

<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

CTRL+CLICK CAST #064 - Build Automation Using Gulp with Dan Tello

[Music]

Lea Alcantara: From [Bright Umbrella](#), this is CTRL+CLICK CAST! We inspect the web for you! Today we are talking about automating front-end and CMS workflows using Gulp with special guest, Dan Tello. I'm your host, Lea Alcantara, and I'm joined by my fab co-host:

Emily Lewis: Emily Lewis!

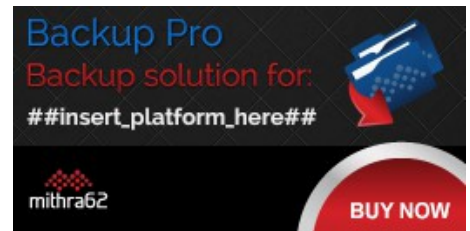
Lea Alcantara: This episode is brought to you by [mithra62](#)'s Backup Pro, a complete backup solution for WordPress, ExpressionEngine 2 and 3, Craft and PrestaShop. We use this one ourselves for some of our client sites. It's insanely customizable and includes automated backup integrity, eight built-in storage locations, console routing. Basically, Backup Pro was built to make disaster recovery as painless as possible. Just visit [backup-pro.com](#) to get started.

[Music ends]

Emily Lewis: I'm pretty excited about today's topic because it's one that I personally haven't made any time for. So by virtue of having an episode dedicated to it, I may finally have a practical understanding of task runners like Gulp. So joining us to talk about these workflow automation tools is Dan Tello, an ex-graphic designer who fell in love with code. Dan is a front-end developer at Viget Labs. Welcome to the show, Dan.

Dan Tello: Hi, thanks for having me.

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



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Sure, yeah. I grew up doing graphic design and web design kind of on my own. I built my first site when I was 12 with Microsoft Word, the premier development tool of my time.

Emily Lewis: [Laughs]

Lea Alcantara: Nice.

Dan Tello: I studied design in school. I loved it for a while, but then when I got into the workplace, I was doing both design and front-end dev and really started rushing through my designs just so I could build.

Lea Alcantara: Interesting.

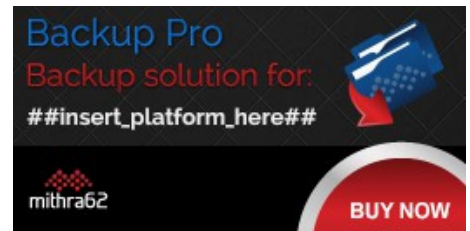
Dan Tello: And that's what I fell in love with and that's what I do full time at Viget today.

Emily Lewis: Yeah, I think that's probably how I would describe my entry. I'm not a formally-trained graphic designer, but I definitely started doing design, and once I saw the code, I was like, "I like that so much more." [Laughs]

Dan Tello: [Laughs]

Emily Lewis: So today we're going to be talking about workflow automation, and specifically Gulp because that's one of the tools you're using a lot these days. But before we get into that, can you talk a little bit about what your workflow used to be when it was more manual, before you found this sort of automation options.

Dan Tello: Sure. So maybe when I first started at Viget, we weren't using task runners and we really didn't start using task runners heavily until maybe three years ago. But we were still using some of the build tools that we're using today just in isolation. When LiveReload came out, that was a big



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

deal to be able to save our CSS and have that reload in a single browser while we're developing. When I first started, we weren't using pre-compilers yet or we were just learning about Sass and scss. Now, everything, I can't imagine my development process without these tools. Nothing is manual anymore.

Everything that I write goes through some sort of transformation or compiler to allow me to write how I want to write and how the future of the web wants to be written, I suppose. And yeah, there's no more waiting for... cut this, I'll tell you what, I'm going to go in a completely direction. Icon fonts are a thing that I used to build sort of manually.

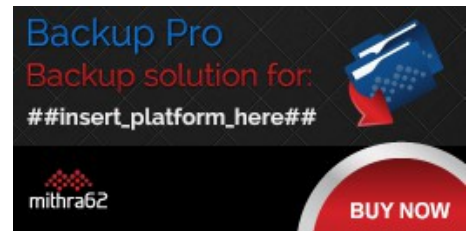
Emily Lewis: [Agrees]

Dan Tello: I'd still use a tool to do it like IcoMoon. I'd upload a bunch of SVGs and then download a font. Now, I can throw all my SVGs into a folder. As soon as they hit that folder, a task automatically runs, generates new SVG sprite for me to use in my markup immediately and all that just updates on the page if I've already referenced it. It's a huge time saver.

Lea Alcantara: Oh wow. Oh yeah, I know, totally. I mean, I feel like when we're thinking about workflow automation, I feel like you get to the point where you get so busy and until that point, you just keep doing everything manually because it's what you know.

Emily Lewis: [Agrees]

Lea Alcantara: So what kind of like prompted you to start or what prompted Viget to look into something to automate, like what was kind of the tipping point where you're like, "Oh, I'm just copy pasting this over again," or you know, that kind of stuff.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

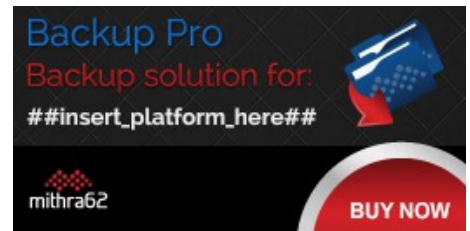
Dan Tello: Sure. So Viget for a while has traditionally been a [Ruby on] Rails shop, and Rails has its own set of build tools. They're kind of bit chips with it called Sprockets. So for a while before Grunt and Gulp and some of the other stuff was out, we had this thing that ship with Rails that would compile our JavaScript. It would combine and minify. It would compile CoffeeScript. You could use Sass or scss, and then there were RubyGems to add some extra layers of processing like Compass that would kind of add vendor prefixes and give you some fun mixins to use, but not all of our projects are Rails, we also work with ExpressionEngine a lot.

We work with Craft or now we work with Craft, and we wanted these tools on other projects. Also, a lot of build tools were coming out that were better and faster than what came with Rails and we wanted to use those on our projects. So we started, a few of my coworkers have been using Grunt, and for JavaScript, they've been using RequireJS which allows you to write a little more modularly instead of using global variables everywhere and just concatenating in one giant file that has to be in order.

Emily Lewis: [Agrees]

Dan Tello: And they'd had some luck with that. They're using it to transform images and output them in different sizes and we were doing some cool stuff. So I sat down on a Friday and worked through the weekends to learn Grunt and RequireJS, and I found it pretty cumbersome and difficult, but I got through it. I'm like, "All right, I know this thing now." I sat down at my work and worked on Monday and an article came out called "Grunt and RequireJS are out. Go open Browserify and I was like, "Come on, I just learned this."

Emily Lewis: [Laughs]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: [Laughs]

Dan Tello: So...

Lea Alcantara: And I know the developer.

Dan Tello: Yeah, that, I had a light week so I was like, “Whatever, I’ll learn this other thing now and see what the difference is.”

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs] Nice.

Dan Tello: And I immediately loved it. I thought it was way easier for me to reason about. I enjoyed building these tasks. It was more straightforward. I really liked working with Node [Node.js], and they use common JS instead of AMD [asynchronous module definition], which are kind of two competing modular JavaScript systems. So since it was relying on Node, you could use a lot of what Node had already done in a lot of packages that were built for Node.

You could use NPM, which is Node.js package manager, which is maybe the single most valuable thing in my workflow today. Say, on an old project, you wanted to add jQuery, you would have gone to jquery.com and downloaded the latest and saved it to a folder and then added a script tag to your page or if you’re fancier, you would have concatenated it manually somehow into your giant JavaScript file. Now, all I have to do is type, “npm install -- jquery” and then require jQuery and whatever file needs it and start using it, and it’s not just jQuery, and I actually rarely use jQuery now because there are so many modules for doing specific things that I need to do.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

If I need Ajax library, I can install SuperAgent. If I need something just for down parsing or like some ReactJS utility or component, all I do is npm install the thing and I can start using it right away, and sharing this with other people is easy having servers can run these same commands and install dependencies on the fly. So yeah, I live NPM and not having to manually manage my dependencies anymore.

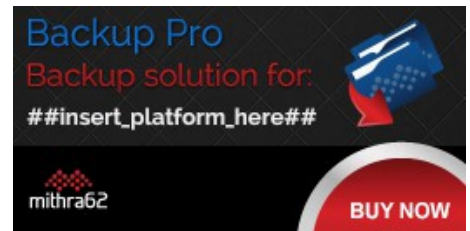
Emily Lewis: So I definitely want to talk more about those specifics, but before we get into that, let's take a step back and go a little high level here. So you've used a couple of terms. You said build tools and you've said task runners. Is that one and the same? Is that just phrases to describe these tools that you're using to automate, or is there a difference between task runners and build tools?

Dan Tello: That's a good question. Yeah, I would say they're different. Task runners like Gulp or Grunt or Broccoli [Broccoli.js] or you can use NPM as a task runner or you can use Makefiles, these tools are for stringing together kind of sets of build tools. So in a typical project, I need to compile my JavaScript. I'm writing using ES6 [ECMAScript 6] or ES2015, compiling that with Babel. I am using webpack to use JavaScript modules. I'm compiling my Sass to CSS. I am running optimizations on my images. I am creating my SVG sprites from a folder of SVGs, and then for production, I'm doing things like hashing file names, minifying and compressing files, uglifying them, removing unused code, and tidying everything up to ship.

Emily Lewis: And those are all task runners essentially?

Dan Tello: No, those are all individual, I would say, build tools that can do these specific things. Like I'm using node-sass to compile my Sass...

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: [Agrees]

Dan Tello: I'm using Babel and webpack to compile my JavaScript.

Lea Alcantara: Got you.

Dan Tello: I'm using imagemin to compress my images. I'm using svgstore to create my SVG sprite, and so on. So what task runners do is give me a way to run a series of these things, like every time I start my app, I want to get rid of my old stuff and recompile everything just to make sure I'm starting fresh.

Emily Lewis: [Agrees]

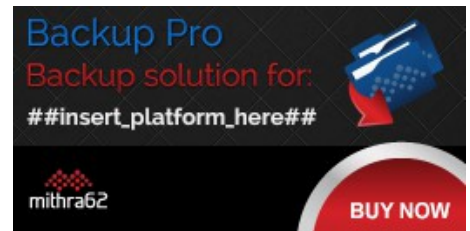
Timestamp: 00:09:57

Dan Tello: I have a watch task that watches all of these folders and then can call these individual tasks anytime I am throwing SVG in that folder, rerun my SVG task that uses that specific build tool to compile the thing I need. Anytime I change my JavaScript, rerun webpack and run it through Babel and then output it and also trigger Browsersync to LiveReload the page that I'm looking at.

Emily Lewis: [Agrees]

Dan Tello: So task runners string together all these really powerful tools into one interface and kind of the composable task that you can run things in parallel, you can run things in sequence, and you can pipe your files through multiple transforms.

Emily Lewis: From what I'm hearing, basically the build tools that you choose to use as a developer are up to you, and then use a task runner like Grunt or Gulp to trigger those at different times based on your own workflow.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Yeah, I think that's...

Emily Lewis: So it's really personalized in a way to how you like to work?

Dan Tello: It can be, and that's kind of what I love about it. I can customize this as much as I want to, to do specific needs of my projects, whereas when we were using what came with Rails, we were tied to whatever they gave us.

Lea Alcantara: Right.

Dan Tello: And if you wanted a new release, you would have to wait for like Rails 5 to come out and they ship like a whole new monolithic, "Now, here's how we do build tools," instead of like all the other stuff is fine, I just want to use webpack, but that's hard to do when it's a closed box.

Emily Lewis: [Agrees]

Lea Alcantara: Right, so that led you or Viget to Grunt and Gulp. I'm still not clear over the difference between the two, like why would you use one over the other?

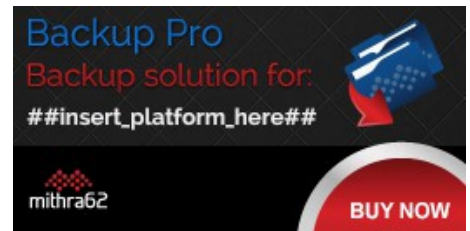
Dan Tello: Sure. That's also a good question. There are a couple of reasons for me. One was, like I said, I learned both within a week's time period of each other.

Emily Lewis: [Laughs]

Dan Tello: And I felt really initially confused and not great about using Grunt. It uses large configuration objects and it just isn't super clear how everything is strung together, at least as a newbie coming in and playing with it for a weekend.

Emily Lewis: [Agrees]

Lea Alcantara: Right.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: If you got into it every day, I'm sure it would be fine. But the other reason is one of speed. So with Grunt, every time you transform a file, it saves that file to disk and then it rereads it and it makes another change and then saves it to disk, which can be slow and we had noticed the slowness on some of our projects. Gulp keeps everything in memory. It transforms a file, keeps it in memory and then pipes it to the next task.

Lea Alcantara: Oh.

Dan Tello: It transforms it, pipes it into the next task. It doesn't write until you're completely done messing with that file.

Lea Alcantara: Aha, right.

Emily Lewis: [Agrees]

Lea Alcantara: That seems a lot more efficient for sure.

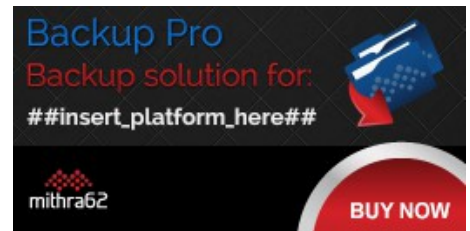
Dan Tello: Yeah, it is, and this is not completely unique to Gulp. You can do this with straight up bash in terminal pipes have always been the thing.

Lea Alcantara: Sure.

Dan Tello: But Gulp takes advantage of this feature and it's nice.

Lea Alcantara: So whenever I read about these two, it always seems like they're used sometimes together. Would that be something that you would do or is it an either/or thing, you choose Grunt or you choose Gulp?

Dan Tello: I think when Gulp first came out, people were doing that because they already had a bunch of Grunt tasks that were serving their purposes just fine.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: Aha.

Dan Tello: And then maybe they saw something they wanted to add or that Gulp could handle better or there is a plugin or maybe Gulp could do it faster and they want to tuck that on.

Emily Lewis: [Agrees]

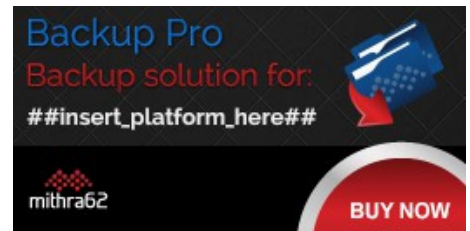
Dan Tello: I've never used them together and I don't see a reason, if you're starting a new project, that you would need to do that.

Lea Alcantara: Sure.

Dan Tello: There's a pretty good plugin parody if you're just looking for plugins, and then as far as flexibility and configuration and kind of writing your own tasks, I think for me Gulp was easier to make it do what I wanted, and yeah.

Emily Lewis: So the Gulp Starter setup that you have on GitHub, is that basically what you were talking about earlier about how you have all of these different things set up for your icon fonts, for your JavaScript, for your Sass or whatever your pre-compiled CSS is? Is that just sort of you've put it together and documented it so that, I'm guessing, other people on your team use the same workflow?

Dan Tello: Yeah, exactly. When I was trying to wrap my head around this, I was like, "Okay, what are all the things that I need on almost every project. I'd love to just be able to hit the ground running every time I start up. So I had some time, I put this together. I wrote a blog post about kind of what we're talking about now, my experience at how I got to these build tools in the first place and what I loved about them, and I always learned through example and seeing things put together. So I did one for myself on other projects and then as well as for my team, and we do use this on all sorts of



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

projects, all of our Craft sites, many of our Rails projects. Even our ReactJS projects use a variant of this, depending on who the developer is, like I use it on all my stuff, but it's not the way that we do it.

Emily Lewis: Well, I guess since these are build tools, it kind of ultimately doesn't matter what the process is as long as you get to the final assets that the project needs.

Dan Tello: Yeah, and I would say we all use the same build tools. We just use different methods of running those tasks in those tools. Some of my coworkers use Make, which is kind of like basically just bash scripting. If you're on a smaller project or you're building like just a JavaScript module for NPM or something, you really don't need a complex task runner. You can use NPM itself as a task runner.

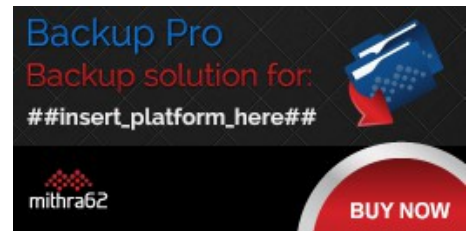
Emily Lewis: [Agrees]

Dan Tello: So if you're not familiar with how Node modules work in NPM, there is a package.json file that lists all of the dependencies that you've installed and their version numbers. There's also a script object and anything you put inside that script object, you can run via `npm run name of your script` on the command line, and from there you can call the build tools command line interface directly. So if you are running node-sass, you could use the terminal command to like compile your node-sass from that NPM script, so it can do `npm run sass` and that would run like `node-sass path to my files and output destination`.

Emily Lewis: [Agrees]

Dan Tello: And it would just be a shortcut for doing that on the command line.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: And then you can string those together with a && to make them run in sequence and now you can start composing tasks. You can use a single & and it will make them run at the same time. So those are some cool stuff you can do and some great articles on how to do it.

Emily Lewis: So I want to ask a little bit more about the command line and the syntax in a second, but I'm curious, a couple of things that you've mentioned really stand out to me as pros. It sounds like you're building faster. You're not only like while you're building, but when you're starting up a project. It's doing things in a way that suits the way you, as an independent developer, like to work but it also works as part of a team.

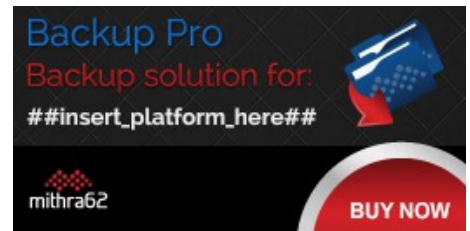
I'm curious though, with all of these tools that you're using, is there more – I'm not sure of the right term – like maintenance, like making sure that whatever package you're using or whatever tool you're using is up to date and maintained and keeping track of any changes within that tool's code base?

Lea Alcantara: And then on top of that, is your team also keeping up to date, like how does that work?

Dan Tello: Yeah, that's a really good question and those things do become issues. Let's see, so far all of the tasks that we've been talking about are JavaScript modules on NPM.

Lea Alcantara: Right.

Dan Tello: So webpack, node-sass, Babel, like all of these things I'm doing, npm install, the tool that I need. So you have this package.json file with all of your things in it, and there are version numbers on all those, and by default, when you npm install -- save something, it will put a little caret next to that version, and that means I think roughly compatible with the minor version, and you can run into



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

issues sometimes with that where someone runs npm install and they're getting a slightly different version than what you have on your machine.

Lea Alcantara: [Agrees]

Dan Tello: So we have started either removing all of those carets and specifying specific versions that we know work on our setup and will work on the server when we deploy it, or there's also a command called npm shrinkwrap, which kind of creates a version of the package.json file, but it's got all of the dependencies specifically locked down and the child dependencies, so the dependencies of the dependencies are frozen at a specific version number, and then you can be completely sure that everyone has the exact same code base.

Lea Alcantara: The same...

Dan Tello: And then throughout the project, we like to keep up to date. If there's a new version of React or something, we want to upgrade and sometimes there are hiccups there, but...

Emily Lewis: So you'll do that in the middle of a project?

Dan Tello: Yeah, yeah, we will. Some new bug fix will come out or a new feature that we really like and want to use, we'll upgrade and that's fine whoever I'm sharing that project with, like the package.json file is committed, they run npm install and everything updates and so long as I haven't broken anything in that process, it runs smoothly. Never just blindly update everything to the latest version because things will break.

Lea Alcantara: Sure.

Dan Tello: But if you kind of stay on top of it.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Timestamp: 00:19:55

Emily Lewis: So you feel like the tradeoff of that little extra taking care of that and making sure everything is up to date and everyone is using the same thing is balanced by the benefits you get?

Dan Tello: Yeah, and I really don't feel like it's much overhead. You don't have to update your dependencies. If what you started working with still works, there's no reason to mess with the versions and update.

Emily Lewis: Right.

Dan Tello: Sometimes developers will release like a warning that says, "Hey, this version is deprecated. You really should update," and that will show up in the console when you run install, and then you can make a decision, "Okay, maybe I'll look into this one thing because the developer thinks it's important."

Emily Lewis: Okay, so before I had you go into that, you were sort of going into a little bit of the commands that you might use, and do you work a 100% in the terminal or do you do anything in some sort of client GUI application or something?

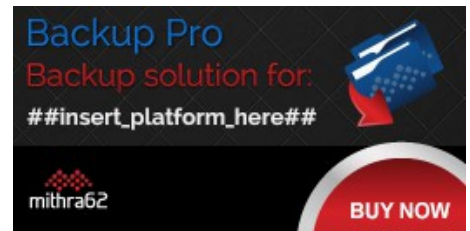
Dan Tello: Yeah, I'm a 100% in the terminal. Being a designer, my terminal is very colorful. [Laughs]

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Dan Tello: It's not like a green monotone screen.

Emily Lewis: Mr. Yuck colors. [Laughs]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: [Laughs] Yeah. So I think that takes away some of the fears some people have of us like just text put onto the screen that you don't know what's going. That was an adjustment when I came to Viget. I had never used a terminal before starting this job, but very quickly I came to love it and for most of the stuff, at least for installing things, you're going to have to run a simple command.

Emily Lewis: Even just looking at your GitHub entry for Gulp Starter, the syntax in the terminal looks pretty straightforward, but do you know of any applications that could be used for more designer-friendly interface?

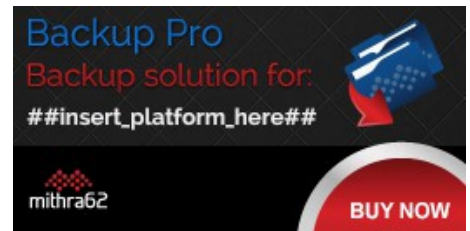
Dan Tello: Yeah, I know of at least one. I believe it's called CodeKit, and one of the other front-end devs on the team have used that at his last job on production things and enjoyed it. But again, using something that's kind of packaged up, I feel you lose some flexibility, like if you want to add something to it, you'd have to hope that they decide to ship that new feature versus just writing a few lines of code and now you've got it.

Lea Alcantara: Sure.

Emily Lewis: Yeah. I feel like that's true with everything web. You really should understand the foundation before you use the stuff that makes it easier. We actually used CodeKit, or I used CodeKit, but I just use it to compile and minify my Sass and JavaScript, and I never made the mental connection about that being part of build automation, and I don't even know how I would leverage it to do anything more because I'm just doing the bare minimum of what I can with it.

Lea Alcantara: Sure.

Emily Lewis: But I think it's interesting, now I probably should poke around in it a little bit more and see if there's more I can take advantage of.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Yeah, I actually have no idea if there is more to it. I just know that that's a thing. I've never used it myself.

Emily Lewis: Yeah, it's got stuff in it. [Laughs]

Dan Tello: [Laughs]

Emily Lewis: And it's like the best I can describe it. It's got some stuff in it. I just don't really know what that stuff does, but as far as using it to compile and minify my Sass and JavaScript, it couldn't be easier, but I did have to go into terminal to get started because that's basically kind of how you trigger where you've got a project set up and then it picks it up and recognizes it, and so on some level, there's still some working in the terminal, on the command line.

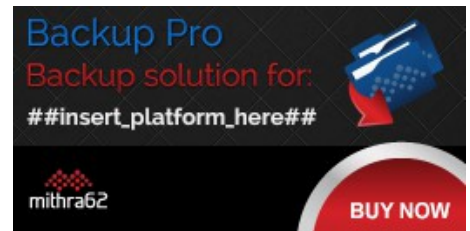
Dan Tello: Yeah, I was a designer and I came from that background. I have no computer science background at all, and it's not a scary thing. It's a lot of fun. It's usually typing `npm start`, it's how my stuff kicks off.

Emily Lewis: [Agrees]

Dan Tello: And then always with Gulp, task configuration is pretty simple JavaScript, nothing to scary.

Lea Alcantara: I like the fact that we've made the distinction that build tools and task runners. It isn't necessarily just like writing things in terminal. Something like CodeKit which we never even considered could be a task runner essentially, at least not on front-end, but that's what it is. It's just kind of switching your mindset that the entire point is to automate things that you used to do by hand.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Yeah, and under the hood, all CodeKit is doing is running. It's probably using the same Node modules that we're installing and running the same commands. It's just doing it for you and puts an interface on top.

Emily Lewis: So let's talk a little bit more in detail about Gulp. So first, is there a particular server environment that it's best suited for that automating using Gulp is best for?

Dan Tello: Not particularly. We've had some issues on really old servers, not being able to install some dependencies. I know the image optimization plugins that I use require some kind of low-level system stuff that wasn't on one of our servers, so our choices were like cut that out and optimize them before and commit those optimized images instead of doing it on the fly, which is probably a better idea anyway or upgrading our server. So you're going to have some issues on old servers. I get a lot of issues on Gulp Starter from people on Windows machines saying things aren't working quite well.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

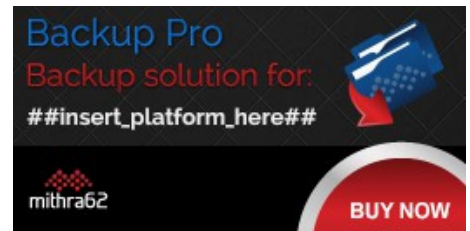
Dan Tello: There are some differences in Windows like slashes are one direction.

Emily Lewis: [Agrees]

Lea Alcantara: Right.

Dan Tello: Forward slashes versus back slashes.

Lea Alcantara: Back slash, right.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: And there are some other little quirks with the file system that not every package is perfectly handled. So I wouldn't recommend a Windows server, but it's not possible. There are plenty of people, at least locally, that have stuff like this running.

Emily Lewis: Yeah, what about just project types. Are there some that are best suited for automation or some that you really just don't need to invest that extra setup?

Dan Tello: I would say every project that I work on I need some sort of automation, and I guess by automation, I mean, some sort of build tool to compile my code. Unless it's like a very small demo, I'm always going to be writing Sass and at bare minimum I need that. There are projects where I don't think you need a full-blown task runner like Gulp and you could get away with just running the commands either manually if they were super small or set up an NPM script to do it and you don't need all the overhead of Gulp-specific plugins or all the files that come with it.

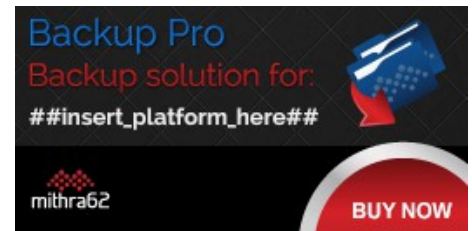
Emily Lewis: [Agrees]

Dan Tello: You can just use that one node module that you need to use directly.

Emily Lewis: So we've talked a lot about kind of the front-end assets they're going to serve a project. Do you utilize any of this as part of a CMS project? Is there any automation that enters creating templates and even many minifying HTML?

Dan Tello: I haven't done this personally, but there's really nothing limiting you from that. This isn't in a CMS, but Gulp Starter uses Swig to compile HTML, which is pretty much identical to Twig if you've used Craft, or I think WordPress has a plugin that uses a Twig syntax as well.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: So yeah, I've compiled HTML with build tools and task runners, and that's been super helpful. It's great for building small static sites. If you go to code.viget.com, that site was compiled with Gulp Starter using Swig and it's just nice to be able to break things up into partials and layouts instead of having to copy paste the same thing everywhere.

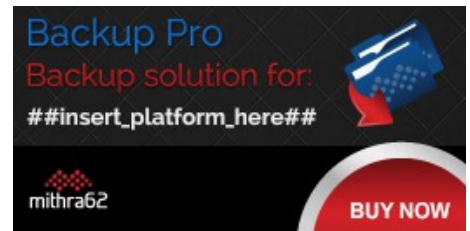
Lea Alcantara: Well, one thing that I discussed with someone on Craft Slack recently, actually I believe it was Andrew Welch. He was chatting with me about creating this generator to just auto create a Craft instance because right now we've been talking a lot about a lot of the front-end things, but, for example, installing something like Craft or any other CMS means normally the manual thing is you open up a MySQL like Sequel Pro or something, then you have to install, like create a database and then you have to go through the entire install process and then you have to install all these plugins and then you have to go through the database plus the files, and then put in all these configurations, but if you know that you need to do a bunch of all those installs right away, apparently you can also automate importing databases through a build tool and a task runner. Have you gone through something like that too, Dan?

Dan Tello: I have not taken it to that level. I know coworkers have, and we've done stuff like that with just simple shell scripts.

Emily Lewis: [Agrees]

Dan Tello: We'll have like a script folder that does those things like copies the database and imports it and does a bunch of that setup stuff, and so all a task runner does is kind of it gives you another way to call those commands.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: And I guess what I like about Gulp is, coming from a front-end background, I know JavaScript and I can write those tasks in Node, which is JavaScript. I can manipulate the file system. I can save and move things around. I can copy them. I can modify them, and then put them somewhere. So it's pretty easy always for me to understand how I would do that. It's pretty easy to google what you don't know too.

Emily Lewis: Yeah.

Dan Tello: But yeah, you can even call like a bash script that you've written in a file somewhere else from Node through a Gulp task. I guess the possibilities are endless the ways you can run things.

Emily Lewis: Yeah, and Lea, what you were describing to me sounds a whole lot like basically like bootstrapping in install, getting it up and running.

Lea Alcantara: Yeah.

Emily Lewis: We talked, gosh, a long, long time ago with Leevi Graham about that topic, about how Newism does that.

Lea Alcantara: Because sometimes, like for example, you already know that there's going to be, for example, a layout's group or something like that if we're talking about ExpressionEngine or..

Emily Lewis: Yeah, you have your standard conventions.

Timestamp: 00:30:01

Lea Alcantara: Yeah.

Emily Lewis: Like we follow the same file naming conventions and template group naming conventions and such.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: Yeah, and our old school way is just copy pasting. [Laughs]

Emily Lewis: [Agrees]

Lea Alcantara: And then manually going through documentation, just like manually, physically going through the control panel and doing it.

Emily Lewis: Yeah, it would be interesting to see if there would be a way to automate, but then again it goes back to that point you were saying earlier. I'm like, "Oh, there's a time to figure that out, or I'll just keep doing it the way I know how to do it." [Laughs]

Dan Tello: Yeah. This might be the topic for another podcast.

Emily Lewis: [Laughs]

Dan Tello: But I know we have an EE master repo and a Craft master repo that I think is like our base setup

Emily Lewis: [Agrees]

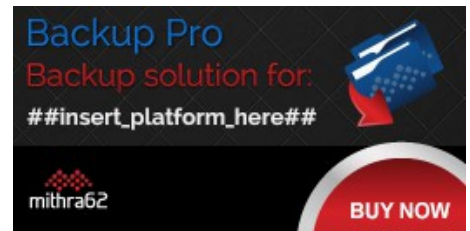
Lea Alcantara: Right.

Dan Tello: And I think there are some scripts in there to do a lot of the initial like permissions changing and all of that.

Lea Alcantara: Yes, yeah.

Dan Tello: You just run a command and you're ready to go.

Lea Alcantara: So when you're saying like run a command, would that be something you do manually with terminal, or is that part of your Gulp Starter?



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

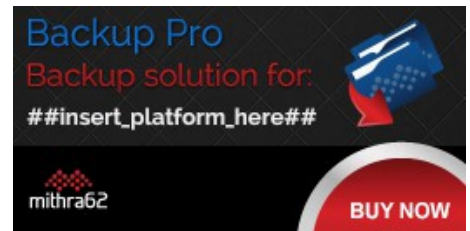
Dan Tello: No, I haven't actually worked on the CMS projects in a while, but I think at the time, at least, those were you just run from the command line like script/the name of the file and then it's just a bunch of terminal commands saved in a file.

Emily Lewis: So maybe the best way to describe it is how you kind of develop your Gulp Starter workflow. When you were looking at things, what were all the different pieces that you had to identify that you needed in order to define an automated workflow and then kind of patch together all of your build tools with Gulp? What were the pieces you had to kind of know in advance for someone who wants to put together their own automation?

Dan Tello: Yeah, so it was helpful coming from Rails because they had thought through all these problems and they had assembled their system for handling these kinds of things. So I could look at all of the features from Sprockets and make sure, at a minimum, I had feature parity with all of those things because I was proposing, "Hey, let's dump Sprockets and start using our own custom asset pipeline."

Emily Lewis: [Agrees]

Dan Tello: So there what you would think, "I need something to process JavaScript, something to process CSS," and that's pretty much the main things that Sprockets would do. It would do some stuff with images as well, but for production, kind of the tricky thing was Rails will hash all of the file names. It will add a unique string to the end of every file anytime it changes. If the file doesn't change, that hash doesn't change and what that allows you do is cache your files on the server forever.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

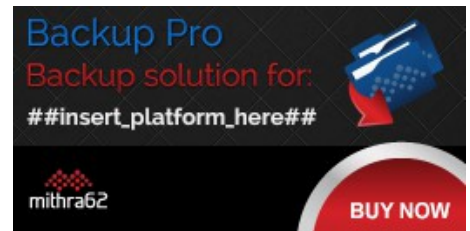
Just tell your server, “Never let go of these. Always cache it. Send down the cache version.” And then you can get away with that because if you make a change to that file and you upload it, it’s a completely different file name. The server isn’t holding onto it and it caches that one forever, and it’s a performance optimization. So I had to recreate that functionality with. There wasn’t one tool that did it all. I have one tool that will add the revisions, but then I have to update. So if you’re changing file names and you’re referencing them in other files, you have to make sure you update the references.

So if I’ve got an image that I’m referencing in CSS, I need to go into my CSS file and replace that file and that happens in kind of step two of this revisioning task that I have, and same with JavaScript, if I’m referencing an image or anything else, it’s mostly around images, I need to update that and then in your templates, in Rails, it’s ERB [Embedded Ruby], in Craft, it’s Twig, and you’re also referencing asset names, the JavaScript files, the CSS file, images, fonts.

Emily Lewis: [Agrees]

Dan Tello: You need to update those references in production too. So for those, you usually need some sort of asset path helper that reads a manifest file. It’s usually a JSON file with the key being the original file name and the value being the new hash file name, and it’s usually some function that you pass it, so we usually call it asset path.

You pass it, the original asset path, and then under the hood, that manifest.json file gets read. It finds the original asset path and then replaces it only in production with the hashed file name. So that was a little bit tricky, but we have in Gulp Starter, if you click on gulpfile.js and then extras, there are documentation and/or asset helpers for Craft and the ones that we built for Rails as well.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Emily Lewis: [Agrees]

Dan Tello: So that's ready to go and we use them on all of our projects of those types.

Emily Lewis: Interesting.

Dan Tello: And the concepts would apply to whatever system you're using. You just read this file and do a string replace.

Emily Lewis: And in terms of version control, where does this fit in? Is any of this part of a team-based version control?

Dan Tello: Yes, so we use GitHub for everything.

Emily Lewis: [Agrees]

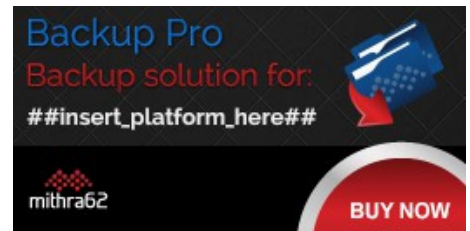
Dan Tello: And since we're compiling files, this kind of problem comes up. If you were to commit all of the compiled files, so you need some changes to JavaScript and I've got my source code and then I've got this compiled minified single file that's output that someone else does work in some source code and they commit something and then when they try to merge, even though there are no conflicts in the original source code, that compiled file will change and there will be conflicts in that.

Emily Lewis: [Agrees]

Lea Alcantara: [Agrees]

Dan Tello: So you get all these unnecessary conflicts you have to resolve, and to resolve them, you basically have to delete the file, recompile it and commit it and it just goes over and over again.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: Right.

Dan Tello: So if at all possible, avoid committing compiled code.

Emily Lewis: [Agrees]

Dan Tello: It makes your diffs and pull requests a lot prettier and it just keeps everything nice and clean, but if your compiled code isn't committed, you have to have a way to serve that. So what we do is, as part of our deploy process and depending on if it's a Rails or a Craft or a standalone like JavaScript app that we have different ways of running these commands, but basically Node is installed on the server, the first thing we do when we push up code is npm install runs on the server, all the dependencies get installed, our production tasks run and build all of those files on the server from the source, and there are all your stuff.

Emily Lewis: Wow!

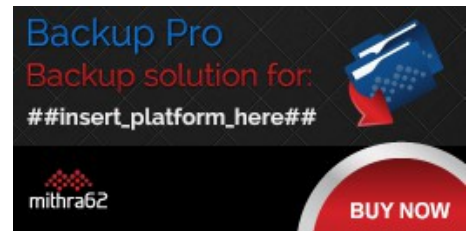
Lea Alcantara: Interesting.

Dan Tello: And it's nice and clean.

Emily Lewis: I think with Lea and I, we haven't encountered those kind of challenges because essentially we have ...

Lea Alcantara: You're the one. [Laughs]

Emily Lewis: I'm the front-end developer, so there isn't another one I have to coordinate with, but as you add more people into the process, the project, that becomes even more of a pain that you could avoid.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Yeah, yeah, you can. Definitely, you can get away with it for a while if you're the only developer on the project, but as soon as you start working with someone else, that conflict will come up.

Emily Lewis: [Agrees]

Dan Tello: I would say if you are the only developer and you know that's going to be the case for a while, it is simpler just to commit the compiled code.

Emily Lewis: [Agrees]

Dan Tello: And maybe even a little safer, you know you can test the code locally. You'd make sure it's all good, and you just push it up. You don't have to do extra server configuration or run tasks on another box.

Emily Lewis: So I think the other thing I was really curious about, and you talked about it a little bit earlier on in the conversation, but can you talk a little bit about how you're handling your images and your image optimizations?

Dan Tello: Yeah, I can. So when I first built Gulp Starter, I thought it would be cool to optimize all of my images as part of this series of tasks. Before, we were manually running like dragging our files into ImageOptim and if there were 24-bit PNGs, we use ImageAlpha because you can convert them to 256-color PNGs, but keep that perfect Alpha channel. It's an awesome tool if you haven't used it.

Lea Alcantara: Oh.

Emily Lewis: Leah is writing it down. [Laughs]

Lea Alcantara: I am literally going there right now. [Laughs]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Oh, it's fantastic. You'll like cut your PNGs in 10 if you can get away with that with few colors, which a lot of times you can. Anyway, there's a Node module and I think it's called imagemin that combines like four different modules, one for PNGs, one for GIFs, one for JPEGs, one for SVGs and just applies this Lossless transformations to all of them.

Emily Lewis: [Agrees]

Dan Tello: It's just kind of a nice thing just to every time ensure if I forgot to run some manual optimization, at least they're going through that.

Emily Lewis: [Agrees]

Dan Tello: In retrospect, it's probably a waste of time to reoptimize those things every time I run my tasks. It's kind of something that would be okay to commit to source. I think if I were to redo it or update it, I would still have that image optimization task, but it would be something that you would run manually every once in a while and you'd just save your optimized images to your repo and that's what gets served up.

Emily Lewis: We have a number of clients and images have become, like no matter how much we advise them on how to get them compact and they don't all need to be enormous. Because they're uploading these to ExpressionEngine or Craft from the control panel, I wonder if there is a way to build a task runner that once something has been pushed through to there goes to these extra efforts to optimize the image even further. That would be interesting.

Timestamp: 00:40:13

Dan Tello: Yeah, I think there are. If it's not for ExpressionEngine, I'm pretty sure there is for Craft some optimization plugins that run these same tools, just a different version of them.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Emily Lewis: I think that's one of those things that at some point a project goes out of your hand as the developer and the client has ownership of their own content, media and everything, and then they start complaining about, "All these pages are slow," and I'm like, "Well, you have fifty 2-megabyte images on the page." [Laughs]

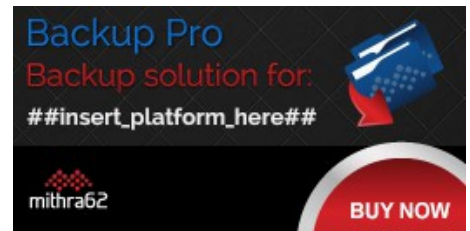
Lea Alcantara: Images, right.

Emily Lewis: "And you shouldn't be doing that." It all depends on how you have that particular feature built into the control panel as to what you're doing in terms of limiting file size, but taking it even further through some of these tools might be an interesting way to help a client who can't seem to help themselves.

Dan Tello: Absolutely.

Lea Alcantara: So we've talked a lot about all the things that automation and Gulp can do. So what are your favorite Gulp resources if I want to dive right in and learn how to do this and automate things right away?

Dan Tello: Oh, that's a good question. The Gulp.js documentation is a great place to start. They've got good examples and good resources. If I can promote my own thing, Gulp Starter, I think it's pretty helpful. People seem to think it's helpful. I'm actively in there answering issues and looking at pull requests if you have something to add with those examples of the things that I need to use on every single project, and there are some great articles, which I forget the names of, but if you google using NPM as a build tool, there's a fantastic article on that, if you want a little more lightweight solution to use modules directly.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Oh, that's one thing I failed to mention that I did want to say. The disadvantage of using a build tool that relies on plugins like there are a lot of Gulp and Grunt plugins to do things, and these are plugins are just wrappers around the modules themselves that are doing the work, if the plugin developers aren't constantly updating and keeping their plugin up to date with the latest version, you can get behind, and if there's some new feature or bug fix that you want to use and need today, you're kind of at the mercy of the plugin developer to update their thing, whereas if you're using the modules directly, you always have access to the latest, and that can be useful.

Emily Lewis: Why would someone choose a plugin over just using the modules themselves?

Dan Tello: Well, to get something into a Gulp stream, which is what kind of like Gulp sucks in a bunch of files, keeps them in memory and then pipes them from one transform to the other, they have to be kind of in a certain format and the plugin just wraps that original module, puts it into the format that Gulp needs to ingest it and then carries on. You can write Gulp task that do use modules directly.

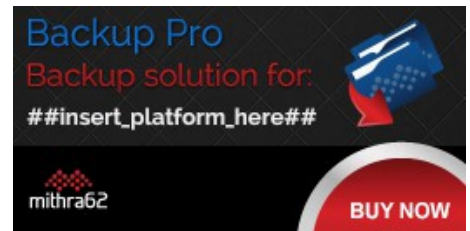
Emily Lewis: [Agrees]

Dan Tello: But you just wouldn't be able to stream together and you wouldn't necessarily know that this task is done before starting the next one.

Emily Lewis: Oh.

Dan Tello: So yeah, it's a little bit of both, but...

Emily Lewis: And as we wrap up, what's your final bit of advice to those of us who are looking for automation to make the workflows more efficient?



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: The time it takes to get up on the stuff and learn it is definitely going to be worth it in the long term. So if you feel like you're wasting time getting all this stuff set up, your efforts are not wasted. It will pay off. You're going to love it. It's a ton of fun and you'll get to focus more on what you enjoy doing, which is making cool stuff work.

Emily Lewis: Throughout this discussion, I just keep having a déjà vu moment when we had that episode, Lea, on version control.

Lea Alcantara: [Agrees]

Emily Lewis: And we hadn't started doing it yet, but everything they were saying just started sounding like we needed to look into it.

Lea Alcantara: Yeah.

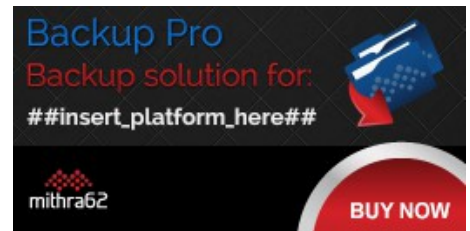
Emily Lewis: And then, honestly, Dan, I feel like the Gulp Starter, having no knowledge of it when you sent me this link last week, I got a sense immediately of what was involved, and I literally have no familiarity with Gulp and Grunt and task runners in general. So I think it does seem like it's one of those things that it might feel a little intimidating initially if you aren't comfortable with some of this, but that it's probably one of those professional life-changing kind of things, because that's what version control was for us once we finally spent the time to figure it out.

Lea Alcantara: Right.

Emily Lewis: And now we can't imagine our workflow without it.

Lea Alcantara: And I also have déjà vu on discussing our starter files.

Emily Lewis: [Agrees]



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: Because we also spent a lot of time in figuring out like, well, we can't keep copy pasting things and/or starting from a blank page each time. It's just too much.

Emily Lewis: [Agrees]

Lea Alcantara: So we needed to create a starter file, and now this is just taking that concept to the next level.

Emily Lewis: Yeah, to automate all the final production codes.

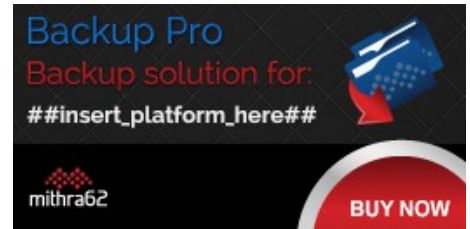
Lea Alcantara: Yes, yes, or even just the initial, just have the initial setup so you can just hit the ground running instead of like manually going to control panel or manually copy pasting things here, there and everywhere.

Emily Lewis: Yeah, and I also really liked, Dan, how much I feel you conveyed that your workflow can really be personalized to what you like to do with assets that you build for every project. I think that's the other thing that I never liked anything that's a little too prescriptive.

Lea Alcantara: Yeah.

Emily Lewis: I think I just get a little like, "Oh, that's not how I like to work. I want to work the way I want to work," and I kind of like that vibe about this.

Dan Tello: Great, absolutely. I should say one disclaimer to Gulp Starter as it started off as a very basic kind of intro repo, and since then some of the tasks have gotten more complex and making it into a system that's configurable for multiple platforms like it has to work on Craft or Rails or standalone, there are some abstractions in there that like I moved a bunch of stuff out into a configuration file to let you change where stuff compiles to more easily so you're not in the individual



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

tasks so much. So, if you actually dig into it, it might look a little more complex than it needs to be, and some of that is just because it needs to be flexible.

Lea Alcantara: Right.

Dan Tello: But it is something that I feel is ready to go. If you just want to start somewhere and see what you can do, that's a good thing to play with or use in production.

Lea Alcantara: Very cool. Very cool. So before we finish up, we've got our Rapid Fire Ten Questions, so our listeners can get to know you a bit better.

Dan Tello: [Laughs]

Lea Alcantara: Are you ready, Dan?

Dan Tello: Sure.

Emily Lewis: [Laughs]

Lea Alcantara: Okay, first question, morning person or a night owl?

Dan Tello: Night owl until I had kids and now I'm forced to be a morning person.

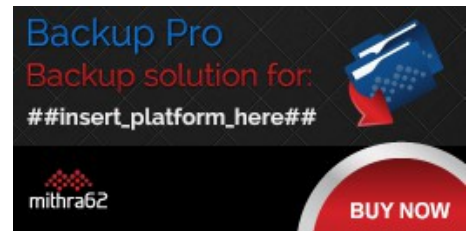
Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Dan Tello: And I get tired really early.

Emily Lewis: What's one of your guilty pleasures?

Dan Tello: *Star Trek* and scotch.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Emily Lewis: [Laughs]

Lea Alcantara: Nice. It's an interesting combination. [Laughs]

Dan Tello: My favorite combination.

Lea Alcantara: What software could you not live without?

Dan Tello: NPM.

Emily Lewis: [Laughs]

Dan Tello: And GitHub.

Emily Lewis: What profession other than your own would you like to try?

Dan Tello: I would like building things. Honestly, I feel like I'd have fun being a contractor or I'd love to be a musician. I never learned an instrument and I've always were good at that.

Lea Alcantara: What profession would you not like to try?

Dan Tello: Accounting.

Emily Lewis: [Laughs]

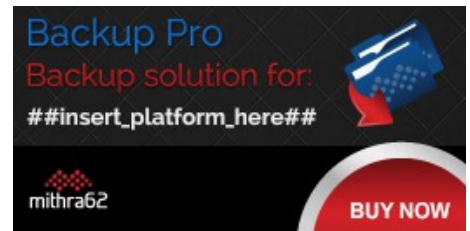
Lea Alcantara: [Laughs]

Emily Lewis: So many people say that. [Laughs]

Lea Alcantara: Totally.

Dan Tello: I'm sorry, accounting is maybe... yeah. [Laughs]

Emily Lewis: If you could take us to one restaurant in your town, where will we go?



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Dan Tello: Oh, there are so many. Well, I guess I'll say the first one that came to mind is called Blanca's. It's a Salvadorian place and they have the best pupusas like cheap and greasy and delicious.

Emily Lewis: [Laughs]

Lea Alcantara: [Laughs]

Dan Tello: And I love them.

Lea Alcantara: Amazing. If you can meet someone famous, living or dead, who would it be?

Dan Tello: Oh, I was recently asked this, so now I know, C.S. Lewis is one of my heroes.

Emily Lewis: Oh.

Lea Alcantara: Ah.

Dan Tello: I'd love to have a pint with him in a pub in Oxford and smoke a pipe.

Emily Lewis: [Laughs]

Lea Alcantara: Nice.

Emily Lewis: If you could have a super power, what would it be?

Dan Tello: Usually, flying, I think. I would love to fly.

Lea Alcantara: What is your favorite band or musician?

Dan Tello: Oh, so many. I'll say the three that have the most physical space in my record collection, Radiohead, Sufjan Stevens, and Sigur Rós are all taking up large portions of my shelf.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Emily Lewis: All right, last question, pancakes or waffles?

Dan Tello: Neh.

Lea Alcantara: [Laughs]

Emily Lewis: [Laughs]

Dan Tello: [Laughs]

Lea Alcantara: I like that. I like that answer. [Laughs]

Dan Tello: I'm more of a sausage, bacon, eggs and more bacon kind of person.

Emily Lewis: Savory. [Laughs]

Lea Alcantara: Savory breakfast.

Dan Tello: Yeah.

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

Dan Tello: Thanks. I had a great time.

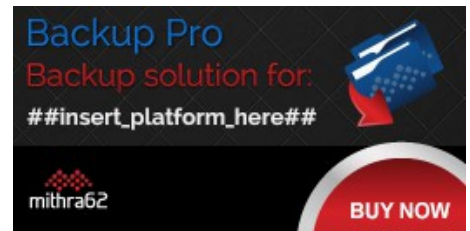
Emily Lewis: In case our listeners want to follow up with you, where can they find you online?

Dan Tello: I am on GitHub as [greypants](#). I'm on Twitter [@dantello5](#), and you can find me at [Viget](#). If you go to our website, I'm there and there's probably more contact info that way.

[Music starts]

Emily Lewis: Awesome. This is a really great discussion, Dan. Thank you so much.

Dan Tello: No problem, anytime.



<http://ctrlclickcast.com/episodes/build-automation-using-gulp>

Lea Alcantara: CTRL+CLICK is produced by [Bright Umbrella](#), a web services agency obsessed with happy clients. Today's podcast would not be possible without the support of this episode's sponsor! Thank you, [Backup Pro](#)!

Emily Lewis: We'd also like to thank our partners: [Arcustech](#) and [Devot:ee](#).

Lea Alcantara: And thanks to our listeners for tuning in! If you want to know more about CTRL+CLICK, make sure you follow us on Twitter [@ctrlclickcast](#) or visit our website, [ctrlclickcast.com](#). And if you liked this episode, please give us a review on [iTunes](#), [Stitcher](#) or both! And if you really liked this episode, consider donating to the show. Links are in our show notes and on our site.

Emily Lewis: Don't forget to tune in to our next episode when our friend, Aaron Gustafson, returns to the show to update us on progressive enhancement. Be sure to check out our schedule on our site, [ctrlclickcast.com/schedule](#) for other upcoming topics.

Lea Alcantara: This is Lea Alcantara ...

Emily Lewis: And Emily Lewis ...

Lea Alcantara: Signing off for CTRL+CLICK CAST. See you next time!

Emily Lewis: Cheers!

[Music stops]

Timestamp: *00:50:42*