

Tomcat 02 IDE Setup Aug 31st

This document is a continuation after building a simple Spring MVC project in the Spring Tool Suite IDE. This document follows after the “Tomcat 01 spring-mvc-demo Aug 31st” document. This document installs a Tomcat server within the Spring Tool Suite IDE.

Apache Tomcat is a free and open-source implementation of the Jakarta Servlet, Jakarta Expression Language, and WebSocket technologies. Tomcat provides a "pure Java" HTTP web server environment in which Java code can run.

Table of Contents

Install a Tomcat Server in Spring Tool IDE	2
Download Tomcat server zip file	2
Configure Tomcat server in Spring Tools IDE.....	3
Open a search window.....	3
Click link to add a new server	4
Select apache-tomcat-9.0.52	4
Select “jre” from “JRE:” dropdown	5
Add the Java JRE from your machine.....	6
Add an MVC project to the server.	9
Start the server	10
Test server using the browser.....	10
Appendix: Recording Transcript for This Document.....	11

Install a Tomcat Server in Spring Tool IDE

Download Tomcat server zip file

<https://tomcat.apache.org/download-90.cgi>

Download Tomcat server zip file to a folder on local machine.

Make note of the folder location.

Unzip the Tomcat server zip file.

There is no installation of files after unzipping.

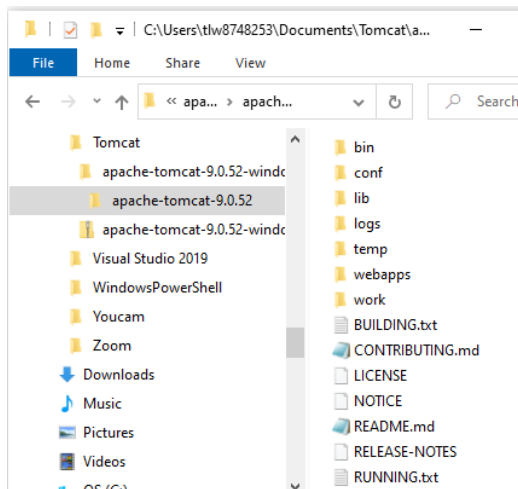
9.0.52

Please see the [README](#) file for packaging information. It explains what every c

Binary Distributions

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

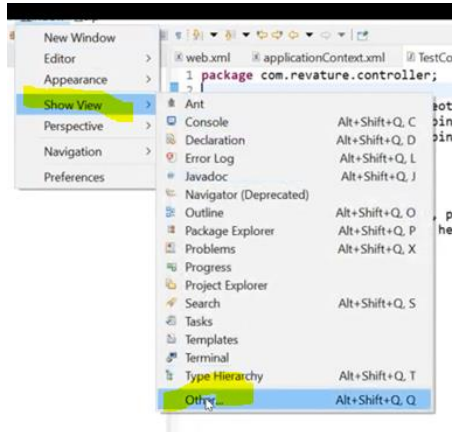
Unzipped files.



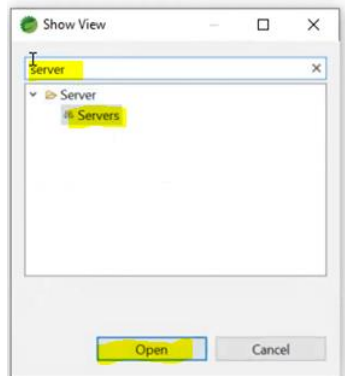
Configure Tomcat server in Spring Tools IDE

Open a search window

Window → Show View → Other



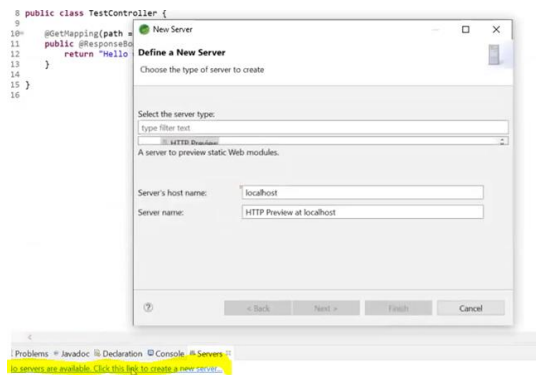
Search for Server



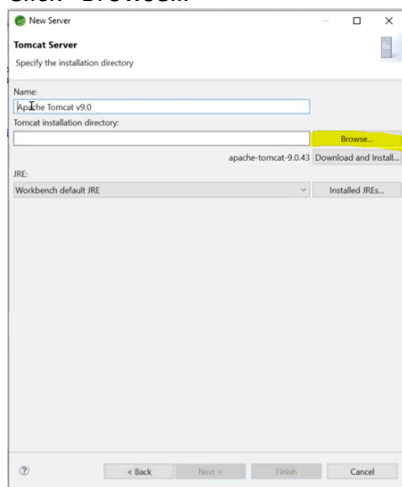
The location of the “Servers” tab can vary depending on your window layout.



Click link to add a new server

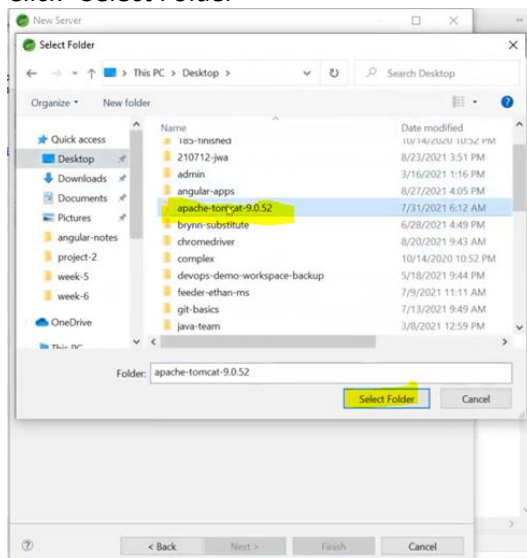


Click "Browse..."



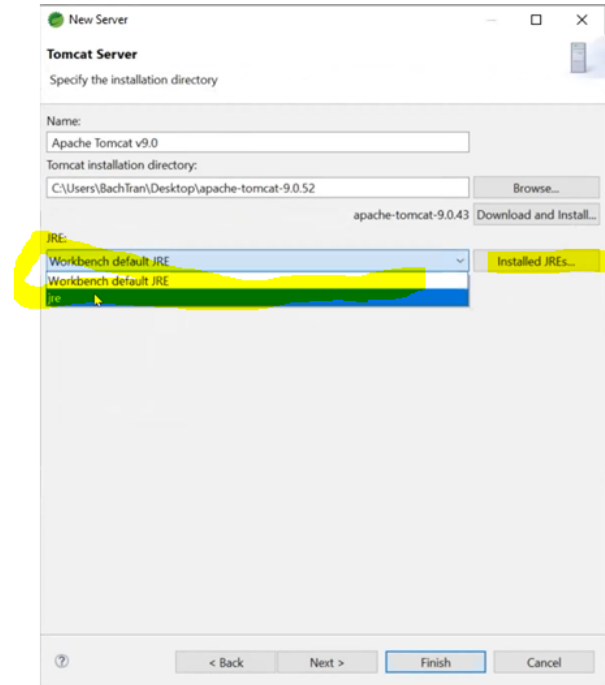
Select apache-tomcat-9.0.52

Click "Select Folder"

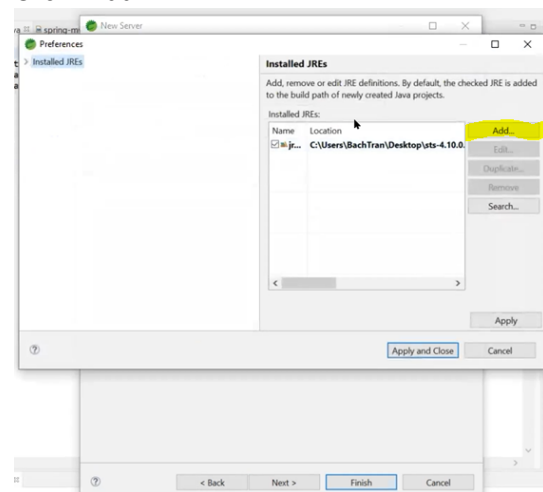


Select "jre" from "JRE:" dropdown

Click "Installed JREs..."



Click "Add..."

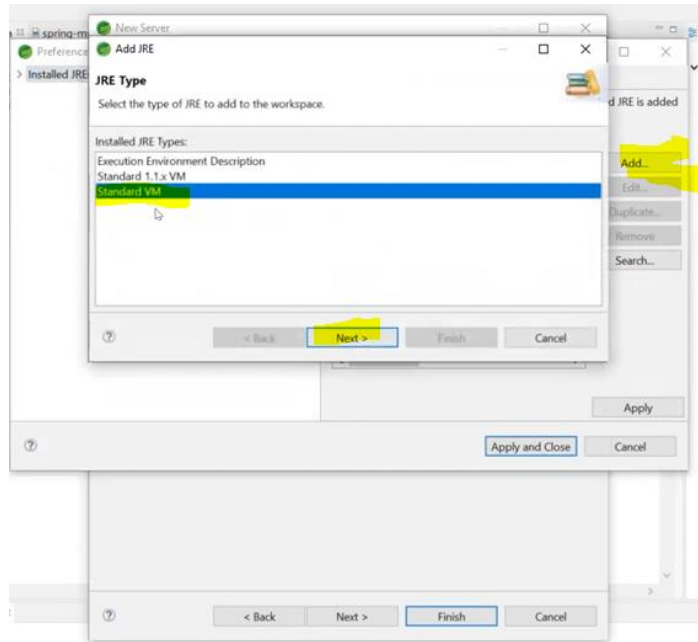


Add the Java JRE from your machine

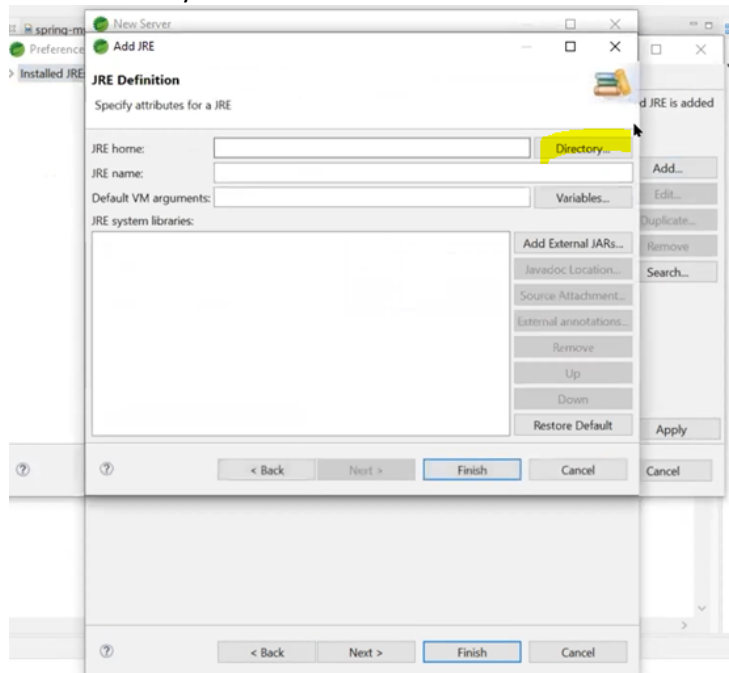
Usually located: C:\Program Files\Java

Select the jre 1.8.nnn version

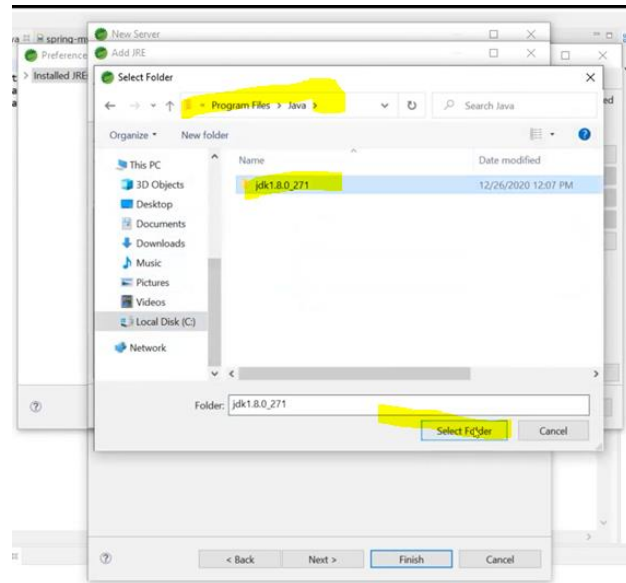
Click "Next>"



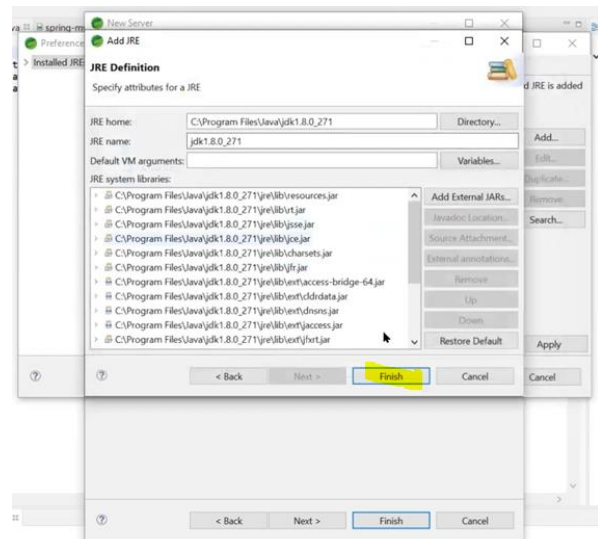
Click "Directory..."



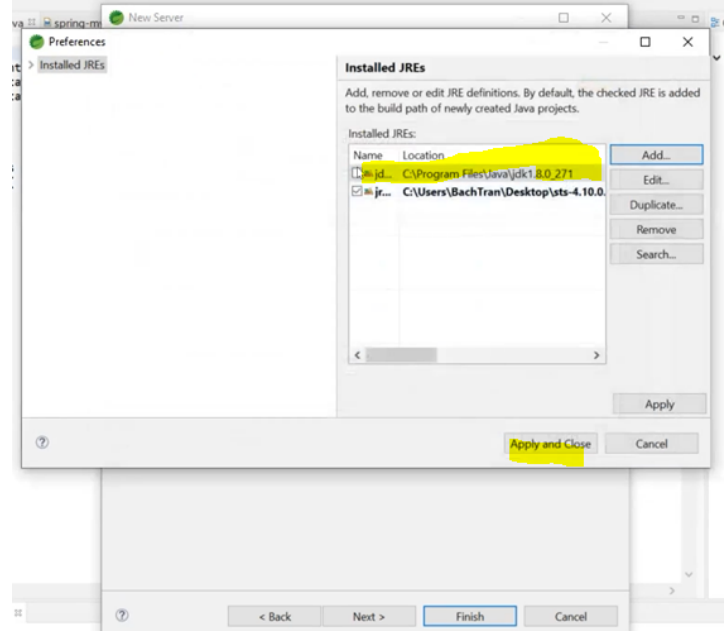
Navigate to the folder where you downloaded the Java jdk1.8.nnn version
Usually located: C:\Program Files\Java
Select the jdk1.8.nnn folder
Click "Select Folder"



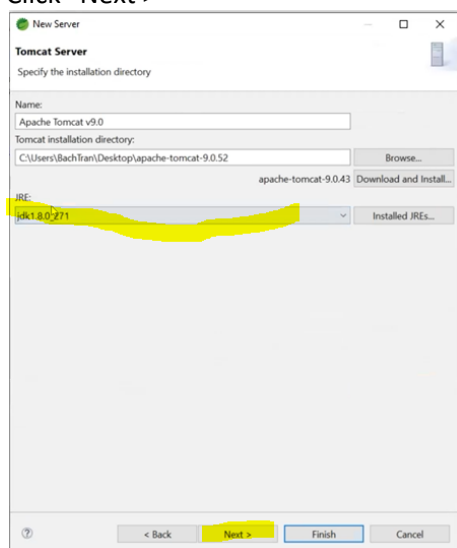
Click "Finish"



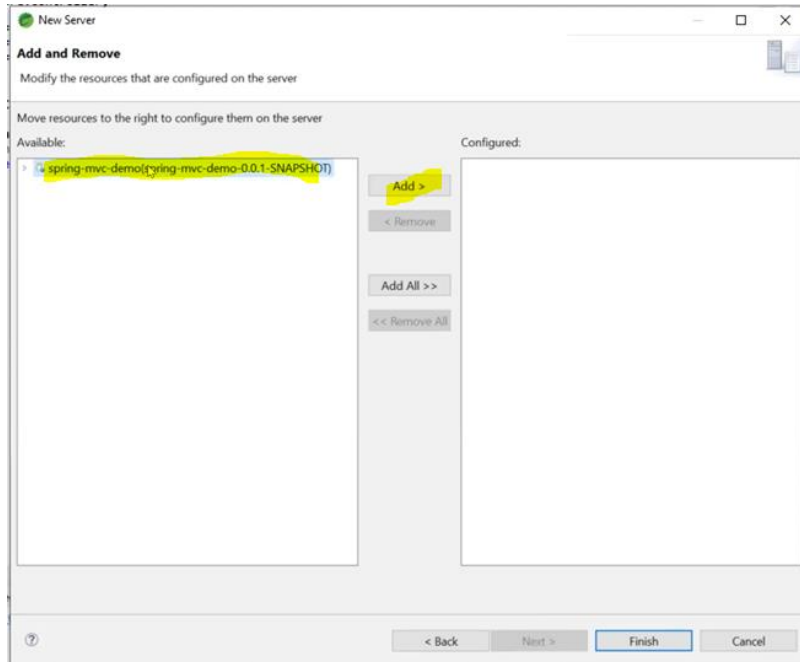
Click the checkbox of the jre just added to make the default.
Click "Apply and Close"



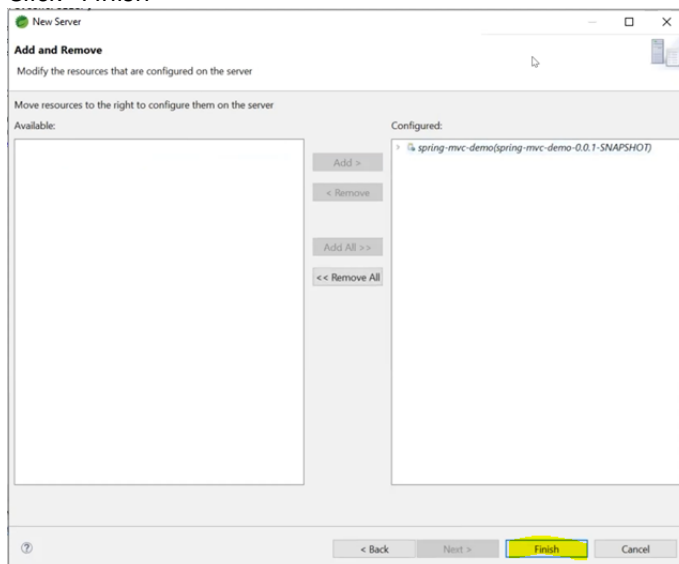
Make sure the jdk1.8.nnn check as the default is selected
Click "Next >"



Add an MVC project to the server.
Select the Java MVC project to add
Click “Add >”

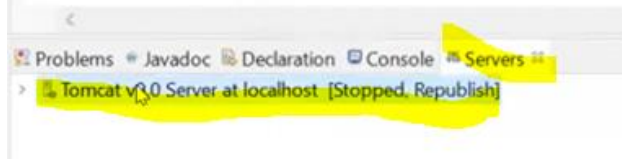


Click “Finish”

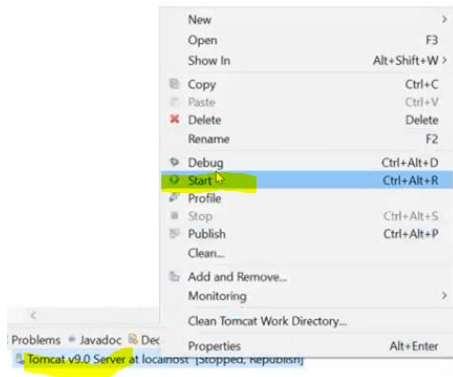


Start the server

Click the “Servers” tab, location will vary depending on your IDE layout



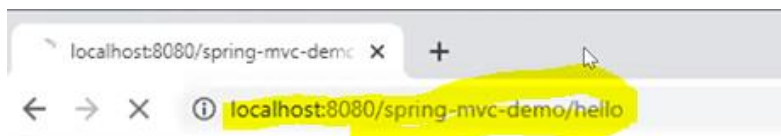
Right click on server then click start



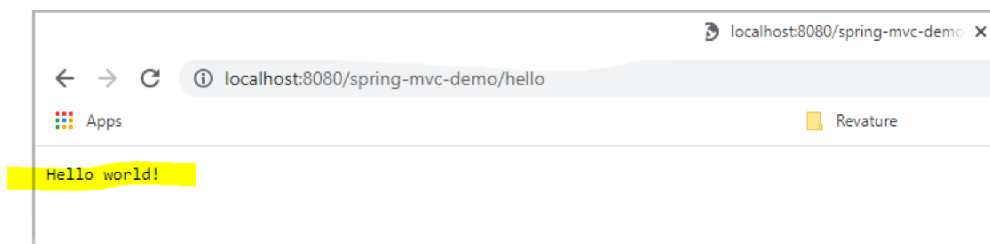
Test server using the browser

This test will vary depending on your MVC project.

For the Lecture spring-mvc-demo there is a hello endpoint and this test should work.



localhost:8080/spring-mvc-demo/hello



Appendix: Recording Transcript for This Document

Bach Tran: What we need is to download tomcat itself right which is going to be some kind of server that this will run on so i'm going to search for tomcat nine.

Bach Tran: So yeah let's let me go ahead and give the link here to tomcat nine.

Bach Tran: and basically you're going to want to install those that version I would say, you could install it technically to your machine but i'm just gonna like unzip it to a folder.

Bach Tran: that's going to be a bit easier it's kind of like with spring tool suite where we don't actually install spring tool suite we just have spring tool suite inside of the folder.

Bach Tran: So same concept here.

Bach Tran: So yeah while that's finishing up there.

Bach Tran: yeah we're just going to wait and then we'll unzip it to the desktop.

Bach Tran: And then we need to actually configure.

Bach Tran: The server to basically integrate it with our ID here so that we can actually like have a server running while developing.

Bach Tran: So what we would do is go to the window tab to show view and then you want to go into the other section right so window show view other.

Bach Tran: And then you'll want to search for server.

Bach Tran: And then click on servers which will add a servers tab to the bottom of your console here right, so you can switch between the console and server.

Bach Tran: Right in the top here, so you just want to add this servers tab here and then let's go ahead and check on the download now it's done I just need to unzip this folder.

Bach Tran: place it on to my desktop here.

Bach Tran: And there is my tomcat server available right there.

Bach Tran: Right so with this tomcat folder I will want to go back to the IE.

Bach Tran: And click on no servers are available click this link to create a new server.

Bach Tran: Right, so one time in there.

Bach Tran: navigate over to a patchy.

Bach Tran: And then look for tomcat V nine.

Bach Tran: Sir.

Bach Tran: Right, then you can just go ahead and click next.

Bach Tran: From there you want to find the installation directory of tomcat and we just download it so we want to click on routes.

Bach Tran: And then just click on Apache tomcat right the folder that you just unzip.

Bach Tran: it's like that folder.

Bach Tran: and

Bach Tran: yeah for Jerry let me select.

Bach Tran: me actually add my other jr you right, you can do that, or you can stick with whatever's in the ice, but I will look for.

Bach Tran: This one right here.

Bach Tran: and switch over to that fine clothes and.

Bach Tran: yeah there we are.

Bach Tran: Alright, so once you do that you want to click on next.

Bach Tran: And now on this screen you'll see that there are resources that we need to move to the right to configure onto the server right So here we have our spring mvc DEMO project that we want for the server to actually run.

Bach Tran: So we need to actually move this project over to the right side here right just click add and now it's available to be run by tomcat.

Bach Tran: And then we can just click on finish.

Bach Tran: And down here it's going to say tomcat version nine server at localhost is stopped, we want to go ahead and start the server and if everything works, then we should not see any sort of exception, so let's start the server.

Bach Tran: and

Bach Tran: let's see here any issues.

Nicholas Hailey: i'm just not sure about anybody else.

Nicholas Hailey: But I was still downloading and so on, a missed just the part where you got to server spot where we all went to windows and we added the servers.

Bach Tran: Yes, so you want to add the server so go to Apache select tomcat or denied click on next then you'll want to go through and look for your tomcat folder that you just unzip.

Bach Tran: And then add add this to the right side and then you're done.

Nicholas Hailey: Actually tomcat be nice server click on that it next.

Bach Tran: yeah then browse for your folder once that's done click next and then add the project from the left side to the right side and then click finish.

Nicholas Hailey: You can find it on my desktop.

Bach Tran: So it's the I could open this up in.

Bach Tran: A post man and then send a get request oh i'm gonna go ahead and try it out here so local host.

Bach Tran: What port, are we using at at.

Bach Tran: It says it right here, so at at is the port by default.

Bach Tran: and

Bach Tran: You would think that maybe I could do Hello like this, I actually can't I need to specify the project name first.

Bach Tran: So.

Bach Tran: The project name is spring dash mvc dash DEMO so I need to do that slash spring.

Bach Tran: mvc.

Bach Tran: slash Hello.

Bach Tran: And let's see it gives an internal server error something's going wrong so we're gonna.

Bach Tran: See right here.

Bach Tran: what's going on class not found exception apparently this dispatcher servant does not exist inside of.

Bach Tran: Our depends.

Bach Tran: They must have moved the location so i'm gonna have to dig around and find the actual location of the dispatcher certainly because what happened is I sent the request and then it.

Bach Tran: tried to fire up this service mapping here and create the dispatcher survey, but apparently this path doesn't exist so.

Bach Tran: On yeah let's see what's.

Bach Tran: going on.

there.

Bach Tran: let's go to maven dependencies go to spring Web.

Bach Tran: org dot spring framework dot web yeah they got rid of the surplus package.

Bach Tran: Actually first thing i'll do is switch back to 5.3 point nine for the palm here, a while ago I switched it to 5.3 Point six i'm going to make sure.

Bach Tran: That that dependencies right there so yeah this should be the dependencies here and the version.

Bach Tran: So yeah I just stop the server as well, and let me find where the dispatcher servile it actually is so spring Web.

Bach Tran: yeah they got rid of the surplus.

Bach Tran: package.

Bach Tran: So.

Bach Tran: I would.

Nicholas Hailey: Well, are we supposed to get a exception or errors, because I don't think I have in your mind.

Bach Tran: You will, if you try sending a request.

Bach Tran: Right I got here i'm sending a get request to spring mvc Dallas Hello i'm.

Bach Tran: Okay, I think I think they they were completely restructured, a lot of the modules.

Bach Tran: In the past web was just all by itself, but based on this documentation here, it looks like they created a new service module.

Bach Tran: that's separate from web, so I actually need to plug in.

Bach Tran: I need a plugin yet another.

Bach Tran: Dependency here.

Bach Tran: Which is the service module, so I am learning about the new changes on the spot as well.

Bach Tran: So let's search for spring serve it up.

Bach Tran: let's see.

Bach Tran: Let me see here spring.

Bach Tran: web me see.

Bach Tran: Ah, OK.

Bach Tran: Alright alright alright, so this is what we're going to do inside of the palm will actually get rid of spring contacts.

Bach Tran: We will get rid of spring web as well, and what will actually grab is.

Bach Tran: This dependency no not that see where is it.

Bach Tran: will have the right window this one.

Bach Tran: So you actually want these three dependencies here.

Bach Tran: Alright, so yeah it's the latest message in the chat there.

Bach Tran: Alright, so if we look at spring web mvc.

Bach Tran: Now we see those server packages.

Bach Tran: Right now we see, there is a dispatcher sort of available.

Bach Tran: So that's good cool cool.

Bach Tran: Alright, so if I start up the.

Bach Tran: server again.

Bach Tran: Now I don't see anything about dispatcher certainly here.

Bach Tran: And i'm just going to go into my browser and.

Bach Tran: Type this in right everything that you whenever you go to the web browser and press enter the link that's that's a get request alright.

Bach Tran: So here we execute the get request.

Bach Tran: And yeah it gives us Hello world.

Bach Tran: and actually that took a little bit of time because it didn't start up the dispatcher certainly until I sent my very first request that's not necessarily something I want to do I want it to load up immediately.

Bach Tran: Instead of lazy loading the dispatcher server, so we should be able to.

Bach Tran: edit our web dot xml a little bit, so let me stop the server again.

Bach Tran: We will go back to the.

Bach Tran: web dot xml and right here, I will specify load on startup.

Bach Tran: To be quiet.

Bach Tran: We just added a new line here load on startup.

Bach Tran: That means that, whenever tomcat starts up, it will actually load up this this factor service immediately instead of waiting for the very first request to come in and then starting it up.

Bach Tran: So this is another God not related it's not related at all to lazy and he are loading of spring beans, this is its own thing with surveillance and tomcat but, as you can tell eager versus lazy loading applies to many different technical concepts in software development.

Bach Tran: So now, if I start up.

Bach Tran: A server again.

Bach Tran: it's going to load up the dispatcher service it right from the start right notice how it says here.

Bach Tran: initializing servile it dispatcher server.

Bach Tran: And then from there.

Bach Tran: If I go over to the link host at spring and see demos Hello and boom it's immediate that's time because the herbal it's already available didn't have to load up at that moment.

Bach Tran: Alright, so if we wanted to do a post.

Bach Tran: let's do like login and point it consume.

Bach Tran: Application json so it takes in some json and.

Bach Tran: We want to exclude a return a response body of.

Bach Tran: a string it could be like.

Bach Tran: The anything and.

Bach Tran: So yeah post mapping, we just need to get that imported.

Bach Tran: Inside of here, this is where we place a.

Bach Tran: Request body.

Bach Tran: Right, we could do with us body and it's going to take the body.

Bach Tran: and basically turn that body into an object so like we could turn it into a login D to object.

Bach Tran: Right, so if I created a login D to kind of COM or dot pto.

Bach Tran: username and password and let's do.

Bach Tran: getter.

Bach Tran: setter.

Bach Tran: I think it was people and hash code.

Bach Tran: And know our constructor.

Bach Tran: And to strength.

Bach Tran: Now, we do need lombok as a dependency so we'll also grab that as well.

Bach Tran: So long box there and let's go ahead and import.

Bach Tran: Right so that's our very simple setup for a login D to, and then we can go back over here.

Bach Tran: and

Bach Tran: let's just return the login D to.to string right, so the the string representation of this object.

Bach Tran: And notice how it automatically reloads.

Bach Tran: My application whenever I make any changes that's not always it's not always reliable, so you do want to probably restart.

Bach Tran: But right now it's probably working just fine so to test out the post mapping of course I need to actually bring up post man, because I can't do that from the browser unless i'm writing some javascript so.

Bach Tran: let's test out the login.

Yves Bouele: Will I be able to do all these things, plus man, if my server by server slashed HALO did not work so i'm not sure if i'll be able to work.

Bach Tran: yeah if you're Hello doesn't work then yeah this one will work.

either.

Bach Tran: let's see we'll go ahead and try this out and see what issue you're having.

Bach Tran: So username test 123.

Bach Tran: s we're.

Bach Tran: password.

Bach Tran: And yeah so right all it takes in is this json request body this annotation tells.

Bach Tran: spring to basically just take that json and turn it into a login D to object and then pass it into a method and then just return that object.

Bach Tran: To string.

Bach Tran: method here.

Bach Tran: And so we get that string back as the responsibility.

Bach Tran: Alright cool so yeah let me go ahead and stop the server.

Bach Tran: And yeah us let's take a look at your problem.

Yves Bouele: So.

Bach Tran: You will be putting out controller on the top of login detail because it's not a controller.

Yves Bouele: What, of course, the issue was before.

Yves Bouele: Before.

Bach Tran: Just make sure to get rid of that so that there's no syntax problems and then just save the file.

Bach Tran: go over to test controller, is that it looks like you have some syntax here.

Bach Tran: yeah you need to import those annotations.

Bach Tran: And you need a semi colon at the end.

Bach Tran: of it.

Bach Tran: So make sure your palm not xml matches up with my.
Yves Bouele: In he does he does a copy the last one.
Okay.
Bach Tran: So yeah go ahead and import those annotations.
Yves Bouele: yeah.
Bach Tran: Just just do control shift though or highlight your mouse.
Nicholas Hailey: Control oh that's how you do that.
Bach Tran: yeah yeah yeah that's the shortcut instead of having to.
Bach Tran: highlight over every single one and.
Bach Tran: Then clicking.
Roberto (Rob) Granciosa: So mad.
Nicholas Hailey: Oh, my God.
These are things.
Nicholas Hailey: That you should be.
Saying.
Bach Tran: All right, you want to right click on your server and restart it right so yeah just just make sure it restart here.
Bach Tran: And yeah now now try to do the.
Bach Tran: Get request it should be your project and then.
Yves Bouele: yeah my project is exactly that space ABC the.
Yves Bouele: slash.
Yves Bouele: localhost eight.
Yves Bouele: Are you ready to go and then i'll come back on the video is to doesn't work I don't want this to happen.
Yves Bouele: yeah it's not working.
Bach Tran: controller kidnapping.
Bach Tran: The only thing we can see is your idea, so if you can share your entire screen, that would be more useful.
Oh.
Yves Bouele: What you mean.
Yves Bouele: That oh OK.
Bach Tran: We can always see spring tours we can't see anything.
Yves Bouele: Oh.
Yves Bouele: Okay.
Yves Bouele: same.
Bach Tran: spring and we feed DEMO flash Hello or.
titian.
Bach Tran: Now okay so let's see.
Bach Tran: Go ahead and right click on your server down at the bottom.
Bach Tran: And let's go to properties so going to the properties there.
Bach Tran: And that's not right double Double Click on the server instead.
Bach Tran: scrolling down.
Bach Tran: clicking on module.
Bach Tran: I don't think you ever added the application to the server so uh.
Yves Bouele: Yes, yes.
Bach Tran: Try right clicking.
Bach Tran: Right click on the server again.
Bach Tran: um.
Bach Tran: let's see publish.
Bach Tran: going to add and remove.
Bach Tran: yeah you need to add that over to the right side that none click finish.
Bach Tran: All right now yeah just close out of this you don't need to be.
Bach Tran: restart your server.
Bach Tran: You probably have something incorrect in your web dot xml i'm thinking go go over to the web dot xml.
Bach Tran: yeah so you have the wrong parameter name, it should be context context location not context configuration so change that to
what I put in the chat.
Bach Tran: context and location.
Yves Bouele: yeah.
Bach Tran: yeah go ahead and save that and then restart uh yeah.
Bach Tran: All right.
Yves Bouele: Good.
Bach Tran: yeah I think that.
works.
Yves Bouele: Thank you.
Bach Tran: Alright cool yeah so a very common issue where if it can't find the endpoints even though you know that you.
Bach Tran: set everything up correctly project it's probably because.
Bach Tran: You did actually add the application to the server itself right so that's very important you need to actually.
Bach Tran: Right click on the server click on add and remove and move whatever is on the left side to the right side here.
Bach Tran: right then click finish, and then it should be available, because if you don't have the application on the tomcat server itself that hey

it's not going to talk that's not going to be running that yeah that's definitely an important part of there.

Bach Tran: Are other things that you can do is to clean so like right click clean that will.

Bach Tran: discard all published state and replenish from scratch right and do that I could also clean the tomcat work directory right there's there's also this option here.

Bach Tran: So you'll just want to try different ways to.

Bach Tran: troubleshooting it to work, this is, this would be the first step, right here, you need to have the APP from over here to over here.

Bach Tran: Alright, so.

Bach Tran: yeah let me go ahead and really quickly just dive into some notes here on the topic of web mvc.

Bach Tran: And then, for the rest of the day we'll just be moving along to project to work and also QC right so i'll put you on two separate breakout rooms once I am done here with the different notes, and then we will.

Bach Tran: cool out the the form mentioned this morning one, at a time for one on ones.

Bach Tran: All right, training talk nation week six.

Bach Tran: spring mvc.

Bach Tran: spring mvc notes.

Bach Tran: So spring mvc.

Bach Tran: What is spring.

Bach Tran: spring mvc.

Bach Tran: or spring model view controller.

Bach Tran: Is a framework built around the concept.

Bach Tran: Of bottle.

Bach Tran: controller.

Bach Tran: That enabled web application functionality.

Bach Tran: Within the text.

Bach Tran: based around the circle API with.

Bach Tran: dispatcher sort of look and so.

Bach Tran: stature starlet is the single server.

Bach Tran: That we register.

Bach Tran: With the.

Bach Tran: web server in this case.

Bach Tran: That will forward http requests to the corresponding.

Bach Tran: classes with controller.

Bach Tran: class.

Bach Tran: Alright, so.

Bach Tran: controllers.

Bach Tran: At as the entry point to the core business logic.

Bach Tran: The application.

Bach Tran: Service layer

Bach Tran: etc.

Bach Tran: So we haven't really shown this, but there is also another special component.

Bach Tran: To be resolved.

Bach Tran: which can be used to handle beautifully.

Bach Tran: Like gsb right really old technology tsp or static files like html, but you can also use it Sir up static files as well.

Bach Tran: For our purposes.

Bach Tran: we're really only making use of the.

Bach Tran: model and roller our spring mvc.

Bach Tran: park.

Bach Tran: But the view.

Bach Tran: Is there to provide.

Bach Tran: server side.

Bach Tran: rendering of.

Bach Tran: static.

Bach Tran: So we're just making use of models passing data around controller receive requests right that's that's really the part that we're mainly using.

Bach Tran: The whole view part gets into more complicated territory with like, for example, you might use dsp which are Java server pages, which do look like html but they have like this dynamic functionality behind the scenes, but we're only using model and controller.

Bach Tran: For us, the view is handled.

Bach Tran: client side.

Bach Tran: Using.

Bach Tran: Alright, so the general flow of spring mvc requests right so have a request.

Bach Tran: Alex do general flow of http requests to spring.

Bach Tran: So the http request sent to the search.

Bach Tran: engine is, in this case tomcat.

Bach Tran: tomcat creates an http.

Bach Tran: request is actually lower case here.

Bach Tran: So http service request object and http server response.

Bach Tran: i'm cat forward these objects.

Bach Tran: dispatcher.
Bach Tran: Which is an org dot spring framework web dot zero dot dispatch.
Bach Tran: Is configured and our web dot xml to be the.
Bach Tran: default server that.
Bach Tran: contact you for all of our.
Bach Tran: map.
Bach Tran: to any other.
Bach Tran: Using flash as the URL mapping means that all yeah actually.
Bach Tran: yeah yeah so so using flash as a URL mapping inside of web dot xml is what determines this right here.
Bach Tran: So, theoretically, we could have other service what's.
Bach Tran: Inside of our web dot xml with others are like mapping.
Bach Tran: Those would go to those other services if there were specific mapping, but any that are not will go to the dispatcher certainly which, right now, we only have the dispatcher serve.
Bach Tran: As the single server that handles requests so yeah it just goes to the dispatcher sort of like and only the dispatcher server.
Bach Tran: And then from there, the dispatchers are lit.
Bach Tran: consults the handler.
Bach Tran: Oh.
Bach Tran: In our case.
Bach Tran: Were.
Bach Tran: Actually, making use of the org dot spring framework taught web server live dot NBC dot method got annotation dot question mapping mapping.
Bach Tran: right behind the scenes.
Bach Tran: We configure this handler mapping.
Bach Tran: In our application contact.
Bach Tran: got xml.
Bach Tran: Using what we saw mvc and.
Bach Tran: So.
Bach Tran: Using the controller annotation along with.
Bach Tran: Our.
Bach Tran: Other annotations such as get mapping.
Bach Tran: At coast mapping.
Bach Tran: etc, is what provides handler map for.
Bach Tran: The handler mapping.
Bach Tran: The information.
Bach Tran: Right, so the handler mapping.
Bach Tran: returns.
Bach Tran: To the dispatcher circle it.
Bach Tran: Which controller.
Bach Tran: delegate.
Bach Tran: let's see i'm i'm going to get rid of this part of using the using annotations such as get mapping mapping, etc, is what provides the handler the handler mapping the appropriate.
Bach Tran: Right, and so the handler bobbing returns to the dispatcher service which controller to delegate the request to.
Bach Tran: The controller will then receive.
Bach Tran: and send a spaz back to the dispatcher sort of.
Bach Tran: The dispatcher circle it well, then send the data back to the client.
Bach Tran: In whatever data format is required right so, for example, json that's the most common to be images the.
Bach Tran: In general, just other files X amount.
Bach Tran: Alright, so as for the annotation like these are some annotations that you can definitely.
Bach Tran: That you will definitely be using so add controller specifies a class as a controller.
Bach Tran: Which is a special being.
Bach Tran: used to handle web application requests.
Bach Tran: At request body is another allows spring to map data from the body of a request.
Bach Tran: To a job on.
Bach Tran: Within the.
Bach Tran: Request method.
Bach Tran: But, so what I mean they're.
Bach Tran: The managed to get rid of this part within the methods parameters, I mean whatever I had.
Bach Tran: right here, right so turn the request body into the login do object by utilizing at request body.
Bach Tran: And then we also have at response body so.
Bach Tran: used to convert the Java object.
Bach Tran: or some other specified.
Bach Tran: Like xml technically.
Bach Tran: So let's actually test this out if I do application, instead of application slash json I did like application slash xml.
Bach Tran: Right, well, it gives me X amount instead of json look let's try that out.
Yves Bouele: Yes.
Bach Tran: let's see.

Bach Tran: um.

Bach Tran: unsupported media type date all right, let me see.

Bach Tran: Oh, I might be getting this wrong here, it might not be slash xml let me actually search up the.

Bach Tran: Entire so it's going to be xml mime.

Bach Tran: I.

Bach Tran: i'm all right, it might be text slash xml so let's try.

Bach Tran: Oh, this is consumed, because this is the wrong one consuming json.

Bach Tran: by default it produces json as well, so I don't know actually put that there, but like if I wanted to do xml produces xml right and then let's try to restart this.

Bach Tran: see if it send us back.

Bach Tran: So maximo instead.

Bach Tran: But then again what we're sending back as a string which.

Bach Tran: Just looks like.

Bach Tran: looks like that.

Bach Tran: Is we're not sending back like a an actual object, let me, let me send back the D to again oh that's gonna be better right here, where we're taking in the D to as json and then we're going to return it back.

Bach Tran: let's see here we do that.

Bach Tran: No converter.

Bach Tran: let's see know convert from our end to present content time.

Bach Tran: let's see produces consumes application json produces application slash xml.

Bach Tran: yeah I may need to do some additional configuration like if I did produces application splash piece on that should definitely work.

Bach Tran: yeah let's let's see.

Bach Tran: yeah apparently xml is not immediately.

Bach Tran: Available immediate try text slash xml.

Bach Tran: yeah yeah so it looks like I need to do some additional configuration if I want to send back some xml instead but yeah basically we can send back json and that already happens, by default, we don't really need to put produces here but.

Bach Tran: we're just putting it explicitly here, just so that you know that we can have these different values right, if you look at intelligence and you'll you'll see all of the other possibilities as well.

Bach Tran: Alright, so back to the annotation.

Bach Tran: Alright, see.

Bach Tran: APP response status so.

Bach Tran: used to mark a method or exception, with an http response.

Bach Tran: stethoscope.

Bach Tran: I kind of shy away from this one, because you might run into different situations where you don't simply want to have a random status code.

Bach Tran: that's basically set in stone for that particular request, and so I might have to wait a bit differently and we'll we'll see how we might do that by creating our pilot project probably tomorrow, within the context of spring mvc.

Bach Tran: But here let's do requests per Ram.

Bach Tran: used to bind query parameters right remember the difference between query parameters and path parameter.

Bach Tran: We had this and the very first project right, this is a query parameter.

Bach Tran: Meanwhile, we have another one called path variable.

Bach Tran: used to buy and half or.

Bach Tran: So.

Bach Tran: For example.

Bach Tran: slash reimbursements.

Bach Tran: select one.

Bach Tran: Right not not might be an endpoint to get the image of reimbursement with idea one, this is the parameter.

Bach Tran: right here.

Bach Tran: Where.

Bach Tran: ID is the path.

Bach Tran: parameter.

Bach Tran: Alright request header so.

Bach Tran: Let me also put here.

Bach Tran: And request header, we can also mapping headers.

Bach Tran: Variable.

Bach Tran: And then at rest.

Bach Tran: controller Oh, this one.

Bach Tran: same as.

Bach Tran: At controller.

Bach Tran: But implicitly ads at response body.

Bach Tran: To the controller.

Bach Tran: Oh.

Bach Tran: notice how right here, I am doing at responsibility and that responsibility.

Bach Tran: Two times, if I just replaced controller with rest controller.

Bach Tran: I do not need the response body.

Bach Tran: annotations anymore, it will automatically return these in the response body itself.

Bach Tran: With just by changing from controller to rest controller.

Bach Tran: So I changed the annotation from at at controller to address controller.

Bach Tran: So I do not need to put that response body.

Bach Tran: That response body.

Bach Tran: To specify that be returned.

Bach Tran: serialized.

Bach Tran: into.

Bach Tran: For example.

Bach Tran: A soft.

Bach Tran: place to the.

Bach Tran: body of our.

Bach Tran: http response.

Bach Tran: To we all need at response body here anymore, because of rest controller.

Bach Tran: So yeah those are the annotations you should be aware of for.

Bach Tran: spring mvc.

Nicholas Hailey: hey buck real fast arm.

Nicholas Hailey: Could you go to your login dt oh.

Bach Tran: yeah I have this really simple.

Bach Tran: Word public law in the Tito.

Bach Tran: string user.

Bach Tran: password.

Nicholas Hailey: I think that's what I was missing because oh.

Nicholas Hailey: When I returned login dts out there was underlined so maybe it's because of that.

Bach Tran: So I put lombok in as a dependency and I put the.

Bach Tran: annotations up here.

Nicholas Hailey: and training.

Nicholas Hailey: And no.

Nicholas Hailey: Okay.

Nicholas Hailey: able to finish off in there, thank you.

Bach Tran: sweet.

Bach Tran: So.

Bach Tran: yeah I think with that we have covered what we needed to cover today.

Bach Tran: um.

Bach Tran: yeah Let me also move around, some of the topics on the calendar here.

Bach Tran: Okay scope, so it would be in stereotypes annotations.

Bach Tran: can be see architecture annotations for input spring we don't really need to.

Bach Tran: Cover that one.

Bach Tran: Maybe maybe yeah maybe i'll come over an example for that just to show off the the view side of spring mvc know we don't really need for project to maybe it's a good idea to.

Bach Tran: Introduce that a bit.

Bach Tran: So yeah let's see about that one.

Bach Tran: Okay, so yeah we should be pretty much good to go, so you have the basic outline of how to go about building the backend for project to now.

Bach Tran: Right, so you can actually map out your endpoints you can create your service layer and classes, you can even get started with hibernate there's nothing wrong with just doing hibernate our old way we're going to.

Bach Tran: Probably tomorrow switched over to spring or em but we're still using hibernate it's just going to integrate.

Bach Tran: spring or into our usage here, so we can treat our session factory as a spring being itself and just auto wire it.

Bach Tran: Right so yeah everything applies here right, you can do dependency injection get those services at auto wired at component of those annotations to register your beans and the I just get started with that and i'll be good practice as well for just understanding spring.

Bach Tran: The only.

Bach Tran: difference between spring mvc and our older demos here is that we have this controller layer right an actual controller the layer we can do dependency injection on services and down just pass those activities right.

Bach Tran: All right, so yeah i'm gonna.

Bach Tran: Go ahead and construct some breakout rooms for each group and then.

Bach Tran: yeah let's look at that order for one on ones again.

Nicholas Hailey: hey Bob my mom my return login D to it just keeps telling me that I have to change the method return type to a capital login to my.

Bach Tran: Do you have a string dizzy I need to change it to the return type that's.

Bach Tran: Appropriate right.

Bach Tran: Just just go back to the basic Java concepts like your return type needs to match with what your attorney.

Bach Tran: yeah so coating haiphong rob and on yeah that looks like the ones who are today.

Bach Tran: i'll probably get started on those like.

Bach Tran: i'll basically tell you when i'll get started with them, but for now just just focus on project to.

Nicholas Hailey: Okay, well, can you push this example, then, to the own to get home.

Nicholas Hailey: Look at it and.

Bach Tran: put these over there.

Nicholas Hailey: Thank you.

Bach Tran: Alright, should be all cash so yeah let me go ahead and create those breakout rooms here.

Bach Tran: Alright, should be open.