## CTRL+CLICK CAST #069 - Connecting to 3rd Party APIs with Mitchell Kimbrough

[Music]

**Lea Alcantara**:    From Bright Umbrella, this is CTRL+CLICK CAST!  We inspect the web for you!  Today we are talking about connecting to 3<sup>rd</sup>-party APIs with Mitchell Kimbrough.  I'm your host, Lea Alcantara, and I'm joined by my fab co-host:
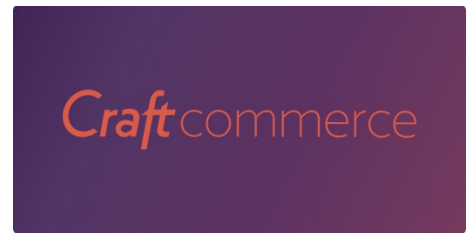
**Emily Lewis**:  Emily Lewis!

**Lea Alcantara**:   This episode is brought to you by Craft Commerce, a brand new e-commerce platform for Craft CMS.  If you're a web shop that likes to create custom-tailored websites for your clients, you're going to love Craft Commerce.  It's extremely flexible, leaving all the product modeling and front-end development up to you, and it's got a simple and intuitive back end for content managers.  To learn more and download a free trial, head over to craftcommerce.com.

[Music ends]

**Emily Lewis**:  Before we get to today's episode, I wanted to remind our listeners that we have a donate link on our site, so if you love CTRL+CLICK and have a little spending money, consider donating to help us keep the show going.  A dollar or five dollars, whatever you can spare, will help us continue delivering great content, high-quality audio and transcripts for each and every episode.

Now, to today's topic... Mitchell Kimbrough returns to the show to discuss a necessary topic in today's integrated web, connecting with 3<sup>rd</sup>-party applications.  Our EE listeners probably know Mitchell through his company, Solspace, which makes some top ExpressionEngine add-ons in addition to providing custom EE dev.  For listeners who don't know Mitchell, he graduated from Stanford with

degrees in Philosophy and Religious Studies before working for a tech startup, which taught him how not to run a company and led him to creating Solspace, a sustainable tech company operating with integrity and a dedication to excellence.

Welcome back to the show, Mitchell.

**Mitchell Kimbrough**:  Hey.

**Lea Alcantara**:  [Laughs]

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  Hi Mitchell.

**Mitchell Kimbrough**:  [Laughs]

**Lea Alcantara**:  Can you tell our listeners a bit more about yourself?

**Mitchell Kimbrough**:  Yeah.  Thanks for having me on the show.  I really appreciate this opportunity to visit with you.  Don't forget Craft.  Solspace is working on Craft stuff nowadays too.
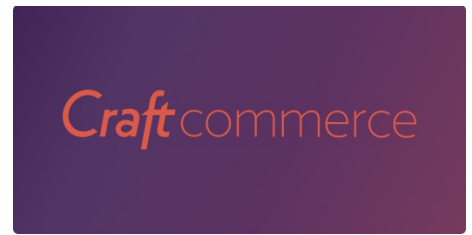
**Emily Lewis**:  Yeah.

**Lea Alcantara**:  Well, absolutely.

**Mitchell Kimbrough**:  And we just released that calendar add-on for Craft and we're really enjoying the response for it.  Yeah, I got a couple of kids.  One of them, the 5-year-old work me up at 2:30 and thought that was a good time to throw a tantrum about getting a drink of water.

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Laughs]

http://ctrlclickcast.com/episodes/connecting-to-3rd-party-apis

**Mitchell Kimbrough**:  It didn't occur to her that her two legs are perfectly functional and she could go get her own drink of water, but yeah, so you are talking to a sleepy daddy today, but I'm going to pretend like the caffeine is working.

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  Yeah, I've got a 5- and a 6-1/2-year-old, two little girls, and thank goodness I have the job I do because it gives me lots of flexible time to just schedule whenever I want to work and whenever I don't want to work, and I get to play with them all the time.  I'm really grateful for that. It's been a good lucky break in my life.

**Emily Lewis**:  Nice.  I'd like being able to dictate my own schedule too without, you know, not the kid factor, but just the flexibility of working when you feel like working and not working when you don't feel like working.  [Laughs]

**Lea Alcantara**:  [Laughs] That's true.

**Mitchell Kimbrough**:  Yeah, it's good.  We make some tradeoffs to get that, by the way.  I'm sure we all know what those things are, but if you're willing to pay those dues, they kind of get organized. You're guys are really organized.

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  Then you can create a pretty good life out of it.

**Emily Lewis**:  So let's talk about one of the things that you do to help you have that great life, which is you at Solspace, one of the things you all specialize in is API integration, and before we talk about that specifically, can you define what is a 3<sup>rd</sup>-party API?

**Mitchell Kimbrough**:  Sure.  A 3<sup>rd</sup>-party API is an open window or door into another system, usually a Cloud-based system, something like Salesforce or HubSpot or NetSuite or MemberSuite or all those sorts of things, but many of the tools out there have pretty good APIs.  Because there has been such a demand for API for the various Cloud systems out there to provide APIs, there have been standards that developed around that.  So yeah, an API is just a way to access the data and business rules that sit inside some other system on the web.

**Emily Lewis**:  And an API integration, that's taking that data and pulling it into a totally different system?

**Mitchell Kimbrough**:  Pushing and pulling, yeah.

**Lea Alcantara**:  Oh, right.

**Mitchell Kimbrough**:  So you may have several different systems doing different types of work for your organization or our clients may who might have a CRM, and CRM (Customer Relationship Management) tool is in charge of managing humans and your relationships with the people who buy your stuff or work on your stuff or what have you, but then you might have another set of tools that manage inventory and you've chosen a certain inventory management system for its best qualities and then you need to glue these two things together because your customers and your inventory need to have a relationship, and if both of those systems offer an API, you can get them to talk to each other, usually you need an intermediary so you'll need some sort of a platform upon which you

http://ctrlclickcast.com/episodes/connecting-to-3rd-party-apis

can write custom code to make a connection to both of those APIs and make the sharing of data and business logic make sense depending on your client's context.

**Lea Alcantara**:  And I think at its simplest, the social media integration is something that clients are always asking, but are confused over what that actually means.
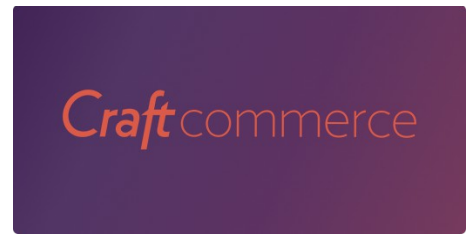
**Mitchell Kimbrough**:  Yeah.

**Lea Alcantara**:  Can you tell our listeners what some examples could be?

**Mitchell Kimbrough**:  Well, one of the products that we've sold for ExpressionEngine for a long time is the Facebook integration tool.  People want to be able to come and log in to an ExpressionEngine website using their Facebook credentials.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And Facebook got pretty aggressive about creating SDKs (Software development kit) and APIs and stuff to allow people to do that.  Basically, back in the day, Facebook made the good choice to treat itself like a social OS, so it's the operating system for your identity across the web, and their API work in that regard was really good, it makes it easy for you to write some code to integrate with whatever sort of web system you've developed your site on that allows someone to use their Facebook credentials to log in.

 That's called single sign-on (SSO), and that's one of the more frequent API integrations that we do, a single sign-on.  We often have people coming to us saying, "All right, we got a client, they have 10,000 users.  Those users are in some other system and their usernames and passwords are in this other system.  When they come to the ExpressionEngine or Craft website, we want to be able to log in with those credentials from the other system, but we want it to be invisible to them.  We just want

them to see the username and password screen that they're accustomed to seeing and enter those credentials. We want you to make a call out to that other API, validate those credentials securely and come back and give that person a session in ExpressionEngine or Craft or whatever the case may be."

**Emily Lewis**:  Interesting.  So how come integrations became something that Solspace was focusing in?  Was there just a big need or was this something that your team was personally interested in looking into?

**Mitchell Kimbrough**:  It's a combination of things, like anything at Solspace, we accidentally fell in to most of the stuff that we do.  I've never really had had it for seeing a market looking into the future and sending the ship off in some direction of what the future is.  It's much more natural for me to have a client come and say, "Hey, do you know how to do so and so?  Is it possible for you?  I mean, have you ever built this thing and made this other thing connect to it and make them talk to…"

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  That's how the business was built really.  It's kind of based on listening.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So we had a client come to us and say, "You know, we've got a lot of stuff sitting in Salesforce.  A lot of our business processes are sitting inside Salesforce, and in particular, our entire staff uses Salesforce to interact with the people, our constituents and our data.  They don't know anything about ExpressionEngine and we don't want to have to train them on a whole new system.  Is it possible to connect ExpressionEngine to Salesforce so that you capture some of the data in there and pull it into ExpressionEngine and show it on the website under certain

circumstances following certain business rules?" And I said yes, like I like to say to really difficult questions like that.
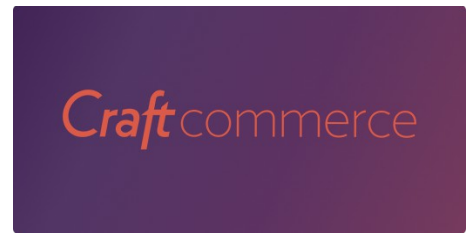
**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  And once I got into it and started doing the work of writing the code that was capable of establishing a secure connection.  In Salesforce's case, they use a REST API.  They also offer a SOAP API, and we can get into those details, but writing all that code, something about it was really fun.  Something about do you know how it feels when you get the website running and it's ready to show an alpha version to a client, and now the pieces have come together in stuff that it's behaving properly or pulling out the database and you could populate different fields with certain kind of content, and you could show your client how to do it.

It feels like that when you finally get the connection to work.  When you run your tests and you have successfully connected to the other system and the other system just say, "Hey, welcome, nice to see you," there's something really fun about that.  So it's a combination of things that we fell into doing this.  I had a client asked if we could do it.  I took the project on to see, it sounded fun.  It sounded like some difficult challenges.  It turned out to be really fun work.

The other thing that I really liked about it was the work of building websites of various levels of complexity on a CMS like ExpressionEngine or Craft or WordPress.  Some of that has gotten commodified.  It's really easy for a client to find someone to build that in any part of the world, and it's kind of pushed my ability to make a living on that down so I needed to work my way up market.  I

needed to something that took my team's skills and expertise and took advantage of it to try to push us further up the market, up the food chain, and this API stuff made a lot of sense in that regard.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Not everybody can write this code.  If you apply yourself, you can learn how to do it, but not everybody has any depth in it, so that was another appealing thing.  If we developed an expertise in this, knowing how much the Cloud, as it were, was emerging as a permanent part of the internet, a permanent concept on the web, and how you need to clue these different pieces of Cloud together, I thought, "All right, we got something here because, first of all, the work is fun.  We've got a demonstrated expertise in being able to write software and ship it, and we can write really difficult, complicated software to do hard things, and there's another API being released every day."

*Timestamp*:  *00:10:05*

**Emily Lewis**:  Right.

**Lea Alcantara**:  Right.

**Mitchell Kimbrough**:  So it started to make sense.  So we kind of fell into it, but I was trying to listen to the marketplace and I heard the marketplace say, "Hey, man, we need you guys to be able to write some API integration stuff and get pretty good at it."

**Emily Lewis**:  Yeah, I know.  I think we've even certainly seen an increase in integrations in terms of not only within our RFPs, but also with existing clients as they expand their businesses and start using tools like Salesforce for like their marketing activities and they want things more tied.  I'm just seeing it more and more.
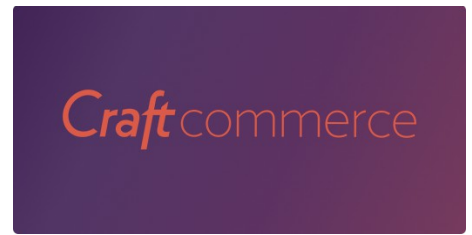
*Produced by*

**Mitchell Kimbrough**:  Yeah, they need these things tied together, and clients are getting more comfortable with the idea that several things are happening.  First of all, they're getting comfortable with the idea that they need to spend thousands and thousands and thousands of dollars a year on some big mothership system that does everything for them.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  They're more comfortable, they're getting more tech savvy, and they're getting just a little bit more relaxed about technology and their engagement with, and so they're willing to say, "Yeah, forget it.  I like this tool for this thing.  I really love how Craft feels to manage my content.  The fact of the matter is Salesforce is our de facto CRM.  We've adopted it.  It's been in place for a few years and we're glued to it for better or worse. In most cases, people really like it, and this other system is managing our inventory or managing our flight data or what have you, and it's fine.  All these other systems, they can do what they specialize in and we can get them to talk to each other or we can find a developer who can do that, and we can have what we want."

**Lea Alcantara**:  So you're listing a lot of examples of API integrations and how you built your business based on client requests.  What are the most requested API integrations, and are some services easier to integrate with?

**Mitchell Kimbrough**:  Yeah, some services are easier to integrate with others.  By far, the most common thing that we see is imagine someone is running a membership-driven website, so the website has content that's restricted to certain members.  You have to pay or the organization to which you belong is a paying member of this other organization, and because of that membership, you have access to content, you have access to special functionality on that site, you have access to communicate with the other people in the system, and that access is gated by something.  It's not

always built into the CMS.  It's usually built into some other system like Salesforce or a HubSpot or a NetSuite or a MemberSuite or there's LinkedIn.  There are a number of other systems.

So the business logic, the business rules of who's allowed to see what and what are they allowed to log into is held in another system, and the interesting thing is sometimes the money part, the commerce component, for some organizations, the membership dues are so expensive in that the person paying them is a large organization.  They just do a PO or a check.  Sometimes, it's an e-commerce thing.  So you've got three-way integration at a minimum going on.  You've got a CRM like Salesforce, you've got an e-commerce tool, and then you have content management system like Craft or ExpressionEngine, and these three things have to talk to each other so that someone getting access to the website, seeing what they're allowed to see or not see, is presented with the right picture, and again, a lot of that is governed in a system that's not really appropriate to be a CMS.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  The management of content is really different than the management of humans and their dues and the groups to which they belong and the privileges that they inherit based on that.  So the answer to your question is a lot of these or most of these are membership-oriented websites.

**Lea Alcantara**:  [Agrees]

**Emily Lewis**:  I know, let's say, for example, ExpressionEngine, you can manage members in ExpressionEngine.  So when you're talking about an integration with a CRM, is ExpressionEngine maintaining member records as well and they're just talking to each other, or…

**Lea Alcantara**:  Or is there duplication at some point?

Produced by

**Bright** UMBRELLA

**Emily Lewis**:  Yeah.

**Mitchell Kimbrough**:  To everybody listening to this podcast, I swear to you I did not pay these people.

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  I did not pay.  This is not a paid advertisement.  You guys are asking all the questions that other developers are going to be asking of someone like me.  To answer your question, the way I like to set this up is I like to tell you, who owns the client, you're the lead developer, you're bringing the client to us and you need us to help out, I tell you that ExpressionEngine or Craft is going to behave the way for you that it always does.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  When you built your templates, when you built up your channels, when you do your member groups, all that stuff is going to feel to you like it always does.  When we integrate with these other systems, we're going to make those integrations talk to Craft or ExpressionEngine in the way that you're accustomed to, and it depends.  The question of, is there a lot of duplicate data?  It's going to depend on the business rules and the plan that we come up with.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Let me give you an example.  In some cases, the API that we're integrating with is a really strong API, like there's serious money behind it, lots of servers, lots of server nodes, CDN (Content Delivery Network), all sorts of stuff is backing that up like in the case of Salesforce.  If

you know the API is that strong and it's fast and reliable and it's going to have a really good uptime, you don't need to duplicate a lot of data in ExpressionEngine.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  Because whatever data you might need about a person from the API, for example, the CRM, you can get that in real time.  As you build the page, you can go and fetch that and pull it into the page.  You don't have to do a lot of data syncing.  There are a lot of cases where I will prefer not to do that though.  You get a performance hit if you're making a bunch of API calls.  Here's the really important thing to consider, almost all of these APIs, they're kind of rate limited, so there's a maximum number of API calls you can make in a 24-hour period or a monthly period after which you have to pay, which is fair.  You're going to use up a lot of server resources.  They have to gate that somehow.

So it's better if you can cache some of that data locally in some way, and the best and cleanest way to do it is just to cache that data as a normal Craft or ExpressionEngine object, whether that be a channel entry or a member records or whatever the case may be.  So yeah, there's sometimes duplicate data, but because we've written software to do the thinking for us, we don't care how much refreshing and resyncing of that data has to take place on a daily basis, we don't care.  As long as we work the bugs out, the system could run for years without any trouble.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So what we tried to do is make all of these tools act the way they should act right out of the box.  So ExpressionEngine and Craft, they should feel to you like they always do, so

the data they were pulling from Salesforce to populate the channel that contains a list of products or certifications or whatever, that's just going to feel to you like a channel with channel entries and a bunch of fields.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And it's my job, behind the scenes, to make sure that data comes in and behaves appropriately and doesn't break anything for you.  So you just build the templates the way you're used to.

**Emily Lewis**:  Cool.  So you mentioned a strong API such as Salesforce.  Can you talk a little bit more about what makes an API strong, and if it's strong in terms of server support and uptime, and that's what I gathered from what you said, does that also mean that it's easier to integrate with?
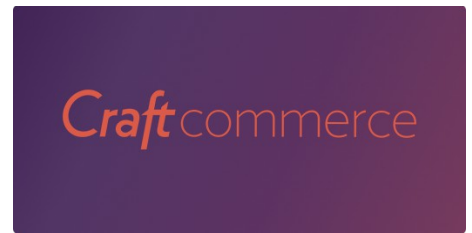
**Mitchell Kimbrough**:  Yeah, the question of strength is there are several points.  I've never really thought about this because I've never talked to anybody about it, but the strength of an API is dependent.  One factor is uptime.  Is it one box?  Is it like a virtual machine somewhere or is it a bunch of server clusters globally?  That's part of the strength.  Another part of the strength is it modern?  I mean, was it built in the 90's?

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  Right.

**Mitchell Kimbrough**:  And it's still kind of being kept glued together by chewing gum and dental floss, like I've seen those.

**Lea Alcantara**:  Like Access databases?  [Laughs]

*Produced by*

**Mitchell Kimbrough**:  [Laughs]

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  I mean, there is a long list, right?  So some of the APIs are old and they're not well maintained, and they don't do simple stuff like exchange their data with JSON, like a JSON string is the common way that APIs talk to each other.  Maybe it's XML.  Maybe it's SOAP.  I mean, there is some really nasty stuff out there, and if the API hasn't been updated, it's nothing I could fix, I can't do anything about it.  If my client is saying, "Look, we're not going to move to another system, this is the one, can you do it or not," I don't consider that as a strong API because it's going to require a lot of extra work and pain and suffering on my part, and the client is going to have to pay.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  There was one job we did about two months ago where, and we'll go into how the process works, but I was in the discovery phase on this job.  This is the phase where I come in and I say, "I don't even know if I can make a connection with this API.  I don't even know if it speaks English, so we're going to do a little discovery.  I'll try to write a proof of concept scripts and make sure I could connect."  When I did that, I started to notice that the API was not well put together, not very well documented, the kind of methods that were available to do queries and writes and reads and stuff were pretty thin, and so I stopped.  [Laughs]

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  I stopped and I kind of hit the big red emergency button.  I said, "This API is garbage.  Are you sure you guys need to use this system?  Can you use something more

contemporary?" And they're like, "No, we've got 20,000 members in here who's not going anywhere. Can you make it happen?"

**Emily Lewis**:  Wow!

**Mitchell Kimbrough**:  The strength of an API is how many machines are supporting it, how big is the Cloud?  The other part is, is it modern?  Is it old?  Is it a REST API?  Is it SOAP?  Is it sending JSON? Does it adopt some of the API standards for create, read, update, delete or is it kind of stupid?  There are all kinds out there.

**Emily Lewis**:  And when you were mentioning like REST, JSON, SOAP, is JSON the more preferred method these days to be exchanging data?
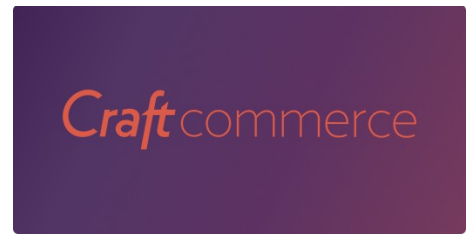
**Mitchell Kimbrough**:  Yeah.

**Emily Lewis**:  Okay.

**Mitchell Kimbrough**:  Yeah, JSON is preferred because it's so easy to parse.  JSON object lands in PHP and you just say, "Hey, PHP, JSON decode or JSON encode.  Turn this thing into an array or an object that I can talk to," and it's done.  There's no parsing layer.  If you get XML, man, that's most of your work.  If you got this XML coming in and you've got to parse through it, and XML has all these standards and a character like the wrong kind of glyph or something that can break the whole document, it's ridiculous, like the stuff we used to think was smart is so dumb.

*Timestamp*:  *00:20:14*

**Emily Lewis**:  [Laughs]

---

*Produced by*

**http://ctrlclickcast.com/episodes/connecting-to-3rd-party-apis**

**Lea Alcantara**:  Yeah, absolutely.  I've run into stuff like that just like a tiny glyph thing whenever I tried to do imports from different systems into Craft or ExpressionEngine, and it's usually like a missing glyph out of like thousands of lines of code.  [Laughs]

**Mitchell Kimbrough**:  Yeah.  Yeah, absolutely.  Because of problems like that, the approach that we take to a lot of this type of work and a lot of these API integrations are actually data migrations, so it's maybe a one-time thing, but never think of it as a one-time thing.  These scripts run much better if you architect them and think about them as, "I should be able to run this thing 10,000 times until I'm satisfied."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  You run it over and over and over again.  If you architect it so that you can just run it over and over again, you will eventually finally work all the different kinks out and the last run is one that you tell your client, "All right, we're done, you can launch."

**Emily Lewis**:  So you've sort of gotten into a little bit about that you have discovery part of your project.  Before you even get to discovery, like during the sales process, what kind of information are you gathering in order to determine whether this is even worth your interest and time and if it's even possible?  What kind of questions do you ask the client?  But what we've found is that clients very rarely even know, and so we have to end up going to the system they're trying to integrate with to try and see what's possible.

**Mitchell Kimbrough**:  Yeah, there's a big gap.  There's a big chasm between your client, their knowledge and expertise and the documentation that tells how this API functions.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  There's not really any intermediary stuff like marketing material that would bridge the gap between the clients where they live and where this API lives, so someone like me has to come in and say, "All right, so I read the documentation and it looks good.  They support all the stuff we need."  I want to test it and make sure it's not BS, but sometimes, actually, every time, the client is feeling pretty uneasy.

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  There's a lot of fear because, like I said, there's a chasm between where they are and this API documentation that looks like total Greek to them.  I mean, it's manna for me, it's milk and honey for me because it tells me everything I need to know.

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  But for them, it's like, "I don't know what this is, it's really terrifying.  Are we sure we have to do this?  Maybe we can punt this three years down the road."

**Emily Lewis**:  Right, or we could just do manual export/imports.  [Laughs]

**Mitchell Kimbrough**:  Yeah, let's just hire an intern from Bangladesh or something, I don't know.

**Lea Alcantara**:  Yeah.

**Mitchell Kimbrough**:  And a lot of those are perfectly decent choices, but really one of the main things I find that I have to overcome is the sense of uneasiness and fear and discomfort that people feel about the prospect of integrating systems together because it sounds scary, and honestly, sometimes it is.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  But I learned from other colleagues in other territory that if you can show a potential client that you have a process, that you have a way, you can just put them in a little car on a train, put them on a track and send it down the track, and you say, "I got you.  I don't know what's coming.  I don't know what to expect.  I don't know how good or bad this is going to be, but I have a process for helping us."  Asking and answer all the questions, it helps a lot.  It helps everybody, and so we do, and I recommend that.

**Lea Alcantara**:  So how can clients better articulate to you, the developer, what their intentions are, so you can accurately ask the right questions, that they're actually asking for the right integration, because if there's like a language barrier there, they might be actually asking for the wrong integration in the right system?

**Mitchell Kimbrough**:  Yeah, that's probably one of the first things I try to do is give the client permission to not know.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Give them the permission to not know how to tell me what they want.  That's one of the most important things.  That's why I make the big bucks.

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  If I come to them and I say, "Look, you're not supposed to know how to do this.  When you take your car into the shop, do you expect to be able to tell the mechanic how to deal with your fuel injection and like which point is off or whatever?  You're not supposed to have to do that, and the same applies with me.  You don't have to know this stuff.  Let's talk more broadly about what

**http://ctrlclickcast.com/episodes/connecting-to-3rd-party-apis**

you want to do, like what do you want to accomplish?  What kind of problems, what kind of systems are given that we know we have to work with no matter what, and what ultimately do you want to do?"

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And I tell them what we're looking for is, what can we do as a first step?  What's the first version of this, and what's the second version, and what's the third and fourth?  So you just start breaking a big problem down into smaller chunks.  Any problem works this way.  You guys know this.

You can break it down into smaller pieces and now they just feel more relaxed and you'd have something to talk about, "Oh, well, the first thing we really want to do is we want to make sure single sign-on works.  Maybe down the road there is some other stuff we could do with this integration, but for now, people just need to be able to log in.  They're so confused right now."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So one of the first things I want to do is make sure they feel like they have permission to not know.  It's my job to probe.  It's my job to ask the questions to determine what they need to do, and one of the most important things is to be really effective at the people skill level.  Saying it in a way that they can hear, "Are you sure you want to do this?  This sounds ridiculous.  What if you did this like 10% instead of this 100% you're telling me?  What happens if you do this little tiny piece and then you have somebody do data entry in this other thing once a month, would you consider that?  Because she could potentially save $10,000 if you do."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Finding some really nice approachable ways where they can hear that kind of thing.  I feel like that's my job, let's say, calling BS on some of this stuff and saying, "Are you sure you want to glue this stuff together?  You might not really want to do this ultimately."

**Emily Lewis**:  When you're having conversations like this, who are you talking to?  Are you talking to marketing people, or are you in front of executives who are the decision makers, but don't have tech savvy?  Is there a typical point of contact?

**Mitchell Kimbrough**:  Yeah, there does seem to be a pattern.  The point of contact is almost always a developer who is the lead for the client.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So they're the designer.  Maybe the design is done and now they're implementing the site on ExpressionEngine or Craft, and now their client has said, "Hey, can you do so and so?  Can you integrate with this other system?"  And they say, "No, but I'll find out if someone can."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  My first point of contact is the developer who says, "I bought your Freeform or whatever and I heard that you guys do integration stuff.  I've got a client who needs something done.  It's on so and so system that you've probably never heard of.  Do you think it's possible?  Here is the documentation.  Can you check it out?"  That's our first point.  I have a couple of conversations with that developer, and this is cheating, by the way.  I'm insulated from crazy nut job clients, right?

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  I'm protected because I can talk to a developer who is most likely a rational human, and I can say, "Do they pay their bill?  Do they show up for meetings?  Do they show up and answer questions?  Do they participate in QA?  Are they going to be a good client?  Are they a red light or green light client?"

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  And I get that scuttlebutt in the very beginning so I'm protected from really bad situations.  Now, the other part of the answer to your question is, who on the client side are we interacting with?
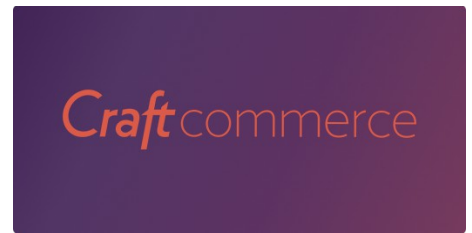
**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And keep in mind, one of the things I try to insist on is that I need a developer relationship with a developer's client.  The nature of the work is sufficiently complicated that the developer shouldn't just try to have me in the background.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  It's just too hard.  The work is too difficult, complicated and the business rules and logic that we have to go through to get everything right, we need to not play game at telephone, so I need to talk to the client, and the billing relationship is there too, so I bill the client.  I don't bill the developer and wait for them to get paid to pay me.  No, thank you.  I go straight to the client.

*Produced by*

**Bright** UMBRELLA

The relationship I have on the client side is often with someone who is kind of like of an office manager type person, if you know that type in an organization. So they're not an executive, they're not a VP of anything, and they're probably not even a director of anything, but they're in charge of a certain set of things, like the website is something they own or the CRM is something they own or some kind of an operations manager person. They don't know any code. They barely know technology. They sometimes do, they usually don't, and they say, "Well, I get these three systems and I've got to get them to work together or one of them has to go in the garbage. Can you tell me if this is possible?" So we go through a process to determine that.

**Lea Alcantara**: [Agrees]

**Emily Lewis**: And in terms of that process, let's say, you've gotten past that process where you've decided this is a project you want to handle. You're working with this other dev and you have the direct relationship with a client. What are the steps that you go through to begin an actual API integration?

**Mitchell Kimbrough**: The steps are initial contact with the developer, and then after that, we usually have some sort of a phone call with their client or some representative, like their main point of contact over on the client side, and we just talk about it like, "Who the hell is this Solspace person, and have you done this before? Can you assure me that you know what you're talking about?" So we have that initial contact, and then I say, "Here is how the process goes. What we want to do is you need to pay me to figure out if this is possible at all. I'm not going to do it for free. I'm going to write you a document that says, 'Here is my initial estimate for how much it will cost to do a discovery and specification process on this project.'"

*Produced by*
Bright
UMBRELLA

And that one-page document, we turn that into a contact, we sign it, it says, "I need five hours or I need ten hours or I need three." It just depends on the system and my guess at the complexity. I'd say, "All right, here's what you're going to get in that discovery. We're going to talk further in more detail about what you want to accomplish, like what the business rules are, where the logic lives, where the data lives, frequency of updates, and is it bidirectional, like are you just reading from this other system into Craft or ExpressionEngine? Are you writing too? Are you doing both?" You'd kind of get some of that terrain laid out.
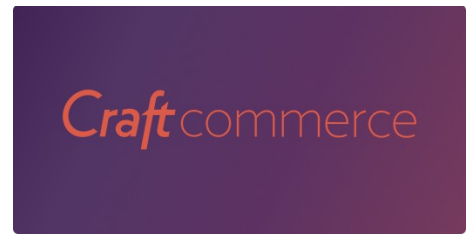
You conducted discovery and they get to sign the contract. They're going to pay their money, pay their deposit, and then we the block of time, the five hours or whatever. That includes flow diagrams, and I'm going to that later, that's a really fun one, but flow diagrams that sort of describe and walk the client through how each of the different points of integration behaves. So if someone logs in, what's that flow? If someone's data gets updated on the ExpressionEngine site, what's the flow of pushing that out to the API? So I capture all those different flows, and the really important thing that we do in these integrations is I write a proof of concept script. I call it "crud script" sometimes, create, read, update, delete.

*Timestamp*: 00:30:23

**Emily Lewis**: [Agrees]

**Mitchell Kimbrough**: So you write just a simple PHP script that is capable of proving that you can do all the stuff that the client is hoping you could do, and this is me proving to them that it can be done, and it's also me proving to myself that it's possible.

**Emily Lewis**: [Agrees]

*Produced by*

**Mitchell Kimbrough**:  This is me testing the API and testing to make sure that it behaves the way the documentation says it will and make sure that I can actually get real credentials from whoever owns them.  Honestly, sometimes somebody's API integrations for larger companies, a lot of politics is happening, and so it takes you forever to get the certificate you need or the credentials you need or the secret key you need or whatever.  You're like, "Okay, I have all the credentials.  Why is there not a connection happening?"  "Well, it must be because you're dumb."  "No, I don't think you opened up my IP address or firewall or whatever."  Just a lot of sense happens.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So the crud script is my opportunity, and this is really good, it's my opportunity to say, "Does this thing behave?  Can I connect to it at all?  Can the human beings involved in giving me the credentials I need, can they show up and do their job or are they going to mess around?"

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And the other thing that happens, you get flow diagrams, you get a crud script or proof of concept script and you get a proposal.  So you get the final proposal on how much I think it will cost to build this entire thing.  In some of the complicated ones, I try to write the versioning plans, "So let's do Version 1 here, get this thing out the door and running and launched and I will do Version 2 three months later or whatever."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  One important note about this proof of concept script, I go back to this over and over again almost on every single integration.  By the time you've got the integration working between the CMS and the CRM or whatever tool you're using, there are a lot of other pieces that got

*Produced by*

involved. There are a lot of other moving parts that are in the way. Three months after you launch, the client comes back and says, "Hey man, this is broken. The code is not working anymore. What happened?" One of the first things you do is you go back to your crud script and just test that sort of baseline thing.

**Emily Lewis**: [Agrees]

**Mitchell Kimbrough**: Is the connection working at all? Have the credentials changed? Because it's so much more simple, it's not ExpressionEngine, it's not anything, but a little PHP script running on a server somewhere and you can validate that and that usually shows you where the problem is, so they're really effective little tools.

**Emily Lewis**: And in this situation with your crud script, is there ever a scenario where you've done this discovery, you've done this initial test script and the client doesn't go with you? Do they have ownership of that script and they can take it and use it some…
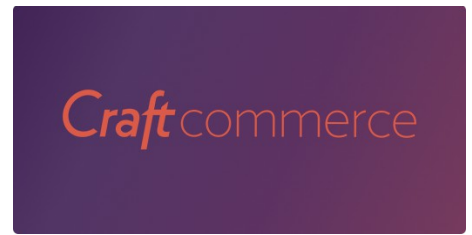
**Mitchell Kimbrough**: Yeah.

**Emily Lewis**: Yeah, okay.

**Mitchell Kimbrough**: Yeah, and they never do.

**Emily Lewis**: [Laughs]

**Mitchell Kimbrough**: But yeah, that's the idea. You see, our proposals are really specifications.

**Emily Lewis**: [Agrees]

**Mitchell Kimbrough**:  I mean, it has dollar figures attached to each thing we're going to build, but they're really specifications on how you would put this whole thing together, and because of that, I think you ought to pay for it.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  But because you paid for it, you own it.  You can take it to somebody in Latvia and have them build it for you for cheap, if you want to.  They're probably going to have to go through the same process I did to take what's out there and stick it in their brain.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  You have to do exercise to get all that complicated stuff into your head, so it's not necessarily that they can just take it and have it built, but yeah, they own this stuff.

**Lea Alcantara**:  I'm a little bit curious about the human- and business-related challenges you kind of alluded to about when you do the test and suddenly the credentials no longer work, it could be somebody fussing around with new API keys and you didn't get a hold of it.  How do you plan for that really?  How do you plan for that, and what are the common challenges besides that example?

**Mitchell Kimbrough**:  This is one of the important things to talk about.  Now, if it's a smaller client, smaller website, smaller API system, not necessarily mission critical, like nobody is going to die if this thing goes offline.  The stakes are lower, so it's not that big of a deal.  On some of the other jobs we've done, I mean, these are multinational companies that are going to rely on this integration, and one of the problems that we have is that, and it's a smart thing to do, you want to be in the development environment while you're building and testing, so you have a development site and you're connected to a sandbox version of whatever system like Salesforce or whatever the single

*Produced by*

sign-on system is or whatever.  So a sandbox is kind of a fake version with fake credentials.  If you get all that stuff to work, it doesn't mean that production version is going to work, which is pretty terrifying honestly.

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  So I usually try to argue to the client, especially in the case where we're doing only a read, we were only going to pull data from the system into the website.  When we're in the situation like that, I say to the client, "Can I just have production credentials right now because that way when we launch, nobody has to freak out and quit their vacation and come back to work and fix the problem.  Let's just do production stuff."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Usually you can't because it's usually bi-directional, you're going to write stuff into the system. And then it wants you to write it in the production system until all the bugs are worked out.  That's perfectly fair.  The problem is that you don't know if the credentials are going to work until launch day.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  We did with Nokia.  We did a single sign-on integration with developer.nokia.com.  Nokia has this single sign-on tool that they used across all their organization's properties.  Of course, they've got eaten by Microsoft, but it no longer really exist...

But back in the day, the single sign-on integration required a bunch of credentials, including a certificate that actually sat on the server that is making the calls, and the people responsible for generating that certificate, it took them a while to get us a valid working certificate for our staging

environment.  By the time it was time to do one for the production environment, there was no way to know if it's going to work until launch day.

And so you're on launch day and you have people from all the different global offices dialed into a conference call doing their little pieces to get ready to launch and switch the site over to a system, namely, ExpressionEngine back then, and you're just kind of crossing your fingers hoping that certificate works.  Well, my developer colleague on the other side over there, at the Nokia offices, we came up with a trick, like we came up with a way to trick the system so that we could test the production credentials a few hours before launch, and thank God we did, because the thing that had an invalid certificate, and the way some of these political dynamics work is the consultant brought in from the outside to do some work is they're usually the one that get blamed.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Okay.

**Mitchell Kimbrough**:  Because there's nobody to protect them.  There's no body looking out for them, namely, us, and so they pointed the finger at us for hours.  They're like, "You guys built garbage shit.  Why isn't this working?  Are you sure your code is correct?"  And they started digging into your code and they're like, "Look at this call, this call doesn't look correct.  What is this in here for?"

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  And so you're regretting all the comments that you forgot to take out, all kind of garbage, and so you have a crud script.  Remember this crud script thing I was telling you about?

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Yeah.

**Mitchell Kimbrough**:  That's where I learned the lesson because I had a crud script running on a server somewhere that's just dumb.  It was just making a little cURL call to the other system, just to see if it was alive or not, and I could prove to that team that everything I had done was correct.  Something was wrong with their certificate, I could prove it, and then the knucklehead had to look in the certificate and he finally admit to like 30 people on the conference call, "Yeah, I made a mistake.  The certificate is wrong."

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  And they generated a new one, and then all system worked.  [Laughs]
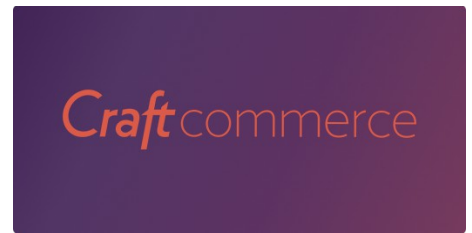
**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So it can get pretty nasty, and that's part of your job if you're doing these integrations.  Not only are you integrating servers and systems and software, but you're integrating humans and various teams together to try to see if you can get them to behave and be nice to each other.  I try to avoid those, by the way.

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  If I can walk away from these big nasty ones and see if coming, those aren't any fun.  I want to write the code.  I don't want to write humans, you know?

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  So let's say, once you've figured out everything during discovery, do you make statements in the contract saying like, "I will get access to production and that they're going to give

you sign-off privileges," like essentially try to mitigate any of these issues right away and have them approve all of that right away.

**Mitchell Kimbrough**:  Yeah, in the contract, I don't really try to foresee everything.  I'm not that organized.  My mind really doesn't work that way.  I'd rather just jump in and be on good terms with the other humans so that we're all working toward the same goal.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Yeah.

**Mitchell Kimbrough**:  But yeah, I certainly have some clauses in the contract, technical requirements like, "This whole contract assumes that I will have access to your ExpressionEngine instance, that I will have access to and you'll create a Salesforce account with their appropriate privileges to make an API connection, and that you will provide me the credentials that I ask for in a timely manner," like that sort of baseline stuff is written into the contract, yeah.

**Emily Lewis**:  And when you do that initial discovery and present a proposal at the end, is your proposal reflecting in terms of I guess the bottom line estimate?  Does it reflect like if it's a client with 30 departments that are distributing globally and you're going to have to work with 60 different people, that the estimate is reflecting a higher cost because it's a higher investment of your energy?

**Mitchell Kimbrough**:  Yeah, yeah.  [Laughs]  When I was young and dumb, it wasn't that way.

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  When I was young and dumb, I was like, "Well, I mean, I got to write so and so lines of code and it's going to take me so and so hours.  I don't care how big you are and how much

*Produced by*

money you have here, I just think it will take this much." No, that was when I was dumb. I'm smarter now, and I know, yeah, I know that most of my work is conference calls at 2 in the morning with some team in Bangalore with people whose accents I can barely penetrate. It's really challenging sometimes, and yeah, you've got to pay.

You've got to pay for all those meetings, and I can't necessarily predict them. Some of these groups insist that you do fixed price bidding, like some of these really large organizations will only engage if you guarantee the price, so you quadruple it or whatever you have to do to protect yourself, you know?

**Emily Lewis**: [Agrees]

**Lea Alcantara**: [Agrees]

**Mitchell Kimbrough**: But yeah, when you're dealing with a larger organization where there are a lot of teams with a bunch of different stakeholders. First of all, I don't do it if I can. The bigger the company, the more terrified I am, because the bigger the company, the more irrational their decisions are. I would rather work for a church, you know?

**Emily Lewis**: [Agrees]

**Mitchell Kimbrough**: We're doing something for one of those mega churches in the South, and there is basically a church-oriented CRM that they use, that they've been using for years. It manages all of their parishioners and so forth, and it's a handful of people and they're all really nice and they want to succeed and they want to get this done and get it out of the way so they can move on to other stuff. These really large organizations have a bunch of different teams and departments and part of their job is to get into each other's way sometimes, I think.

*Timestamp*:  *00:40:07*

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So if I can avoid those, I can.  If it's a really cut and dry and simple thing and I'm dealing with like my point of contact is really good, like they're really good at navigating the political stuff and will protect me from that, yeah, let's get this done.

**Emily Lewis**:  And you mentioned that that might be something you would stay away from.  Is there anything more on the technical side that you might discover during discovery that would make you walk away from a project?

**Mitchell Kimbrough**:  Yeah, one of these jobs a few months ago, I got in to do the crud scripts and I got in to really look at the documentation carefully and I noticed, "Oh, they don't support anything like my client needs.  They support a tenth of what we want to do."  And not only that, but the manner in which they're doing it looks insecure and really old, really old school, it looks like something from the 90's instead of something from just a couple of decades later.  So like I said, I freeze the job.  I said, "Stop.  There's your problem.  I just wanted to let you guys know, this is about to get really expensive and painful.  Are you sure you want to do this?"

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  And in that case, they're totally irrational.  They were like, "Well, we can't throw this system in the garbage.  We're totally tied to this other CRM.  It's a given.  We're not going to move off of it because it's too late.  What can you do?"  And so I went back and I said, "Well, you can have this, this and this, and if you want to do this other thing, we have to find some way to do it, but you can't accomplish it the way we initially planned."  I tried to call it out.

The discovery process, the very first part, I tell them is, "If I can get in there and in ten minutes realize that this thing is impossible, I'm not going to charge you."  I'm looking for the opportunity to get in there and realize there's a deal breaker problem, a show stopper problem, and come back and say, "I don't think it can be done.  Do you want to talk about a Plan B?"

**Lea Alcantara**:  So you've talked about a lot of typical ways APIs can be integrated.  Do you have any odd challenges you've encountered with integration?

**Mitchell Kimbrough**:  Yeah.  [Laughs]

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  With every one of them.  Some of the odd challenges that we faced are performances as an issue pretty frequently.

**Emily Lewis**:  Right.

**Lea Alcantara**:  Oh, right.

**Mitchell Kimbrough**:  Like I was telling you that a lot of these APIs are read limited so you're allowed so many calls or your client has to pay more money, or they have to pay more money to have separate user dedicated to the API connections so you have to try to trick the system so they can save a few bucks.  In a lot of cases, you try to make an argument to the client, "Well, you can pay me $2,000 to work around this problem or you can just pay an extra $50 a month to give yourself more API calls or whatever you need."  So you try to help them do the math that way.

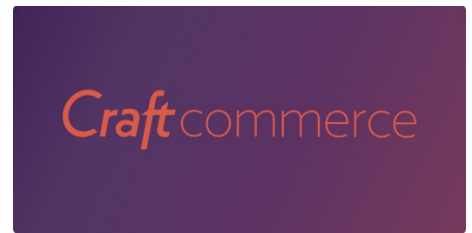**Emily Lewis**:  [Agrees]

*Produced by*

**Mitchell Kimbrough**:  Some of the other things we have to work around are if one system was really bad at reporting errors, like if you've put in the wrong password as you're trying to log in the single sign-on or whatever, the errors that you would get back, if your password was wrong, is really bizarre. It wasn't like a string that said, "Your password is wrong."  It was a bunch of funky headers and some strange strings.  It's like they used an HTML template to push the error through and I'm getting that in machine language that I'm trying to parse it.  It's really peculiar.

If you try to capture that stuff in the beginning, so that you're telling your client, "Well, because of these weird things I encountered when I got in there and played around, ultimately building this for production is going to be more expensive because they didn't carefully build this API."  So yeah, the parsing of the data, the responses that you get back from the API are sometimes weird.  Sometimes the APIs go offline or they're slow.  We had this one, this one happened on the job with NetSuite.  In the development, the client was saying they were integrating Freefrom, hosting Freeform, like contacts and stuff as a lead into a CRM.  In this case, it's NetSuite, and the client was saying, "You know, I hit Submit and it takes forever for the page to finally load and tell me that we're done and I get a thank you message.  Why is it taking so long?"

And so we go back to the crud script and test the crud script and see how it performs doing that same action and we're able to say, "Well, it looks like NetSuite is really slow or the combination of the calls I'm making to NetSuite plus this development server is a crummy $10 a month development server that we're on is the problem.  Let's move to a better server and see what happens or let's call someone over at NetSuite and see if they're limiting us on their sandbox calls versus on the production box calls," which they were.

**Emily Lewis**:  So a lot of what I'm hearing is that in a way, it sounds like every integration you do on some level is unique.  I mean, you've got the unique client business rules, perhaps a unique system, but I'm curious, if there's an aspect to API integration, any integration X is always present and you always have to factor for it regardless of the scope or the client or the systems.

**Mitchell Kimbrough**:  Yeah, I've had module that connects to Salesforce for four or five years now and it's an ExpressionEngine module, and there was an EE 1 version, there's an EE 2 version, and now there's an EE 3 version.  I've never been able to sell it in the store because every engagement, just like you said, every engagement is unique.  The common piece, in that case, is making the connection to Salesforce, but even that is difficult because in the context of Salesforce, they lock it down pretty tight.  It's a pretty secure system.

The process that you have to go through inside your Salesforce account to get all the credentials you need for the API to make a valid connection, that's a pretty long list of instructions and there are a lot of little exceptions and weird stuff.

**Emily Lewis**:  Right.

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  It's just hard enough to eat my lunch if I sold it as a product in the store.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  Like the support burden would completely outweigh whatever I could sell that product for, so I never went in that direction because every time it's a little bit unique.  There are some things that happen pretty regularly over and over again.  It's pretty common that we'll have a contact form like a Freeform form running on the site that posts as a lead type of a record into a CRM,

whether that's NetSuite or Salesforce or whatever. They all have a concept of a lead, like a new business opportunity, and that's pretty consistent, to say a lot.

It's always going to be the case that you have to find some way in your code to abstract out the way that you connect to that other API. If you do this well, you've written the code in such a way that when I try to, I can reuse code I've written for other integrations and I can make a new connection to a totally different system, and if you abstract the code away correctly, then I'm just saying to that code, "Hey, log me in. Here's the username and password, go make it happen."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And then you have this intermediary library that connects to that specific system that handles the login action and returns what I'm expecting, so you're just kind of adopts some modern software development paradigms to do that, and those, that's pretty consistent so you've got to make the connection to the API and how you abstract that away is something I do almost every time.

**Lea Alcantara**:  So considering that these integrations can vary so much even with a thorough discovery, I'm sure you've met some scope creep eventually throughout. I mean, how do you deal with the unexpected? How do you mitigate for those situations?

**Mitchell Kimbrough**:  I try to have some padding in there.

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  Whether the client knows there's padding in the budget or not, almost all these bids are just hourly, "You know, I'm going to bill you what I work and here's what I think I'll work." In some cases, if I see it's pretty predictable, I can offer a fixed price bid and that protects the client from
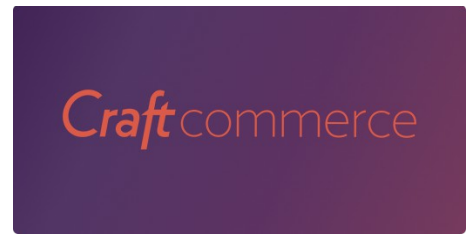
overages, and that's why the process, the planning process, is so critical. You've got to do it no matter what. If you do it really well, you know exactly what it's going to cost, you may as well do fixed bidding. You protect the client and make it easier for them to say yes.

There are definitely overruns and changes and modifications. This is where really effective client management is critical, and that's why you can't play game at telephone. That's why the developer can't be my client. The actual client has to be my client because I have to get them on the phone and say, "What do you mean he wanted to add the ability to also delete records? We never talked about that. That was not in your contract. It was not part of our plan. I see how critical it feels to you right now, but it's not part of the plan."

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  So you have to play bad cop. You have to protect your budget and the client's timeline, the client's budget, and those are client management skills that are really essential in these kinds of jobs because they're so hard, they're so complex that something bad is going to happen every single time, and you can't just be a code monkey. You've got to be a real human who can connect with people and write the code at the same time. Hopefully, that's why there aren't that many people who can do this, like it's sort of a specialty that's kind of a rare set of skills.

**Emily Lewis**:  Well, that's like a perfect segue way into one of our questions that we have about what kind of expertise is needed to do these kinds of integrations, and I'm hearing the code, the technical stuff. I'm hearing the human aspects, but is there an organizational or a planning mindset? Does design enter the picture in any way?

*Produced by*

**Mitchell Kimbrough**:  Everything is designed.  You adhere to that idea.  Anytime you're going to build a thing, you've got design at some point.  Whether you design it as you threw it together or whether you actually planned it out and had some way to represent the thing you were going to build, everything gets designed.  So yeah, these systems get designed.  Most of them are small enough to fit in my head, so I don't need to write all of it down.  All the design doesn't need to happen on paper necessarily, but yeah, you definitely design these systems and the more organized you are the better, but let's be honest, the really organized mind and the mind who's a good designer and the mind who's a good coder and the mind who's really good with humans, they're not necessarily all of the same mind.

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  Yeah, yeah.

**Mitchell Kimbrough**:  So I've had to lower my expectations for my level of organization, but I know which things to organize and which things to just not worry about.  So the set of skills required are, like I mentioned a number of times, the human factor, the ability to tell someone, "I know you think the thing you're asking for is simple, just a simple change to the contract, but you want to add that field, you have no idea how bad that is, and let me explain it to you in a language that make sense to you without talking down to you."  Those are really critical skills.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  You need some planning skills.  You don't necessarily have to be super organized, but you need some way of forcing yourself to see the problems that are coming your way, and that's planning, and you need to have some coding chops, like you need to be able to code well.

In particular, a lot of these APIs offer SDKs, so they offer a software developer kit that you can download in PHP or Node or whatever, and that gets you started and basically does a lot of the work for you, you just need to find some way to connect your system to that SDK and make that SDK behave.  That means you have to get into that SDK and make sense of the bugs that are still in there and be able to fix them, so definitely some coding experience and chops are important.

*Timestamp*:  *00:50:13*

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  QA is really critical too.  The kind of QA that's required to make these systems show their bugs before you launch, that's some seasoning and some expertise that you just develop over time.  The planning piece is really interesting part, and I recently felt like I made a breakthrough on that.  I noticed that I was running flow diagrams.  I believe flow diagrams are really effective way to communicate complicated systems.  They take a really complicated set of things and break it into smaller chunks that can be digested.  I was doing them on the OmniGraffle all these years, and I was just noticing that I never did them.  I never did the work.  I never completed the flow diagrams because I was so itchy to write the code.

**Emily Lewis**:  [Laughs]

**Mitchell Kimbrough**:  I just wanted to go write the code, and let me just open up a BBEdit and start writing some code and see if I can get these APIs to talk to each other.  It was so hard to resist the temptation of getting like halfway into a flow diagram and say, "Ach, I know what I'm doing.  I'm going to write some code.  Forget this, I'm good."

**Emily Lewis**:  [Laughs]

*Produced by*

**Bright**
UMBRELLA

**Mitchell Kimbrough**:  What I do now is I know that if I can make the planning part also include code writing, I'd get it done.  So my little trick, and I think maybe two or three people out there might like this, but not too many, my trick is that I basically built a website.  I built a mini-website that is the flow diagrams.  So I take Bootstrap, Twitter Bootstrap, and I just have sort of a flavor of it that I use that has arrows and stuff and boxes and whatever, and it's HTML, and I use Jekyll, this static website generator, Jekyll's templating tool, to let me do stuff like to simplify some of the things I use over and over again.  So when I'm doing the planning process, I'm actually writing code.  I'm in the HTML writing some code, and I do it because, okay, I'm writing code.  I'm moving my fingers on the way I like to, and I'm actually doing the planning and writing the code at the same time.

**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  When I didn't do that, I never got the planning done.  I never finished.  But now that I'm doing the planning, these projects are going so easy because I'm seeing everything I need to see.  I'm planning, I'm designing.  I'm doing what I'm supposed to be doing, but I trick myself into doing by writing code.  I'm going to write a blog post about this so people can see what I'm talking about.

**Emily Lewis**:  Yeah, it's like you're leveraging one of your strengths too, because I know when I'm working in code, I'm planning at the same time.  There is something about the process that you're seeing the solution form itself in front of your mind, so to be able to leverage that naturally while your mind is working and apply it to an area where maybe you're not as strong, it's brilliant.

**Mitchell Kimbrough**:  Now, it's accidentally brilliant.

**Emily Lewis**:  [Laughs]

*Produced by*

**Bright** UMBRELLA

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  I'm not taking credit for it.  It was just something like a little light bulb went off, "Hey, what if I build a website instead of trying to run all these freaking OmniGraffle diagrams."  And I did it one time, I was like, "Bingo, now we're good.  Now, I can plan, I can build anything now because I will be writing code as I'm thinking about it."
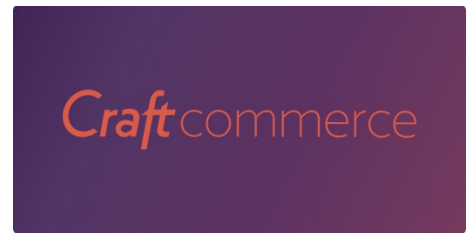
**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  It's really a good insight that you saw there that, yeah, my mind works and starts to architect well when I'm writing lines of code, and so I tricked it into doing it.

**Lea Alcantara**:  So cool, so cool.  Finally, I'd like to know, it seems like such a complicated process, especially because, like you mentioned, that each integration has its own flavor, its own need.  Are there actually any resources to understand 3rd-party APIs and integrations, like what's the best way to learn how to do this?

**Mitchell Kimbrough**:  Well, the best way is to start with some of the more predictable and easy to use APIs.  Facebook is really good.  Facebook, like I mentioned earlier, they put a lot of effort in a few years ago to just penetrate the entire internet and they knew that doing that would be most effectively done if they have a presence on everybody's website, so they integrated themselves into every website.  So they do a number of things that make API integration stuff easy to pick up and learn really good documentation, tons of like message board support from other developers working on stuff.

So starting with Facebook and just giving yourself a challenge of "Can I trick ExpressionEngine into accepting a Facebook identity as a valid user?  Can I figure out how to write that code?"

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  And you'd be starting with an SDK, like they have a PHP SDK you can start with.  We're talking mainly to people who are most likely working in the PHP realm so going in that direction is pretty good.  Salesforce is good.  It's just a little bit more challenging to get up and running because the credentials set of things you need to have before Salesforce will even allow you inside is kind of more tricky.  Facebook is a little bit more friendly that way.
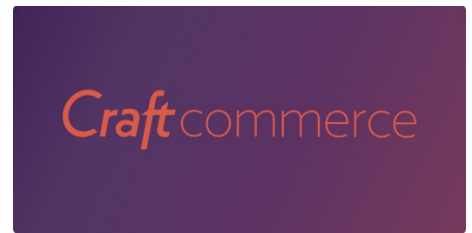
**Lea Alcantara**:  [Agrees]

**Mitchell Kimbrough**:  Another really good one that I like to work with is Stripe, and Stripe is an e-commerce tool, right?

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Oh, right.

**Mitchell Kimbrough**:  And Stripe was built for us.  It was built knowing that we the developers are most likely these days able to win the argument with the client of what system to use, what e-commerce system, what CRM, because we can make arguments from the point of view of how easy is this to maintain over time, how hard is it to just build initially, how does it cost you, that kind of thing.  Stripe is really strong with respect to its support for developers, so the API is totally industry standard.  It's one of the best out there, and it's so easy to get a sandbox version up and running.  It's perfectly designed for developers.  They really were thinking about us when they established themselves.  So I would recommend that one so you can start doing some API integration at an e-commerce system level with Stripe.

*Produced by*

http://ctrlclickcast.com/episodes/connecting-to-3rd-party-apis

**Emily Lewis**:  And I'm wondering if, by chance, you've come across any resources that detail integrations, but from a client perspective, something that you can share to educate and/or help a client understand something?

**Mitchell Kimbrough**:  It's usually conversational, so I'm sure there is material out there.  If I was smart, I'd write some blog posts from that point of view to help people put at ease when they're looking up somebody who can help them develop, like it's a nice little segue way into them making a phone call, having read a blog post that you've written.

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  I should do that.  But really, it's mostly conversational.  I mostly rely on the years of experience talking to clients and making stuff sound simple and easy, when it's not, to get conversational with them and get them to a place where the anxiety is gone from their voice, I like to hear them and talk to them and get them to a comfortable place.  It's scary.  What they're trying to do is pretty scary, and to get them to a point where you take away each of the individual things that they are fearing and get them to a happy place, I think that's the way to go.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Very cool.  Wow!  Thank you so much for being part of the show, and especially really emphasizing the human aspect to all of this, because despite the fact that this is a technical topic, 3rd-party APIs, getting the information you need in order to do the development is very human skill.

**Mitchell Kimbrough**:  Getting the information you need really after you launch, getting a relationship with someone where they feel like they can call you, and sometimes they're embarrassed the thing is

*Produced by*

**Bright** UMBRELLA

breaking and they're like, "I don't know if it's us.  I think I broke it.  I don't' know if I broke it.  How do I know if I broke it?"

**Emily Lewis**:  [Agrees]

**Mitchell Kimbrough**:  If they know they're talking to a human who they already have a friendly relationship with, I can come in and say, "Yeah, you did break it.  Oh, no, you didn't.  My code broke or whatever."  The human factor is critical from start to finish on these jobs.

**Lea Alcantara**:  Yeah, awesome.  Well, thank you, Mitchell.  But before we finish up, we've got our Rapid Fire Ten Questions, so our listeners can get to know you a bit better.

**Mitchell Kimbrough**:  Oh no.

**Lea Alcantara**:  Are you ready?

**Mitchell Kimbrough**:  Okay.

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

**Mitchell Kimbrough**:  Night owl.

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

**Mitchell Kimbrough**:  Coffee.

**Lea Alcantara**:  What software could you not live without?

**Mitchell Kimbrough**:  Oh, BBEdit.

**Emily Lewis**:  [Laughs] What profession other than your own would you like to try?

**Mitchell Kimbrough**:  Schoolteacher.

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

**Mitchell Kimbrough**:  [Laughs] Oh man, there are so many.  I don't want to be a mid-level manager at any large company.  Thank you very much.

**Emily Lewis**:  [Laughs]

**Lea Alcantara**:  [Laughs]

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

**Mitchell Kimbrough**:  Man, we'd go to Taqueria Vallarta, my little burrito place.

**Emily Lewis**:  [Agrees]

**Lea Alcantara**:  Oh, yummy.  If you could meet someone famous, living or dead, who would it be?

**Mitchell Kimbrough**:  You know I'd like to meet Churchill.

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

**Mitchell Kimbrough**:  I would like to be able to like an artist, sort of read minds, so that's good, and sorted on that one, I'm kidding.  Super power, it would be cool to fly.

**Lea Alcantara**:  What is your favorite band or musician?

**Mitchell Kimbrough**:  Lately, I'm liking Miles Davis again and Bob Marley.  I'm really big into Bob Marley, especially if the weather gets warm.

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

**Mitchell Kimbrough**:  Waffles.

*Produced by*

**Lea Alcantara**:  I love it when guests are so decisive.  It's just like, "Of course."

**Mitchell Kimbrough**:  Some of these are obvious, like anybody would answer waffles.  It's not hard.

**Emily Lewis**:  I do not like waffles.  [Laughs]

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  [Laughs]

**Emily Lewis**:  I am a pancake person!

**Mitchell Kimbrough**:  Aren't you the one who lives with three cats and sneezes all the time?

**Emily Lewis**:  [Laughs] Yeah.

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  [Laughs] Yeah.

**Emily Lewis**:  I didn't say I was reasonable.

**Lea Alcantara**:  [Laughs]

**Mitchell Kimbrough**:  Yeah.  [Laughs]

**Emily Lewis**:  I said I like pancakes.  [Laughs]

**Mitchell Kimbrough**:  Yeah, well, I'm telling you that you're not right about liking pancakes, and the evidence is that you have so many cats in your house.

**Lea Alcantara**:  [Laughs]

**Emily Lewis**:  [Laughs]

*Produced by*

**Lea Alcantara**:  Well, that's all the time we have for today.  Thanks for joining us, Mitchell.

**Mitchell Kimbrough**:  Thanks for having me.  I'm really honored.

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

**Mitchell Kimbrough**:  At solspace.com, mitchell@solspace.com.

**Emily Lewis**:  Thanks again, Mitchell.  This was a really great conversation and it's always great to hear your voice.

**Mitchell Kimbrough**:  Thank you.

[Music starts]

**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, Craft Commerce!

**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 we're going to talk about emotional intelligence in design with Beth Dean.  Be sure to check out our schedule on ctrlclickcast.com/schedule for more upcoming topics.

*Produced by*

**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:  00:59:49*

*Produced by*

**Bright** UMBRELLA