

**TOLE KHESIN:** Welcome, and thanks for joining us. This webinar is called, "Developing a Custom Video Solution with 3Play Media's API." My name is Tole Khesin with 3Play Media, and I'll be moderating this webinar. I'm joined by Bryan Ollendyke, who's the instructional technologist at Penn State University College of Arts and Architecture. From 3Play Media, we also have Mo Zhu, who's a software engineer, and our VP of Products, CJ Johnson.

We have about 45 minutes for this webinar, and we plan to leave the last 15 minutes for Q&A. The best way to ask questions is to type them in the Questions window in the bottom, right corner of your control panel. We'll keep track of them and address them all at the end. Also, feel free to email or call us any time after this webinar. This webinar is being recorded, and you'll receive an email with a link to watch it tomorrow.

For the agenda, Mo will begin by giving an overview of the benefits of an API and how it works. He'll then dive into the specific features of the 3Play Media API and do a demo with code examples. Then Bryan Ollendyke will do a demo of the automated captioning workflow that they've built at Penn State University. We'll then open it up to Q&A. So with that, I'll pass it off to Mo.

**MO ZHU:** Thank you, Tole. So again, my name is Mo. I'm a software engineer here at 3Play Media. And I'm going to tell you a little bit about APIs and the 3Play API today. So first of all, what is an API? API stands for Application Programming Interface. And the way I like to think about it is it's sort of in opposition to a graphical user interface, which is the website that you use to click around.

And specifically, a graphical user interface is for a human to talk to a computer. We care about fonts. We care about nice color palettes. But if a computer is talking to a computer, they don't particularly care about how things look. All they need to do is to pass data and commands back between each other. So that's what an API allows computers to do.

So what's so great about APIs? Why do we build them? They allow customers to design workflows that suit their own business needs, so that the functionality behind an API-- for example, our 3Play functionality-- are all exposed to the customers who can then mix and match those in certain ways, however they like. That allows them to automate repetitive manual tasks, especially if you have a lot of things that need to be done at a large scale. And that lets you reduce costs and labor and workflow and organizational complexity.

So for this demo today, I've created a demo web app that we can use to illustrate how this API works. Here it is. This is just a place where we can see all the assets that belong to a test account that I have made.

There are basically three things that you can do on this website. You can view all of the videos that you have. You can click on one of the videos, and you can watch the video itself, along with the captions. And then finally, you can search via this toolbox up at the top here.

I created this demo website. But as I explained, it's basically just a computer talking to another computer. All they need to do is pass data between each other. And so that is built based on this data that I received from the API. And you can see here that there is a name, a duration, description. All of these things can then be parsed and turned into something that people like to look at, which is this.

So how exactly does 3Play's API work? We built it on what's called the RESTful architecture, meaning that every piece of information that is stored in 3Play can be created, read, updated, and destroyed-- also called a CRUD app. So for example, in Facebook, you can create a new post, you can view other people's posts, you can edit your posts, and you can delete your posts.

Now, not all of these CRUD actions are available via our API. And not all functionality from the 3Play Media GUI account system can be accessed from the API. But really, a lot of the core things that you want to do with 3Play are accessible via the API.

And those things, speaking broadly, we allow you to manage captions, translations, and alignment jobs that exist in 3Play. You can request them. You can view them, whether you want to see their status, or you want to see other metadata about them. And then you can download the finished captions and the transcripts. Finally, the API also allows you to add our interactive transcripts and other plug-ins to your video.

So before we talk specifically about how to use the API, we just need to do a little bit of housecleaning beforehand. First, for each project, you need to create an API collection. And that allows you to group your media assets together into one accessible package for your API.

And each of those collections have a key that gets created. And you can find those in the Settings API Managements part of the account. There's an API key and a secret key for each

collection, and you need those for your various calls to the API.

So the API key is a random string of characters that identifies you to the systems, kind of like a password. Some calls only require the API key. Other ones, where you're creating new objects, creating new data, you also need the secret key.

And then one other bit of housekeeping is types of HTTP requests. So these four types-- POST, GET, PATCH, and DELETE-- correlate with the CRUD actions. So a POST is a create. A GET is read. PATCH is update. And DELETE is a destroy.

All right. So to get to the API, there are two ways that you can view files in a project. The first is you view all the files at once. And that's when you issue a GET request to `api.3playmedia.com/files`. And you also include the API key parameter. So the example in the third little bullet there is a fully functional call there with a fake API key. And that correlates to this page here of our demo, where you can see every single one of the videos, including the duration and a description.

Another way to get access to your files is to view a single file. And that's when you issue a GET request to `api.3playmedia.com/files`. And then you also provide the ID of that file. And of course, you include your API key. And that corresponds to the action of this page where, now, you can see more information about this one specific media asset.

Here, we have a link to the YouTube video. And there will be captions available. [INAUDIBLE]. And also, in addition to that, our interactive captions are playing along on the side there.

And finally, another GET request that you can do for the API is querying the files by searching. And for that, you would issue a GET request to `api.3playmedia.com/files/search` and provide the API key and your search query. So that roughly corresponds to this home page here. If we've searched for something like "Olympics," it will only return the files which either have the title or the transcripts that have the term "Olympics" in it.

To create or request captions, you can issue a POST request to `api.3playmedia.com/files`. This time, you want to include both the API key and the secret key. And you can either upload a full file using the File parameter, or you can upload an HTTP link to the video using the Link parameter.

And this is for captions, but there are also very similar calls that you can make to translations, to alignments. And so all of those you can see in our support document. I won't get into the

details of them right now.

And the same goes for downloading the captions. You issue a GET request to this URL here. Include the file ID and what format you want and also your API key. Some of the common formats are listed on the slide here. And 3Play supports, basically, all common formats.

So other things that you can do-- you can order translations and transcriptions. You can view them, download them. You can also create an auto callback, which is when our API tells your system when the files are done. There also, Transcript Alignment where you convert a transcript into timestamped captions. And finally, you can import existing captions and translations so that you can manage all of those assets in one place in the 3Play system.

So just to go over what I talked about-- what is an API? It's a way for computers to talk to each other and to give commands back and forth. It exposes almost all of 3Play's functionality to customers, so that they can build around it how they like. And so the benefits of it include being able to customize workflows, being able to automate manual tasks. And basically, in our API, you can do most of the key things that you want to do, using the 3Play GUI.

So next, I'm going to pass it off to Bryan, who's going to talk about what they did at Penn State.

**BRYAN**

OK. So as was mentioned, my name is Bryan Ollendyke. I work at Pennsylvania State

**OLLENDYKE:**

University in the e-Learning Institute. It's out of the College of Arts and Architecture.

My main job responsibility is working on a system called, ELMS Learning Network. It's a series of web-connected systems that stitch together course experience. And the reason we do this I won't go into the great details about, but you can find out more about the project at [elmsln.org](http://elmsln.org).

So what I'll be showing today is one system in that network, which is known ELMS Media. So ELMS Media is more of our asset management piece to the overall system. You can see, being in an arts college, we have a lot of media. So we've got about 40 courses that are online. And in 40 courses, we have almost 2,000 videos, which are lectures, clips from movies that are relevant to the course. Some of them range in time from five minutes to as long as an hour and a half.

So obviously, needing to maintain accessibility is very important. And so it's where 3Play Media has really helped out in that process, because we had a lot of video-- lots of hours of

video. We couldn't possibly transcribe it all by hand in a reasonable amount of time.

So just very quickly what the system does is we were able to look through all of our media. We could upload a video, for example. And so this would be the ELMS Test for 3Play. We could classify it as the course that we want it to be a part of.

And then I'm going to just upload a quick video here. All right. Good thing over VPN. That goes fast. And so I'm going to add a thumbnail to the video, as well. And you'll note that, below here, we've got support for a caption file and transcription file.

So what we've done is, in Drupal, we have this content type that is video. And so we're trying to replace the process of someone having to know object code, HTML5 embedding, all of that. We're trying to automate that entire workflow of how do we get something from being produced media into a course.

And so if we had a caption and transcription file to help maintain accessibility, we could upload those. We don't want to do that, though, because, as I mentioned, we have 2,000 pieces of media in this system alone.

So the nice thing we've built here using the 3Play Media API is-- you see there's Transcription Status. So I'm going to mark this as needs transcription. But you can see, there's these other steps in here. So we've got Submit to 3Play for Transcription, the fact that it's been sent, whether or not it's currently being processed, and whether or not it's received, which would complete the process.

So I'm just going to mark this as Needs and hit Save. So just add a new video to our system. You can just see, quickly, we've got that thumbnail in place. I can click Play, and it'll start with the video that I can jump through.

The real time saver in the system-- as we mentioned, trying to automate these things away-- is this. So we have a short code that anyone can copy and paste in. You don't need to know HTML to use this. If you do know HTML, you could fire this up. And you see, it's basically just rendering an iframe, which is a very similar approach to the way YouTube does things.

So let's go back to our Search Assets. And we'll use Search for Films, which will filter our list down. And you see, there's our new one that we just put up, the ELMS Test for 3Play. So we could then take that and embed it in the course and show that very quickly. I'll just copy. I have a course space here already. Test ELMSLN-- how convenient.

And so we're just going to go here. And this is pretty much the way our workflow works, copy, paste, you're done. We do have a WYSIWYG button we can integrate and do it that way that I won't show right now. And so I hit Save. And you'll see the media shows up. It's all happy, responsive, all kind of cool stuff.

So that's not the piece we're trying to solve, though. The piece we're trying to solve is making this accessible, because hypothetically, we've just gotten a request that says, hey, we need caption and transcripts for that. So what happens, once we get that request, we come in. There's a Transcription Workflow button. And so I'll go to the transcription workflow.

And you see it's similar search, but we've got some extra fields. So let's search by ELMS again. Filter this down, and we'll look for our 3Play Test. I probably should just type 3Play. 3Play. I'll just type Play. 3Play. There we go.

And so you see we have a video here, and it's been marked that it needs transcription. And then there's fields for caption and transcript. Now, you may have noticed on the larger page, when we had all of them, that Caption and Transcript are here for some of these, right? So when the process is done, they show up here.

And you'll see this is marked as received. We're not going to be able to transcribe and caption something in the five minutes here. So I just wanted to make sure we know that that's what it looks like at the end.

So what I can do-- let's say I had a whole lot of these. I could just click this box and ship all of them off. And so I'm going to click this, just to select this one. And then I can switch this. From Needs Transcription, I could set it back to None, to eliminate it from this list, if it was an error.

Or I do this. So we get a request. We say, hey, we want to submit this to 3Play Media. It confirms it. As part of our workflow, this page is only available to a few people, because this obviously has financial implications, because we've got this here. It's going to tell 3Play, hey, get our videos and automatically transcribe them.

And so, right now, that doesn't do anything. That just marked it as such. And I can go to my 3Play Media account, I can hit Refresh, and we'll see it hasn't actually sent anything there yet.

So what happens is the servers have this thing called Cron. And Cron is a job that just runs on an interval. And so, for performance reasons, we batch these up and we ship them off nightly,

because we never have a request that needs to be finished in five minutes. You could do that, we just don't have need for it.

And so what happens is, nightly, there is-- there it is-- there's this Cron that runs in our Drupal system and says, hey, ask 3Play about all this stuff that's already processing. So this is how we look up things after they've already been sent.

And so you can see in here, in invoking the 3Play Media APIs, that we're basically just making requests against parts of the API. We've got our private and public key, which I've starred out. But we've got a Settings page where we just copy and paste those in, so that 3Play knows, hey, these people are who they say they are kind of a thing.

And so we've got a couple of calls we can do in here that we've implemented. We've got the ability to send them a file, which tells them, hey, this exists, which is what we're about to do. And we can get transcripts and captions after they've been finished. And for our purposes, we get transcripts as text files. And we use the DFXP caption format, because that's what we use.

We can also check status. And so what we do every night is, hey, look up the things that we've shipped off. And so we just keep requesting status. When status is finished, then we actually go about doing the get transcript and get caption, which stores the files locally for us.

So that's the ongoing job that happens. What ends up happening after that is then we look at the things that are approved, which is I just put that into the approval state. And so when we look at something that's been approved, we're going to get the list of them, and then we're going to invoke the API call here of Send File. And we'll ship this off.

You'll see call structure's pretty simple. It's the keys in question, a link to the file, so that 3Play knows to download it-- you're not sending them a multi-gig file-- and then a video ID-- so this is how we're able to look it up after the fact-- and then batch and name, which is, pretty much, just help for filtering.

So because we're not going to wait until midnight to run this, I'm going to trip the Cron on the system to run. And since the DNS appears to be working, now, when we go back to 3Play, success. So we've got this over here. It's the ELMS Test for 3Play. If I open it up and get some of the metadata, we'll see the file ID is such. Look in the metadata. There's a video ID. It's 6673.

And so what that lets us do, from a later looking something up standpoint-- everything in Drupal has a node as an identifier. And so I can go to 6673, and you'll see it's taking me right there. So that allows us to keep our system somewhat in sync, a primary key of sorts. So if we go back to basic, you'll see it's pending. This hasn't actually gone through yet.

When we were doing some set up work just recently for this presentation today, I had to remember how to do this stuff, which is a testament to how much this API integration has eliminated me from the process. I don't do this anymore, because we've been able to leverage the API with our existing workflow and empower our staff to not need to know HTML, not need to understand captions and transcription, and honestly, not really need to think to make things accessible, which was one of our goals with this API integration.

And so the nice thing I mentioned-- you have to simulate this in your mind-- is that three days later, our system is constantly asking, hey, is it done? Hey, is it done? Hey, is it done? And then when it's done, it would automatically download the caption and transcript file. And then playing this same video-- you notice we have closed caption grayed out-- this just self-updates, and so, pretty much, kick it off.

Three days later, you know that the material's accessible and the process is completely automated. We pay the bill. And that's about all we have to worry about, as far as captioning is concerned. It's been pretty transformative for our process.

So that's the gist of it, really. That's pretty much the integration we use. We're working on it, going forward, in ELMS Learning Network with a Drupal 7 publicly available project. And I don't have a lot to say about it, other than this has been awesome. [LAUGHS]

**TOLE KHESIN:** Awesome. Thanks, Bryan. All right. So we will resume with Q&A here. Bryan, there's a question here for you. How long did it take to develop the API captioning workflow for this project?

**BRYAN OLLENDYKE:** Developing it on our side, once I got access to it and got my head around it, was less than a week. I think we tested it in one day. And it took about two or three days of building things on our end, because we didn't have this workflow in place. So as far as actual API integration, yeah, maybe a day.

**TOLE KHESIN:** Here's a question that maybe, Mo, you can answer it. Will the API ever become deprecated?

**MO ZHU:** So if we ever enhance the API or change how the API works, what we'll probably do is make



this Version 1, and then we'll have a Version 2. So what we don't want to happen is for us to change how the API works, and then suddenly Bryan's app doesn't work anymore, and everything breaks in his app.

And so we'll probably send out notices about the fact that our API is changing. You can switch up to Version 2. But we probably won't get rid of our API, the way it currently is, at any time.

**TOLE KHESIN:** There is a question here. I'm unclear about how the transcripts get made. Is that an automated service that 3Play sells?

Yeah. So 3Play Media is a captioning and transcription service company. Basically, the way that it works is you set up an account, and then you just upload content. You can do it over the API, or you can use it through the graphical user interface.

But you upload files, and we process the captions and transcripts. And our process is hybrid. So the way that it works is, when we receive a video, it first goes through speech recognition, which gets it to a point where it's about 70% or 80% accurate.

But then we have professional transcriptionists who will go through and clean up the mistakes left behind by the computer. And we actually even subsequently have a QA process, which somebody will go through and double-check grammar, punctuation, and things like that. So by the time we're done with it, we create captions that are pretty much flawless.

CJ JOHNSON: Yeah. I think it's worth mentioning also, Tole, that we recently launched what we call progressive delivery. So we have some organizations using the API to send a lot of content to us who have extremely time-sensitive deadlines in terms of getting even things like key words for search out of the transcript content and even just a base level of captioning.

And what progressive delivery does is, when that automatic transcript is finished-- so within a matter of hours or even minutes, if it's a shorter video-- we'll send the ASR drafts-- so basically a transcript or a captioned file that is the automated transcript-- back to the client. And then, as the revisions are made by our editing staff, you can start to access the more and more accurate versions of the content, until it's deemed complete. So that's been something with folks who have very time-sensitive needs that's proven pretty valuable.

**TOLE KHESIN:** Awesome. Great. There is a question here. When would you recommend building a custom API workflow? And maybe Bryan, and CJ, and/or Mo, if any of you guys can address that

question?

BRYAN OLLENDYKE: Yeah. So I'll jump in on that. So we used to do it manually with 3Play. We had this system in place. We were uploading video to it. We were putting video in courses. And that was saving us a lot of time.

And then I was talking to our media staff one day and realized that, while I saved a lot of people downstream from them time, they were still going in manually. When we would have a course get audited-- we had a course that had, I believe, 60 to 100 videos. And really, once you start to hit any type of scale, using the interface, you almost end up having a dedicated person to do that.

And then that also leaves a lot of room for potential error, as far as, oh, did I actually upload that video? Oh, shoot. We meant to do these 10. It became an almost unmanageable process when we had an accessibility audit.

And so, in my mind, it's a big question of scale, as well as if you already have a process in place to align with. If you're just using 3Play for a couple of videos here and there, and the videos are coming from lots of different places, maybe you don't need to build a custom API integration. For us, with a pretty well-refined get media all in one place type of process, and then the scale, it was a no-brainer. This saved us from, probably, hiring two or three other people to be a dedicated part of the media staff to pretty much just click the buttons.

**TOLE KHESIN:** Awesome. Thanks. CJ and Mo, do you guys have a weigh-in on that?

**CJ JOHNSON:** Yeah. But I think Bryan mentioned a keyword of scale. If you have a lot of content and you're going to be processing it over time, and you're adding to your library, that is a really good signal that an API solution might be good for you.

And the other key that Bryan has is that centralized content management system where he is driving off of their internal records and able to use the association we have through the video ID field to make 3Play assets a part of the master assets that live at Penn State. So it's a way to just have one place to manage your content. And it makes it a lot easier for having a solid workflow.

**TOLE KHESIN:** Very good. Very good. Thank you. There's a question here. What types of metadata can be extracted from the transcripts? CJ, do you want to--

**CJ JOHNSON:** Yeah. So there are two main things that-- beyond the metadata that you can put yourself, like the video ID field or wild card attributes, once transcripts are completed by 3Play, we do do a keyword analysis of each transcript. And we have a really interesting blog post that maybe we can share as part of resources around how we actually extract keywords and key phrases from the transcript text. And it can be very useful in terms of interpreting what that transcript is about, as a whole, just a list of keywords.

And since they're also timed back to the transcript, you can merge that with a longer file. So you have an hour-long video, you can merge those keywords with the transcript to see how topics evolve over time. It can be a pretty interesting way to view the content on a more macro level. We have examples of that with our interactive transcript where it's implemented in, I think, we call Scan View that I recommend you checking out.

The other thing we do do is-- and this is, I guess, sort of metadata-- but we do index all the content, so it is all searchable. And so, I guess, that index is sort of a piece of metadata in a way.

**TOLE KHESIN:** That's great. Thank you. Bryan, question for you. Is ELMSLN open source? And where can we get it?

**BRYAN  
OLLENDYKE:** Yeah. So ELMSLN is built on top of Drupal, Drupal 7 primarily, and a lot of other open source projects. You can find out more about it at ELMS-- E-L-M-S-L-N-- dot org. It's a collaboration between Penn State, Wisconsin, and pretty much anyone that contributes to Drupal, because we're leveraging so much of the open community.

Not ELM specifically, but projects as part of ELMS learning network-- because they're all openly available on drupal.org-- have been downloaded over 775,000 times and are actively used on more than 11,000 websites. So ELMS has this really good symbiotic relationship with Drupal. And it's 100% open source. Everything that's out there-- the larger project is on GitHub, but I prefer you go through elmsln.org, if you want to find out more about it instead of skipping straight to GitHub.

**TOLE KHESIN:** Great. Thank you. There's a question here. Can we use the API if we already have existing captions? I can answer that. We actually recently released a caption import feature. And what that allows you to do is, if you already have captions, you can just import them into a 3Play Media account system.

And when you do that, that gives you the exact same capabilities that you would have, had you processed the files with us originally. That means that you can create translations. You can output over 50 different caption formats. You can use the interactive transcript and video plug-ins.

And so this is a new feature. And you can upload your own existing transcripts and captions. And this is all available through the API as well.

All right. So here's a question about the API documentation, probably best for Mo. Where is the best place to start with the API documentation?

**MO ZHU:** Yeah. So in our Support area, [support.3playmedia.com/forums](https://support.3playmedia.com/forums), there's a section called API Management and Documentation. So you can go there and take a look. It's pretty thorough, so yeah.

**TOLE KHESIN:** Great. Here's a question here that maybe CJ and/or Mo could address. What are some other use cases for the 3Play Media API?

**CJ JOHNSON:** Well-- this is CJ here-- one great use case is simply for, I'm saying, scalable workflows. And hopefully, Bryan can show us an example here in a little bit. But we have a lot of folks who are basically baking the 3Play API into their own content management systems as a black box where they have hundreds or thousands or even 10s of thousands of media assets that are cataloged in a database on their own system. And they buttons set up that can do things like request transcripts, request captioned files, request metadata from the 3Play system to support on different needs for publishing and content management on their ends.

And that includes things like searching for videos based on keywords, finding results based on that specific phrase that you want to find the different videos that, maybe, match a relevant news topic of the day, or even diving deep in and saying, of these thousands of hours of content, here's 20 minutes that pertain to the word I'm looking for.

**TOLE KHESIN:** Great. So there are a handful of other questions here, but they're actually pretty specific. And I think it's probably best to address these offline. And we'll reach out to you after the webinar to talk about them.

I wanted to thank the speakers and everyone for joining us today. Thanks again, and I hope you have a great day.