ALLAN O'LEARY: All right. It's about time to start locking the doors. For anyone who is unaware, this is a internet of things class. If you're in the wrong place, this is your last chance.

How many people here have been to some of the other classes on the IoT track? A few people. Good, good. So--

PRESENTER: That guy in the back, I don't know your name. Thank you. I think you said now you've been to all five. Is that right? Yeah, that's our winner right there.

ALLAN O'LEARY: Do we have a--

PRESENTER: Thank you.

ALLAN O'LEARY: What a groundswell. Surely we can give him a prize of something.

PRESENTER: We've got to find something.

ALLAN O'LEARY: Yes. But up till this point, we've had a look at how to justify a business case, how to look at new revenue streams with connected products, how to lower the cost to manufacture. We've also had a look at the hands-on lab, how we actually connect to a wood chipper. We've looked at some of the more high level use cases, again, for business justification around [? spare ?] parts management and predictive maintenance.

And we've also had a few, I would say, support sessions around use cases, how you guys can get to market with your IoT solution and what benefits you can hope to get out of it, navigating some of that-- some of the implementation and integration. So today, we are going to take a step back. And we are going to have a look at the platform itself.

We'll dig in, look at the object model. Hopefully, we're not going to be spending too much time on the deck. We want to get as much hands-on as we can. And I have my beautiful assistant here with me today fresh from the nightclub, Brian Sherman.

So digging in, I am a product manager. This was actually an AU photo, and not a recent one. I've aged terribly since then.

But I've been with Autodesk since 2008. I have now been in the States for five years. So obviously, you can't tell. But I'm not American. I've lost a lot of that accent.

But I enjoy running. And I now am fully committed to fantasy football. So I'm waiting to see some injuries. I might have to stop some halfway through the class and shuffle my order.

Likewise, Brian Sherman is a fantasy football fan. He's much better at it than me. But he's a senior software engineer. And he's been on the Fusion Connect team for around two years, a Computer Science degree at Berkeley and Political Science as well. Pretty fancy.

So that's who you're going to be dealing with today. In terms of the class summary, basically, introduction to the development platform. We're going to learn how to make a digital twin. We're going to learn how to connect it to a device emulator so we can start feeding information into our application platform. And we're going to look at how you actually build an application for your customers, for your dealers and how they can visualize the data that's coming in, process it, and take actions like notifications.

So we're going to be starting from scratch and show you just how easy it is to get up and running with an IoT application inside of this environment. So in that vein, there's four outliers here that we're trying to achieve. So we want to get you guys to understand the Fusion Connect object model. How will the parts in the background link together?

We want to talk about how to build a basic application based on a real world product. Does anyone here working in iRobot or have any affiliation with floor cleaning equipment? Perfect. Learn how to create a rich application reporting interface.

So how are we going to present that data visually that's coming in from our real world product? And learn how to emulate that to actually give our application a test run. See how it behaves.

And that's really a big part of our value proposition here with this product is it gets you up and running very quickly without all the nasty connecting to a physical device in the field, troubleshooting the connections before we can actually get in and build an application. So we'll show a nice shortcut to getting in and test and build an application before you finish your implementation of hardware. So let's jump in and have a real quick overview.

There's a couple of areas that we're going to be focusing on primarily throughout this presentation, which is number one, the Dashboard. This is the landing page for your application. It is probably the primary source of information inside the application. We're going to put some widgets on here, which are going to give us information about the current status of our machines. We're going to see some historical reporting data up there perhaps as well as a map with some locations.

But this does not form the complete application. You'll see across the top as we go through. We're going to be building out some custom menus. We're going to have some forms for inputting data and some other management capabilities.

We're going to talk a little bit about white labeling. So how do we put our name at the top of this application? How do we integrate our coloring schemes? And how do we integrate our logos, for example, inside of the application so it looks like it's something that we've built inside our own company? Important thing when we were looking at how to build a business on this platform.

The second part of this, we're going to be spending a bit of time in. And this will be the exciting part. We're going, and we're going to be filling out forms.

We're going to be specifying units. And there's going to be lots of OK buttons and Update buttons. So hold on for that.

But this gives us an overview of what's going on inside of our application. How much data are we processing? How many objects do we have in the system? And we can come in here and edit and create new object model and reports.

So with that said, I think we're going to flick across now to Brian. And he's going to take us through the prebaked example here. So we've got a very, very simple application that we've already built. And this is going to form the basis of what we're going to show you today.

So this is the IoT bot. It is an industrial grade automated floor sweeping system. So we are interested in a bunch of information about this. I want to know where they are, maybe not to the millimeter but at least where I've deployed them. Which customer owns them?

I want to get some unit details. So I want to see what currently is happening with that IoT bot. Is it actually charging? Is it running around the office?

Is the battery running low on this thing? What's happening with the filter hopper? Is it filling up?

So we can start to gather a bunch of live data here and present it. And we can do some fancy things. We can turn it red when the battery level is getting really low, perhaps orange when we start to see the hopper fill up. And we can also put visual warnings there about how many cycles we've had on the filter, for example.

Underneath that, we've got a report there which gives us some historical view of the readings. So you can see my battery's been all over the place. Motor temperature had a couple of bad days.

But we're starting to now get a visual overview of what's been going on with my units. And this is very interesting when we get this into your environment and you start looking at this data. A lot of people have never seen some of their machine detail actually sitting side by side adjacent. And they can start to see correlations all of a sudden.

When the motor temperature goes up, I tend to start to see some issues with the battery level reporting. Or my filter-- when the hopper fills up, I start to see the motor temperature rise because it's not filtering the air through correctly. So just doing even some simple reporting like this, which we can scale and move around, soon in, we can start to see some correlations with how our product behaves in the real world.

Now if we click through on one of these reports, you'll see there on one of these widgets it'll take us to the full report. And when can start to now dive a little bit deeper. And you'll see up at the top there we've got unit status.

So we can have a little bit of telemetry here feeding in live data from our iRobot, IoT bot running around the field. So again, we have a bunch of custom gauges and different scales that we can put up there so you can see what's going on in each of those units. Obviously, we have the unit location. And switching over to that map view, I'm able to click on the left-hand side and have the unit highlight, show me where it lives. We're focusing primarily on San Francisco area, Bay Area for this particular product.

Along the top, you'll see that we've also got a trend view. We'll explain a little bit about how the trend view works. But this, again, allows us to overlay data readings information coming from the field on a common scale.

So it's a time scale there, and I can zoom in and out, filter on specific time periods. And again, we start to look for some correlations with that data overlaid there. One of the things that-- we're just going to show here. So yeah, there's some longer term data that we've been receiving over time.

Also at the top, we have some asset management CRM type forms. So I can come in here, add a customer to my system, which will automatically generate a sub account. And I can start associating customers and assets with it.

But if you guys already have a CRM system, some other application to manage your customers, we can actually use this form and via the API start to transact that information directly into the system. So we don't really need to be coming here and entering information. We're cloud-based. We've got all sorts of adapters that will plug into your other systems. So we could automate this process.

But here, a nice little form to build customers. I can also add a unit and a device. So when I commission my new iRobot, when I sell it, I'm going to associate it with a customer and start collecting data on it. And likewise, I have some device provisioning there.

So these custom forms, which you guys can build out, a great way to very quickly and easily get information into the system and build your model. Again, we have an Import here, another automated way to get data into the system, manually in this case. And we have a Messages view in this particular application.

The Messages view is showing me what data is currently coming in from the field. I can come in here and view the raw data, which is being processed and which is ultimately filling out those reports on the front page. And we're going to have a look at how that works and how we set that up.

Now if we switch across to our Design mode here, this is the backend of the application. And here is all the objects that we're about to go in build in our brand new instance in a moment. So in here, I've got a custom group set up. I've got several types of resources.

I've got some activities, device profiles, routines, and reports. So when you are the super manager of your application-- and we're really focusing here, as you'll see when we go through the demonstration, someone with product knowledge, a product manager, an engineer, site [? manager, ?] someone who understands the business and understands the product but not necessarily someone who knows how to code an application. So we're removing the ability-- the requirement to have a software programmer to come and build your IoT application.

You'll see with some simple routines, if you've got a skill set, if you've ever used Visio, you know how to use a macro in an Excel spreadsheet, then you are going to be an IoT programmer by the end of this class. Well, maybe two classes. But this is a quick look behind the covers on our existing application.

So I'm going to get Brian to switch back. And we'll have a talk about the object model. And then we're going to go back and start our new application.

So there's three, I would say, critical components that form the backbone of our application. In the middle is the resource itself. So this is the thing that we actually want to record information on and understand. Underneath that, we have a physical device, which is installed on that thing. And it's going to send us the information that we need to know about it.

We're going to have a look in a minute and a bit of a discussion about why these are two different objects. You may think that typically the device could be considered the resource itself. But we split this up for a couple of good reasons.

Above that, we also have our groups. So this is a container for our resource. Who owns it? Where it is in the world, that type of information.

So this is a very simple model that we're going to build. Now as you get a little bit more advanced, we can start to add some more additional object types and build more complex model here. In this example, I've got a customer group, which has site locations underneath it. And then I've got the resource attached to that.

I've got a couple of devices reporting from my thing information back to the system. And I can even add a resource underneath my IoT bot here, something like an alarm, which is going to allow me to record information about, I got a warning message from the IoT bot. How long was it until someone switched off the alarm and reset the machine?

We can also attach devices to groups. So we can be recording the ambient conditions, for example, where our machines are operating as well as the machine itself. Over on the right-hand side, we've also got a couple of other object types here, which is an activity and a process. And we'll get into those in a little while.

But I want to break down and just iterate where we are with what these three main object types are that we're dealing with. So as I said, first one here, group. This could be a dealer.

So it's a sub account with white labeling enabled. So you can see we've got Goodtime

Distribution, which actually sells its product along with access to the application to Cleantime Systems and ABC Systems. So a group could be a dealer, someone that sits underneath the manufacturer and distributes.

Underneath that, we of course have the customer. This is a pretty common one. If you're selling your device to more than one person, we want to be able to know who owns the IoT bot. And we want to be able to restrict access to only members of that customer to view the data. So this is a multitenanted application platform.

You as, I suppose, the super administrator will see the data for all machines in all places in the world. When we start to build customer groups, we're going to have users that can only see information underneath that group, that account. So they can see for themselves how the machine is behaving. But they cannot see anyone else's. So secure access to the assets-- and again, we're building some sub accounts based on this.

The third example I'm going to use here-- and there's probably many others that you can think of. We'll have a competition at the end and see who can come up with the most imaginative group. But we've got location.

So we can start now to filter assets by site or region. And again, even within a company, the service manager for the US probably doesn't need to know information about the machines that are running in Australia. So we can break these groups down into regions as well and isolate access within a company.

Now the resource [INAUDIBLE] is the most exciting thing, as we've discussed. So you might hear of this being called the digital twin. It's essentially going to be the object with the information that we want to know about it modeled up in the system here.

As we've sort of said, a quick example of another resource is an alert or an alarm. So this is a little record which is created when something goes wrong. And we send a message to someone.

They come in, change status, put some notes in there, maybe fill out a form to reset your IoT bot when it gets stuck in a corner. And we can come in here and do some reporting on that resource. So although we're going to primarily focus on the resource being the thing, it can be some more abstract ideas like an alert.

And finally then, we've got the device. So we have a cloud-based adaptor layer, which has a number of components in it, which we can interchange to speak different protocols and with different specific devices. So we are essentially device agnostic.

We can set up or configure a new adapter to read in whatever information you are sending us in the correct format. And that's going to depend on the device you're selecting, how much power it has, how long it's lifecycle is. There's a number of factors there that will decide how we receive the data.

But we're going to model that device up. We're going to give the device a unique ID. And when you switch on your machine, it's going to start sending messages to us. And we're going to ingest it and associate it with your resource and start to fill out your reports. So this is the virtual-- we're going to build a virtual device, which is going to connect your physical hardware and send us information.

There's one more thing I want to highlight here before Brian jumps in and starts building this. We have, inside of this application, the concept of a device profile, a resource type, and a group type. What these are are the templates for our actual records, the actual things.

So we would-- when we get into our resource type that Brian's going to build in a minute, it is essentially a list of all the values that help us to identify the asset. So what's the asset's serial number? What's the build?

What sort of filter options does it have? What sort of motor type does it have? And it's going to also have placeholders for the data that's being sent to us.

So the group type at the top and the resource type and the device profile, they're essentially templates. Now you could also think of them as the column names in your Excel spreadsheet. Then once we've got a template set up, the type set up, we just start inserting the actual records themselves, so filling out the information.

What is the actual serial number of this unit? What is the actual temperature? So we're first going to go through, build the types, and then start to add in some records.

The really cool thing about this application is, this is all the heavy lifting. Once we get this object model set up, we don't have to go back to the well and redo this every time we add a new product into our system, every time we add a new customer. This is all just going to start to flow immediately once we've got the object model set up.

So we covered this stuff really quickly. Template or the record structure, what information should we have? As I said, the record is the asset identifying information and the status information that's coming from the field.

So we're going to go and put this all together. I'm going to switch back to Mr. Sherman. And you can see here we're actually starting in a blank application.

So I've got a welcome message here on the Dashboard. I've got nothing else to display. So this let's someone know that, hey, there is an application here.

We click through to the Design mode view. And you'll see all empty. So nothing built yet. We're going to start off and build ourselves a resource.

So we'll call it IoT bot. You'll see as we go through here, we're going to get a resource type code. This is going to help us to identify this particular object as we go through and do any scripting you want to do through expression language. We can reference back to this.

You'll see at the top, I also have globally unique keys. And I can have some performance options with this as well. But let's start adding fields.

As I said, this is the template. So what do we want to know about the IoT bot? We need a unique identifier. So we're going to call it serial number.

Down here we make it the resource P. We also maybe make it to the label. But we can have a more friendly name if we don't want to display serial number.

Over on the right-hand side, we've got the data type there. This is just going to be a text field. But we've got a whole host of options there.

So [? integers are ?] numbered Boolean. We can select media, which we'll do in a short while to add the logos in on a section. But we can also in here choose, I'll say, a complex and start having address fields coming in here, phone numbers, email addresses. And we can also modify a lot of this.

So we can create [? patterns, ?] capitalize it, different units for numbers, things like that. But let's just leave that text. And we'll select Add.

And then we're going to add a few more fields in here. We want to know the location. We want to have and icon to put this on the map. So we'll put those two values in there.

So I did kind of cheat. It's not quite empty. I loaded up some icons because I didn't think you'd really want to see the upload icon functionality.

In here, we're also going to add some information about the things we want to monitor. So I think for today, we'll probably want to monitor the motor temperature. And let's look at-- yep, and say underneath the modifiers here, you can see we're selected Units. We'll change that to Temperature and Celsius please.

Because I use Celsius. And I really don't understand Fahrenheit. I have no idea how hot it is. That doesn't mean anything to me.

[LAUGHTER]

So in here we're going to put in the hopper level. And this time, yeah, we're going to have a number. And we're going to have, obviously, a minimum and maximum for this, 0 to 100. Still working with the base of 10. And we'll add that in there.

Maybe we want to put in here a battery type just so we've got one option. And let's give it three options. So type one, type two, and type 3.

We've now got some very basic information. There we go. Had to redo that.

So we've decided what the label and the key is so we can identify it once it's in the system. We've got some placeholders for information that we're going to start to get from the field. And we've also got a section here where we're going to get some location information to see where this thing is moving.

So that's it. Now if we go back to our Design view, you'll see that we have one IoT bot resource now created, the type created. You can see there's none currently living inside of it. And we've just created the template.

And up on the right-hand side, you can see we don't have any records. And we're not taking up any space in the application yet. So we're going to continue that. And we're going to-- we'll put together now a group to associate this with.

And we do it in this order because when we create the group, I want to know what goes into the group. And it's going to be the resource. So you'll see container for the IoT bot. So the system knows an IoT bot belongs to a customer.

This is going to be a customer. We'll add some information here. So obviously, customer name.

We want to probably have an address for the main site if not the actual site that we're selling to. So we're going to create that as a complex. And here we're also going to go and add a contact name.

So again, super simple. You can capture as much or as little information against the customer as you please. As we've suggested, you've probably got another system which manages the bulk of the customer data.

So we can actually just bring in the little pieces of information we need to set up the object model. But if you prefer, we can store all your customer information in here as well in the group types. So we'll go back to view. And you'll see now we've got a group type set up.

The next piece of the pie-- does anyone remember what the next one is? Device. That's right. We've got to build a device.

So underneath the Connect menu, select a Device Profile. Now as we go through this, we'll give this a name. We'll call it the IoT bot device. It's a custom one.

Underneath the device type, we'll pull that down. You'll see this is the list of-- if you pull that down, Brian, this is a list of the adapters that we can use out of the box. And as I said, we can certainly add to this as well.

But we've got some custom-- some specific devices listed in there as well some more generic protocols. So we're going to use HTTP abstract. And you can see this device, we need to know what it's going to be installed on. So is it going to be installed on our IoT bot? Or are we going to install this device at the customer site? So this is going to be installed on the IoT bot.

So we'll add that in and start to add the message content. So we know there was some information that we are definitely going to get from this device. We have to know what the device is. We need an ID.

We have to know when the message was sent from this device. So we prefill this already with some message fields. Now it's up to us to add what our custom values are that we know we're getting from the machine depending on the sensors that are installed.

So we're going to go through. And we're going to get the motor temperature. Again, we can specify what sort of units we're expecting to see in here and what we expect the minimum, maximum to be.

We'll add in the hopper level. Again, it's going to be a number. And we're expecting that to be between 0 and 100.

And we'll add in the battery level. So everything that we're creating here has a direct correlation to sensor data that's going to come from the real machine in the real world. So we've got a fleet of IoT bots running around outside sending us data now as you'll soon see.

But this is the virtual device that we've-- the device profile we've created here. So if you go back to our Design view, you'll see there we've now got our devices identified.

So let's go and see how that works when we create a record. So we'll go through. Select Resource. We're going to put in-- you'll see there. Yeah. That's [? really ?] [? going to ?] [? help ?] [? them. ?]

So here's the information, the type that we identified. We said the customer has to have a name. The customer has to have an address. And we want a contact name to associate with the customer.

So when we go and create one, create these records, it's now just a matter of filling this in. And we can do this in more clever ways. We can build our own custom forms as we saw on the application that I had previously. But this is just a very manual way to start filling out the records.

So let's call it New Day. I will let you choose which address you would like it to reside at. That's a favorite of mine too.

And the contact name is also at your discretion. Perfect. I'm sure that apostrophe is going to come back to bite us. So underneath the resource type now, we'll add that in.

And you'll see, we said the IoT bot has to belong to a customer. So from our menu here now, I'm going to have a pull-down list of what customers I can associate it with. This one's going to be associated with New Day.

Again, we can build a form that does this automatically for us. But through here, well, that serial number is 1111111112. And again, we can put some controls over that. So if that number is too large or it doesn't meet a certain pattern, we can reject the numbers as well.

We can do order numbering. So we can generate a number based on the sequence starting point in increments. So the battery type on this, we have three types of batteries. This is a type two.

You'll see on that front page-- if we get back to the main, we also have here a placeholder for the motor temperature and the hopper level. Well, we're not filling that out. That's the placeholder from the information we're going to be getting directly from the machine itself.

So underneath the location, somebody has put the wrong icon on the map. That's fine. Try [INAUDIBLE] logo. You know what? Maybe I did it.

[LAUGHTER]

No, it wouldn't have been me.

BRIAN SHERMAN: We're good.

ALLAN O'LEARY: We're good?

BRIAN SHERMAN: Mm-hmm.

ALLAN O'LEARY: All right. Do we have to put the serial number in again? [? Oh my god. ?] There you are. You missed a one.

So type two. And on the location now-- here we go. That's a little bit better. Now we can see where our little dude is on a map.

So we can put an address in there. We can put in some GPS coordinates, some long lat for those of you who prefer to work in longitude and latitude. And locate that on its initial position on the map. But again, this is going to be coming from some sort of a GPS sensor sitting on the machine itself.

So here we go. We've just created our first resource in there. We've got the product. We [INAUDIBLE] on [? the ?] left-hand side. We know which customer it's associated with.

There's its unique identifier, its serial number. There's the battery and motor temperature and hopper levels. And we can click across and look at its location in here as well.

Super easy, huh? So now the last thing to do is to attach a device to this thing. So at the moment, we know what the serial number of this is. But that's not associated with a physical device sending us information.

So you'll see here, I come into the new device area. I'm going to select the protocol that we had configured previously, which HTTP abstract. The device code is the unique identifier for the device in the field.

Now this is very important. This is the number or the value that is embedded on that device itself. So if those numbers don't match up, we're just going to ignore your message. We're not going to listen to it. Sorry.

This one, we'll just call it 1012111. No, it's fine. And you'll see this is our gateway. We have the address where that's being sent to.

And we can add descriptions in here if we want. But this should do us for now. So we'll select Add. And now we've got our object model laid out.

All we have to do now is sit and wait for messages to start coming into the system. If we go back to our Accounts Center, so you can see there's our diverse profile set up. You can see we've got one customer and one IoT bot in there.

And our little dials on the left-hand side are starting to go up letting us know that there's data in the system. So that's it. We finished early. Switch back to me.

I did have some bonus material. So some of the additional objects, which we didn't cover here-- I'm not sure you're ready for this. But I wanted to go over these. And we can certainly, if we get time at the end, have a bit more of a look at-- or answer some questions about what these do.

But the first one here is the process type. It's essentially a collection of events. So if I'm driving a truck, it's interesting when the trucking ignition gets switched on. And it's interesting when the truck ignition gets switched off.

But I don't probably want to build a report about how many times the truck was turned on and how many times the truck was turned off. It's much more interesting to me to know when the truck was turned on, how far did it drive? How long did it take to finish that journey? And where was it when it was switched off?

So I can actually bring a series of events together and turn into what we might call a trip. And we can start reporting on, how often does the truck leave the building? How long does it typically drive for once it's out there? Which is much more useful information than how many times has it been turned on and off. So this is a process.

The other type we have in here is the activity. So right now, we've set up a model where we're going to be collecting live data and displaying live data to the users. As the information comes into the device, we put it into the resource. And we can see what's happening.

If we, though, want a historic-- if we want a history, a historical view, of how it's behaved in the past, what we do is we create an activity. So an activity is sort of like a read-only record that says, on this day at this time, we received a message. The motor temperature was 25.

An hour later, it was 47. An hour later, it was 39. And we can start to build some [? training ?] graphs and historical reporting.

So I don't think we'll do a demo on that one. And I want to set up an activity time yet. What I do want to do is get into some of how we're going to display this data. So as I suggested, we've got an object model laid out. But at the moment, we're not seeing anything.

We can go in and manually view what's happening on our object or our resource. But I now want to build a report. So what we do with our reports is we go and query the information that we've collected.

And we can start to present it in some meaningful way. So we can show current asset status. We can start to show historical performance. We can do some events or errors.

We can have a list of things that have gone wrong. And we can also start to record location. So in terms of what we're actually able to show on the Dashboard in our reporting areas, we've got just a simple table view. And you can see this one has some decorations.

So we've had some failures or some warning messages coming in. You can see battery level is high. So it's green. So we can just show some current status of our machines in a table view.

We can put it on a map, which we've already seen. We can start to bring in the gauges for a telemetric view of how it's behaving. We can show charts. So now we can start to break things into pie charts.

What's the breakdown of the number of alerts that we've had? Is it typically a high temperature? Is it typically a filter issue? Start to show some bar charts.

And if anyone was in Paul's class, he actually gave us some pretty good examples of how to build charts up to capture [INAUDIBLE] information. Timeline, so we can start to look at how things have changed over time with our resource. Trending information, schematics.

Schematics is kind of cool. We can put a picture of our model in there and start to overlay it with live data. So any dummy can kind of start to figure out what we're actually talking about when we're talking about motor temperature if we've got a picture right there for him. We can get it in the 3D schematics as well.

We can build composite reports and custom. So as we've already suggested, we don't have any-- there's no real requirement here for you to be a coder, developer, have any of that sort of expertise. It's all no coding. But if you do want to have a more advanced interface, some more advanced reporting tools and images, we can start to build out some custom reports as well.

Here's an example of the trend. This is a little bit different from our other reports, as I said. And we touched on this briefly at the start. So common x-axis, this time, allows users to overlay a number of measurements and compare them. We could also filter in zoom along the timeline.

So now I think we'll go and do a demonstration.

BRIAN SHERMAN: For real?

ALLAN O'LEARY: For real. And what we want to do is we want to set up some reports to start displaying information. So you'll see we're in the Queries and Reports section. We really consider this two sides of the same coin. We're going to do a query and display it in a report.

So first one, let's just do robot status, or IoT bot status, whatever you like to call it. Now we're going to add a target. So what is it that we're collecting the information on?

We can just have a list of customers here in our report. So we could select a group. But here we're going to grab the resource.

We're going to give it a code. So we'll call that IoT bot maybe. And we need to know what information we want to display on our report. So I want to know, obviously, which unit it is. So I need to know the serial number.

I want to know, I'm going to say the motor temperature, the hopper level, and the battery level just at this stage. And let's add the icon in there as well. So there is our target. This is the information that we're going to query and put into our report.

We can add criteria here. So we can start to filter on this. So if we were to bring in, say, a customer value or something like that, we could use that as a filter.

Underneath that, is the actual area we're going to create a report view. So this has defined what information is coming in, how are we going to filter it. And now we're going to decide how it's displayed. So there's our table, map, gauges, chart. This one's just going to be a simple table.

We're going to have it appear on the report page so we can view it when we go into reports. I want to also have this on the Dashboard. Now there's some more advanced options here too.

When I start building my custom forms, I can say, run a query. And give me the results of the query so I can look it up when I start filling in a form. I've also got the ability to export this, the table view. I can export it as a report. And I can do that on a regular basis, send out emails with the report attached.

I can have monthly billable hours, things like that. I don't [INAUDIBLE]. We've got a suggestion box in there. And we can actually add this to the API as well. So when we add a report to the API, that means we can actually ask the system to return us the results of the value and do something with it in another application.

But we're just going to do a report page in Dashboard widget. Slick Add. And now show them how easy it is to finish this table.

You're doing it the slow way. Gosh. This is terrible. There. So basically, I've grabbed my object, dragged it down to the section here.

And you can see it automatically fills out my table view for me. So I can go in and start to change the names of these headers, reorder that. But for now, that should do us in terms of our report.

So we'll add that. And if we click on the View Report, that will take us straight there so we can check what this thing looks like. There's our serial number.

There's our motor temperature, the hopper level, and the battery level. If we go back to the Dashboard, let's go and add some more widgets. And I want to bring you on my front page this report, Robot Status. I also want to get rid of the welcome message. We don't need that anymore.

Now I can decide how many columns I want here, one, two, three, across the front. Start to work with the layout of the reports. So there we go. Now we're starting to display some information to the user when they first log in what their assets are doing.

Let's go and create another report. This is everyone's favorite. We're going to do a map.

BRIAN SHERMAN: [? Do you want me to send a ?] [? report on ?] [? this thing? ?] [INAUDIBLE]

ALLAN O'LEARY: What would you prefer?

BRIAN SHERMAN: [? I say do that ?] as a new view to the same report.

ALLAN O'LEARY: We're going to modify our existing report. And we're going to add another view. So we've still got all that same source information there.

We're going to add in another target. And now when we come down to the view level, we're just going to simply add another view. So that same report, it's going to come up as a second tab on that report.

Here we go. So a little bit more of a change here. We need to know the information about what icon is going to appear there, what we're going to call it. So we can start to bring in variables like the serial number so we can identify what's going on with this thing.

And in here we're going to start to put in the very specific information we need to locate it on a map. So obviously, location, what the icon's going to look like, and the label is the serial number. So update that. We should be able to view our report again.

And you can see now second tab on that report, I can go and see where my IoT bot is in the world. I don't think that's quite 1 Market Street, though. That's down here. Right? All right.

So moving back to the Dashboard. [? There we go. ?] We're going to add another widget here. And we'll do robots on the map. There we go. So starting to gradually build this out.

One more? What do you think?

BRIAN SHERMAN: Customers?

ALLAN O'LEARY: Yeah, let's do a customer report. So similar to what we did last time, we've selected our source this time as a group. We've selected what information we want to display about the customer, dragged it into the table. And we've now got a list of all our customers that we can start to move through.

So here's the report view, New Day. There's their address and their contact there, Super Guy, Allan O'Leary. Back to the Dashboard, more widgets. And we'll add in our custom report and start to move that around.

[? Here we ?] [? go. ?] So that's again, a very basic set of information. But we're very quickly able to put that together and display it in an easy to consume way. Flip back to me, and we'll move on to our next [INAUDIBLE].

So the next thing we want to do is to actually start to emulate some of the data on the field. So as you saw with our initial values there, there's no information in the system. Everything is set to zero or a number that we've manually put in.

Now as I said at the starting point there, we can now sort of start to sit and wait for the hardware guys to finish wiring everything out to get the prototypes finished, get it all plugged in and start sending us information. But obviously, that's a lot of downtime that's wasted. So we are going to now go and have a look at how to send some information as if we were really connected to a machine. You might have also picked up the sarcasm with the IoT bots running around outside. There are none. It does not exist.

But we'll see today using the emulator we can start to fill out that data as if there were. We've got one more thing though we've got to do before we can actually send information into the system. We need to tell our application how to route that information.

When you get a message, what do we need to do? And usually, it's going to be updating a resource. We're going to want to send the information from the message and display it in our reports against that resource.

But there's some other things we probably want to do here as well. We might want to add an activity so we can record, at that point in time, historical value, which is going to be used in a report later. We might also have some mission critical information that we need to know about.

If the temperature recorded from the device comes in at over 50 degrees Celsius-- I'm not sure what that is in Fahrenheit-- then we want to set an alert to someone. Someone gets an email that says, your IoT bot's running hot. It might be more critical than that. And we might say, if we receive this reading at this particular temperature, then I want you to go and actually send a directive to the IoT bot to switch it off. Stop it from running so we don't do any damage.

So this is where this logic comes into play. We build a routine that looks at the information in the message, does some real-time analytics, makes a decision based on the data that's coming in what it's going to do next. Is it updating a resource? Is it sending us an email?

Or is it just going to throw this message away because it was nothing-- no useful information in it, for example. We can also through here start to call out some other services. We can perhaps start to send some of this information into a machine learning application.

We can start to look up real-time weather conditions from a weather service. We can start to create tickets in a service system or in a CRM system. So as we're connected-- you know, fully cloud native with a really strong API and adapter layer, we can do a lot of really cool things with other web-based services in here.

So as I said, we can add our objects, process messages, add logic operations, set alarms and notifications, update the resource status. And we can do some very specific things that relate to the IT world. So we can now start to add a device automatically. We can connect with other web services and send a directive back to a machine telling it what to do.

So this is all done through the routine engine. Yes, thank you. Oh, and I really should point out there in that routines area, the example that we're looking at and what's certainly an example you're going to have is just processing message information. But we can also use that same routine to build the logic out for a form. So when I submit a form, what do I want to do with the information in the form?

And we can go in and fill out-- automatically create our customers, automatically connect it to specific machines, provision numbers, and devices, all that sort of stuff. So it's not simply messages. But we're going to be focusing on the message routines today.

Now once I have a routine set up, it tells me how that information is going to be routed through the object model, where it's going to land. The next thing to do is go into our device simulator and start sending some made up information. Now this is going to allow us to test that out routine actually works, that our reports are useful and really validate all the work we've put in, which is, what, 45 minutes? Yeah, lot of work.

So we're going to-- you'll see here on the right. We've got a little view of the interface here. And we'll show this live. But we're going to expect to see our fields in there. We're going to put in a mean value and a deviation on that. And it's going to start sending varied messages into the system to appear on the reports.

Sorry. So we'll jump in, and we'll do a demo. So underneath the-- no, we're going to need to configure our routine. So underneath the Configure, we'll select Routine, at the top we're going to give this routine a name. We'll call it Incoming Message.

Again, we could do more descriptive information there if you want. Now the origin of this routine is a device. If I had a form built, the origin could be submitting a form.

I also have a timer option in there, which means that maybe once a month, once a week, I can run a routine that will figure out, statistically, what are the numbers for the month? How many new customers have been added?

Or it can automatically send a report out to someone based on that timer. So a few options to start our routine. But we're going to go with the origin being the device. And the event is that we've received a message.

Once we've selected that, it loads a bunch of variables into context for us. So we can start to directly call out variables and start to build expressions with those variables [? there. ?] So super easy. Again, this comes back to the no coding.

So we've got an empty sheet on the right-hand side here. And we're going to add an action. So the actions are many and varied. And a really important one at the top is the if-then statement. Obviously, this is going to start to do some logic and rerouting of our message depending on some of the input values.

So this is how we would set an alarm level. If it's above 50, we go to option B. The next one down is to set a variable. So this allows us to, just in the context of this particular routine, to set up a variable, which we're going to call later on.

So maybe you want to define pi. Or you want to have some other constant, which you're calling through your routine. Underneath there, we have Find. So this allows us to go and select an object in our ecosystem and bring it into the routine.

So we might want to go and find a specific resource based on the battery type and take some different action on that resource that we found. And you'll see there, we can find the first element. Or we can do an interative find and basically loop our routine over every unit that we find.

Select, similarly, it's going to go and allow us to select an object to carry out some action on it. I could do some calculations in here. You'll see here, we've got some what we perceive to be some obvious summaries that we might want to withdraw here. Let's do a count on what actions, how many things we've found.

Let's find out what the min/max temperatures are and display that in a different way, for example. We've got Add here, which is typically what we're going to do when we're using the form method. When we submit a form, we're going to go and create a customer. Or we're going to go and create a group, something like that.

Next one down is Update. So this one is the one we're going to be using in this situation. So this is just going to take the information and plug it into our actual resource. So if we look at the fields here, obviously, I'm not going to change the serial number.

Someone's already put in that asset information. I'm not expecting to get a new serial number for my device. Likewise, the battery type, it probably could tell me what battery is in the machine. But when I ship that out, I already specified what the battery type was. So we're not changing that.

However, there are a few things, which you're changing. Motor temperature is changing on the way in. So we're going to-- and you'll see, all of those fields that we defined are available in here.

So we can plug that to the correct variable coming in from the device. We can see location changes. Start to update that.

And interesting here too, just if we go into one of those and set that to an expression, you'll see this gives us a little bit of the expression language that we're able to work with. So here's our variable list on the front page. If we move across to the Functions, you'll see here's a bunch of mathematical functions and expression language operations that we can carry out on our data.

So we can do calculations on time period. We can look up weather data. We can do geocoding. So we can actually put in lat longs and go and look up what the address is, where it is in the world. A whole bunch of functions available for you in there.

So without knowing how to code, we can still do some really powerful real-time analytics on the data as it comes in using this. And we've also got our basic operations, times by, divided by, equal to, not equal to, those types. So that should do us for now.

We'll put in location. We're not going to change the icon. Our device doesn't know anything about that. So there you go. There's our first basic routine.

We're going to get a message from the device. And we're going to fill out the current conditions on the resource. Start to display it in some reports.

So we'll add that. And it's now time to look at actually sending some information. So if we go to the Monitor tab there, let's go and have a look at our Message area.

So nothing up our sleeve. No messages in there. All empty. We haven't received anything in the last 24 hours.

We don't have a total message count. We don't know how often with the frequency of the messages we're getting. It's all empty.

Now if you go to the in-bound area, I can see the content of what's coming in in those messages. If I go to the Outbound tab, we can look at any information in messages we've been sending to the machine itself. But as we say, it's fairly dull right now. There's not a lot in there.

So let's go back to Connect. And you'll see down there we have the device simulator. So we can select a device here, which is already set up in our system. So we've selected 11111111112.

You'll see it gets a message flow name associated with it. This is the only one in our system, so it's fairly easy. And now we can add a message underneath that.

So we select the message type there. And it knows what information the device is expecting to receive. It knows we should be getting battery level, hopper level, motor temperature. And we should actually be getting a location as well.

So what we're going to do is we're going to start to fill this thing with what values we expect to see. Let's move it somewhere else so we can-- let's take a holiday from San Francisco. Go to Sacramento.

BRIAN SHERMAN: [? You got a holiday. ?]

ALLAN O'LEARY: Do you know that Sacramento is the capital of California?

[LAUGHTER]

I was amazed. In terms of the battery level, we know that this has to be a value between 0 and 100. So we're going to make the mean 50. And we're going to have a range of, say, 50 on either side.

Same dealio for hopper level. Well do 50, 50. And motor temperature-- and now remember, it's Celsius. So we're going to go 30 with a variation of 17.

So what's going to happen now is as we send off that emulator, it's going to alternate those values and start pumping in and seeing live data. So we might add a delay to this thing. Please don't do this at home.

But we're going to set up our delay really quickly so we get a lot of messages start to come in. So let's make this one second. Should we make it longer? No. Five? You want-- OK, we'll make it 5 seconds.

We can also, of course, set this to minutes, whatever you want to to replicate your real-world application. So we've defined our message. We've defined the variables in there. We've got a delay.

The next thing to do is to start the emulator. So this will allow us to pick a starting time, how many times we want to loop the emulator. We'll just do it 50 times. [? Something like that. ?]

We can also set the start at time and decide what to do with the messages that are coming in there. So we've got a little green dot here, which lets me know that my emulator has started. So somewhere in the world an IoT bot has fired up and started sending me information.

If we go to the Monitor tab-- here we are-- I've now got four messages already coming in. I'm starting to see inside of the message. If you go back to the table because I can't read it.

When it was received, when it was originated, what device it came from. And you can see there I've got the message type and when it was finished processing. So the processing time is super quick here because we've got one routine that does one thing. It just updates the IoT bot.

Obviously, as you build out more complicated routines that start sending you out emails and making analytical decisions, that processing time increases. But that's it. We're starting to see some messages coming in.

So if I click on the magnifying glass-- thank you-- we can start to see what information we're getting. So this one came in with our battery level at 57.8, 53.09 for the hopper level, 35.02 for the motor temp. And it's in Sacramento.

So if we go and have a look at our reports, see, we should be seeing now updates, the new location. Our customers haven't changed. But down here, we're going to start seeing our temperatures, battery level, and hopper level alternate. Now we've got 71.7, 31.18 degrees.

[INAUDIBLE] There we go. [? Flicking ?] across again.

AUDIENCE: Can you emulate other record types?

ALLAN O'LEARY: Can I emulate other record types?

AUDIENCE: [INAUDIBLE]

ALLAN O'LEARY: This is just device message information coming in. So this is a real device being emulated-- a real device sending us information to fill out our reports and make sure our routines work. So it works. Any one got any questions at this stage?

So I've been talking a lot. I haven't given you guys a chance. Got any questions so far about what we're seeing?

AUDIENCE: Yeah, the XML button [INAUDIBLE] emulator. Does that [? give you a ?] [INAUDIBLE]?

ALLAN O'LEARY: So everything that we have done here is being written to an XML. So if you prefer to do away with our user interface and just type everything into a XML format, you can do that across the whole application. We can actually take that XML file and go and build a new application using that XML inside of our platform.

So as we go building it, it's just creating an XML in the background there that can be edited and accessed.

AUDIENCE: So that is not the emulated data [INAUDIBLE].

ALLAN O'LEARY: No, that's not the emulated data [INAUDIBLE].

AUDIENCE: So how do you [INAUDIBLE] your interface, your different message types that [INAUDIBLE] or [INAUDIBLE] HTTP [? whatever it ?] was. [? Where does that ?] become [INAUDIBLE]?

ALLAN O'LEARY: OK. So your question was where do we see the content? If I understand correctly, where do we see the content of the messages? So that actually happens before it gets to us.

So when you're actually setting up your device itself, you'll be deciding on the protocol and deciding all that. And you will actually arrive at a format that that message is going to be transmitted in. So at our adapter level, we come in.

And we build-- we would have to use one of the standard ones that we've already got in there. Or if you have some custom hardware that we need to modify, we would build that adapter with the format that you know you're sending. So when we get it, we know what to do with it.

AUDIENCE: [INAUDIBLE] You would [? at ?] [? the service ?] [? build ?] [? that? ?]

ALLAN O'LEARY: Yes. And that is typically not a long-- that is a very short engagement.

BRIAN SHERMAN: We do have documentation on [? all different ?] standard [INAUDIBLE] adapters [? that ?] [INAUDIBLE] exactly how [INAUDIBLE]. You're going to use [INAUDIBLE] transfer that over to [? HTTP. ?] There's a document that says here's how to format your [? adjacent ?] [INAUDIBLE]. [? And you'd ?] use your [INAUDIBLE] [? to do this. ?] [INAUDIBLE] So they're using standard ones now. And they can be customized as well. But [? he'll keep talking ?] [INAUDIBLE] format the [? information. ?]

ALLAN O'LEARY: So yep, thank you for your question. We can certainly catch up afterwards and talk more about what happens before we get to this point with the application. We're more than happy to talk about that with your offline too. Anyone else at the moment? Oh, [? we'll ?] [? go with ?] you.

AUDIENCE: What's the limitation of the devices on the platform?

ALLAN O'LEARY: The question is, what is the limitation of the devices on the platform? Are you talking in terms of numbers?

AUDIENCE: Yeah.

ALLAN O'LEARY: Oh, this will scale to more than you need.

AUDIENCE: [? About how ?] many?

ALLAN O'LEARY: We would start to have a discussion with you at a certain point with the number of devices. I guess the important thing for us is the size of the messages, how often they're being sent. So certainly the number of devices plays a role in how your application is going to behave. But quite as critical is yeah, what is the amount of message data, how often it's being sent, and what you're doing with it.

Are we creating 50 activities every time we get a message coming into the system? And we're going start to see that dial about the information that's in your application, and the number of records start to fill up. Now once we start to get into those performance thresholds, we do have some other options as well though.

Instead of storing the data live, storing the data in our system at rest, we could jettison the information off to another system and hold the records there for historical purposes. We can also do real-time analysis on the message data that's coming in in memory and just basically take a summary or a daily average, something like that, if that's applicable and, again, get rid of the raw message data. So there are-- we can scale up our ports that are listening so we can actually process the number of devices coming in. And we can deal with the size.

And then at the backend, we can also reduce what's being stored in here and the side of the application. If you ask me a question, it will take me five minutes to answer it. So just be wary.

Leave some time.

All right. So next, I want to get into a little bit of the nice stuff and have a look at now we've got our basic object model in place. Now that we are able to display some reports, I want to do a little bit of customization and make this look like my own application. So if we go back underneath our Sub Accounts Center, we might start there.

So we're going to create a new sub account here. At the moment, we didn't automate the groups to create sub accounts. So we're just going to do this manually for the demonstration. So we'll call it New Day.

We're going to associate this with an account logo. And we should have a New Day account logo in there. Doesn't look like it.

So the customer that we're going to give access to here is the New Day data. So you can see now, we're starting to divide up access to the objects which get put into the system-- we're creating in the system. So as we've seen, we don't have to go and build that resource type.

We don't have to go and build that group type for each new customer that gets added to the system. We don't have to build a new application from scratch. All that's sorted out in the background. Now it's just a matter of sectioning off the data accordingly.

So we've got our customer there, New Day, set up as a sub account. We can change the logos, and we can start to change the coloring and access. So if we go into our Settings-- there we go-- here's where you can come in and start to fiddle around with what the end users are going to see. And this is the starting point of white labeling.

So when we talk about white labeling, this can include your own customer URL. So you see at the moment, this is pointing at our fusionconnect.io. If you want to actually keep us completely hidden behind the covers, we can give you a different URL to your account.

And underneath that, we can change the login screen. So we remove Fusion Connect from the login screen. And in here, we can start to change the coloring, as we said, the logos, et cetera to make this fully customized.

So if we go in underneath the user interface, let's select our Sub Account. And you'll see now, I've got a list of all the sub accounts that are available. I'm going to select New Day. And we're going to mess around with their user interface.

So let's make the background gray, maybe. Yeah, let's make it light gray. Color scheme at the moment is Fusion Connect. Now we can start to put in our own custom colors. And there's some information here about what color maps to which elements in the screen.

But we'll change from Fusion Connect. And let's make it blue cherry. That's pretty obnoxious. Down here we've got our filter box locations. So we can start to move elements around the screen in our application.

Let's leave it on the left. I'll get confused. We can change font style. We've got gauge styles in here. So you can see, we've got orange lines. Let's do a thin red.

And down here, we can also start to mess around with the scheme. So I'll let you choose which scheme you like. If we go into User Experience now underneath that, we can start to look at the header styles. We can also do some modification to how dialogues are going to appear, how charts are going to behave.

We can implement a splash page and enter in information on that. You can also customize your welcome message. So we've got all those options there. And we've got our default values and report settings in here as well.

So this allows us either at an account, a sub account, or even down to the user level to change what each person is going to see when they log in. So the next thing we want to do is create a user so we can go and have a look at that account. So pretty cool in here once we get into our users. Let's really quickly locate the user role first. Sorry.

So we're going to go through here. Our new user will be Brian Sherman. Do you know how to spell that?

BRIAN SHERMAN: [INAUDIBLE].

ALLAN O'LEARY: Yes, sorry. We're going to have a customer in here. And you'll see now, we get a really refined grid of roles. So we can select whether you're able to view or modify.

We can start to break up different areas. We don't want this user to perhaps be monitoring transactions. We don't want them to be able to add customers into the system. But we probably do want them to be able to add specific resources through provisioning.

There might be some reports there that they need access to. But you'll see at the moment when we-- next to reports, it has [? all ?] [? in ?] brackets afterwards. So if we click on that, I don't want this person to see the Customer Report. That's none of their business who the customers are.

So we're removing that. We can get right down to the sublevel in each of these roles. So typically, yeah, you're going to have a few profiles set up here, service manager, site manager, accounting. And they're all going to get different levels of access to the objects that we've built up.

And you'll see, this also reflects in our menus and areas like that. If you don't have access to a report, you don't get a button to go and click on the report, fairly simple and straightforward. So let's add this role. And now, Mr. Sherman, we'll add a user.

So the dependencies here, we're going to have first name, last name, emails, phone number, if you need it, login name, and of course, a password to get into the system. In here, we've got some localization settings next. So this is our preferences for the time zone that we're working in, the system of measurement, metric obviously.

Do we have an Australian language option.

BRIAN SHERMAN: [? Isn't it ?] [? the ?] [? same thing. ?]

ALLAN O'LEARY: There's no Zs. We don't use Zs. And finally, we have access. So when we get into the access page, this is which sub account is this particular user allowed to get into and what role is it that they're taking off.

So now we're sort of starting to slice and dice the security. They've only got a certain level of permission. And they can only see some records. So let's finish that.

And we now got a user that we can go and log in and see our sub account. You can see here, this is the standard Fusion Connect interface again underneath the white labeling. This can be configured with your own logos and color schemes.

We've got our IoT bot logo up in the top left-hand corner. And there is our blue cherry color scheme. So this guy comes in. No widgets laid out for him yet. But he can go in, and he can select those two reports.

He can not select the Customer Report, can't do that. But we can now see his status and where it is in Sacramento. Any questions there about that level of the white labeling and setting up our application? Yeah. Oh, you have thumbs up.

Well, with that, I think we're nearly at an end. Anything else you want to show on the application before we move on?

PRESENTER: [? No, I'm good. ?]

ALLAN O'LEARY: You think we're good? Oh, no. I want to show menus.

BRIAN SHERMAN: All right.

ALLAN O'LEARY: My apologies. I knew there was one more thing.

AUDIENCE: So when you created your login there, I kind of [? noticed there ?] [? on the screen, ?] can you use other authentication systems? Can you use single sign-on or something like that if the customer has their own?

ALLAN O'LEARY: Yes, we have options. So by default, we're storing the customer information in here. And the authentication is done through our system. But yes, there are other options to sign into the system if you have other methods.

So Autodesk ID. I think I saw LinkedIn there. And so there's a question in the back.

AUDIENCE: What about hosting, local hosting?

ALLAN O'LEARY: No. The question there was, can this be hosted locally? No. We're a full cloud native architecture. And the way this is built, it just lends itself to being pure cloud-based application.

So the last thing I wanted to do before we moved on was show you the last bit of the white labeling here, which was building out some custom menus. So as we saw, the Dashboard is really the focal point for the end users [? around ?] where they're going to go and consume information. But there's other things we want to do as well.

So you'll see there, at the moment, the only menu item is the Dashboard. But I can now come down here and start to add views. So let's put it in a Report. And let's put it in the Customers.

Yeah, so we'll just say-- we'll call it Customers. You can see, I've got a new menu item there. Add another one. And we'll call it-- let's say, View Report. Robot Status. Add.

If we've got any additional forms to be filled out, if our uses have got permissions to go and view messages, you'll also see those options come up. So we'll do maybe a Monitor Messages option. And these can be renamed to a more friendly pull-down menu.

And we can also create a pull-down in here as well. So we can start to group some of these options together and put into a pull-down menu. So real basic stuff, but no hard coding required to do it. It's very easy.

So there you go. We've put our two reports now underneath a pull-down. So Update that. And if we get out of our Design mode and look at what we're allowed to see here, here's our new menus at the top.

So Report Status. I can click through to Messages. And I can go back to the Dashboard.

Now if we want to quickly log in as user number two, we'll see that there's a couple of changes there. So you'll see now, we've only got one option on the menu for this guy, which is his Robot Status. He can't see the Customer Reports. And we're not letting him see Messages. Thanks, Brian.

[APPLAUSE]

So quick summary here. What we saw today was really just enough to get you into trouble. But I think we showed very quickly and a-- [? all ?] be given a simple example, we've got a functioning application up and running in under 90 minutes.

Unfortunately, we charge by the minute for our service. But we've seen really just enough to get you in trouble of the hands-on how to build up a location. You're going to need a lot more information before we take the next step. But certainly, we're going to be here to help support you and provide you with resources and whatever you need to get up and running with that.

But what you need to do before you really get to the application level is build out for yourself, what's the product value? What's the business need? What are you going to do with this?

Are you looking to lower costs? Close the design loop? Get instant feedback from when we put our IoT bot in the field rather than waiting for it to break?

Do you want to build up new service revenues and actually start to make money within your business revenue stream there? So I know when the filter is going to break. I know when the battery's end of life is. And I'm going to send someone out a battery, a filter just when they need it.

But now once we know that information, that's that next step to taking on some of the risk with the customer and actually managing the whole kit and caboodle and actually doing your-- instead of selling IoT bots, you're selling clean floors. So I know when the machine is going to break. I know when it needs servicing.

So I can actually take on that risk and sell it as a service for $50 a month knowing that I'm not going to lose money on it. Good for the customer. Good for us.

So once we've got that value out and we're ready to build our application, as we saw, we're going to model our digital twin. Get that object model fleshed out. We're going to set up the application with routines so we can start routing messages and form information to set us up.

We're going to test it with our emulator to make sure everything works before we plug it in. That can be embarrassing. So best to emulate it first.

And then finally, we're going to connect it to the real-world asset. And that's the point we're at now. We're going to sit now and wait until that device actually gets clicked on in the field and starts sending us messages to that device.

So a little bit of-- a few housekeeping notes there. If you want to know more about IoT, the software will be down on the expo floor. I believe that's still open.

So we can go and see them at 19:12. And they'll certainly have a bit more information around the IoT industry in general, how you guys can get engaged and what you need to be looking for, what sort of advice you need if you're about to embark on an IoT business model. Also this guy's great. He wrote a book that goes really in-depth about all the layers involved in setting up an IoT or a connected product.

And he'll take you through step by step in here what the terms of reference are, what you need to know about when you're selecting a provider, when you're selecting protocols, all that sort of stuff. So that's very useful. And we have an online course that you can actually do-- I believe it's a free course-- to get your knowledge up in general with IoT.

Most importantly, you need to fill out your survey. And it's important you do it because if you don't want to hear from me again, you need to really ding me on that survey. But if you liked it, then please fill it out. And give me a high rating so I can come back next year.

If you've got more questions, the answer bar is open. But we'll certainly be around. We'll stick around out here if you've got some more questions.

If you want to get contact details with us and have a long conversation later, more than happy to do that. So thank you, everyone, for your attention. I appreciate you guys coming on day three in the afternoon. Thank you.

[APPLAUSE]