

Zendesk Developer User Group

Developer Tools: Postman

Fri, May 26, 2023 2:26PM • 1:01:56

- Zack shares his screen and gives some housecleaning tips. [0:15](#)

- Starting with the basics and some housecleaning.
- Zendesk podcast, deep dive into Zendesk AI.
- Great power comes with great responsibility.
- API keys should be treated as passwords.

- What is an API and how does it work? [5:00](#)

- What an API is and how it works.
- How robust APIs tie systems together.
- How to get access to the Zendesk API.
- Creating an API key.

- What we need to do to get an encoded output. [9:58](#)

- Generating the base64 encoded version of the user.
- Using environment variables in Postman.
- Store keys for different instances of Zendesk.
- Add an API key to Postman.

- What's a Base64 encoded string? [16:29](#)

- Getting rid of the API key.
- The Base64 encoded string is important.
- Apologizing for not being able to authenticate with Postman.
- API endpoints in Zendesk.

- How to use Postman to get code examples in other languages. [21:12](#)

- JSON object returned from the Zendesk API.
- Postman provides code examples for the request.
- Running the JSON response in Postman.
- How to use the JSON library in Python.

- What are the squirrely brackets? How do they work? [26:25](#)

- Squirrely brackets and how they work in JSON.
- How to loop through groups.
- Extracting the group and ID from the data.
- Printing the information to the screen.
- Extracting information from the API and transforming it.
- Loading information into the ETL framework.

- The first step you did for encoding is that important when Postman already stores that as a secret? Is it still a good thing to do? [33:17](#)

- Open to questions.
- Encoding base64 encode for Postman as a secret.
- Debbie has a way for Nicole and Drew to export chats.
- Pagination in Postman.

- How to create a ticket in Zendesk. [38:18](#)

- Two ways to get information into Zendesk.
- Content type and authorization for getting information.
- Creating a ticket in Postman.
- Using the HTML body to create a ticket.
- Getting a response from Zendesk support.
- The benefit of using OAuth tokens.

- A neat trick I use when I want to change authorization for a whole collection. [46:49](#)

- Using the auth tab to change authorization for a whole collection.
- How to access the collection.
- Running requests individually or as a collection.
- Learning to encode yourself is valuable.
- Zendesk postdoc resources and trial and error.
- Google and Chat GPT for concise information.

- How to use Zendesk to get the list of triggers and all this kind of information. [54:27](#)

- Zendesk is like a relationship database.
- Troubleshooting and troubleshooting in Zendesk.
- Monthly cadence of events and events.
- Pagination and events API.
- Advice on a better understanding of Postman's functionality.
- Upcoming community events coming up.

SUMMARY KEYWORDS

postman, encode, api, zendesk, developer, pagination, request, information, group, give, ticket, json, call, troubleshooting, token, endpoint, user, click, api key, id

SPEAKERS

Ahmed Zaid, Stephen Belleau, Zack Olinger

Zack Olinger 00:15

awesome

00:23

cool

Zack Olinger 00:31

okay, so I'm gonna go ahead and start then want to share my screen and see if I could mimic a little bit of what Michael does for his admin user group to kind of give some cadence to this. I won't spend a lot of time in the beginning on going over a whole lot of things because I get a sense that there's a lot of interest in today's topic. So I want to give a lot of time to that. So I'm gonna go ahead and share my screen and kind of get started with kind of the basics and some housecleaning. So let me see here. All right, everybody able to see this? Okay, I actually can't see. So I'm assuming you can. audio feedback would be great. Ken, if somebody wants to give a shout out to that. Oh, good. Awesome. Thank you. So the agenda is basically going to be like this, do some, Hey, everybody. And then I'm actually not going to spend a lot of time on relate. There's a lot of other great people covering what happened that relate. And I will just publicly state that I was grateful to be able to attend and it was fun. And I want to publicly state thank you to Zendesk for hosting that. Along with that, I also just want to announce that Nicole is starting and has started the conversations with Zendesk podcast. And in the email, like the follow up email to this meeting, I will include the links where you can find that and listen to it. So that's a very cool new thing that the community is starting. So thanks to Nicole for that. In addition, Zendesk just did a deep dive into Zendesk AI, I'll have a link to that as well, the events already done, it's passed. Yet, there is a YouTube video for the replay if anybody's interested in that. So I will include that in the follow up email as well. To get to the heart here real quick, I wrote this up, actually, I had Chet GPT write it up for me, to be honest, this was not requested by Zendesk for me, for this particular group. And what we get into, I just thought it might be wise, it's not really not so much like to the point of like, legalese. It's more to impress upon us that where we get into some of these things, it's great power comes with great power comes great responsibility, right. So the things that we do here, I always just caution, like, make sure that you feel comfortable with your own production or sandbox environments. It's a trust and verify kind of a thing. It's it's as is, and I'm only putting that on specifically on this meeting. Because there are things I'm going to demonstrate that I would never suggest somebody to do. Primarily sharing of the API key, I'm going to be pretty open with that on my own developer instance, so that I can illustrate how to get these things set up and the steps involved to to perform these tasks. Yet at the same time, you would never openly share this information with anybody outside of your organization, or even anybody outside have a need to know basis. API keys should be treated as passwords, and are very much like the keys to the kingdom, we could go into a little bit more specific, there are OAuth keys, which can be scoped down as far as their ability to impact the system. Regardless though, if it's an OAuth key or an API key, both should be treated as passwords. And then caution is just meant to be impressed here. So enough of that. Just wanted to put that out there. And then basically, right now, I'm going to stop sharing the screen, and we're ready to kind of get into the bulk or the meat of it. So the sorry, what we're going to get into here is the usage of a developer tool called postman. And before we get too far into it, we should really just need to find what an API is. And I still use Google sometimes. And I tried to find what would be a good explanation for this. And I was kind of dissatisfied in a way. I couldn't necessarily find a great visual aid for this. So I'm just gonna go ahead and give you my take. What an API is. So specifically, we'll just connect with Zendesk, since that's the product that we're most familiar with here. Or at least that's what we're here for. So on the back end, we can visualize a desk basically as a database. And each piece of Zendesk has like an object associated with a probably a table inside

of a database. So like the users, right users are an object, they have information associated with them, it's probably represented maybe as a table in a database forms, same kind of thing, fields, same kind of thing. Any type of object macros, these kinds of things. So what an API does what it is, is it's a service that a company provides that allows developers to access those pieces of their database or of their system in a programmatic fashion. So API actually stands for Application Programming Interface. So just like a GUI, or a UI is meant for an end user and meant to be clicked on and, and used that way by the end user. And usually, the interface is aimed to be clean and easy. The API is for developers. And it's their interface to interact with the system. So for example, a good way to do this would be instead of adding a user by clicking on the screen and going through the normal way of maybe adding users through the the user interface, and a developer can write some code to access the API. And instead of just creating one user, we could bulk import or bulk create, we can adjust different fields on that user. And the API is what allows us the ability to do that. And what's really nice is that when systems have robust API's, then we're able to tie systems together and integrate them so that they can talk to each other. And this is where things like JSON will come into play. And then also terms like ETL, the extract, transform and load type of like phenomenon. So that's just to give you an idea of what an API is. And so in thanks, thanks isn't desk, they actually have a very robust API, they have a great service that they provide, that allows developers to access a wide range of their functionality. So to start here, to get access to the API, what we're going to need to do is come into these in DESC Admin Center. And we're just going to search for API and come down to Zendesk API. And I'm going to just go ahead, and I'm going to use the one that I have already created back in 2021. But basically, what we would do is we would add an API token, we'd give it a name, I personally like to date them, and then give it like a some type of identifier here, right? So this will work, then right here is the API token. And this is the thing that you do not share. This is the keys to the kingdom. So basically, we then save this, I'm going to copy it. I'm gonna come over here to my notepad, save it here real quick. That saved if you click on this, it's never going to show you the full API key. So when you first create that API key, you have to copy it out, or else you'll you're not going to see the full thing again, you just delete the key and create a new one. And again, this Wait, I thought I gave it a name. That's okay. So now what we need to do after we have this API key is that with this, we have password access and API token enabled. This means that we have to base and code the Speak out the basics before encode the token and the user that's able to execute these commands, then that's usually an admin agents can execute API calls, although most of the time, you're going to want to use an admin user for the API calls. So what we need to do and this is where I'm actually going to, I was a little unprepared for this, because I forgot I had a basin code. So we're going to do this live. I personally like to use PowerShell for this 64 ENCODE, show. Okay, so I'm going to just do this here real quick.

Zack Olinger 09:58

There we go. Oh, I was wondering why that was not coming up, okay?

Zack Olinger 10:08

Take my key, actually, I need to do this, I need to do this gonna be right now that's encoded, sorry, everybody to 64 this is what

Zack Olinger 10:34

I'm gonna do my user

Zack Olinger 10:49

and this is how it has to look here with the forward slash token. And then we're going to take API key that we generated. I think you might have deleted that one.

Zack Olinger 11:11

deleted.

11:13

Because created the developer user group one, and I think you clicked on the Delete, because it disappeared from the list. I think that's what I saw.

Zack Olinger 11:21

Oh, let's see, I'm still having trouble here with the converting to see.

Zack Olinger 11:43

Yes, this right here, oh, is it not? Here we go. If you could be right about the key, I just want to make sure that I can actually get an encoded output real fast. So.

Zack Olinger 12:09

We go. Okay, so let's give this a try. So basically, what we need to do is generate some type of output like this. This is the base 64 encoded version of the user that has access, that's going to be using the token. And then we put the forward slash token and then the API key that we generated. So we're gonna give this a try and see if it actually works or not. So we're going to copy this, I'm going to highlight it. Right copied. So then we come into postman. And what I like to do inside of postman is to use the environment variables. And so over here, on the left, we have the working environment. And this is where you can store keys for different instances of Zendesk if you have them, at least for your sandbox or production. And it just makes it easier to keep those tokens or that that API information in a concise spot, and then also keeps the requests inside of postman more secure. And a way that way, it's not every request doesn't have your key embedded in it. It'll make more sense as I demonstrate this. So we'll just call this. And I'm going to put it as a secret. And paste that in. And then I'm going to save the environment. And then I'm going to come over here to my collections. And basically inside of postman, what a collection is, is it's a collection of API calls. And so what I'm going to do here is I have my own set of get requests for my own developer instance. So let's just try it here. I'm going to choose to get the list of ticket fields. Maybe Actually, I'll do list of groups. And then I come over here to the headers and for authorization. This is where we use the variable. And I'm actually going to go ahead and change this to Kairos to save it, and then when I click Send, this is where we'll know if it actually is valid or not. Okay, so it could not authenticate me. So then that means that either

Zack Olinger 14:44

Yeah, the key. So let's delete this. I'll try it again. Add an API key.

Zack Olinger 15:08

Save this come over here

Zack Olinger 15:38

so now want to come back over to postman, I should be able to come back over to my environments here to be able to just come in here and delete

Zack Olinger 15:53

try this and it was groups couldn't authenticate. So that is curious. So this is somewhat embarrassing here, because

Zack Olinger 16:19

am I missing anything from anybody in the group does anybody have Let's try it this way. I would normally never do this either. See if it's any different. Okay. Since I'm going to get rid of this API key, I will do things that I normally would not do. Okay, that looks different than what I received here. That is something for another time. point is is that it is a base 64 encoded string. So that's the important thing to to know. And the main thing about that base 64 encoded string is this particular format right here. The token part is what hung me up when I was first learning to do it. But anyway, just know, this is required to be a base 64 encoded string. Again, I would never put this in in like a production thing, I would never put my API key into a website. There are oops, other ways to it's encoded, but I just wanted to show the principle here. So we'll come back down to the groups. Yeah, sorry, list groups will do this again. Still could not authenticate me. Let's come here. Delete it.

Zack Olinger 18:06

Right, it looks right. Maybe I didn't save it. Possible. I sometimes forget when I update the keys here too, just to save the environment. So I may not have actually done that the last time still can authenticate me. That's crazy. Okay, well, I know this key does work. Okay, we're gonna go with that. And I am unsure as to why this is not working. And I feel embarrassed if I'm going to be straight up honest about why this is not working here like that. So I'm going to, I'm going to just move on, so I apologize about that. So the main thing, then, unless you guys would like me to troubleshoot that I don't necessarily feel like you guys would like me to continue, but you let me know. Okay. So once we're able to authenticate with postman, we're going to get, we need to put the authorization header. And then we're going to use this environment variable that contains that base 64, encoded user and API token. And now we're starting out we can make calls to API endpoints. And basically the endpoint for here like is for for the groups. And there's all sorts of different endpoints within Zendesk. So we can come over here to let's see, not this one. But there's the Zendesk a API reference. And if you come under here, there's all sorts of general topics. So the capabilities, if we come under ticketing, this is going to give you all of the different things for the tickets, you know, that's a general heading. And then each one of these is a different endpoint that we can access. So what I was just accessing was down here under groups right here. And so I wanted to list groups. And then this is the API endpoint. So you can see here, that in the URL, I have my base, my base URL, and then I have the endpoint right here. And then basically, all I have to do is to make sure that I have the authorization header. So if this wasn't here, I would just click inside of here and type authorization, choose that going to be basic, and then be my variable, click save it, and then send. And then when we would get a 200, you know, HTTP 200, which is an okay, response means everything was good. We're gonna go to our response back, and in here is going to be a list of all of the information for the groups. And this is the JSON object that is returned from the Zendesk API. And one of the things I was going to demonstrate here, one of the nice things about postman is that you can get code examples. So to perform this exact same request and receive this output, and a Python example, this is the code that would allow us to do that. And then postman can change this over to a large variety of different programming languages. And basically, as a developer, and my experience, it's been once we're able to wrap our head around how to procedurally tell or like, attack a problem, logically, languages are just a matter of syntax largely about how we procedurally write those instructions to get things done. So if we needed to, you know, make the same request in JavaScript with jQuery, here's the code for that if the project that we're working on

and the new environment that we're using calls for this application of this code, then we can have postman provide us a starting point for this request. Same thing with go or curl. So it's a pretty sweet couple. And so to kind of make this more practical, one of the things I was going to do is show Python, and then basically take this code snippet, we can copy that. And so say, we wanted to know the IDs of all of the groups and the names. Like we see this object here. And we're like, Okay, we've got URLs got the ID we've got is the public the name, we've got all of this information about the group. But maybe we're only interested in the ID and the Name. So what we could do is, we could take this code snippet over here. And we'll go back over to, again, I'm using Visual Studio. So I would just go new, new file. Now I'm going to do new Python file. And then I'm just going to paste that in. And then I'm going to go ahead and save this file. And I'm just going to call it let's see list groups is what I'm going to call it. Okay, so what we can do right now is we can click this play button. And we can actually run this and we'll get some information. So this is what we get as a response. Doesn't look super pretty. But that's a JSON response. And really, that's all we need from a developer perspective, because it's got all of the data that we wouldn't need to have access to, although for reading it has we're troubleshooting doesn't look that pretty. So there's a way around that. So inside of Python, there are things called modules or libraries, and you can import those. And basically, again, they're just libraries are just large amounts of functions that people have written to do specific tasks. So for the JSON, the JSON library or or module that I just loaded in, is specific around handling JSON files or JSON data. So what I'm going to do now is I'm going to do JSON dumps. I don't get JSON, and I am making assumptions you guys don't Need to Know this, I'm just showing you how this can be done. So what I've basically done is I've called this library to load this text version, because the text version of the response loaded up in a JSON escape, this gave what needs to be done. Sorry, there's, we're getting a little ahead about some of the things that you may need to know, I need to load the text version into a JSON format, and then dump that JSON format. And then I'm going to tell it to indent each of the keys by four spaces. So that way, it makes it easier to read, this will make a lot more sense when you visually see it. So we run it. Instead of that block of code that we had here, this block of JSON, we now have what we pretty much had in postman, which is this right here, let me let me close this up. This right here, it's easier to read. So by adding these couple of things, I was able to take this and convert it into this. To go a little bit further now that we know that we have a response. And we know what the response looks like. You know that in our JSON, we have an element called groups. And to get into a little bit of specifics, there are things called arrays or lists. And inside of JSON, they're going to be denoted as these square brackets. And then each of the squirrely brackets is going to be an object. So each one of these is a group or an object, group object or an object of a group. Here, then, this is a it's one of those. And then this is going to denote that there's a list or multiple of these and say this comma, and there's another squirrely bracket, this is another group, then a comma squirrely bracket, another group. So that's just to give you an idea of how this is formatted and how this works. So what I'm going to do here, to get just the ID and the Name, what we're going to do next is we are going to do, I'm trying to remember I did this before, and let's see, just do groups, I'm going to do

Zack Olinger 27:30

Richards going to be json loads. That I'm going to do for group and groups. Actually, I need to do it this way, this is going to be slightly funny. The reason I'm doing it this way, basically, what I'm doing here is I'm setting this variable to this entire thing. And what I need to do is I need to just access this, like I need to I need to set the variable when I'm looping through, since this groups is this entire data set, I need to just loop through the groups. So I'm going to tell this for loop to assign group as a variable. And it's going to loop through each one of these items here, right. And so what that means is that group right here, is going to end up equaling as it's looping through, as it's going through each one of these

that's defined in this list, the variable group is going to be one, this one, and then when it loops through the next, the group group variable will be reset and equal this one, and so on and so forth. So, here's what we're going to do. So then we're going to do ID equals oops, I cannot type. Oh, wow. Sorry, everyone. Okay. And the reason that this is going to be group and ID is because again, we're looping through this right here, right, so we're working with this data right now. And then ID is right here. So we're referencing, again, that group very of this group is equal to this data. And so to extract out the ID, we need to tell it that this is the information this key is what we're looking for. And then for the name, we're going to do the same thing. Look for this. Okay, so now what we can do is we're going to print that information out. And again, I'm making assumptions about like the level of where people are at for coding and such and I don't necessarily want to get lost in the weeds on the details. But what we're going to do next is we're going to use a formatted A line or print statement, so that we can print this information to the screen. So I'm going to use this, and we're going to do ID, and then we're going to do ID, then I'm going to go and do this as well, like this.

Zack Olinger 30:24

Okay, so we should hopefully be able to get, and I'm going to comment out this line, so we don't get that large JSON output. Hopefully, we'll just get a few of these. So I'm gonna go ahead and click run here. And there we go. So we extracted out the ID, and the name. And if I wanted to make this a little bit prettier, I could do something like put a carriage return at the end. To the slash in is for newline. Run it again. There we go. So this way, I'm able to see, okay, cool. I've got basically four groups, here's the names. Here's the IDs. And this is more or less what the ETL framework looks like I've extracted I've made a request to the API. So I've extracted some information, stored it in the response here, I'm now transforming the information. So I've extracted it and I'm only, I haven't necessarily done like real transformation with it, but I've extracted out what I want. And then now I can transform it if I'd like to. Meaning I could take this ID and pull it into something else or look it up for somewhere else, I can take this name and do something with it. If I wanted to, I could import, you know, Import CSV and do some CSV stuff, where I can start to actually create my own comma separated, separated value files, my CSV files based upon this information. So I could, you know, I know sometimes for information that admins need out of Zendesk, it's not necessarily easily accessible. This gives you kind of a way to be like to get an idea of how developers would go about that, I know for sure, in my days, as a system admin, there'd be times where information would be needed. And it was much quicker for me to write up a quick script, to extract the information, and then format it the way that I needed it to be in a CSV, and then turn it over to, you know, the stakeholders, rather than, you know, trying to manually do it or massage the data some other way. But this is basically the quick way to do it. And then from here, the other thing I was thinking about doing is like to load it in, basically, what you would be doing is making another API call, but instead of doing a get right here, which is for pulling data, you could do a post or a put, which is a way to send information into another system, or back into Zendesk. But that would be the loading part of the ETL. So this actually went a little bit faster than I meant, or anticipated. So I am going to go ahead and open this up to questions. I'm going to quit sharing my screen here real quick. And, yeah, let me know if there's anything that you guys have. Oh, my goodness, there's a lot in the chat. I missed all of this. Let's see.

Zack Olinger 33:45

I am sorry, I'm just taking a look at the chat that certainly feel free to please on mic and ask questions or speak to anything that is coming up for you.

Stephen Belleau 33:56

Yeah, let me jump in and ask the first question here. So the first step you did for encoding? Is that important to do when postman already stores that as a secret? Is it still a good thing to do just as a best practice? Might we need to do that?

Zack Olinger 34:14

So the way it does, it still will need to be base 64 encoded. And the reason that's needed to be done is that when you're passing the and I'll go ahead and share my screen again, I guess, here real quick. So when we come in here to like this right here for the authorization part and that HTTP header, and we're telling it, it's a basic auth. And then we're sending that base 64. Encode what is on the back end for Zendesk, what it's doing is it's taking that and it's understanding, it's decoding it and it's saying, Okay, this user and this key and it's it's a way for Zendesk to authenticate that request. So it does need to be basically four encoded even though it's being stored inside of postman as a secret. So as we're kind of looking at two different things like the secret inside of postman is separate, it's a way of securing that information inside of that tool. Whereas when you're sending that basically for encode that is for Zendesk to authenticate you so we're kind of looking at at two different kind of security features, I think, in a way, if that makes sense.

Stephen Belleau 35:31

Absolutely, thank you. There was another question about lots of people asking if this was recorded, it is, is there a way to send out the comments afterwards?

Zack Olinger 35:41

Yeah, absolutely. That's one of the things that I kind of made a wrestle about. I know other user group admins did too. So Debbie does have a way for Nicole and drew to export these chats for us. So yes, we can certainly, certainly do that. Because that's a big thing for me, too. I was like, this is a community thing chats a huge part of the community contribution and experience, it's very important that we have that. So they are able to pull that for us. So

Stephen Belleau 36:09

awesome. And then some questions around pagination. We found some resources in the chat here. But I think the sentiment is it would be nice to have a more official resource around how to paginate in postman and get some Doc's around pagination in general. But

Zack Olinger 36:26

yeah, that's a that's an interesting one, as well, because I know well, even the latest. Like I believe that on the developer inst, or patch notes that they have, in their use, their change that they're doing for cursor based and pagination, and showing that in postman can definitely be done. It's really an interesting thing to do programmatically as well. So that I feel could use a little bit more time, just to make sure that we go through like the nuances of that it doesn't need a full, you know, necessarily, like its own thing event, so to speak. But, yeah, for sure. pagination is a thing.

Stephen Belleau 37:12

Awesome. John is asking, Where can we access the recording after it's uploaded? I think it gets sent by email to everyone.

Zack Olinger 37:19

It does, yes. So basically, I let drew know that the recording, evens done, he takes the recording, he goes to somebody else's desk, they do their thing, and then they upload it to YouTube. And provided the link usually takes a few days. And then within a week of this of the event, I write a recap email and send it out. And I make sure that that YouTube link is in the email, and then also in the community post as well. That's in the community forum. So sweet.

Stephen Belleau 37:50

Okay. And then Christo, I think is asking, Is it possible to demo a test of pushing data into Zendesk using a post or put?

Zack Olinger 37:58

Absolutely, yeah, we can. Yeah, we got time to do that now, for sure. We can absolutely do that. So let me go ahead and share my screen again. Okay, so with this? Let's see. So I had a couple of different things. So like, yeah, if we wanted to create a ticket, you know, this is a pretty common thing, depending on the type of developers or application. But more or less, you've got two different ways of getting information. Or, actually, you've got more than that. But you've got these different options here. Primarily, it's going to be post and put are the two API methods that you're going to use or, or HTTP requests that you're going to use to push information or put information into a system. One is for like creation. And then the other is for updating primarily, I'm honestly going to get the two a little bit mixed up as far as which is which. But to demonstrate more or less, it's going to be the same kind of thing. As far as like what's what you need to have in your headers. There is though, something to call out for pushing information into Zendesk and which is this content type. This particular HTTP header is not required when you're doing a get. So if you'll notice here, if I did, if we go back to the list the groups these are the headers here, and we don't have a content type. You know, there's, there's nothing like that here or list a single user headers. There's no content type. Now, though, on the on the post and put, you didn't need a content type, because basically what you're doing is you're sending A JSON body to Zendesk or to somebody that some other service that's receiving it. And since it's an HTTP request, the receiver, the server that's receiving that request needs to know what type of content it's receiving. And so when you set it to content, type application JSON, then it's like, okay, I'm receiving a JSON object. And it's prepared to parse that object and do something with it based upon which endpoint you're you're sending it to. So like, if you try to post like a JSON object to an endpoint, that is like a get only type of endpoint, you'll get an error, it'll be like, I don't understand the response. So the main thing that's different is the content type. So make sure you have this specified. And then the authorization, those are the two main things when you're putting information into Zendesk. And then for the body, it's basically this will be like where the trick kind of comes in, because you can come into the developer docs again, and let's say we wanted to do tickets, right? And we want to, I'm going to hide that, we want to create a ticket. So it's going to tell you right here that we need to post. So that's what we have over here, post. And then it's going to tell us the endpoint. Okay, so I've got that, right there, too. Now, down here, hide this, again, it's gonna say, It's gonna give an example body, like, Okay, this is a generic kind of give you a basic framework for what we need to send to the server when we're creating a ticket. So you could see here, I wonder how well this'll work. Maybe I changed my resolution. So hopefully, it shows up easier. I didn't necessarily like how my screen was showing up on the last recording. So hopefully, this is easier to read. But you can see over here on the example, and then what we have in postman, now I do have some additional things such as the requester. And the group ID. Because I wanted to because I was testing some things. So there's even what I prefer when I'm actually creating a ticket, or even posting to the ticket is using the HTML body, because then we can do DML that makes this we can stylize internal comments a little bit and what is it we want to do? Actually,

I think if we wanted to make the public if we wanted to make the comment, public or not. And I'm going to make it an internal comment. And let's see. And then there's one more, I think, yeah. So let me actually verify that, you can click this beautify button in postman, and it will format your JSON. And it will help you determine whether or not it's formatted properly. So basically, we're ready to go ahead and send this. So hopefully, what should happen is a ticket will be created that says hello from postman, and the body will have some question in it. And it will be as an HTML one, there'll be a header. So we'll send it and we did get a response back. And we can come down here and let me bring this up. Here's the ticket ID. So it's 272. So if we come over to my Zendesk instance, and we go to support. We look up ticket 272. Actually, it's right there. And here's my internal comment that is formatted as a header. So that's basically how we would do that. And then again, like with any of these fields over here, we could do the same, not exactly the same endpoint. But if we wanted to, say update a ticket, right? Ticket, we come over here, and we see that it's a putt instead of a post. And we have a different API endpoint. And then we need to know the ticket ID. So we can come over here to postman. And let's see, I already have a let's see. Let me do this here. Come comment, add comment the ticket. So I'm going to I like to get a little cheat cheeky. So then 272 You right? Where's the ticket? Yeah, we can update that and add another comment. And we're gonna get another response down here. And it's forbidden. Oh, this is because okay, I was testing. Yeah, I was testing my, my O auth. token. So that's a whole different thing. So let's change this real quick. Let me come back over here. And we'll do this. And there you go. So this is, this is what you would see this is kind of the benefit of doing Oh, auth tokens, you can see that the token I was using for OAuth wouldn't allow me to write because the scope, which is good, depending on what you needed to do. However, for this demonstration, I need to switch it back to using my API key, which has readwrite on everything. So now, I should be able to come back, post this comment to 272. On ticket 272. Okay, so with that went through, and we can come back over. And here is the updated comment. So that's two different methods that we went, we did a put to update, and then we did a post to create the ticket. So let's see. I'm gonna go ahead and stop sharing that. Anything else? I mean, I can certainly continue on that path. Sorry, I'm taking a look here at the chat again.

Zack Olinger 46:44

Okay, Zach, here, hey, how are you? Hey, thank

Ahmed Zaid 46:49

you great presentation, I just had a neat trick that I use when I want to change the authorization for a whole collection. So instead of adding the header to individual requests and encoding, I tend to use the auth tab next to the parameters, and then normally inherit the authorization from the whole collection. And then I believe it's possible to add the user name and the password, or in that case, the email or slash talking and the API key in plain text. And that would have postman encoded for you save a few steps.

Zack Olinger 47:36

That would be much easier. So I appreciate that. Let's see. Do you want to walk through that process? Yeah. Okay. So let me go ahead. And so this will be fun. See. So this is what I like, just because I'm in this particular role, certainly don't feel like I know everything by any stretch. So I will open myself up to learning here. And let's do this. All right. So you're going to walk me through.

Ahmed Zaid 48:08

So under the request URL, you can see the tabs, and right now you're at the headers tab, and on the left of it, there is the alt tab. So right now, there is no authorization here. And you can inherit it from the OB its parent, parent leave the parents would be the current folder it is under. So you can keep if you're nesting folders, there, at the very top, you have a collection, you can just leave everything into that one collection to make the switching much easier. Cool. And that collection, then you can select the basic of authentication. If you click on the

Zack Olinger 48:52

folder itself, got it. Okay. I see. And then under here, got you. And then basic. And then Oh, I see. And then you could just put it here and here and it will do it for you.

Ahmed Zaid 49:04

Yeah, so it will do exactly the same thing that you did, and encode it for you here.

Zack Olinger 49:10

Gotcha. Okay, well, everybody that's going to be a pro tip, we'll have to make a call out to like, take a look at the end of the video to do the beginning. Because that's probably going to be the best way to go. So then, I guess if we wanted to go ahead, sorry. I was gonna ask,

49:27

Will that still run those requests individually? Or does it have to be run as a collection,

Zack Olinger 49:34

it'll still allow you to run them individually. So it there may be a collection inside of postman is basically just a collection of API calls that you've grouped together. So it's not something that it's going to execute like one right after the other. Like as a collection, we'll put it that way. So you can see here like I've grouped together like my posts and put a An API calls in, in a collection that I call Congress nature post. And then I have another collection called get, but these are each individual ones. So it's not like it would run all of these, it just would, I would set the authorization here under the parent collection. And then what I could do under that is till each one of these requests to inherit the authorization that it's going to use from right here, it just makes managing the authorization much, lot cleaner and easier. So thank you for that. Hopefully, that that makes sense. My explanation.

Ahmed Zaid 50:44

Yeah, I still believe that learning to encode is a very valuable skill, though, especially for troubleshooting. Because depending on the protocol, or if you're depending on curl, or whatever library you're doing to do the encoding for you. And then you somehow you have special character in your password that will drive you and your developer crazy. So learning to encode yourself is extremely valuable. That's

Zack Olinger 51:10

true, I can totally attest to that. So learning to encode and then decode is well, to troubleshoot. I've absolutely had that where I was like this was working just a minute ago, and why is it not? And it was a simple, yeah, one extra character. And that totally threw it for sure. So yeah, as a developer, single characters matter for sure. It can be a little crazy how much it matters. So

Stephen Belleau 51:39

quickly back to the topic of the collections versus individually panic calls, I believe you can and correct me if I'm wrong. You can click on a folder and actually run everything inside that folder. So if you needed to do things back to back, you can definitely do that.

Zack Olinger 51:56

That's true. I've never had to do that. But I imagined that's probably very true. Yeah, yes, there is there is a run collection. Yeah, there is that option. I've never done that. But that is an option. Which is pretty cool. Because then you can set up automations. Okay.

Zack Olinger 52:21

Let's see. I'm just gonna check something here real quick as far as like the other items that I had on my slides, but yeah, nothing too big. But yeah. Are there any other questions? Or certainly, if there's anything else, like a deeper dive a follow up or anything else that's related to this? I'm open to going into and exploring more and another talk or an extension? Certainly. Yeah, so this last question about resources and the Zendesk, postman Doc's. Honestly, it's about, I think, when it comes to stuff like this, trial and error is about getting kind of into the, into the weeds with it. So that's where my biggest learning has come from is just through experience and trying things and then spending the time to learn what I didn't know. So kind of like what I was doing earlier, where I was, like, I need to base 64 encode this, I had to sit and figure out how to learn how to do that. Apparently, I need to still learn how to do that. But it's an iterative process like that, in my opinion, that's how I have learned how to develop it's a lot, spent a lot of time of just trial and error. And then just being mindful about where I'm playing. So that way, if I happen to error, it's not, you know, catastrophic. So it's Google and chat. GPT. Honestly, I use I use both chat GPT is great for really saving me the time to concise down the information that I would have found with Google Sometimes, though, Google is still my go to. Yeah, that's true, too. Yeah, what type of real world applications folks will be using this as a starting point? Because he I'm curious about that as well. Because I know my own use cases, they will my own reasons for needing to kind of get into this and find out that information and manipulate it and put it into a format that was meaningful to me and my stakeholders. So but yeah, there's a there's a lot there.

Zack Olinger 54:37

Yeah, that's a good. That's a good way to get the list of triggers, and all this kind of thing. One thing I will tell you too, just so you guys are aware that because it is basically like this relationship database on the back end. With Zendesk, there's all of these IDs, right. So there's going to be some information like if you're pulling something from like the groups or brands or even users you And there's some information that you think may be in there it will be, but it may be the IDs. So like, it's like on a ticket like contributors, the requester, the user like the assignee, all of those are IDs, right? You're not going to get like the person's name, you're not going to get the group name, although that's probably what you're looking for. So there's going to be these lookups, right, you're gonna get information from the ticket, get that information about the requester, or, you know, the collaborators or whatever, get that Id go hit the endpoint for, you know, the user to get that information saying grab the name. So there's a lot of these like, you know, looking things up, which is it, which makes sense. It's just something to be aware of, as you get into that. So. Yeah, audits are a huge thing. Yeah, that's the big, big deal, too. It's pretty. Yeah, it's good for that. Like I say, if you want to know what the information or the data looks like inside of Zendesk, that's one way to really do it. I've had to use it for troubleshooting. Why does this HTML comment, look the way that it does? So like, what is the actual? What's the data of the comment? Like, what's the legit? What is it inside the system? Okay, something's not escaped properly. Something of that nature. So it's, it's a good for troubleshooting?

Zack Olinger 56:21

Yeah, definitely. Yeah, it's definitely the best ways. It's easy to get started testing stuff, and then having to code options to be able to kind of take it into a more practical application is good to. Gotcha. Yeah. Yeah, for sure. Yeah. And if you get comfortable for you know that you'll be writing all sorts of Python stuff to do things.

Zack Olinger 56:54

Well, I guess what I'll do here real quick, just to be mindful of everybody's time, I'll share my screen again, real quick to kind of wrap it up if there's no other questions. And just to kind of talk about some things that are coming up here. I was gonna say, actually, I'll leave it on this one real quick, for our next event, that we have a monthly cadence. And I believe that June is actually a month that has five Fridays, and if I recall correctly, and if I remember properly, June 30, is the last one. And that's rolling right into the Fourth of July weekend here in the States. So I'm going to assume most people probably would rather not have the event on the 30th. So I'll do it on June 23. The Friday prior is when we'll have our next meeting. And as far as the topic goes, I'm still actually up for grabs on that. If anybody would like to actually I'll go ahead and stop. Let me do this here real quick. I'm a little all over the place here. Let's see. Okay, yeah. So

Zack Olinger 58:10

yeah. So you want to cover that as in pi? Yeah, we'll figure out what the next things will be. I've thought about covering the OAuth tokens, and then are maybe going further into some of the stuff that we talked about here. But I will definitely make it relevant. So sunshine API, that would be that'd be brand new for me, too. So that would be interesting. Yes. Thank you pagination. That's what it is. I think probably that might be the one pagination. I'm gonna write that down. Because that's applicable. It's a thing. It impacts a lot. Sunshine is definitely a thing too, for sure. So that would be a great topic. I think I'll put that one a little further out. I'll see if I can get somebody from Zendesk to talk about that. I know that. So I'm talking out loud to myself. And sideloading off tokens. Okay, cool. pagination seems to be getting a good vote. Events API. Gotcha. Okay. Cool. And sideloading Sorry, I'm writing I'm taking notes. Okay, perfect. Thank you for that feedback. See, so advice on a better understanding of postman's functionality, good resources. For me again, it was just a matter of just kind of experimenting. This is where I'm probably gonna have to lean on some other of our community members where they may have some good resources that I can help share with you guys and I can certainly look some up. For me, so much of my own personal growth and development. It's just been around like playing with it and seeing what I can, what I can figure out, I spend a lot of my time doing that. And for others, I understand that's not the cup of tea. So I'll see what I can conjure up for you. Yeah, absolutely. Yeah. Visual Learning. YouTube is great for that kind of stuff, for sure. And thank you. So I guess real quick just to end this, to tell you guys about some upcoming things is I didn't mean to launch slack, what I meant to do. All right. So there's a couple of events coming up. And I'll include these links in the email as well. So this has got some community events coming up on the adaptable agent workspace. And then also platform security, which I'm actually pretty interested in as well. And I have questions around this, and other things. But those are the two next community events coming up. Again, those links will be in the email. And then we've already actually covered this about discussion topics for next time. So thank you again, for the feedback. And then of course, thank you for joining. I appreciate everybody's time for sure. Because I know this is a an investment. And and I trust that what we do here is meaningful and impactful for you so so thank you for the time and attention and all of the community and help. So but yeah, if there's nothing else I can give you back your afternoon. So thanks.

That was awesome. Absolutely, for sure. Thank you appreciate everybody's help and the contribution so All right, well, I will like I say within a week, I'll get an email sent out and we'll get the YouTube link and so people can watch the replay. All right, cool. Everyone. Have a great Friday. Good afternoon, or good morning, wherever you're at. See you Bye