Switching back to switch back and forth here should be fun. So I got my my data and the driver that interacts with the data. I got my interface which tells me the rules of what I can do and those things are implemented in my driver. And then I have my model. Right. And this is where it gets a little interesting. OK, so this is dim. I don't know why but anyway this is the model. Right. And that this is this is as far as we got with the DL. The data access object pattern. pattern we see. And the M in MVC stands for model. So in what it stands for, and again I don't like the order of it, but everyone uses it, so it's model view controller. And the idea behind model View controller is I have a model. And I want to make it accessible in different ways. There are different views that I might have for the model. And see the controller is the thing that kind of mediates like a traffic cop between the model, which is the data I have. And the view of it. And try to think about good analogy on this. It's a little bit as if you had. Your company has a whole lot of of. Stuff, storage data, things, all kinds of information coming in, sales reports, whatnot. And you are a reporting division. So. As the reporting division, you need to have data. but you don't but you need to have all your data in the same consistent format and it's going to be what you're actually working with every day you are the model in this situation right you are storing you are not storing Remember that stands for. Might remember these. OK, so the reporting division, so the Reporting division has a model. It has data that it uses, it gets data using the interface and it always interacts with the interface. But the interface is implemented by this data level data layer. And that driver pulls in the data. OK. But what are you going to do with this? The reality is that you're reporting division. So what you're going to do with it is you're gonna take your model. I'm going to repeat it down here intentionally, but you're going to take your model. OK. And you want to present it in different ways. You wanna say the the CEO wants a an executive white people. And so they're gonna. You're gonna take your data, and you're gonna write out a report that primarily tells tells your CEO whether he or she is going to be able to buy a yacht this year. And that's that's all the CEO really cares about. You were going to create another report which is going to be more of a spreadsheet. Alright. And you're taking it from your data and that is what you were gonna do for your? Auditors. Amy. They need all the raw data, but formatted consistent. Right. That's another view of the same data. I still have exactly the same data, I'm just. Doing it one way for one of these and another way for for another audience. I might also have a glossy magazine. Looks like a magazine. Who? OK, I glossy magazine, which is what I'm gonna tell my invest, show my investors. Same data. Shown different ways. OK, this is the view part. Probably. OK. This right here, the thing that goes between the model and the view is the controller which tells me how do I take this data and put it into this queue? And so that's the controller. Can't stop control over this, but you're gonna have to believe me. Oh, I might get it. Oh, I'm so proud of myself. OK. So anyway, the MVC model is built this way for sort of the same reasons that VAVAO model is built its way. It's a separation of concerns. The model doesn't have to worry about how it's going to get projected. Now I will tell you by the way, that the view that we're going to look at now, well, it doesn't feel like any of these is it is Jason. Hi. The cat was very upset that I was gone for so long and she has been very clean OK. The view we're gonna have is Jason and Jason may not feel like abuta you, but from a computer's point of view, it's not the data objects, it is something that is rendered out there in a different format for use by other other computers. So sometimes the audience is actually another. Somebody who's consuming my API for instance. It still has a model which is that Java objects. And a controller which is the thing that says. How do I do this this thing and then the view which is the actual JSON or whatever you just put out there. Alright. MVC is a very, very popular. Pattern. Diego is 2 but, but NBC is talked about more. NBC is very popular. You will hear about it. People might ask you an interview what is MVC and you memorize model view controller and they probably won't ask you anymore. Alright. It's about taking our data and presenting it in different ways. To different audiences, that's what it's all about. So in today's. Session. She's lasher. We're gonna have. You're gonna see both adieo and an MVC, because from now on, everything is just going. We're going to start piling on all the stuff you've done for the whole cohort. It's just going to get more and more like. Integrated them. I was gonna say complicated, but that's negative integrated it's it's, it's like we're getting to use it all. So now I'm going to switch my share one more time. I'm going to go back to even those shows sort of the same thing. We go back to this right here, right. We talked about this whole pattern. You got an application that's got his data, which is the model. It's got this interface level and it's got the drivers. And the one that we've been using has been JDBC and JDBC was the driver that lets you interact with a, you know, a relational database. But today, we're not gonna use JDBC. And the reason we're not going to do it is because we have to leave something to teach you, you know, tomorrow. No, we have to do something for a little later. But I also wanna show you that we are using a different implementation this layer, the DEO layer, will look exactly the same and the application should look. Exactly the same. Because it's only the driver that really changes. So the driver that we're going to look at today, if I can figure out which thing I'm actually looking at. Server side my server? Yes alright. If you look here. Just before we even look at the the details over here, we look over here, we have controllers and we have DAO. And we have the model right now. The reason we have that this level is the model we remember is really shared between the Dale level and the controller level. Right. You may remember, like you might have had hotel, VA. Oh well, now you have what's called memory. Hotel Diego. That's takes the place of JDBC hotel. Memory. DL is a thing you will never use in real life, but it's good for testing somehow. OK, memory just says I'm just gonna store this stuff as data. I'm just going to make arrays of stuff and just store it right here and then it'll just appear magically. But it doesn't really matter how it's implemented. What matters is that it does the same things that we are used to doing. With our hotel or. Yeah, that's our hotel. Diego. Looks like this. I can have a memory, hotel video or I could have a GBC hotel, video or whatever, and they're all gonna look just like this at the interface level. And when you go into your application, it is going to interact with them. Not sure what's going to interact with much of anything right here, but it's going to interact with them the same way. No matter what the implementations matter where my data is stored. That's what the DAO model gives is that I don't care how you store it, I just care that you store it. And in this case, we're just storing in memory. It won't go alot into that today, but I just wanted you to be aware that that's why you've got these memories here, Hotel Villa or memory reservation here. Or is it just another place to store it? What we are going to talk about today is controllers. Controllers now are the thing that goes from one place, but goes from the model. And. Serves up the information. To somebody who needs it. Right. So we have a hotel controller. And the gist of this hotel controller is that when people, whether they're running from our client that will run, or whether they're running through postman or whether they're running from some other application. When they want to do their getting their coast and their put in, whatever else we're now in charge of that. Alright, so we are going to. We're going to implement. What we need to in here so that someone can use our API to get access to the model data? But one other thing that I wanted to emphasize though, and again this is back to big picture. Love the screen sharing things. OK. The thing that is that is. The thing to bear in mind, and I mean it's one quibble. I have sort of with the way we teach. We teach everything at a simplistic level first, and then make it more complicated. Alright intentionally to sort of be able to teach, to get through the concept without getting too confusing. But on a very important part of both the DAO level, where you've got the data here and the NBC level where you're putting the data out is that. I don't necessarily have to be putting out all the data. The model has a bunch of information in it. I may only put a little bit of it out. I may put all of it out. I may put, you know, parts of it integrated with something else out and in The Dalles model, there's a whole bunch of stored data and my model may not include all that. Like if I have a company and I'm, I'm the report ready to division, that doesn't mean I'm going to pull in every bit of data. Company has data ohmygod. They have tons of data and I probably don't need it. I don't ever need to write a report which includes the addresses of our customers, but we need the addresses of our customers. I just don't need it to report on it and so. Part of the DAO model. Is a filtering process where it says. What is it that I'm going to show or what is it that I'm going to store? And the other way around the MVC model part of the goal is to say what is it that? What do they need? I may not. I may have stuff in here that is confidential to say that the that both the CEO and the auditors may have access to this information. I don't want the the investors to know particularly this negative. I I don't want the investors to know any of this stuff, right. So there's stuff in your model. That gets filtered before it goes out here. Right. In addition and you'll see this soon as we get into capstones and we start getting into the web or whatever else it may be that someone puts information in through like when they post information. It may go into my model. But never come back out. If someone wants to read it. Classic example of that is if someone is registering and gives a password, it's gonna create a hash value. It's gonna store that my model, but I don't wanna give it to anyone. So it's not. What we've been showing is so far has been a 100% essentially a 100% representation of data. So in the VA we have with the you know the whatever it is, a hotel that would store it in your database and or State Park. You know if you've got a park it all the information for the park stored in the database. Then we showed all of it in the model, but that is not really the way it usually is. There's usually some either the modeling gets built up of multiple data sources, or it filters down information, but that's in the real world. The complexity that you have here is the model both pulls information from different places and pushes it to different places. But I I just wanna make sure you understand that 'cause. We're gonna still be. Pretty close to mapping one to one, but that is not a requirement of the model. That is simply the way the examples are going to work so that you can see what we've got, you know, see how it works. Alright, so let's go back to sharing. Alright. So. Do you really want me to ask you this? I feel like this is a gotcha kind of thing or whatever. This is going to be our controller. Can anybody who did the reading tell what is missing in this controller just by looking at? I fail to see the pedagogical purpose of this, but but OK good nobody answered. I can just tell you in order to make a controller or controller, this is what. Now specifically the type of controller we're going to do is going to be a rest controller. So it is going to serve up a rest API. You guys heard us talk about spring or spring boot. Spring is a. Very large framework of of. I again back, I think you may have heard the term frameworks before frameworks a whole bunch of libraries and stuff put together that you can that you can use to to build your applications far more easily. Spring has all kinds of features and that we were using spring stuff as part of our day to day oh. Spring boot. Is well, spring MVC is a subset of it. That is all about the MVC thing and it has a lot of parts of it and a subset of spring MVC is spring boot. Spring Boot is the thing we are using. To drive. External APIs. That is what it's called. So when you hear someone say spring boot or if you are clever enough in an interview to say. Well, we use spring boot to drive our web API's. They will go. Yeah like you know. You will sound knowledgeable. We call it spring shoe. Then you will sound unknowledgeable. So remember spring boot. In order to make it this into a an actual controller, we need an annotation and I will tell you that in the MV in the spring boot model annotations are almost everything you do. Right. We used annotations a little bit before. This is the. This is where they really come into their their own right. So we're going to just say. It's gonna have to be a rest controller. Rest controller says tells the the. Java with spring boot package this. Is implementing. Is it controller for a rest API? And then it makes that process. Really really easy. Like even easier than the stuff we were doing yesterday, this is easy, easy. This is like OK, all I have to do is tell it for each method. I have to tell it what is the path I'm going to. And what kind of, you know, they call verbs, but which which kind of HTTP method like get or post or whatever else. And then you just little little tiny bit of stuff. To make sure that you are returning the stuff that it's looking for. Alright, so when I have if I have this rest controller up here. Then I can put it in a tation before any method and this just says when someone comes to my server. And they use the path slash hotels. With a GAT, what do I give them? And? You're literally in this case like this may look to you like we are simply, you know, like when you we have Subs of things that are not filled out, but this is actually the entire thing. This is all we do. We say we're gonna return. We're going to use the Hotel Diego, which is the interface method I interface. So we're not interacting with the hotel. The memory one. We're going to say, use that DEO to give us a list of hotels. Let's hop over here for a second too. Hotel vaeao. Alright. Hotel Vaeao has list of hotel returns. A list of hotels and it's called list. If I go to memory, hotel, DAO. Then we're going to. For getting some of the rest of it we get list. What does it do? Well, it turns out it's super simple because all it is is the ray of hotels. That's the entire memory storage. Is that a ramp? Hotels. So what do we do? We return the hotels. But all of a sudden, by putting this annotation here. Right. And and doing this I can. Push that out. To the user, but remember the user doesn't know isn't they're on a, they're in postman or something. They don't know anything about what I have as my internal model. All they know is that they want a chainsaw object. Spring Boot then takes this whole thing. And then looks at the DL and whatever else and does the mapping for you to turn this into a Jason object. Alright, so let's see how that actually works in reality. Alright, so let's run the hotel reservation application. Alright. And it's going to do this, it's going to start up seeing spring boot. The only reason it does this is because I want to put the word spring boot in front of you lots of times so you remember it, but it's just a, you know, a logo or whatever and then it puts out a bunch of stuff that is happening behind the scenes mostly. So you have an appreciation of how much is happening behind the scenes. There are a whole lot of stuff that is going on here. Then you don't need to really understand, but which is all happening. It's all spring related things, so it's just doing the logging. The one thing I will mention is this Tomcat. Apache Tomcat is a servlet engine. Hey, that just means it takes a bunch of Java stuff and it it treats it like a server, it reads, it does all it's, you know, it knows how to read on a port and all that kind of thing. The only thing of all this that really matters. Is that right? There Tomcat was started on Port 8080. Once you know that. That I can go out and I can say in my. OK. You may have noticed yesterday when I was typing. Sometimes it says 8080 and sometimes it says 3000. Reason it said that was because I've been running both of them and you know that that's what you get for testing room. So yes, when I run this. Now look again at the. So the last thing here, it started hotel reservation I've seen. I think it'll tell me what bushings are having. I don't remember for sure, so if I go out here and I run this. And it's gonna return something, let's say reservations. I don't have reservations. I haven't implemented. So I'm gonna say. What is it we actually used here? We're going to say we're actually implementation is hotels. So I have an implemented reservation. I have implemented. Hotels should return a list of hotels. So let's go over to our. I'm trying to model what you are going to do because you will make all these same mistakes because that's what we do in life. Alright, so when I do this and I get my hotels right, it's going to give me much like before a list of hotels. If I go look at my. Go to the right place. Let's see if it showed me. Oh my gosh. Yeah, I don't think it's. I don't think it is. I don't think logging is turned on yet, so. You won't be able to see it that way, but if I did not have this running. Wife were just shut this down now, which I will go ahead and do. And I go back to postman and try to do this. It's gonna say. Error could not send any request and it it just connection was refused. That means nobody is listening on the port. So I'm gonna fail. So one of the things that will happen is you start running these things is that you will just predicting one of the things that is likely to happen is you will forget to start your server. This is what it looks like when you forget to start your server. If you go into your server again. Can you start it up again? Now my server is available. And I can send again. Alright, my server is available and reading on port 88. Right. So it gives me the information I need. Now I want to just point something out to you because. You probably catch it, but it will be important later if you look at one of these hotels. It has an ID, it has a name, it has an address, a state rooms available. Cost per night and a cover image. Reason I show you all that is OK, now we know this is running. Now is where I'm going to switch to, hopefully the right one of my different, you know, things. The other thing will be very hard to do. This is my client. My client similarly has a model. It has the services that we saw before. But if you look at your model. And you look at your hotel, I have an idea named Stars, Rooms available on cover, which I have no address. and store that information. In my model on the client. So remember the client well, it happens to be on the same machine for you guys. The client is actually it could be a totally different company. You know accessing this API they only need the stuff they need for their own application. For their application, they don't need mattress, they just need to know which hotels there are. So the model that I have for my client. The model is all based on your application, your personal application, my reporting division. In in my example with MVC. Needs one set of data out of all the data the company has, but my sales division probably needs a completely different, well, an overlapping but different set of data. No. So. This is where it all gets a little hard to keep in your mind about what what is what. Everyone has their own spheres of what they need. APIs are available that will give you information. In the. It's at this level when the services and I have my hotel service right? I went out and I got my staff. And. What did it do when I tried to turn that? Jason is a reservation. If you had a hotel, what did you do with the address that it doesn't have? It just ignores it if somebody sends you Jason, that has a whole bunch of things that you don't care about. It only maps the fields that actually match. So you can have a subset that you need. That's a lot of talk, right? Frankly, tired of talking. But you know, I've got the rest of the lecture. Yeah. So does anybody have any questions about? What we have so far before we take a break? You should be the last thing you said about about the the Jason. I'm sorry. I think I just missed it. If I am in here and I'm gonna map me, see if the hotel is in here. OK. When I do like list hotel and it goes out and gives me an array of hotels right? When I call my API. It is going to, it is going to return. All of this data. Well, it's good to turn all of this data because it's it's it's an array, but for each hotel it's going to return. That data. Right. My model. For my hotel doesn't include all of that data. It only clues the ID, name, stars, rooms, available coverage. So if I look at that data that I'm getting back, I've got the idea in the name, the stars, the rooms available. Got cost per night and coverage was prosper night in there. I don't remember right. A doctors never cost, so both the cost and the address. Are not here. When I get my Jason in and it deserializes it to store it into my object. It's going to do math, whatever it can. But if she's gonna ignore the rest. The fact that my Jason that comes in has an address, an entire address object with all this information, and it has a cost doesn't matter at all. I just won't use it 'cause I don't need them in my model so. Well, we may have been doing a one to one say with the database and the you know model or or the API. The model that is not necessary and here my model and my client it just doesn't need cost per night and it doesn't need address. It might not need other things even though they're in the JSON. Spring Boot will handle all the mapping back and forth to deserialize it into an object or serialize it the other way. What it does mean? I just wanna emphasize and you won't need to worry about it right now. But if you on the client wanted to put that data back in, you wanted to update something in that hotel. You would screw it up. And the reason he would screw it up is 'cause. You don't have the address, you don't have the cost per night, but when you do a put. You overwrite all the information so. By the fact that I'm not storing everything, it also tells you pretty much that I'm unlikely to be doing anything with these or that there are default values, so it can use OK. sorry beth said let us go until about 10 let's take a break until about 10:10 and we will come back 'cause this is a

To the server we're creating. And. We should start filling it out so that it handles not just hotels, but also handles reservations. Stop my server. I have to tell you that they're getting more and more magical with the way Intellij interacts with all this stuff. And I am. Throughout my entire programming life, you have always had to stop things and restart them to make anything work, and they're making more and more of it possible to run, to change and keep running. I couldn't tell you for the life of me which things I'm allowed to keep running, and not because I haven't had to do it so much, and I'm usually just writing all the code and then starting it and debugging whatever else. But I I think somebody showed me last cohort that there was like magic. That happens so I could change code in here and then even though it was already running it would it would figure it out and start using the new stuff. If you guys wanna play with that, see what happens then you know have at it. You are far more like, I mean, far, far, far more likely to go out, have a job where that actually matters to your job than that I ever will. I'm going to teach you guys stuff, but I'm not going to use this stuff in a professional capacity other than that. Except my own stuff and my own stuff. I have total control it. I can just stop with him saying I don't care whether that works. So just telling you that if you see somebody else do it, or if you make a change and forget to restart and somehow it still works, that is not. There is some magic going on there. I just and I this may be one of the places, but I'm not positive. I know that when we get into the web stuff. More of that magic does happen once you get into JavaScript, a lot of it is sort of interpreted on the fly, so I know that'll happen, but I am going to stop, stop and restart it. Oh, also I did have a question that is so if you Scroll down just to where you were just a second ago. So here this is this, this spring boot or whatever this framework is. Does that enable us to actually have a an interface call a method? 'cause I thought that that was a big no no. Generally speaking, in China interface column method the onoway to excuse me. So hotel ndio I'm so sorry that is. Go ahead and ask the questions 'cause. Alright, well, there's. I just realized I just, well for everybody else. 'cause, I just made this. Hotel to go if you scroll up, that's that's where is the OK. That is sorry structure. Sorry. It was an instantiation of totally understand the construct. By the way, just so you all know, first of all the constructor is. It does some magic, it does some actually not relatively. On magic stuff, that kind of breaks the model of our not knowing which DL we're using. We are yet to get to something called dependency injection. And dependency injection says that it will magically use the stuff that it's supposed to use instead of my having to say, well, I know it's a memory reservation DO, so I'm gonna put it in here. Because we're having great cut in there. we do it this way all we do we start are and gets the single hotel from the Hotel DAO. Um. Collection. Right. The only important part here is you need to make sure there is always a path area over here for everything that is in the curly braces here. Same thing is with the with the sequel calling it. If you don't have something, it's going to cause nastiness. So what does this look like? I'll actually run this one more time and we'll go out and see what? What is this actually look like? My server is running, I have hotels I can see I'm gonna do hotel #2. That's my favorite hotel. Right. Hotel #2 is just gonna be. Hilton Cleveland downtown. I never stated that I had no reason to like it. It's just it says Cleveland. So you know, right? I used to walk away too. OK so here is the this is a single object because the way we have defined this, it should return a single object. But the way we can relate that back into our code. Is that? The return value form I get. I remember God is now the. Yeah, it's just here. It could be get hotel. If I wanted to get hotel then in here I would call it. Sorry in the controller I would this this method is just a method that is implemented here. Alright, so the fact that it's called get and this is again it's coincidental. It's not important. Like it's not surprising because that's what we're doing. But usually we take at hotels. So. Now let's say, well, what happens if we want to get? Let's expand this. Let's let's make it do a little bit more. So we might want to. I just follow the model. I've always creating the javadocs. Remember, this is the javadocs that we talked about before. This is the thing that lets me hover over it and have useful information. That's not useful information, but however over it again have useful information. Because it will automatically pull this information about the the individual method. Let's go ahead and do that. Let's get. Let's match. Let's model this, but we want to get all the reservations. OK. Gonna look at something, actually. Yeah, I'll just do the the first part. We can do the rest ourselves. In a simple system. We're like this one. We're gonna go ahead and put the reservation stuff in the same controller as the hotel stuff. If you had a much bigger, more complicated system, you would create a separate controllers. That's just kind of a design choice. This is how we're doing it. Right. So we want to do something like this. So we need to do a request mapping and make sure you can see it. Also you know what I'm doing. Yeah. OK. We're gonna do a. Alright, let's do a. Your list. Payless to reservation. She's gonna make it. Is going to be a method that returns a reservation. In order to do that and go see what is the name of that method? Well. Let's do find all that seems to be the one that we're going to implement here. So let's do find all. And then. we wanted to take in nothing dot is memory reservation. Yeah. Or reservation here. Find. In hotel. Just let stop. It shows the same one as the final. Final. testing testing test and or Tatian. I want to do at request. Path Knoops quest mapping. Equals switching our path B. From what you saw yesterday. I'm getting a list of all the reservations. For your reservation? Sure. Yeah. Should be reservation slash reservations. Right, and the method should be what? Request method dot get on top. Yep. Gonna look just like it. We're still getting information. Choir tape this list of reservations. And. Problem means I have to import something. Presentation. I've got all this stuff. So am I doing round here? find out your reservation i think i said let's reservation reservations Find out your reservation. I think I said list reservation or reservations, but it's just a guess. Alright, so now if this works, now let's. We've saved it. We do reservations right now. OK, this is giving the error message status 404 and error message. It also returns a 404 here they just chose to give us back Jason which gives us some additional information. And that's when I said Gate Rose I spring boot will by default pass back a Jason error message like this. It also has to conform to the East. Quest, so we'll also have the 404 here. There's nothing about the standard that says you have to have this. That sounds Spring boot does it, so you're gonna. That's what you're gonna see. So that is now let me stop. And restart and see if I can. OK, so now because I implemented those like 3 lines of code or whatever, it now will be able to pull in my reservations and and put them out. So what else do we have in our reservation D AO that we might wanna do? We have our. We have find all we have the list reservation. Find by hotel OK. Make my life easy here. Say, let's copy this over. We want to controller method which does that thing. I'm gonna copy this. Never changed and say reservations. Turn a list of all reservations. It's only problem with copying and pasting comments. If you have to actually do something with that. Otherwise it's the same comment as the other one. So I'm gonna say. Say, let's return all reservations at a hotel. Ciao. Turn list of all nations. No. OK so here I've got my. This is what I'm going for. Then we want to say what do I need to do? To implement this. Yeah, I will make it easy on you in the fact that. Perhaps something else above you that has. So what do we need to do? Center then at Acquests mapping. Path equals flash. Clash reservations. I'm not sure. No, that's not where. This whole. It would be slash and then open curly based ID. But you need to make sure you've got. Variable part of it, the thing that where they're going to specify an individual one. And then here trying to make this a public method when you get the same name as as before 'cause that's sort of what we seem to be doing. Alright, alright. And now in order to make this into hotel ID associated with this. Crystal. I'm gonna change it to ID, which makes just my life a little easier. Only you could change that. Right. So it's going to make it an int ID and I'm going to do an app path. Variable. By the way, if this is ID and I want this for some reason to be hotel ID. I can do it by could say I could have left that as hotel ID. And I could have made this as name equals ID. So. This is now my method, So what am I going to have inside here? It's gonna be as short as all the other ones. Turn reservation DAO. Reservation Go dot signed by hotel. By hotel. I'm going to pass it in that hotel ID. With having that NP hotel ID and like what you just went over before that is that like six of one, half a dozen or the other like it doesn't really matter. Yeah. I mean, honestly, like I would almost never do this. OK. I'm showing you because there is a way to map them, but. Like. You just leave everything as ID. It's not gonna. It's not gonna matter. Yeah, OK. Because remember, this is just a program. This is just a local parameter. So it's only going to matter for the context of this, the fact that it's called something different in somewhere else doesn't really matter. OK, right. So let's say. Stop. Start. Say the hotel ID one seems to have at least seems to have all that H1. They're all at H1. That is really very boring, right? Is this really the path that we want for? Our hotel by ID. I mean, it is gonna be like I if I do reservations. I think I've got the wrong method here. I think about it again. If I do reservations ID, what would you expect that I am trying to get back? A single reservation. Right. And what would that single reservation be? We gotta be the one that has that ID, right? So I have implemented the wrong thing here. But fortunately we will need to do this one we I just picked the wrong one when I was sitting here looking at the reservation, I said fine, my hotel, what I really want is is they call it yet here. So I will. Make my life easy. I will say, OK, this was all a good thing. It just isn't the right one. So I'm gonna. Copy this. We will go back to that in a moment. We're going to say what I really want is fine reservation. Iid. Right. OK so. This path this mapping is still right. I never put in my. Method. It defaults to get so, but it is good form to put it in anyway. The reason I say it's good form, by the way, it is possible to set a different default and so it's better to just be explicit and say I'm doing it again. This one, we're going to say, I don't want to do. I want to just return a single reservation. And I wanted to return it and it's gonna be called get and then I'm just going to call and get. Alright. Terrible. Alright, anyone see what I'm doing wrong here? Get reservation. Oh, it's having problems because I'm creating a new method. In the same controller. With the name get. I already had it yet. So we can't do that. So if I change it to get reservation. Now that can implement the reservation here yet, but now it is get reservation. This name doesn't really matter at all. It's just kind of in here. You're never gonna call it directly. Run it again, see if I manage to get that. What do you think's gonna happen? Do you think this will work? Reservation ID is one right. I'm predicting this will not work. So let's say you're worked. Reason I was predicting it. But I didn't know for sure was. I am using that path and right up here I think I'm using the same path. Oh no, I never changed it. I never changed it here so I it is not a problem. Or I'm sorry. It's down here. I change it again. I'm down here. If I put them in the opposite order, wouldn't have worked. Be careful not to use the same path. Because the first one will win. Never understand what I mean by that. Right. It's just it's gonna find a mapping and then it's never going to get to the other mapping that happens to use exactly the same thing. So in this case I have to have something different here. What do you think the path really should be for find? By hotel. Tell me. Remember wells/ reservation. Should it be a query string? There are a couple ways it could be done. It could be done like hotels, but now it's got something different. If I am specifying by hotels and I want to get the reservations that I need to put another ID in here which is hotel ID. Right. So it could be done like this and then I could set a variable for the hotel ID and a variable for the window. I want all the reservations start I so I don't need that this one. This is actually a little easier, I'm sorry. I get I was starting to think you would need two both, but you don't need both. So this is the ID and they have find reservation ID or that would work if I had wanted to get inside a resource and an ID and inside another resource and ID that's when I need to change my name 'cause I need to figure out how to what to call it. So this should go out and guest all the hotels that if I say hotels whatever slash reservations it should give me that let's stop it start it again try it. Hey, we have reservations instead. Hotels. Slash 1. Slash reservations now by by normal rest standards this is it says my resource hotels. It gives an ID, so it's just that hotel and then another sub resource or resource within that hotel is reservations. So this should. Do this thing. So it returns all three reservations for that one home that are all happen to be in that one hotel. And then we jump back. Yes, there are so many places here. To make sure that we are doing the right thing if we jump back in, we now have this running and we have implemented. Some of these, so we have we got findall. Yeah, fine. By hotel. We gotta get. We have not created done create yet. If we switch to our client now. Where our client is. We switched to our client. And we run our client service. Hi app. Which things does it want to run? I'm just trying to figure out, I don't remember which one we have to implement, so we should be able to do 1. Testing one by one just to be sure. OK, they've got a list of hotels we should be able to list reservations by hotel. Two. And then we'll pick one. And that list of the reservations. Create a new reservation for hotel and update an existing reservation. So remember this is all the same stuff we did before. We just deal with a fake server and now we're doing with ours. So we're implementing the things one by one. So we can't create if we try to create, what does it say, not implemented yet or something probably? Once this quote all the information it just says know what to do with it. So let's see what it does when we actually do this, because. An error occurred. Check the log for details. Does it actually give along? Maybe not. Yes, it doesn't give a law. I don't know what the log would be at this point. The point that I'm going for more point that I'm going for is, OK, this is not working. We want this to work. And by the way. If you look at the data that it just put in here. It's syntactically correct But should I allow it to create this reservation? It's two years ago. Should I be creating a new reservation two years ago? So the other part to think about when you're create, when you're looking at all these pieces again when you get to your captures, you're going to be putting all these pieces together? Alright. The thing to remember is that is where does it think about where the different kinds of logic belong? If I have business logic. Right. That logic may well happen at the level of well, if I have if I have logic about. What should be allowed to be created? For instance, that's probably really model information. OK. Thank go to my hotel in here. Maybe it shouldn't allow me to create, you know, set one subject instructor. Maybe I shouldn't allow me. To create a hotel or I'm sorry, a reservation. I'm looking to lock it. Maybe it shouldn't allow me to create a reservation. That is in the past. And then I might put that logic here and and and raise an error or throw an error here to make that not work. If I have database. If I'm storing in a database, I may choose to put that data there because it's going to come bubble back up as an error. Think about where your logic belongs. Alright. But let's go back in 'cause the easier thing right now is let's create the thing. Alright, So what is it going to be doing when I do that create? Please, when I do this create. What is the code that it is calling? It is calling my hotel service. Is going to be calling a. First add. Yet Update alright. So what are we doing here? We are doing a post. And we are it is using a. Serialized version of the reservation that we have here. So if we look at the reservation because this is what it should be serializing. It is serializing these things. Notice this thing right here. That's very important. This is another entity. This annotation says that. Even though its name is Hotel ID with lowercase D. When we save it to our Jason, we want it to be hotel ID with another HD. I don't know the rules. I don't know why. I just know that somewhere my server wants me to have that variable. Now I feel offended by having to use that in my Java. So instead of justice using the name, I went ahead and. Used the camel casing that we use in Java. They on the other side don't feel so obsessive about that. They want to be out pictures. Maybe This is why that's there. So when it serializes, what am I going to get out of all this? I am going to what is it gonna post? Well, it's going to look very very similar. 2. Tells. It's going to look similar to this, but without the address and without the. Across parent person when I'm doing a hotel, if it's a reservation, it's gonna be. What am I doing? I'm creating a reservation now, right? Hi I'm losing track of what I created. I'm creating a reservation, So what it should look like is the reservations. Alright and. To have all the same information, I think I have all the same information in my model, so I have the ID hotel ID. Full name checking date. Check out date gas. Think I have everything. Should serialize and deserialize properly, but again, in here it's going to be hotel ID with a capital D on my client it's lowercase. I have to do the mapping all right. Now I'm going to go out and I'm going to try to implement this on my. Server, so I'm on my server. Let's go into the hotel controller. And let's try to create the. Whatever is gonna create that thing. Alright. Sorry. I look at my reservation, DAO. Create takes two things. It takes a fully formed reservation. And it takes a hotel ID. So I want that thing and I tend to do this where I copy and paste this because I'm trying to make sure I have this. And I'm gonna go ahead and put it into. Controller right? And I'm gonna do. For sure I put this. Let's create it underneath. Hotel ID reservations. Put it in here. It's gonna have it's. Comment for it. It's going to create a reservation, create a reservation. At a given hotel. It's going to take as a parameter. At Param. We looking for. Create a new reservation. Just take a reservation. And it is not actually going to return anything. Hello so. Not. Does return the reservation? I should really return the reservation. Trans. Surely modified or newly created? Reservation. So I'm gonna do the same thing. I wanna say public. This is the thing I want to create let. Should I call it create 'cause? It's very confusing here. Let's say add reservation. It is more clear. And it is going to have this these two items, right? What does? Say. Where these going to come from? What is the? What? Did somebody tell me what I'm gonna put in my request mapping? Choose the path be. Do you remember from when we were doing yesterday's? Project we looked at the post and if we went to our consuming API's and we went to. No, we haven't update, we don't have. So remember when if you were doing it, if you were creating a single one, your path should be this part. If you're updating one, your path should include the ID. Because creating one, we don't have the ID yet, we don't know how to what to do. I guess I never created a. Right here so. If we go here, our request mapping then should have passed equals. Slash reservations. Something. We're great. So. We need add reservation. We don't have the hotel ID yet. So we're gonna not pass that in. Instead. We're going to use the thing because we know this is going to get updated. We're going to go ahead and get this value from the reservation that gets created here, it gets deserialized. So what is our method going to be? Anybody. Just guessing, request method dot host. So this says I'm going to when someone posts a message to me. Add this endpoint. I'm going to expect a there will be a body and in order to create this reservation right here from that message in order. In order to do the deserialization where it goes from the Jason that someone is giving me to put it into this reservation that gets passed in. I put in at request body. Is waiting. And I'm sorry I'm having trouble hearing I say. So I like you said. Well, that's just said that they had the reservation. Anybody get that better united. I didn't really get that. Peter, could you type that out? Yeah, 'cause. I I'm afraid you're having too much echoing somehow. Alright, yes, there does need to be a. There does need to be. An ID that is passed into. This create. But that if we assume that the person calling is going to to create that ID. Then what we're going to do here, and this is kind of tricky and there's you wouldn't have any way of knowing this, you just have to trust me on this one. When I do the reservation, not losing my track of where I am. Hotel controller. Basically deserialize stuff thing that came in here, right? And what I'm going to do is I'm going to but it. But that includes everything includes an ID. So what I'm going to do is I'm going to say return. Reservation. PAO dot create. And now I have my reservation as you. OK, let's call it. Yeah, measure reservation. But I need that ID. So I'm going to kind of cheat on this and say, well, it's already part of the thing. So I'm going to get hotel ID. And passionate like that. I kind of question why has the ID passed in there, but because they requested it and I have it as part of this, this is the best I can do. This means that if somebody specified the ID. It will give it the actual idea if someone created a new ID and they and as part of their Jason they included the ID it will use it if it does not, this will return a zero and then the create method. Will actually do that work for me. Then I have a quick question. Yeah. So what key to typed in chat might be the. OK. So there we changed a parameter in one of the reservation dough methods, but did we ever implement the reservation Deo in hotel controller? Reservation Deyo in hotel controller. Or is that what we're supposed? Sorry is basically as far as I'm following the code. We're implementing all the methods that are inside of reservation D AO, right? But we're not locked into the namespace because we didn't actually implement that in hotel controller. Is that correct? Right. Right. Hotel controller is not using these these these interfaces? Right. But is is anything implementing reservation DAO? This is a trick that was up here at the top when I did my constructor. I have a private reservation video right here, and I'm gonna go ahead and creating a the new memory, one. That new memory reservation DAO. Is the is right here or here? It is the thing that implements preservation DA. Oh, so all I'm really doing. I know it's a little roundabout. I have a class variable which I saved which is the new the, the. Memory reservation DL I save it to something which is a just a reservation DL. I don't worry about what type is so this is the interface. So what I am implementing this. This is what implements it. The memory reservation implements it, but what I'm saving just by polymorphism, I'm just saving the the interface version of it. And then when I so I have my reservation DL right there is a class variable. So when I go down and do this I am calling class variable. I don't know that it's memory DAR but I know that it is someday out there that I can use the create method and then I guess what I'm trying to 'cause when we started we said we're just gonna throw the reservation. We're not going to make a reservation controller. We're not going to separate that. We're just going to put it all in one thing, which is fine, except isn't that why we're getting namespace collisions? 'cause, they're like. That is why we're getting OK. That's why it's not always a good idea, but. It is what it is. I can't even tell you why I actually, I just know that. That. If I don't follow the model that they tell me to follow for the for, for implementing them today. Then whatever I do tomorrow won't work. So if I wanted to fix both of them, I could make I could change both of them so that tomorrow's starting point so that today's ending point was tomorrow starting point. But I haven't rewritten the whole all the logic, so I'm gonna stick with the pattern. They're they're telling me to do. I'm not even saying a wise pattern. It's just a pattern they're telling me to do, and I don't want to totally rewrite tomorrow's as well, OK? Yes, that is why we can't just follow exactly the same thing. If we had a reservation controller, we would just call this create to map to this. But I guess part of the point is to show you don't have to do that. This game can be whatever. I don't think it's a great logic, but there it is. I want to be clear though on what is happening here with this reservation to not get hotel ID. When we are out here, let's make sure we restarted and started again. So this should create a new reservation. Right. But I want you to see what? The logic is over on the other side. If I go into postman where it's a little more obvious what's going on. All right. Come on. There you go. Alright, so if I want to create a brand new reservation. I'm gonna create a new reservation. Right. I mean, it's gonna be a post. And I know my body. It's gonna be something like this, and I will go ahead and. Tweak it around a little bit. I'm gonna make a new reservation. If I specify the ID. OK, remember the ones we have right now, it's go back to. When I did this, what are the ideas I have? I have ideas 1/2 and three. If I wanna create 4. If I want to create four and I don't specify the end, shouldn't use it. It should create like if I create one more it should create it with ID 4. In order to do that, I could just eliminate the ID equals three. That's what I'm going to. That's what I'm trying to show here. But. Struggling with if I don't specify any idea at all. Right. The logic we had was I'm going to say this new one. And I'm going to say hotel, I'm going to say John Smith, OK. If I am watch, what should my URL be? Sure. Should be bad. I thought it was OK because I'm posting a brand new one. I'm going into the collection to the the list of resources, not to the individual one, so this is what it should do. I'm going to give it this. I am not giving it a reservation ID. Because I am not giving a reservation ID. It has to make up its own. But we aren't trying to show you in the code is. When it got that in, it tried to serialize it into a reservation object. So it are deserialized, so it took the Jason that I just pushed to the server. It tried to map one by one all the very the the field variables. But if you remember my reservation. Here. Right. It has an ID. I didn't have anything to map into there, so it left ID as the default which is 0. Kind. And then my create method where my create method is. Is the thing that actually adds. Will update the ID alright, so in other words. When I deserialized into an object, I left it at zero because I didn't include the ID. If I had wanted to. So and then it created ID for if I had wanted to. I could have done. The ID. Alright. And I'll say 45, that's going to be my my ID. And when I send it. If you look over to hear what it's gonna is my controller, it's gonna deserialize and it's going to have a value that it put it in as the ID. So when I do my create. Instead of returning 0 here, it's gonna return 45 because I deserialize from that JSON and it actually gave a value ID. So when I didn't give an idea it gave it a zero. When I did give an ID it created it this way with where specifies the ID. So what should have happened and uncle see if it did happen? Is it sure created? So why did? Specified the ID. Anybody have a clue why it would not have used the idea that I just did? This is where when you don't know the actual back and you have to look at, I am calling my memory reservation DAO. And my memory reservation. Deo has a creed in it somewhere. OK. Even though I pass in an ID oh, I see what it's doing. I I'm passing. Sorry, I said. I think I used the other ID. Put into their hold on. Think I may have made a mistake here. Oh, I didn't use get hotel ID. And I just set the reservation ID OK. Different problem and like I I made a mistake there I I was thinking it was doing something different. I thought it was giving you the ability to override the the idea it's help. It's not. This is the hotel idea to use Best Buy and I've been specifying the hotel ID. Right when I set the ID. The reason it wasn't using it is because when I go back to my. My memory reservation day. Oh. I get an object in. It's passed in and then the first thing I do is set the ID matter what the ID value is. I set the ID, so if I go right here. Try to do this again. If I go out here. Oh wait, I wanna have to. Stop this. Stop this and I'm going to start it in debug mode. So now when I create it with my ID of 45. At my cell. I'm gonna stop right here. Now. This has been deserialized from the Jason that was given to me. So if I look at the reservation. Right. Sorry, debug debugger. If I look at the reservation, it's right here. It has the idea of 45 hotel idea of three which I passed in johnsmith the dates and whatever it has all that stuff. I passed in a hotel idea of three. It was already in the reservation, but but they didn't know for sure that it was. So we did that. The very first thing I do. Step over this. Is it gonna change the ID from 45 to get Max ID plus one? Which is some method that goes out and finds the right method, the right ID to use. So I'm going to do that. Alright. And it's gonna make it 4 show, maybe four, because when I stop the server and restarted it, it went back to my original three reservations. Just in case anybody wanted twice stopped high or sticks or whatever. Right? So. No matter what I passed in, it wouldn't make any difference. I could pass in the ID or not. Alright, that's kind of a dizzying amount of stuff. So in that instance ID sorry event. So in that instance like deserializing and re serializing it to what we needed to write like layman's terms or thank you. The thing to realize there is the way I implemented that. Right where I did. I'm just posting into here. Right. The resume, the only place the hotel ID shows here away it should. Continue. So just like it furious with me. Right here, the only way it can figure out which hotel it is is that I posted in here. Well, what is the option? I have another option here which is that I could go back to my. Hotels. Slash 3 slash. Reservations. I could have posted it in here. If I post it here, I wouldn't have ever bothered making that and I wouldn't have ever bothered with that, and I would only get this. If I were going to choose to do that. Then I would get the three. From as an ID from from the the path. Right. They would both go through the same basic process for the differences. I would get that three from the path instead of getting it from the, you know, get ID, the reservation that get ID. I would fill these in when I did that it would have. It would create the reservation without with a zero ID and a zero hotel ID. It would call create. It would use the ID from the path. And it would. Do the plus one on the other ID, so fill those in as part of the create. So both methods work and it's perfectly legitimate to have more than one path that will let you do things and specify more information. Let's go and let's take a break until 11:30 and then we will implement the last little bit of this and then we will play with it with our. With our client and talk about the homework, right? Free