis: Nice. Well, that's about all the time we have today. Thank you so much, Ryan, for joining us!

Ryan Battles: Thanks for having me!

Anna Brown: And hey, Ryan, where can listeners find you online?

Ryan Battles: Oh, you can find me on my blog at ryanbattles.com. You can also find me [@ryanbattles](https://twitter.com/ryanbattles) on Twitter.

Anna Brown: Great, and we hope to find you on the Stack Exchange site, Ryan. [Laughs]

Ryan Battles: Yeah, I've...

Anna Brown: Talk every morning with coffee. [Laughs]

Ryan Battles: [Laughs] All right.

Emily Lewis: Thanks again, Ryan, and thank you, Anna, so much for filling in as co-host.

Anna Brown: Well, I can't thank you enough for the invite. I'm happy to fill in going forward. I am thrilled to have done this. This is my first time on a radio show.

Emily Lewis: [Laughs]

Anna Brown: So it's awesome. I could do this for a living.

Emily Lewis: [Laughs]

Anna Brown: It might be my new job. [Laughs]

Emily Lewis: Awesome you guys!

[Music starts]

Now, we'd like to thank our sponsors for this podcast, [Entry Analytics](#) and [Pixel & Tonic](#). We also want to thank our partners, [EngineHosting](#), [Devot:ee](#) and [EE Insider](#).

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](#).

Don't forget to tune in to our next episode when nGen Works' Carl Smith is joining us to talk about work, life and finding balance. Be sure to check out our schedule on our site, [ee-podcast.com/schedule](#) for more upcoming topics.

This is Emily Lewis signing off for the unofficial ExpressionEngine Podcast. Cheers!

[Music stops]