All right, we're gonna go ahead and get started. Thank you for being here with us today, especially day two of the official AU conference. So some of you probably had a good time last night. You probably didn't win a lot of money at the table, otherwise you wouldn't be here. So thank you for not being a winner, all right.

My name is Keith White. I'm a technical consultant with Autodesk Consulting. Prior to joining Autodesk as a technical consultant, I was an architect for many years. So I've got about 20, 25 years in the construction and design industry. I worked the light on Revit API, Forge platform, Glue, Field, Naviswork, stuff like that. So I'm joined with Saikat.

Hello everyone. My name is Saikat Bhattacharya. I'm part of the technical consulting team with Keith. My specialization is like, Keith again, with the focus on AEC products and APIs, primarily focusing on the integrations and customizations. I'm based out of San Rafael, been with Autodesk for 12 years, with the prior role of EPA evangelist. And again I've worked with different locations. Right now, I'm based out of California.

Very Very good. All right so you guys have had access to this prior to start. So we got a lot of material. So we're not going to cover the class summary or learning objectives. But we hope at the end of this that you'll have a lot more familiarity with what's possible with the Forge API, and especially BIM 360. So just a real quick show of hands, has anyone in here started dabbling with Forge at all? Awesome. Contractors, anyone in the contracting? Business owners or-- very good, very good, OK, facility owners.

All right, so some of the stuff that I'm specifically going to cover is kind of from a point of view of a person getting started with rest API, with web development kind of stuff from the very beginning. So if you're an experienced web developer or an experienced developer, pardon me for kind of covering some simple material. But I thought it would be helpful to get started just showing some of the things that are going to be helpful for a person getting started.

So looking at Forge, we've got three main things we're going to be talking about today. We've got the Forge platform itself. You've been hearing a lot about Forge. Saikat and I presented this past May in San Francisco at the first ever Forge DevCon. There's another Forge DevCon coming up next year, so, maybe something that you want to be interested in. BIM 360 itself, where it fits inside of the forward platform. And we're also going to look at the BIM 360 user
All right. So looking at the Forge platform itself, you could see that the Forge platform is actually made up of a whole bunch of modules, a whole bunch of data layers where we can access data from many different parts of this Forge API. So we’re looking at this top part right here. And we’re going to be focusing on the integrate and extend module. And we’re going to be looking at the BIM 360 API specifically.

So Forge is really large. Lot of moving parts inside of this, so connecting all of the various Autodesk cloud services with data, data at the center, data behind everything. So that's what we’re connecting with.

**SAIKAT BHATTACHARYA:** Just to add to what Keith said, you can think of Forge like LEGO blocks. It provides you with different parts. And you can create your own thing with these different parts. It provides you a framework. And it also provides you with APIs.

Now these Forge APIs are ourselves by Autodesk, being leveraged to create these solutions that you see on the top layer. So essentially what Autodesk is doing is creating a developer platform, which is both secure, scalable, and kind of meets the requirements of integration aspects.

So when you’re writing code, custom code, or integrating any aspects of your ERP systems, or your enterprise applications, you're essentially leveraging the same platform that Autodesk itself is using and is heavily investing on. And as you can see, like Keith mentioned, BIM 360 is a solution that's built on top of those services, which is Forge.

So again, with BIM 360, the portfolio of products is really growing. Again, because it's based on the principles of Forge, it leverages the APIs that underlying. Like Keith said, we have access to the data just like any BIM 360 product does. So it really opens up your access to that data and make sure that whether you’re creating a solution from ground up, or you are integrating into an existing application, you essentially have access to everything that you would need.

So again, the ecosystem is really thriving because of the open-ended nature of this entire scope of the architecture itself. We have different construction workflows that we have our solutions for. But again, I'm probably repeating myself here, because you have access to the underlying APIs, you can build your own solutions on top of it as well and do the integrations.
At a platform level, the intent is to have BIM 360 Docs as the platform data management layer. And will, again, that would be sitting on top of, that is sitting on top of the Forge layer itself. So again, we go into the details as we move on. But essentially this is just to tell you that it's a growing ecosystem. You have different construction workflows that you deal with on a day to day basis. The APIs allow you to go deep dive and access that data, which is sitting at the core. You have a single source of truth that you can envision for your own ecosystem.

KEITH WHITE: And a big focus of this class is kind of going to be on this stuff over here in the bottom, which is connecting Forge and the BIM 360 API to your construction ecosystem, to your ERP system. So that's what this class is going to be focusing on is that particular connection.

All right. So with the BIM 360 HQ platform, right, so we've talked about Forge. So Forge is sitting up here. And in, a lot of you are familiar already with these different BIM 360 services. So we've got Field, Glue, Plan, and Docs. There’s others there as well. But these are the ones that, especially construction firms, are really familiar with already.

So we're talking today as in the BIM 360, our Account Administration HQ. This is what we're going to be talking about. This is our doorway between the Forge platform and these services. So this is where we're going to be discussing today how you can use the BIM 360 API to make those integrations from your CMIC system, or your SharePoint system, into these services here and help you control projects, clients, contractors, things of that nature.

So, kind of as a primer for what we're going to be talking a little bit more detail the BIM 360 HQ functionality, we have a couple of key roles and we have a couple of key API categories that we want you to be aware of. So with the user roles, you have account admins, these are people that have access to the full account. They see all the projects, all the people, everything inside of it.

And then you also have corporate developers. So these are when you have your own development team internal, and you want to develop your own apps and integrations, that's you. And then you can also invite external, third party developers that you're contracting with them, and you have to give them access to your account. So that's another one of the key user roles, are those external partners.

SAIKAT BHATTACHARYA: So from the API categories, as you can see, we will cover the aspects of how to make secure connections, how do you authenticate your calls when every time you try to access the APIs.
We'll show you how we can create projects within HQ using the API, which is again, BIM 360. And then we'll also show you how to activate services. For example, after you've created a project, you only need Field as a service for a specific project. How do you do that?

The other aspects are adding users to the member directory so that if somebody's at Field, they can actually have an access to all the users who are already part of the project at the HQ or the Account Administration level, and have an easy access to allow project admin, to add users, or remove them from that particular service itself.

We'll also cover the aspects of business units and that's about it.

**KEITH WHITE:** Activation of the services. All right, so I'm going to take you through the BIM 360 User Experience. Has anyone been in this at all yet on the web? Have you seen this? All right, so there's a couple. So that's really good. So for those of you who are not familiar with this, when you log in to, basically it's a web access, so when you log in, you're greeted with this, which is a dashboard. And so this is relative to your account that you've logged into.

So at the very beginning, you have this is your dashboard. It shows you some widgets with some analytical data, number of projects. This is one that's really key, right. It's when your account starts and renews. So this gives you a heads up when you log in. You're good until whatever time your contract runs out. So you won't have any surprises that your services ended and all of a sudden you have project related access issues. So it's very secure. It's private. And this is your gateway into BIM 360.

Now you're going to see on the slide there's some blurry stuff. And this is because these are real people that we're working with. So I've kind of hidden and blurred out their names just to protect their privacy. So with this, we actually are logged in. And this is a list of projects.

So in this particular example, there's 27 projects. Here's their names, gives you some information about how many members are in it, the number of companies, if it's an active project, and when was the last time you signed in. So this is our test account, but it does have real people. So that's why you see that there's few members and few last logins. So you get a list of what your active projects are.

All right you also have a member directory. Here's all the people associated with your projects in your account. So these are all the people that have been added to your projects. You also, over here, you can see how many active users, inactive users, and the relative roles, how
many account admins you have, project admins, stuff like that. So this is you’re, basically, it’s just a heads up of what’s going on with your project, or your account. You can add and remove people. It’s very simple, very straightforward for you to take a look at.

Same thing with companies, you have partner companies that you invite in. So you add users through partner companies and you give them access to your account as well. So when you’re working with subcontractors, or with other people working with you on your projects, this is where you add them in.

And then lastly we get to the account settings. And so you’ll see that with the account settings, if you have a project logo, or account logo, you’d have an account logo here, and then you have a list of who the account administrators are. Really simple to see this stuff.

Business units, so, I had a couple of conversations with the different clients about business units and how all of this work. So if you’re a very large company and you have multiple business units and different time zones, different regions of the country or the world, that’s where this comes into play. So you can track, and this helps with your own analytics, so that you know what products are where and how successful they are based on these business units that are being set up. So this is a categorization method for you to set up your projects in your work.

And then lastly, this, we’re going to spend a lot of time here, today. We’re going to be creating, really we’re going to be creating a number of these things today. But we’re also going to talk about apps and integrations. And so you have an account. You have a BIM 360 HQ account. And then you want to do something to access that account. You want to add or modify that data, or access it.

So each one of the apps or integrations that you want to access your account data, you have to give specific permission to. Once you’ve granted them permission to access your account, and the data they’re in, they start to show up here. So this is your way of seeing what has access to my account and my account data.

So once again, the security of this is really important. So as an account holder, if you’re an account holder and you have internal staff, or if you have external third party people, you have to grant them access to your account data. So that gives you the security that not anyone, even if they’ve gotten some of your information, you still have to grant them permission to access your account. So this is a really good way to secure your data.
So, how do we get started on the BIM 360 HQ APIs? These are a typical set of workflows that you would have to follow. The first one is going to the developer.autodesk.com. And Keith will walk you through the details much later. But this is kind of an overview of the steps that are required for you to start leveraging the BIM 360 APIs.

The first thing we need to do is, like Keith mentioned, when you're creating an app, when you're creating an enterprise integration application, or you're creating a solution from ground up, the first thing you need to do is to have it registered. And for you to be able to access the APIs, the Forge APIs need to know that you are an authorized, or your app is an authorized user for that data.

The first thing you do is you create an app in Forge in developer.autodesk.com. You get a unique set of client ID and a client secret. This client ID and the secret is really tied to your application. So every time you consecutively make calls to the BIM 360 APIs, it'll make sure that the auth token that you get in return of your client ID and secret uniquely identifies you and gives you a validity of 30 minutes, after which you have to again, refresh the token. So this is how every API call needs to have the access token embedded in it so that BIM 360 or Forge really knows that your the authorized one. So that's step one.

Step two is the part that Keith just mentioned, which is you as account holders have really have to give the permission for a specific app to start leveraging your data. So in a way, even if I have created a Forge API app, I have my client ID and secret, I cannot access your account ID just because I have your go ahead. You have to give me access on my application and access so that I can start pulling in data or pushing in data.

So once as an account admin you do that step, you're basically tying in your Forge credentials with BIM 360 account, and you're good to go. So this is all you need to get started.

Once you have that set up the next thing would be start leveraging the BIM 360 APIs to do the account level administration activities, like he talked about creating projects. He talked about adding member directory inside of BIM 360. And we also talked about creating companies, business units, activation of services. So those are the account level actions that you can do once you have your app created and authorized to work on that.

And after you have the account level administration tasks done, you can then go down to each of the specific services. You can go to Field. You can go to Glue, leverage those APIs and
perform your service level tasks that you need to work with. So this is an overview. We will go deep down into the details, in fact we'll see some code where we'll actually walk you through the steps on how this happens. You'll see some live code in action. But this is at an overview level.

**KEITH WHITE:** Awesome, all right, so now we're going to kind of start our API deep dive and demo. So the very first thing we're going to do is we need to create our Forge account and an app. Right, so before you can create your app, you have to have a Forge account. So to do that you go to developer.autodesk.com. There's a sign up button. It's very easy. You provide some personal information. And you have your account created. It's very, very easy to do. And by the way, we will be providing the slide deck after the presentation on the class page. So you'll be able to get all of this.

So once you have your, once you have your account set up, you're greeted with this page. It should be very similar to this. And it says, hey, you have an account but you don't have any apps yet. What do you want to do? So what you do is you create your first app. So whenever you create your first app, you're going to be presented with a number of BIM 360 or Forge APIs.

And so at the time that I created this screenshot, these were the three that were available. There are more available now. And there are more that are available coming in the future as they continue to roll Forge forward. So you provide an app name and an app description to your app. And that's what we discuss here. You provide a callback URL. And if you have yours, it's optional. But you can specify your own website, your URL as well.

So once you provide that very simple information, right, and this, by the way, this callback URL is for the three-legged authentication. Gosh, I can never say that word. So at the end of this, you click Create App. And now you have an app. Very, very simple to create, you got a named app. Your app will have some information associated with it, which is a key and secret associated with your particular app. So it's really that simple to just say, here's my app. Here's a name. Give me some identifying information so that I can now start to access Forge.

So you as a developer, you as a corporate owner, an account holder, you have your own app, or a third party person already has their own account and ready to go to integrate for you into your account. So you start off, you need to add an integration to your account to allow either your own company to access the data or that third party. And this is how you do it.
So when you're logged into the enterprise account, we go down to our bottom integrations button here. And you'll see that currently, in this screenshot, there are no integrations already added to the account. So if you're a brand new account holder, this is what it's going to look like for you.

So you click Add Integration. And then you're presented with a screen that says, I'm the developer. That means it's internal. You're going to give yourself access, or you're going to invite a developer. So we're going a walk through the I'm the developer, first. You're doing this for yourself.

So once you click Next, you're going to be presented with some information, relative to adding an integration. I've blurred out some stuff just for privacy sake as well, but you have a BIM 360 account ID. This is associated with your account. You need to make sure that you check this check box that says, I have saved the account ID information securely. You want to keep track of that ID because as that's what you're going to need later. You can come back to this and see what your ID is if you forget, but you really should take note of it and secure this.

You also have a Forge client ID. So when you create your own app, it's going to generate this for you, right. So a third party developer, this is going to be important because they have their own Forge client ID, or they need your Forge client ID. So you give it a name. You give it a description. You add a logo. And then you click Save. So this becomes your key to accessing the data. And when you're done, it presents you with this dialog that says, congratulations, your app is now active. And this is what it looks like.

So now inside of this app, you can go to Details, and I don't think I have a screenshot of it, but you can go to Details and you can get your key in secret. And that's what you pass through in all of your API calls. You're passing in the account ID and the key in secret. And that's what tells the Forge API, OK, this is a good app. It has access to this level of data. And we're and we're good to go.

All right, so the second option is you're an account holder, and you're going to add an integration that's from a third party developer. So you're going to allow somebody into your system. So you click on this radio button. It pops up an email address. So you type in an email address. And you click Send. And then it'll present you with a dialogue that says, invitation has been sent to, gives you the email address. And if you need to, you can click Resend the Invitation if the person doesn't get it, or if the email account doesn't get it.
Seems like I, feel like I missed a slide, there. OK, no, no, no, I'm sorry. So this is an email, right. So this is me kind of walking you through this. So this is what it looks like to the third party developer once they get the email. They get an invitation. They get a Get Started. They have a Get Started button inside the email invitation. You open it up. There's some information that's already populated, right. You check the checkbox. You've secured the information. You type in your information, provide a logo and click Save.

And now your external app has been granted access to that account holder for the level of permission and the data that they wanted you to have. There's a couple more steps in there that you'll see. But that gives you the picture.

In order to gain access to the account level information, the account holder really has to grant you that level of permission, either internal or external. So security is a real big issue because we want to make sure that everything is secure and safe. So at the end, this is what it would look like, again, for you to add a number of different integrations.

So now we're going to start to go into the API itself as a deep dive. So here, I'm going to switch over. We're going to go to do a live demo just so you can kind of see this in action how easy it. Has anyone taken a look at the Forge APIs at all yet on the web? It's a real easy way to navigate.

All right, so all I've done is gone to developer.autodesk.com. And inside of here, this is where you would click Sign Up, or Sign In after the fact, if you want to access your developer account. Menu's at the very top. There's a lot of questions about pricing. I'm not the person to answer questions about pricing. However, there is a pricing link here. You can click on that you can see what all of the pricing information is there.

So these are the current available BIM 360, or Forge APIs here on the left. You'll see that the BIM 360 APIs still listed under beta, but we're using this. This is available for you, so this is what you have access to.

So you click on this link. And then what you have here on the left is you have basically a navigation menu or panel. You have Overview, Step-by-Step Tutorials, and API References. So if you are brand new to this, this is a very helpful resource for you. So with the overview you can see some of these images we've stolen for our presentation, or kind of reused them in a similar fashion. But it's a very simple browsing experience for you to get information that you
So what are the APIs basics, right? If you're a desktop developer, this is where I struggled, going from pure desktop development to something that's using web development, and rest calls, and rest API, this is going to be helpful to you, just understand the basic premises. It's really straightforward and easy to do.

SAIKAT BHATTACHARYA: Do you want to increase the font size?

KEITH WHITE: Yeah, let's see if I can make this a little bit larger so you can see. There we go. All right. So you have your Field Guide. This tells you what's available through the BIM 360 API. We're talking about projects, companies, users, and business units. So again, this is all very straightforward, gives you some ideas and understanding of what the attributes or parameters that you're going to be working with, right. What's an ID? What's the account ID? Things of that nature. What's the format of these parameters that's going to be expected?

So this is a really helpful resource for you to understand what information you need and what you're going to be interacting with. All right so that's the overview. Step-by-Step Tutorials are really helpful as well. So if you want to get started and see, how do I do something? How to access a BIM 360 account? How to create a new project, new company, activate a service? Each one of these are step-by-step tutorials for you to work through the process. So it's very simple.

You should be comfortable diving into this and exploring and finding your way through it. Once you do it once, you start to see just how easy it is. And it's really comfortable to do. Then for me, the most important thing now is the API references. And so the API reference is broken up into each one of those main categories we've talked about earlier. We have projects. We have companies. We have users, business units, and then jobs, right.

So each one of these major categories has API information. So to post a project, to add a project, create a new project, this is how you do this. This is your method, right. So it gives you your address. The supposed method gives you your endpoint that you're going to send the data to.

It also lets you know what the required header information is, what's the required parameters. We’re going to talk about some of that structure in just a few minutes so you can kind of get
understanding. What's the body structure of the REST request that you're going to be sending.

And so you can see here that it tells you that the name of the project, and in this case, project that you're going to be creating, it's a required attribute. You have to send the name. However, the service types associated with it is not required. But if you need or if you're tracking service types, you can push that in as well. And this tells you what kind of data it is. It's a string, right. And this gives you a description.

SAIKAT BHATTACHARYA: So essentially what that means is you can actually create a BIM 360 project without having any services activated. So it's still a shell which doesn't have any activation of any services. It's still out there.

KEITH WHITE: You're going to hear us later recommend that at a minimum, even if you don't plan or know that you're going to use Field, Glue, Docs, any of that at this point, you go ahead and create your project, that way is easy to activate a service after the fact, because the hard work is done that the project has already been created.

So again, each one of these APIs methods to do something with your account data, create data, edit data, do something to it, is very well spelled out here and explained to any user. And so with projects, you can post a project, add a project, you can get a list of your projects. You can update projects.

So you basically have these typical methods of editing and creating data, and it includes all of these. And I'm not going to go through this. I just wanted you to see how easy it is to get to the information you need when you've never been here before. It's really helpful bit of information, very easy to navigate as well. And all of this is available at developer.autodesk.com and then under the APIs fly out.

All right, and here we will make our way back to the presentation. All right, so again, if you're an experienced developer, I'm going to apologize because this is very elementary. But I wanted to do this just so that the people who don't have experience in this understand it's really easy. Don't be afraid of this.

SAIKAT BHATTACHARYA: Can we do a quick show of hands of who is a developer, who's written some .NET code in the past, and versus who hasn't? So this is your written code.

KEITH WHITE: I apologize.
SAIKAT BHATTACHARYA: No, that's OK.

KEITH WHITE: So this is very elementary. And this is something that helped me. You got to remember, I started off in life as an architect. I was in the construction industry. I was a user of software, became a software developer. So some of this transition was a little bit difficult until I realized just how easy it was. All right, so using REST, right.

If I was developing something for Revit, I have all of the information available to me, based on the element that I've selected everything that's available to me right away. Well, with REST it's a little bit different. You have to send a request to a web service. That web service has to get your request. It has to accept your request. Yes, it's formatted correctly. Everything is there. Let me do something. And let me send you data back. And it all happens over a push and pull from the cloud.

So using REST, we're basically, all we're going to do is formulate a request and send it. So we're sending this to Forge. Forge is going to process it. And then it's going to send us something back in an adjacent string. So it's just a JavaScript Object Notation. It's just basically a long string. It's very similar to XML if you've ever seen it.

So we have up here, we have the Autodesk Forge platform. And down here we have a multitude of devices, whether it's a desktop computer, a tablet, or a server somewhere in your environment. So how do we get these devices, whatever they are, connected to the Forge platform? We do it through the REST API and pushing and pulling requests and responses. So it's really pretty straightforward.

So this is an example of a request and a response. So this is the request that we're building. So we're basically we're going to pass, we're going to use kind of a long URL, that we're building up the request in URL. It's going to include, the way that we're doing this is we're going to send a header, so it's got some authorisation or bearer in our token.

So when we send and we authenticate with Forge that were a valid user, where passing our account client ID and secret, we're also passing in the account ID that we're accessing. It validates that yes, we're our active user and yes we have access to this data for this account. And so that's what we're passing in here is that information.

We concatenate. We build up this string based on a specific base URL. So this is the base
URL for sending request to Forge. And then we add to that the endpoint for the very specific method that we’re going to be posting our request to. So in this case, we’re going to hit the account. We’re going to pass in the account ID for the account that we’re accessing. And then we’re going to do something with companies. We’re going to get a list of companies back associated with that account.

And then we’re going to pass in, we’re going to add to that a number of parameters or attributes. And so in this case, it's just the limit and offset which says, we want to start at zero and we want to get the first 20 of these companies to come back. So you can specify ranges of things to increment your data however you need. And this is just a very simple example.

So we send that up to Forge. Forge processes it and sends us back a response. And this is what this response looks like. And it's basically one long run-on sentence, very similar XML, but it tells us this is the ID that we’re working on. And by the way, this is not a real ID. It's just me typing in numbers. This is the account ID, right. This is the name of the company, right. AEC technology consultants, what's the trade. So it starts to give us this information.

Then if you want to make this more human readable, you can post this into an adjacent visualizer. And it just basically breaks it up into these objects. And you can see that this is the parameter. This is a value, parameter, value. So this is all this doing. It's sending it back. And then you just parse that information. It's really very simple to do. You just need to understand how to build up your objects whenever you’re writing your code to handle that data as it gets serialized and deserialized.

All right, so now, having that very generic explanation of build a request, all we’re going to do is basically build out a real long URL with some parameters in strings, and then we’re going to post that to Forge. We’re going to get a response back in adjacent string. And that adjacent string, you can then do something with. So you can review that data, iterate over it, and decide to change it, push back a different request with a new bit of information. So this is how this works.

And so at this point we’re going to walk you through a code demo of a project of how you can create projects, companies, users. So this is how you would connect your ERP system, if a SharePoint, or if it’s CMIC. Whatever it is, when you create a project on your environment, you can trigger an event that pushes this data right up to the Forge BIM 360 platform as well so that you don't have to enter the data twice.
SAIKAT BHATTACHARYA: Thanks Keith. So essentially, like Keith just mentioned, and again for a lot of the audience here who have not seen .NET code this might feel overwhelming. I think if you can walk out of this class knowing that there is a lot of standard code that's out there, that your developers or yourself can look at, leverage, again, we'll be sharing some of these code snippets that you can also bake in with your sample code that you're trying to write.

You can essentially come up with a small list of tasks in your head so that you know, essentially, that these are the things that you need to do to make an authenticated call, get the data, fetch the data, or write something back to it. And again, I apologize because this might be a little bit of the weeds because a lot of the people here are not really written any .NET code. But I think for the audience who have written some code, again, this is for you to say that it's, again, very easy, it's REST. And again, you've seen how the format looks like, what are the APIs calls, what do you pass in, and what do you get.

So at this point, what we have done is we have created a very simple esp.net application. You can imagine this to be an application that you are building up from ground up, or it's an application which is part of your ecosystem, which is constantly fetching, or pushing, or pulling the data between your ERP system and the BIM 360 Forge cloud itself.

So you can have multiple use cases that you can fit for this demo. We said, let's create a simple esp.net application. Let's try and imagine if you are trying to recreate BIM 360 by ourselves. Because if you roll back for the first 10 minutes of this class, we talked about how Forge is giving you access to the web services. And BIM 360 itself is leveraging those web services. So what if we create in the esp.net application, which accesses the same web services, and you kind of have the same functionality within an esp.net application that's all yours. You don't have to go into BIM 360.

So what we have done here is, again, we're trying to tie-in things that you've seen in the slides in the actual call. So you might have a jog in your mind between what you saw in a few other slides before this, and what you will see now embedded in the code itself.

So for example, like Keith mentioned, first is the default URL. We have kind of set this up at a global constant, saying, we know we have to constantly work with this endpoint. Let's just create as a constant. We have also included the specific endpoint for the authentication.

And again, this is part of the Forge web service, this is not BIM 360 at this point. This is the underlying Forge authentication that you would have to do for any app that you create,
irrespective of whether it's a BIM 360 app, or a recap app, or anything that's out there, or just the viewer.

So the first endpoint is the authentication endpoint. And the next one is very specific to BIM 360. Again, we have set these up at a global level so that we can constantly use these instead of having to type in or write in in every instance that we're using this. OK.

What we've also done is we have rolled you, we have talked through how you can create an application, get it authorized to work with your account, and we need to tie all of this together in the code as well. So the three things that you see here are, again, we are tying in the account ID, which is very specific to the account that we are working with.

And in your case, it will be your own account. So your BIM 360 HQ, whenever you log in you enable an app. You've given authorization for an app to start leveraging your data. You'll get the account ID just like Keith had shown you. You embed that in the account ID. And then you put in the client ID and secret. Again, we need this because we need to have an authorized ticket on an auth token every time you make an app API call.

So let me start up Internet Explorer. What we have done here is created a simple esp.net application, like I mentioned before. The UI is very esp.net specific. But the functionality is exactly what you have in BIM 360 HQ. Again, we have not baked the entire BIM 360 HQ user interface functionality into here, but given you enough pieces that you can imagine that this is how HQ works with the APIs working under the hood.

So for example, let me start over the project. And in this case, let me say, we are creating an extension of this portal itself. Let's say it's a convention center. Again, this drop-down is something that's a smaller set of what HQ or BIM 360 really supports, but just for demonstration purposes, and let's say this is a convention center. Let's just say we are building up only few extensions, it's few convention rooms at this point. This is the budget that we have. Let's just go with the contract type. Again, these are some things that you would find in the user interface itself.

We now say, when is the start date of the project. Let's assume the project starts off today. When is the end date? Again, Rome wasn't built in a day. Let's assume we will get this done within a month, which is a month from now. And again, job ID, lets for now, again, this is where you would tie-in, if you have job ID in your ERP system, this is where you would put that data
in so that everything is connected together in your enterprise. So for our sample applications, let's just say its 11/16/2016. Project address, let me just put in our San Rafael headquarters address.

**KEITH WHITE:** So while Saikat types and the address and this bit of information, imagine if your ERP system would trigger an event and collect all this data, you enter it once and it goes to your ERP system as well as Forge. So that's the idea. You wouldn't have to type this in multiple times. You're doing it once, collecting that data. And then we're going to use, we're showing you this in Fields, but you're going to pass it through as Attributes behind the scenes.

**SAIKAT BHATTACHARYA:** So once I have created the project, I can actually just go in and say Create the Project. This is where we were saying, you can always have a shell BIM 360 project without having any of these services, like Field, or Glue, or Docs enabled.

At this point, for the purposes of this demonstration, let me say, it's a BIM 360 Field project. And let me just say that I am the admin of this specific service within this project. And this will enable us to activate the service. To activate the service, you would see that you need End Project Admin. And this is exactly what I'm trying to do with the UI, with the custom UI.

So once I click on Create Project, this is where you'll see a lot of Visual Studio, which was used as an IDE for developing this code. Again, it's an esp.net code. We'll be leveraging some tools, like the RestSharp Libraries, which make it very easy for us to start working with the REST APIs.

But as you can see, the first thing we did was on the click of a button, I put in some custom functionality where I'm actually accessing the client ID and the secret. We need the client ID and the secret, again, repeating, so that we can get a valid auth token from Forge. So let me walk you through the first step.

What we have done in this application is we've actually created some services. So when you go back to the documentation, the way Keith described, we have users. We have projects. We have companies. We have kind of done the same thing the code level itself, saying, let's create services for everything that we might need.

So the first thing we do here is get the valid authentication. We are passing in the default URL, which is the developer.autodesk.com with the API end point, passing on the client ID and secret. And this is essentially where we are instantiating a communication service, but this is
the real gist of where I want everybody to focus on.

This is live code that you saw. This is REST API. Again, we are using RestSharp Libraries, so a lot of the details we can keep ourselves at a distance with. At the very conceptual level for the non-developers, this works this way. If you're using RestSharp, you first of all need to create a client. The client needs to know where it needs to access, which end point.

And then you also need to create a request. The request will have parameters, header, everything else that you need to bundle it up. And you can create this packet and send it off and say, OK, this is the data that I'm passing on to you as BIM 360 or Forge API. And now I'm expecting something back from you, which would be JSON. That's the file format that you would, that's the format that you would get as a written response.

So in this case, like Keith mentioned, again, we have REST request. We have the login URL, which is authentication. And again, this is a Forge authentication. This has nothing to do with BIM 360 at this point. And we are adding in all the parameters, again the client ID, the secret, credentials.

We also have to deal with something called scope. When we are working with an application, Forge needs to know, are you trying to read data, or are you trying to read and write data. So in this case, we are saying because we are creating projects, give us everything. Again, you might want to restrict based on apps and security and all that. This gives you a finer detail of what you really need to do.

And again, this is very standard code where you can see that you can actually do a search on the RestSharp page. You can see that a lot of this code already exists there. This is where we have passed on the request. We had said, REST client, please go and execute with this request. And the request has everything bundled up together. What we got back was content from the Forge API. And like Keith had mentioned, everything is just JSON. It's like this one piece of string which has everything else.

But again using the adjacent serializer, what you can do is we can easily make this into a nice little object so that we can start accessing the IDs, the attributes that come out of it. So as you can see, I'm sorry this is a little smaller for you, but the first attribute, there is the access token. That is the token that we got in response from Forge. Forge got this response with a client ID and secret, realized this is an authenticated application, give back an access token.
This access token has a validity of 30 minutes. So after 30 minutes, if you have to make another call, you would again have to get the authentication token back. But every time between now and the next 30 minutes, you would use the same authentication token and keep passing in while you're making the API calls. So it's like a ticket that says, OK, this application is valid.

And again going down the screen, this is where we are saying, OK, now we have a valid authentication ticket. Forge realizes this app is absolutely authorized, authenticated, and authorized, because we have done the authentication at the HQ level, because we've added that as an application here itself. Now what is it that you need from us?

So at this point we are saying, please create a project for us. What do we do? All we have done is we are passing in the HQ account ID. We are passing in everything that the UI fields have captured. And I get I'm just quickly showing you everything that we had captured from the UI, stepping into the code.

Again, a lot of this will now look very similar because we are kind of doing the same thing, which is we create a REST request. We bundle of parameters, header information, and everything that needs to be part of that bundle. And then we are throwing it out of the fence and saying, Forge, HQ, capture it, give me something back.

So essentially at this point, we are just making sure we have the header. We had all the parameters. And all these values are, as you can recall, these are from the UI that I just typed in. In your enterprise application, if you already have an ERP application, you would essentially just fetch that data. There's no user writing in anything there.

And again, going back to the standard execute command, you can see that I'm making this request with the client, passing in the request. It went in, gave me a response. I can see that, again, the BIM 360 API, at this point, gave me a long JSON and said, this is the project details that came out of your request. And again, going back into all of this.

The next step here is, if you remember, I added myself as a project admin so that Field as a service could be activated. So this is the part we're saying, OK, now the shell project is created. Let me activate Field programmatically.

Again, in this case, we are tracking which service was activated. My email address was set to the field of user ID. And in this case what we are doing is we are saying, OK, let's get all the
companies that exist within this account. The company name that I typed in is Autodesk, which already exists.

All I'm doing here is something similar that actually Keith had in his slide, which was fetching all the companies. So all I'm doing here is not a post, it's a get. I'm getting all the companies back. I'm saying this is the end point that I need. This is the account ID. I'm passing in, again, the auth token that I fetched as the first step, because that's my ticket to make sure that it's all valid.

I'm passing in the additional parameters. I'm getting a request back. And as you can see, the content really gives me a longer JSON because it now has all the list of companies that are part of this Forge account. And in this case I'm just looping through all of them, making sure that the name aligns. Again, this is a very simplistic version.

And at this point I'm saying, OK, once I have the email address and the company, let me go ahead and make another API call to activate the service. So again, as you can see, this is a different end point because here we are saying this is the account ID. This is the project ID. This is the user. This is the project admin role that I want saikat.bhattacharya.list.com to have. This is the company ID and the email address. The service type here is Field, because that's exactly what I want, for me, to be added up as a project admin.

And once this goes through, again, the next part is with, again, this is sample code we haven't done enough exception handling. But as you can see, if I go back to BIM 360 HQ, this is the account that I was trying to work against. If I do a refresh, go into projects, you can see that Venetian project got created. You can see that everything that I fed in through the UI is already part of the user interface itself. In terms of services, I already have a green service, which is the Field got activated. Because I passed in my user ID, I got added as a project admin inside of Field. And again, company Autodesk got added because I am one of the project admins for this project.

And again, this might be really overwhelming because we are stepping through REST code, but again, when you walk out of this class, at least you know the steps of, OK, there is something called REST. There is a RestSharp Library. There is some other libraries that I can use. But essentially I have to create a bundle. And the bundle has request, which has headers. It has parameters.

Pass it on, it gives me a JSON. And I can deal with it. And its fairly simple. Once you do it
once, if you saw, we were kind of reusing the same code again and again. It was the end point and the parameters were different, but it's essentially the same code that we make for every API call.

KEITH WHITE: We're running short on time.

SAIKAT BHATTACHARYA: Yep. So the next step, again, would be creating a company. We can quickly say, let me just say there is ACME Plumbers. And I will not go through the details of the code at this point. Just say these are plumbing services, and just type in some, because we're running out of time, just say there is some location. Again, ERP data, you would essentially get in from your ERP system.

When I see create company, again, essentially, the same thing. We already have the authentication token. We come back and say, let me create a company. And create companies, again, we are making a post. We are saying the end point, like I said would be different, which is the account ID and companies. We are passing in the header information, which is that access token.

And we're passing in the name of the company and their trade. And once we do that here, we are actually making another call saying, let me go ahead and edit the company, because I, all of a sudden, have some different, some additional information that I want to pass. And this is a patch. Because the company already got created, we are patching in more information at this point.

So there is Post, and Get, and Patch, and some other styles which are part of REST. And again, going through all of this. Once this is created, I can go back to our BIM 360. And you can see ACME Plumbers got added.

And we also have something similar, which I will not step through the code but quickly show you from the experience standpoint, which is, you can also programmatically add the users like we talked about. So in this case, let's just say I'm typing in my personal email address. And I'm just saying, OK, I just added ACME Plumbers, Add a User. Once I go through the same set of code, I can go back into users. And you can see that I have the user created with ACME Plumbers.

And again, this is a sample esp.net application. Imagine this will be an integration app, which is fetching in data from ERP and pushing it to be authorized as BIM 360.
KEITH WHITE: All right. All right, so one of the things, we just kind of went quickly through that, but when you get started with your ERP system integration, you need to map out where the data is coming from, where the data is going to, what data do you need. And so this is a couple of examples of creating a new project, right.

So when you create a new project, you need to get your access token to work through the stuff the Saikat just walk you through. But these are the steps that you work through with your specific ERP system. What event is triggered, you authenticate. Does a project already exist? If yes, you do a patch. If no, you push the project to Forge. So you need to do some simple mapping for each one of these processes that you do, so that whenever you get started, you actually have something in front of you that you have a path to follow.

So the same thing with the company, it's very simple, very similar. Events triggered with your system, you authenticate. Does a company already exist? If not, push it, do something else along the way. And then the last thing with the users, it's very similar. You just do these steps over and over and over for each one of these data points that you care about.

All right, so here's some things that you need to think about as far as what are the benefits of doing this and what are the things you need to consider. So as far as benefits are concerned, you want to reduce your human interaction. The more times you have to enter data, the more chances you are going to mess it up. It's just going to happen. So if you can enter the data once, either in your ERP system or in a separate application like what Saikat was just showing, and that triggers a call to your ERP system to add the project and information and do Forge at the same time. If you do it once, that's what you want to do.

You also want to, the idea, the benefit is that you've now integrated your data and it's the same data in both of your systems, in your local system you need for your companies to run and operate, as well as the cloud data that we're going to give you access to access that data and work with.

SAIKAT BHATTACHARYA: Yeah, so essentially the other things that you would want to [INAUDIBLE] the benefits are often we have worked with customers where they have said, we actually have a separate team which goes and activates the services and we give them like this log. So that kind of a manual intervention, really cannot, is avoided, and the delays. So there have been customers who have actually said, it takes about a week because they already have a lot of work. All of that is done.
We've also heard customer success stories where project admins have all of a sudden started saying that, when I'm assigned to a project, I have everything set up. In fact, to the point that I don't remember how to use the user interface to get things done. It's at a point where they get started with the project, everything is consistent across projects, the experience is the same, and they have already everything set up from programmatically from the back end so that they can just go and get started working on the actual project instead of setting it up.

KEITH WHITE: OK, all right, so here are some things to consider. So whenever you are creating your project, right, in your ERP system, go ahead and create your project in the Forge, BIM 360 platform as well. It doesn't have to have any services activated, but the data's there, the project is there, and is ready and waiting for you to do your additional work. That way, you don't have to go back and try to read data. You've already done this once, go ahead do it then.

Understand and map your data between your ecosystem and your HQ project, right. So just those maps we saw earlier, figure out what's required by HQ, figure out what you have in your ERP system. We've seen with customers that sometimes they don't track stuff in their ERP system that HQ require, so you'll have to figure out how do I get that data to HQ that it needs.

You also want to focus on your data quality, right. So this is where this helps, because when you enter the data once, you're going to get that data in Forge as well. And the analytics and the data that's available to you in Forge later will become very valuable as to what you're actually doing in the field itself.

SAIKAT BHATTACHARYA: And again, with the users, you would want to make sure that there are some accounts that we have worked with where they actually have printers assigned to users. So you don't want to really get everybody into BIM 360, you should actually have project members being part of the user member directory and not having everything out there. So if somebody has 15,000, and if 3,000 of them are actually some devices or some users that were added in their Active Directory, you really want to make sure that there is some parity that's happening before you get the data inside of HQ.

KEITH WHITE: This is something we haven't really talked about just yet. We've talked about kind of pushing data from your ERP system up to the Forge platform. But if you are doing kind of a synchronization, both directions, you need to consider kind of a just in time, or moments in time synchronization. You don't want to do full synchronizations every hour of every day or whatever.
The more projects you have and the more data you have, in any service, it's going to take time to process that information. So just kind of keep that in mind because most people say, we want everything now all the time. Can we get this every hour or every 15 minutes? And the reality is, you may be able to do that in the beginning but you're probably not going be able to do that for long.

Another thing to keep in mind is, and this is something that I realized on a project that we were working on, is you need to pay particular attention to the documentation because there are some nuances in there. It's really well documented APIs, but you need to read it carefully because business units that I talked about earlier, if you have east coast, west coast, whatever business structures that you have for your company, when you go to Add a New Business Unit, if you don't read the documentation, you'll think, I'll just patch in a new business unit. But the way it works is, whenever you update the business units, you update the entire business unit structure. So when you replace just one, you actually override all the remaining, all the previous ones. So what you need to do is you need to get the business units, hold them in your memory, add one in, slip it into the array, and then push the whole array back up. So it's documented, but you just need to be aware of it that those are the kind of things that you just need to be aware of for yourself.

The next part of Helpful Tools and Resources, this is from my point of view, personally learning a lot of this stuff. The first thing you're going to do is, if you think something is broken, is it my code, is it the service. This is a cloud service. Is it up? Is it running? You have a health.autodesk.com. And in there you have all of our cloud services with green, red, or yellow to let you know what's happening with the service, if it's available or if it's got any degradation. You can do the same thing for the BIM 360 or the Forge APIs as well. So on the developer.autodesk.com site, you have a Support Page and API status link. And you can see that specific APIs have their own services. So they may be affected as well. It may not be the service that you're accessing through Forge or through the other cloud services. So it could be either one of those two places that may be affected. So it may not be your code. It may be that. Or it maybe your code and it's not that. So it gives you kind of some pointers.

Another thing that I use a lot for testing is I use Postman. It's a Chrome application extension. It allows us to build up our requests in a test environment, push our parameters up and get
responses, visual, right in front of us. So I like Postman an awful lot and I use it all the time.

The other thing is there’s long JSON strings that you get back from your Forge requests returns. I started off with this one, which was jsonviewer.stack.hu. And I was looking at you paste into a text tab, that big, long JSON string. And you click on this viewer tab and makes it human readable.

My favorite now is this jsoneditoronline.org. You paste in your JSON string and on the right side, you can see the objects. If there’s something wrong, if I'm trying to build a JSON string and I paste it into here and there's something wrong, it'll tell you where the errors are, which is really nice to do, to use.

And then we've also, in this demonstration, we've used the RestSharp Libraries. So it basically abstracts all of those long web service calls and stuff. And we’re just adding in data as parameter objects. And then we also use the Newtonsoft JSON serializer and deserializer assemblies as well. So these are real helpful as well.

So how do you begin? Where do you begin? All right, so we've looked at this earlier. You start off with developer.autodesk.com for all of your BIM 360 API documentation. There is a BIM 360 API forum. This is the link to that forum so you can get some help from other users and Autodesk staff as well.

Each one of our partners, you can request your own sandbox. So if you wanted start testing this, get in contact with the Forge team, and they'll grant you, depending on whatever your access is, a sandbox so you could start testing this on your own. Forge.autodesk.com is also the parent website. If you need help, this is a email address that you can send your question. And you'll get a response from Autodesk staff at forge.help.autodesk.com.

We're going to be providing these examples that you've seen here, this slide, on the AU page as well. And then, don't forget, the Forge DevCon in 2017 is going to be in San Francisco, June 27th and 28th, I believe. So this is where you'll get all things Forge, not just BIM 360 HQ, but you'll also get the viewer and everything else related to Forge.

And so with that, are there any questions, anything? We’re kind of running on time. So we can hang around for a few minutes and answer questions. Question?

AUDIENCE: [INAUDIBLE]
SAIKAT

No, so the non-admin users right now have to be handled at the service level. So using the service level APIs, you would have to add the non-admins to the project in that service. It's not something that the HQ is provisioning at this point.

KEITH WHITE:

Questions? Yes.

AUDIENCE:

You’re creating projects inside Forge, is there a way that we can have them create [INAUDIBLE]

KEITH WHITE:

Yeah, yeah. So that’s, what we’ve had happen with other clients, as they have their own accounting software, they have their own ERP system, and either that system triggers an event that hits a console app or something that we built that pushes this, or they’ve created their own app that pushes data in two directions, one to their accounting system, and to Forge. So it just depends on your starting point.

AUDIENCE:

If you have template projects, can you program to copy [INAUDIBLE]

KEITH WHITE:

Field template projects? Yes. Yeah, now, it’s an option now that you can specify the ID of that specific template project. So when you activate the Field service, you can specify the project ID, or template project ID. And it'll create it based on that template.

SAIKAT

This was exposed about a month back. It’s a new enhancement.

BHATTACHARYA:

KEITH WHITE:

OK, you were first.

AUDIENCE:

Is there a release date [INAUDIBLE]

KEITH WHITE:

You can use it now.

AUDIENCE:

[INAUDIBLE]

KEITH WHITE:

So usually the intent is always to make sure that things that are out there doesn’t break. So the intent always is that things that are out there don’t break. So you can always, I think there are definite, you can contact the Forge Developer Platform Group. I think they call it the FPD. They'll hook you up with some notifications, in terms of what's coming and what's not, so that you have an early preview of what's coming.

KEITH WHITE:

Yeah.
SAIKAT BHATTACHARYA: So I would definitely ask you to contact Stephen Preston. He's part of the Forge Partner Developer Group. He'll make sure you're connected to the right people.

KEITH WHITE: Yeah, and you can also see here that within the API, there's versions. And I meant to talk about it but I totally forgot. But up in the top right, you see the version. This is version 1. So the idea, what Saikat was just mentioning, is that when you make a call to version 1, that version 1 should never change. Right, so we're in version 1.

So when version 2 comes out and they make updates, that's for version 2. So if you want to make access, or make use of those updated calls, you specify V2 in your endpoint address. And that's to let to reference those. So the idea is that previous versions don't change with updated versions. That's a good fear. You had a question?

AUDIENCE: In particular the problem I'm running into right now is we're using a third party for commissioning. And they're using CX alloy and things like that. Is there a way to, even just to read that information to pull it in?

KEITH WHITE: I'm not familiar with CX.

AUDIENCE: It's a cloud-based [INAUDIBLE].

KEITH WHITE: OK.

SAIKAT BHATTACHARYA: So I think that the answer would depend on the APIs, if they're exposed with that application.

KEITH WHITE: You would essentially write this app, which will fetch and using that API, and then use the API that we showed to do that connection.

And it may be a good thing to say this, is that what we're talking about is not a button or a service that says, will Forge platform will go get this information from your service and bring it in. It's, here's the Forge service. Here's the APIs to push data and get data from Forge. And then there has to be something in the middle between whatever your other system is. And your system has to support getting that information in and out of it through an API. Yes.

AUDIENCE: You talked earlier about data maps, say, for example, data required by Forge that [INAUDIBLE] services that isn't available from the ERP system [INAUDIBLE]. What about the other way around, where you need to represent data within one of the Forge services that isn't necessarily considered upfront, but is there a provision for user-definable data? [INAUDIBLE]
KEITH WHITE: Right, OK, right now, I don't believe there is. Saikat may know another bit of information. But just think of this. This is a cloud platform that everyone in the world is going to be accessing and using. So tweaks to that platform--

AUDIENCE: Yeah.

KEITH WHITE: Yeah, yeah.

AUDIENCE: --rocky road.

KEITH WHITE: Yeah. It's a good question. And I'm sure that if you've got specifics, the Forge team would like to hear them so that they can maybe have a dialogue. But it's real touchy ground, you know because, yeah.

AUDIENCE: [INAUDIBLE]

KEITH WHITE: Yes.

AUDIENCE: Did you discuss the pricing [INAUDIBLE]?  

KEITH WHITE: I did not. I did make a reference to the fact that there is a pricing link.

AUDIENCE: [INAUDIBLE]

SAIKAT BHATTACHARYA: I think, again, I would reach out to Stephen Preston because they are responsible for the team who's setting up the pricing. We don't know, because we work with enterprise customers. We have a different way of looking at it.

KEITH WHITE: Any other questions?

AUDIENCE: Here.

KEITH WHITE: Oh, sorry.

AUDIENCE: [INAUDIBLE] issues, getting imported exported issues from [INAUDIBLE].

SAIKAT BHATTACHARYA: Right, so, Field, right now, is not sitting on top of Forge yet.

KEITH WHITE: You have to use specific Forge API.
SAIKAT: So next Forge will be the underlying platform, which is Docs, in fact, on top of it's Field

BHATTACHARYA: [INAUDIBLE]. I know Docs inherently is looking at issues API. There are some classes. So that should be natively supported by Docs and not Field.

KEITH WHITE: But as far as current Field issues, you use the Field API.

AUDIENCE: Yeah. So is that going to go away? Or will it eventually all go to Forge?

SAIKAT: So it will get Forge. We don't know if the API endpoints will completely change. But there might be some tweaks that we might have to do.

AUDIENCE: So for the forseeable future, we're bound. Can't even use the the BIM 360 Field API.

KEITH WHITE: The conversations I've had with the product team has been about that. What about contractors? What about folks using this stuff now? The existing Field service is not going away any time soon because those are multi-year long projects.

SAIKAT: What they'll do is your existing projects will probably have a significant overlap. You will

BHATTACHARYA: continue using field APIs the way they stand, but at some point you can make a transition that you say, new projects will start using the new Forge.

AUDIENCE: Do we need to switch like [INAUDIBLE] we make the API calls kind of use that [INAUDIBLE].

KEITH WHITE: Yeah, none not for Field [INAUDIBLE]. Yeah right Yeah Yeah and if you build up your applications the right way you can probably just do a switch.

SAIKAT: So in a way they'll still be disconnected. So the way your work will probably continue working

BHATTACHARYA: with that integration. This will be a completely different platform.

KEITH WHITE: Yeah.

AUDIENCE: [INAUDIBLE]

SAIKAT: Available for?

BHATTACHARYA:

AUDIENCE: The Account Management.

SAIKAT: This is Account Management.

BHATTACHARYA:
AUDIENCE: In BIM 360 Docs.

SAIKAT: So yes, this is the overarching umbrella for every BIM 360 service, including Docs.

BHATTACHARYA:

AUDIENCE: But Autodesk accounts, user accounts.

SAIKAT: Oh, user accounts. Yes, so user accounts are, so the landscape there is really changing. The user accounts will be tied into BIM 360 HQ. I can give you an update on where it is, but I just don't know where we stand at this point.

AUDIENCE: If you go to manage.autodesk.com, you mean--

SAIKAT: Yeah, exactly, that's what I'm saying. So there is something that they are actively working on tying things together. I just don't know where it stands because my data is about two months old. Yeah.

AUDIENCE: We do a lot of that

SAIKAT: Right. So I know Active Directory integration with BIM 360 HQ member directory is already in place.

AUDIENCE: [INAUDIBLE]

SAIKAT: Yeah, yeah. So--

BHATTACHARYA: --approach to services--

SAIKAT: Services that you take up. So if you share me your email address, just if you have a visiting card or something. If you write that question on the back, I'll make sure and follow up.

AUDIENCE: OK, great.

It was very good, thank you.

SAIKAT: Thank you.

BHATTACHARYA:

AUDIENCE: From Forge client token, the authentication token, that has global permissions to all the Forge
within your account, you don't have to be a member of every project?

SAIKAT So--

BHATTACHARYA:

AUDIENCE: It's an account ID [INAUDIBLE].

SAIKAT Right, so that has account ID. That has global permissions. But from HQ you would have to

BHATTACHARYA: say, yes, I really acknowledge this app, which is the part that Keith showed. So then you have access, but it's global.

AUDIENCE: But there's no permissions within that. You say this application has certain restricted--

SAIKAT So that's within the scope. When I'm making the API calls in the application, I can say what

BHATTACHARYA: kind of a scope do I really limit. Read, write, and there are a few more. Yeah, you can take a look, sure. Thank you.

AUDIENCE: You guys were today, you were focusing on BIM 360 and Forge API. But is there an API which we can use to access A 360.

And then transfer it right over without downloading and re-uploading?

SAIKAT Downloading from where to re-uploading?

BHATTACHARYA:

AUDIENCE: --360 to BIM 360.

SAIKAT You're talking with the model?

BHATTACHARYA:

AUDIENCE: Yeah, the BIM model.

SAIKAT So right now, things are changing. The goal is that we have the underlying, they call it the Data

BHATTACHARYA: Management Layer. On top of it you have A360 and BIM 360, which is Docs. So this is going through tweaks. Again, if you can reach out to Stephen Preston at Autodesk, stephen.preston@autodesk.com. He can tell you where we stand. But I know the intent is to have the unified platform so that you have A 360, BIM 360, and Docs. And on top of Docs you have Field and Glue and--
AUDIENCE: Makes sense. Because that's kind of what we're trying to do. Like, would it be beneficial for us to access the model from A360, so again we don't have to download and upload to BIM.

SAIKAT BHATTACHARYA: You shouldn't have to do that. And maybe there is already something that's baked in right now. But yeah, he'll give you the update. Stephen--

AUDIENCE: How do you say that name?

SAIKAT BHATTACHARYA: No, I don't know him. [INAUDIBLE].

AUDIENCE: OK. Because that's a director for BIM 360.

SAIKAT BHATTACHARYA: OK, he might be a good person to talk. OK.

AUDIENCE: Awesome, awesome.

SAIKAT BHATTACHARYA: Thank you. Hey Dan.

AUDIENCE: How are you?

SAIKAT BHATTACHARYA: Good.

AUDIENCE: I wanted you to meet Clayton.

SAIKAT BHATTACHARYA: Hi Clayton.

AUDIENCE: Nice to meet you.

SAIKAT BHATTACHARYA: Nice to meet you too.

AUDIENCE: He's with Austin Commercial.

SAIKAT BHATTACHARYA: OK.
AUDIENCE: He's part-- are you part of their IT group or BIM management?

BIM manager.

So he's really interested in Forge and doing some work in some part with that.

SAIKAT BHATTACHARYA: OK.

AUDIENCE: And I think [INAUDIBLE] is gonna try to get you guys.

SAIKAT BHATTACHARYA: Yes, I think she said three to four.

AUDIENCE: Yeah. I just wanted to get you to meet him

SAIKAT BHATTACHARYA: I look forward to the conversation.

AUDIENCE: Same as here.

SAIKAT BHATTACHARYA: Thank you. Thanks.

AUDIENCE: Here we go.

SAIKAT BHATTACHARYA: Sure, thank you.

AUDIENCE: She left. She escaped on me.

SAIKAT BHATTACHARYA: [NON-ENGLISH SPEECH]

AUDIENCE: [NON-ENGLISH SPEECH]

SAIKAT BHATTACHARYA: [NON-ENGLISH SPEECH]

AUDIENCE: [NON-ENGLISH SPEECH]
SAIKAT: [NON-ENGLISH SPEECH] at own risk type things. But-- this isn't.

BHATTACHARYA:

AUDIENCE: [NON-ENGLISH SPEECH]

SAIKAT: [NON-ENGLISH SPEECH]

BHATTACHARYA: